



تطبيقات الهواتف الذكية

(نظري وعملي)

المسار المهني - الفرع الصناعي

فريق التأليف:

د. بشار طحانية

أ. مؤمن كيلاني (منسقاً)

أ. علاء أبو زهرة

أ. سامية طه

أ. نسرين دويكات



مركز المناهج

قورت وزارة التربية والتعليم في دولة فلسطين
تدريس هذا الكتاب في مدارسها بدءاً من العام الدراسي ٢٠١٨ / ٢٠١٩ م

الإشراف العام

رئيس لجنة المناهج د. صبري صيدم
نائب رئيس لجنة المناهج د. بصري صالح
رئيس مركز المناهج أ. ثروت زيد

الدائرة الفنية

إشراف إداري كمال فحماوي
تصميم فاطمة عبد الحق

تحرير لغوي أ. وفاء الجبوسي
متابعة المحافظات الجنوبية د. سميرة النخالة

الطبعة التجريبية
٢٠٢٠ م / ١٤٤١ هـ

جميع حقوق الطبع محفوظة ©

دولة فلسطين
وزارة التربية والتعليم



مركز المناهج

mohe.ps | mohe.pna.ps | moehe.gov.ps

www.facebook.com/Palestinian.MOEHE

هاتف +970-2-2983280 | فاكس +970-2-2983250

حي الماصيون، شارع المعاهد

ص. ب. ٧١٩ - رام الله - فلسطين

pcdc.mohe@gmail.com | pcdc.edu.ps

يتصف الإصلاح التربوي بأنه المدخل العقلاني العلمي النابع من ضرورات الحالة، المستند إلى واقعية النشأة، الأمر الذي انعكس على الرؤية الوطنية المطورة للنظام التعليمي الفلسطيني في محاكاة الخصوصية الفلسطينية والاحتياجات الاجتماعية، والعمل على إرساء قيم تعزز مفهوم المواطنة والمشاركة في بناء دولة القانون، من خلال عقد اجتماعي قائم على الحقوق والواجبات، يتفاعل المواطن معها، ويعي تراكيبها وأدواتها، ويسهم في صياغة برنامج إصلاح يحقق الآمال، ويلامس الأماني، ويرنو لتحقيق الغايات والأهداف.

ولما كانت المناهج أداة التربية في تطوير المشهد التربوي، بوصفها علماً له قواعده ومفاهيمه، فقد جاءت ضمن خطة متكاملة عالجت أركان العملية التعليمية التعلمية بجميع جوانبها، بما يسهم في تجاوز تحديات النوعية بكل اقتدار، والإعداد لجيل قادر على مواجهة متطلبات عصر المعرفة، دون التورط بإشكالية التشتت بين العولمة والبحث عن الأصالة والانتماء، والانتقال إلى المشاركة الفاعلة في عالم يكون العيش فيه أكثر إنسانية وعدالة، وينعم بالرفاهية في وطن نحمله ونعظمه.

ومن منطلق الحرص على تجاوز نمطية تلقي المعرفة، وصولاً لما يجب أن يكون من إنتاجها، وباستحضار وإع لعدد المنطلقات التي تحكم رؤيتنا للطلاب الذي نريد، وللبنية المعرفية والفكرية المتوخاة، جاء تطوير المناهج الفلسطينية وفق رؤية محكمة بإطار قوامه الوصول إلى مجتمع فلسطيني ممتلك للقيم، والعلم، والثقافة، والتكنولوجيا، وتلبية المتطلبات الكفيلة بجعل تحقيق هذه الرؤية حقيقة واقعة، وهو ما كان له ليكون لولا التناغم بين الأهداف والغايات والمنطلقات والمرجعيات، فقد تألفت وتكاملت؛ ليكون النتاج تعبيراً عن توليفة تحقق المطلوب معرفياً وتربوياً وفكرياً.

ثمّة مرجعيات توطّر لهذا التطوير، بما يعزز أخذ جزئية الكتب المقررة من المنهاج دورها المأمول في التأسيس؛ لتوازن إبداعي خلاق بين المطلوب معرفياً وفكرياً، ووطنياً، وفي هذا الإطار جاءت المرجعيات التي تم الاستناد إليها، وفي طبيعتها وثيقة الاستقلال والقانون الأساسي الفلسطيني، بالإضافة إلى وثيقة المنهاج الوطني الأول؛ لتوجّه الجهد، وتعكس ذاتها على مجمل المخرجات.

ومع إنجاز هذه المرحلة من الجهد، يغدو إجزاء الشكر للطواقم العاملة جميعها؛ من فرق التأليف والمراجعة، والتدقيق، والإشراف، والتصميم، واللجنة العليا أقل ما يمكن تقديمه، فقد تجاوزنا مرحلة الحديث عن التطوير، ونحن واثقون من تواصل هذه الحالة من العمل.

وزارة التربية والتعليم

مركز المناهج الفلسطينية

كانون الأول / ٢٠١٨م

مواكبةً للتطور العلمي والتكنولوجي في عصر المعرفة، أصبحت الحاجة لمهن جديدة ضرورة حتمية، وانطلاقاً من توجهات وزارة التربية والتعليم بتحقيق تعلم وتعليم نوعي وريادي يلبي احتياجات المجتمع الفلسطيني وفق أولوياته بالتحريز والتنمية، كان لا بد من تطوير مناهج التعليم المهني بحلقتها الجديدة لتحاكي التوجهات العالمية بنشأة فلسطينية واقعية تعتمد العقلانية العلمية بالدمج بين الجانبين النظري والتطبيقي، مراعيًا سوق العمل، وصولاً إلى جيل يمتلك الكفايات والقدرة على اكتشاف المعرفة العالمية، والإسهام في الإنتاج الكوني.

اعتمدت المناهج المهنية الجديدة منهجية الوحدات النمطية القائمة على الكفايات المهنية بمجالاتها الحرفية، والمنهجية، والاجتماعية والشخصية، وارتباط ذلك بسياقات واقعية حياتية تصف مواقف تعليمية تعلمية، توظف حل المشكلات بطريقة علمية وعملية، دون إغفال للجانب النظري المتضمن لأنشطة تعتمد على أداء الطلبة، والتأكيد على استراتيجيات التقويم التربوي الحديثة بما في ذلك التقويم الأصيل، والتحول من التقويم القائم على تحقيق الكفايات إلى تحقيق الجودة والامتياز، من خلال التركيز على شمولية أداء الطلبة، وتعزيز أنماط التفكير النقدي والتأملي.

جاء كتاب تطبيقات الهواتف الذكية للصف الحادي عشر المهني في خمس وحداتٍ نمطية، عالجت موضوعات: الاختيار الأنسب للهاتف المحمول، وفق الاحتياجات والإمكانيات. التفكير المنطقي، وتصميم الخوارزميات ومخطط سير العمليات. أساسيات البرمجة، وتوظيف ميزات (البرمجة الكائنية) في تحسين جودة البرنامج. أساسيات تطبيقات الأندرويد، وتوظيف (الأكواد) البرمجية وطريقة دمجها مع التصميم، وبرمجة الواجهات لتطبيقات الهاتف الذكي الذي يعمل بنظام الأندرويد.

وإذ نقدّم هذا الكتاب بين أيدي ذوي العلاقة من معلمات ومعلمين وطلبة وتربويين ومهتمين، نرجو تحقيق الغايات التربوية المنشودة، وب عقلية منفتحة نحو التطوير والتعديل، فإنه يسرنا استقبال أيّ ملاحظة تهدف إلى تطوير الكتاب وتجويده.

المحتويات

٦	الموقف التعليمي التعلّمي (١-١): التطوّر التاريخي لوسائل الاتّصال والتّواصل
١٥	الموقف التعليمي التعلّمي (١-٢): معرفة العناصر الرئيسيّة التي تحدّد كفاءة الأجهزة النّقالة الذكيّة
٢٢	الموقف التعليمي التعلّمي (١-٣): معرفة أنظمة التشغيل المختلفة المستخدمة في الهواتف النّقالة الذكيّة وتطبيقاتها
٢٨	أسئلة الوحدة

الهواتف النّقالة الذكيّة

٣٣	الموقف التعليمي التعلّمي (٢-١): توظيف الخوارزميّات لإيجاد حلول برمجيّة لمشاكل حياتيّة.
٤٦	الموقف التعليمي التعلّمي (٢-٢): توظيف مخطّط سير العمليّات (Flowchart) لإيجاد حلول برمجيّة لمشاكل حياتيّة.
٥٧	أسئلة الوحدة

التفكير المنطقيّ

٦٣	الموقف التعليمي التعلّمي (٣-١): توظيف مبادئ أساسيّات البرمجة في بناء برنامج ١
٩١	الموقف التعليمي التعلّمي (٣-٢): توظيف مبادئ أساسيّات البرمجة في بناء برنامج ٢
١١٣	الموقف التعليمي التعلّمي (٣-٤): توظيف مبادئ (البرمجة الكائنيّة) في بناء برنامج ١
١٢٣	الموقف التعليمي التعلّمي (٣-٥): توظيف مبادئ (البرمجة الكائنيّة) في بناء برنامج ٢
١٤١	أسئلة الوحدة

أساسيّات البرمجة

١٤٨	الموقف التعليمي التعلّمي (٤-١): تجهيز بيئة صناعة تطبيقات الأندرويد
١٦٨	الموقف التعليمي التعلّمي (٤-٢): تصميم تطبيق أندرويد (١)
١٧١	الموقف التعليمي التعلّمي (٤-٣): تصميم تطبيق أندرويد (٢)
١٧٩	الموقف التعليمي التعلّمي (٤-٤): تأهيل تطبيق أندرويد (١)
١٨٤	الموقف التعليمي التعلّمي (٤-٥): تأهيل تطبيق أندرويد (٢)
١٨٨	الموقف التعليمي التعلّمي (٤-٦): تأهيل تطبيق أندرويد (٣)
١٩٣	أسئلة الوحدة

أساسيّات تطبيقات الأندرويد

١٩٧	الموقف التعليمي التعلّمي (٥-١): التعرّف على دورة حياة الواجهة Activity
٢٠٣	الموقف التعليمي التعلّمي (٥-٢): التعرّف على الإيماءات Gesture
٢٠٩	الموقف التعليمي التعلّمي (٥-٣): القوائم Menu
٢٢٤	الموقف التعليمي التعلّمي (٥-٤): التنقل ونقل بيانات بين الواجهات
٢٣٢	الموقف التعليمي التعلّمي (٥-٥): التخطّط بين التطبيقات
٢٤٢	أسئلة الوحدة

برمجة الواجهات Activity

الهواتف النّقالة الذّكيّة



أناقشُ:

العالمُ قريةٌ صغيرةٌ بِفَضْلِ
وسائلِ الاتّصالِ الحديثةِ



يُتَوَقَّع من الطلبة بعد دراسة هذه الوَحْدَةِ، والتَّفَاعُلِ مع أنشطتها أن يكونوا قادرين على الاختيار الأنسب للهاتف المحمول، وفقَ الاحتياجات والإمكانيات من خلال ما يأتي:

✓ معرفة التطور التاريخي لوسائل الاتصال والتواصل.

✓ معرفة العناصر الرئيسية التي تحدّد كفاءة أداء الأجهزة النقالة الذكية.

✓ معرفة أنظمة التشغيل المختلفة المستخدمة في الهواتف النقالة الذكية وتطبيقاتها.



الكفايات المهنية

الكفايات المهنية المتوقعة من الطلبة امتلاكها بعد انتهاء دراسة هذه الوحدة والتفاعل مع أنشطتها
أولاً: الكفايات الحرفية:

- ◀ القدرة على تصنيف الهواتف المحمولة وفق (العتاد، المواصفات، الشركات المنتجة، السعر).
- ◀ القدرة على تصنيف الهواتف المحمولة الذكية حسب أنظمة التشغيل الداعمة، إضافة إلى التطبيقات المتاحة لها.
- ◀ القدرة على تحديد مواصفات الهاتف المحمول الأنسب للزبون اعتماداً على احتياجاته والميزانية المتاحة لديه.
- ◀ تنصيب البرامج التطبيقية المختلفة وحذفها باستخدام أنظمة تشغيل مختلفة.

ثانياً: الكفايات الاجتماعية والشخصية :

- ◀ العمل ضمن فريق.
- ◀ تقبل التغذية الراجعة.
- ◀ احترام رأي الآخرين.
- ◀ مصداقية التعامل مع الزبون.
- ◀ حفظ خصوصية الزبون.
- ◀ الاستعداد للاستجابة لطلبات الزبون.
- ◀ التحلي بأخلاقيات المهنة أثناء العمل.
- ◀ التمتع بالفكر الريادي أثناء العمل.
- ◀ استخلاص النتائج ودقة الملاحظة.
- ◀ الاتصال والتواصل الفعال.
- ◀ الاستعداد للاستفسار والاستفادة من ذوي الخبرة.
- ◀ الاستفادة من مقترحات الآخرين.
- ◀ امتلاك مهارة التأمل الذاتي.
- ◀ الاستعداد التام لتقبل ملحوظات الزبون وانتقاداتهم.
- ◀ القدرة على الإقناع.

ثالثاً: الكفايات المنهجية:

- ◀ توظيف التعلم التعاوني.
- ◀ القدرة على إدارة الحوار وتنظيم النقاش.
- ◀ امتلاك مهارة البحث العلمي، والقدرة على توظيف أساليبه.
- ◀ القدرة على التخطيط.
- ◀ القدرة على استمطار الأفكار.
- ◀ استخدام المصادر والمراجع المختلفة.
- ◀ توثيق نتائج العمل وعرضها.
- ◀ تقويم النتائج وعرضها.
- ◀ توظيف التكنولوجيا والبرامج الإحصائية وتمثيل البيانات.
- ◀ جمع البيانات وتحليلها وتبويبها.

قواعد الأمن والسلامة:



- ✓ عدم استخدام الهواتف المحمولة في مكان مظلم.
- ✓ عدم التحدث بالهواتف المحمولة لفترات طويلة.
- ✓ شحن بطاريات الهواتف المحمولة قبل نفاذها.
- ✓ نظافة الأيدي والأسطح التي يتم وضع الهواتف المحمولة عليها.
- ✓ عدم استخدام كوابل متعرية لتوصيل الهواتف المحمولة بالكهرباء.



الموقف التعليمي التعلّمي (1 - 1):

التطوّر التاريخي لوسائل الاتصال والتواصل

وصف الموقف التعليمي: يرغب طلبة إحدى المدارس الأكاديمية بإعداد مطوية عن التطوّر التاريخي لوسائل الاتصال والتواصل، فزاروا مشغلاً لتطبيقات المحمول في مدرسة الشهيد صلاح خلف المهنيّة للبنات؛ للاستفسار عن مضمون المطوية.

العمل الكامل			
الموارد	المنهجية (استراتيجية التعلم)	وصف الموقف الصّفي	خطوات العمل
<ul style="list-style-type: none">• وثائق (طلب المدرسة الزائرة، نشرات، كتب ومصادر متنوعة، مجلات عن وسائل الاتصال قديماً وحديثاً).• التكنولوجيا (روابط إلكترونية متنوعة من مصادر موثوقة عن مراحل تطوّر وسائل الاتصال، حاسوب، شاشة عرض، إنترنت)	<ul style="list-style-type: none">• حوار ومناقشة.• التعلم التعاوني/مجموعات.• البحث العلمي.	<p>جمع البيانات من طلبة المدرسة الأكاديمية حول:</p> <ul style="list-style-type: none">• معرفتهم السابقة بمفهوم الاتصال والتواصل.• التطوّر التاريخي لوسائل الاتصال والتواصل.• محتوى المطوية.• مواصفات المطوية (عدد الصفحات، الصور، روابط تعليمية...). <p>جمع البيانات عن:</p> <ul style="list-style-type: none">• مفهوم عمليّة الاتصال.• تطوّر وسائل الاتصال عبر العصور.	<p>أجمع البيانات وأحلّها</p>
<ul style="list-style-type: none">• وثائق (طلب المدرسة الزائرة، مكثبات، مجلات عن وسائل الاتصال قديماً وحديثاً).	<ul style="list-style-type: none">• تعلم تعاوني/مجموعات.• حوار ومناقشة.• عصف ذهني/استمطار الأفكار.	<ul style="list-style-type: none">• تصنيف وسائل الاتصال وفق الفترة الزمنية لظهورها.• تحديد طريقة استخدام كلّ وسيلة اتصال.• تحديد ميّزات وسائل الاتصال القديمة والحديثة وعيوبها.• إعداد خطّة العمل التي تتضمّن موعد تسليم المطوية للزائرين.	<p>أخطّط وأقرّر</p>



<ul style="list-style-type: none"> • هاتف سلكي. • هاتف نقال. • هاتف ذكي. • تلفاز. • إنترنت حاسوب. • البيانات التي تمّ جمعها. • قرطاسية. • الطلب الخاص بالزائرين. 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات. • حوار ومناقشة. 	<ul style="list-style-type: none"> • كتابة تعريف لمفهوم الاتصال والتواصل. • إعداد قائمة تتضمن وسائل الاتصال المختلفة في العصر القديم (الإشارة، والدخان، والقرع بالطبول، والنقش في الحجر، والكتابة...) • إعداد قائمة تتضمن وسائل الاتصال المختلفة في عصر الطباعة. • إعداد قائمة تتضمن وسائل الاتصال المختلفة في عصر ثورة الاتصالات (الراديو، والتلفاز، والهواتف السلكية واللاسلكية). • كتابة مميزات كل نوع من وسائل الاتصال والتواصل وعيوبها. • إنشاء جداول للمقارنة بين وسائل الاتصال المختلفة القديمة والحديثة. • إعداد المطوية بحلتها الأولية. 	<p>أنفذ</p>
<ul style="list-style-type: none"> • البيانات التي تمّ جمعها. • هاتف سلكي، هاتف نقال. • هاتف ذكي، تلفاز. • إنترنت، حاسوب. • الخطة المعدة سابقاً. 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات. • حوار ومناقشة. 	<p>التحقق مما يأتي:</p> <ul style="list-style-type: none"> • تحديد مفهوم الاتصال والتواصل. • تحديد وسائل الاتصال والتواصل (في العصر القديم، وفي عصر الطباعة، وفي عصر ثورة الاتصالات). • تحديد مميزات كل نوع من وسائل الاتصال والتواصل وعيوبها. • إجراء مقارنة بين مختلف وسائل الاتصال المتوفرة. • اكتشاف الأخطاء وتصويبها واقتراح البدائل. • اكتشاف الأخطاء وتصويبها واقتراح البدائل. 	<p>أتحقق</p>



<ul style="list-style-type: none"> • البيانات التي تمّ جمعها سابقاً. • حاسوب. • شاشة عرض. • قرطاسية. 	<ul style="list-style-type: none"> • حوار ومناقشة. • تعلم تعاوني / مجموعات. • عمل. 	<p>توثيق البيانات التي تتعلّق بِ:</p> <ul style="list-style-type: none"> • مفهوم عمليّة الاتصال. • وسائل الاتصال. - العصر القديم (الإشارة، الدّخان، القرع بالطبول، النقش في الحجر، الكتابة...). - عصر الطباعة (الطباعة). - عصر ثورة الاتصالات (الراديو، التلفاز، الهواتف السلكيّة واللاسلكيّة...). • إعداد العروض التقديميّة. • إعداد ملف بالحالة (التطور التاريخي لوسائل الاتّصال والتواصل). • أقدمُ ملخصاً يتضمّن أهمّ وسائل الاتصال المستخدمة عبر التاريخ، وميزة كلّ منها، ومبدأ عملها. 	<p>أوثق وأقدّم</p>
<ul style="list-style-type: none"> • نماذج التقييم المعدّة سابقاً. • طلب المدرسة الزائرة. 	<ul style="list-style-type: none"> • حوار ومناقشة. 	<ul style="list-style-type: none"> • مقارنة معرفة الطلبة الزائرين عن تطوّر وسائل الاتصال المختلفة، وميزة كلّ وسيلة اتصال، ومبدأ عملها قبل الزيارة وبعدها. • رضا المدرسة الزائرة. 	<p>أقوم</p>

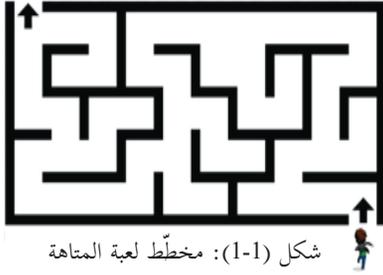
أناقش: اكتشاف الشبكة العنكبوتية أحدث ثورة في عالم الاتّصال والتّواصل.





التطور التاريخي لوسائل الاتصال والتواصل

نشاط عملي (1):



شكل (1-1): مخطّط لعبة المتاهة

رسم متاهة كما في الشكل (1-1) على أرضية الغرفة باستخدام لاصق ملون، ويُطلّب من أحد الطلبة الوقوف على باب المتاهة بعد عصب عينيه، بحيث يتمّ التّواصل مع زملائه بطريقة لفظية لمساعدته في الخروج من المتاهة

مفهوم التواصل

التّواصل: حالة من الفهم المتبادل بين شخصين أو نظامين، يكون أحدهما مُرسلاً في وقتٍ معيّن والآخر مُستقبلاً، وفي وقتٍ آخر يتبادل الطرفان مواقعهما من حيث الإرسال والاستقبال، وفيها يتمّ تبادل الآراء والأفكار والقناعات والمعلومات والمشاعر، من خلال وسائل متنوعة.

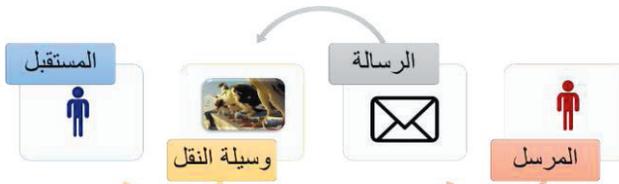
مفهوم الاتصال

الاتصال: عملية نقل رسائل معينة من المرسل إلى المُستقبل، ويتمّ فيها انتقال المعلومات والاتجاهات والأفكار والمشاعر من فردٍ إلى آخر، أو من جماعة إلى جماعة أخرى عبر وسائط متعدّدة.

الفرق بين الاتصال والتّواصل:

الاتصال الرغبة من أحد الطرفين بالتّواصل مع الآخر، ويكون دور الآخر الاستجابة أو عدمها، أمّا التّواصل فهو الرغبة في المشاركة، وتكون متبادلة بين طرفين أو أكثر لتحقيق أهداف معينة، ويُعدّ انفتاح الذات على الآخر ضمن علاقة مستمرة متجدّدة.

عناصر الاتصال والتّواصل



شكل (2-1): عناصر الاتصال والتّواصل

◀ **المرسل:** الطرف الذي تصدر منه الرسالة المراد إرسالها (فرد، مؤسسة، جماعة).

◀ **المحتوى أو الرسالة:** مجموعة الرموز المراد نقلها لإحداث أثر، وقد تكون مرئية، أو مسموعة، أو مشاعر، أو إشارات، أو رسومات، ...

◀ **وسيلة النقل:** الوسط الذي تنتقل خلاله الرسالة (وسائط متعدّدة).

◀ **المستقبل:** الطرف الذي يستقبل الرسالة وهو الهدف من عملية الاتصال (فرد، مؤسسة، جماعة).

نشاط نظري (2):



أناقش مع زملائي بعض الأمثلة المُمكنة حول عناصر الاتّصال والتّواصل

المرسل	المستقبل	المحتوى	وسائل/ وسائط النقل

نشاط نظري (3):



أبحثُ في مجموعات عن مميّزات وسائل الاتّصال المستخدمة قديماً.

وسائل الاتّصال القديمة



منذُ القَدَمِ استخدمَ الإنسانُ العديدَ من طُرُق التّواصل البدائيّة، منها:
◀ الدّخان:

استُخدمت الإشارات الدّخانيّة لإرسال الأخبار، وإشارةً للخطر، أو لتجميع النّاس في منطقةٍ معروفة؛ وذلك بإشعال نارٍ ذات دخانٍ كثيفٍ على المرتفعات (انظر إلى الشكل 3-1).



شكل (3-1): الدخان وسيلة تواصل بدائية





◀ الكتابة الهيروغليفية:

استُخدمت الرسوم التصويرية لحفظ المعلومات، أو التواصُل مع الآخرين، وتعود لعام 3200 ق.م في مصر. (انظر إلى الشكل 4-1).

شكل (4-1): نموذج للكتابة الهيروغليفية

◀ اللغة المكتوبة:

ظهرت في الصين عام 1200 ق.م، وقد لوحظَ وجود تشابه بين لغة بلاد ما بين النهرين واللغة المصرية القديمة، حيث يرجّح العلماء أنّ نظام الكتابة نشأ في الأصل في المنطقة العربية قبل الأماكن الأخرى.



◀ الحمام الزاجل:

كان الإغريق أولّ من استخدم هذا النوع من الحمام قبل 3000 عام وذلك لنقل نتائج الأولمبياد، وكانت تنقل الرسائل عن طريق ربط الرسالة في أرجلها؛ فتطير للمكان المقصود وتُسَلِّم الرسالة. (انظر إلى الشكل 5-1).

شكل (5-1): نقل الرسائل باستخدام الحمام الزاجل



◀ استخدام الرّسل:

مع اقتراب نهاية عهد الميلاد، تمّ إرسال الأشخاص على الأقدام (أو على الدواب) لحمل الرسائل المكتوبة أو الشفهية من مكان لآخر، وكانت هذه الطريقة شائعة الاستخدام في مصر والصين. (انظر إلى الشكل 6-1).

شكل (6-1): نقل الرسائل باستخدام الرّسل

◀ نشاط نظري (4):



أبحثُ عن طرق تدريب الحمام الزاجل: من حيث طريقة التغذية، وزمن التدريب، والمسافة المقطوعة، لكي يتمكن من توصيل الرسالة للمكان المطلوب دون ضياع الطريق، وأناقش ما توصّلتُ إليه مع زملائي.

الصعوبات التي واجهت وسائل الاتصال القديمة

◀ عدم ضمان وصول الرسالة إلى الجهة المعنية، وخاصة في حالة استخدام الحمام الزاجل، فقد يضلّ الحمام الطريق، أو يسلمّ الرسائل إلى الجهات الخاطئة.

◀ عدم تمكّن المرسل من إرسال الرسائل لمسافات طويلة (اقتصر استخدام الدخان على مسافات محدودة).

◀ بذل الكثير من الجهد والوقت من أجل إيصال الرسالة إلى الجهة المعنية.



نشاط نظري (5):

في الوقت الحاضر تعددت وسائل الاتصال والتواصل، أكمل الجدول الآتي:

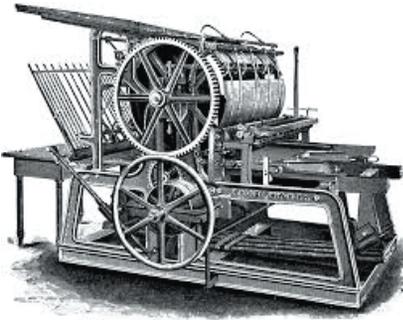
أمثلة	نوع وسيلة الاتصال والتواصل
	المسموعة
	المقروءة
	المرئية

وسائل الاتصال الحديثة



تطوّرت وسائل الاتصال بشكلٍ كبير مع تقدّم الزمن، حيث أنشأ الرومان أولّ خدمة بريد في العالم، كما أنّه تمّ اختراع الورق للكتابة، باستخدام لحاء الأشجار من قبل الصينيين عام 105 م.

أ عصر الطباعة



شكل (1-7): مطبعة يوهان جوتنبرغ

توصّل الصينيون لمقدّمة اختراع الطباعة؛ وذلك بصنّع كتلٍ خشبيّة تُعبّر عن صورٍ ورموزٍ تُغمَس في الحبر، فتطبّع الصور المرادة والرموز، ثم ظهرت الطباعة على يد العالم الألمانيّ (يوهان جوتنبرغ)، وذلك ببناء أولّ نظامٍ معدنيّ متحرّكٍ في أوروبا، حيث تمّ إنشاء مطبعة جوتنبرغ التي تمّ العمل على تطويرها بين عامي 1436م - 1450م؛ ما مكّنها من طباعة العديد من الكتب التي تتّصف بالكفاءة العالية، وفي عام 1605م تمّ طباعة أولّ صحيفة في العالم من قِبَل العالم الألمانيّ (يوهان كارلوس).

ب عصر ثورة الاتصالات

تمّ خلال هذه الفترة اختراع العديد من الأجهزة والوسائط في مجال الاتصالات، منها:

الهاتف السلكي:



شكل (1 - 8) : هاتف سلكي قديم

أول ظهور للهاتف كان عام 1876م، حيث تمّ اختراعه من قِبَل العالم الأمريكيّ (غراهام بل)، ومن خلال هذا الهاتف تمّ التّواصل الصّوتيّ السّريع والبسيط بين الناس عبر المناطق الجغرافيّة المختلفة (انظر إلى الشكل 1-8)





شكل (1 - 9) : راديو قديم

◀ الراديو:

في عام 1896م تمّ اختراع الراديو من قِبَل العالم الإيطاليّ (ماركوني)، حيث نجح في إرسال أوّل انتقالٍ لاسلكيّ للصّوت، وكانت التجربة قد نجحت في إرسال الإشارة على مسافة 2 كم، بعد العديد من المحاولات الفاشلة (انظر إلى الشكل 1-9).



شكل (1-10): اجهزة تلفاز قديمة

◀ التّلفاز:

في شهر أيلول (سبتمبر) من عام 1927م قام مخترع التّلفاز (بيلو فارنسورث) بإظهار أوّل جهازٍ يعمل عن طريق نقل الصّور، حيث اكتشف إمكانية تشفير موجات الراديو مع صورة، ومن ثمّ إعادة عرضها على الشاشة (انظر إلى الشكل 1-10).



شكل (1-11) : هاتف لاسلكي قديم

◀ الهاتف النّقّال:

ظهر أوّل هاتفٍ نقّال عام 1973، وهو جهازٌ للتّواصل، ذو حجمٍ صغير، يرتبط بشبكة اتّصال لاسلكيّة، حيث أصبح الاتّصال لا يرتبط بزمان ومكان معيّنين، فأصبح الشخص يُجري اتّصاله في أيّ مكان، وأيّ وقتٍ يريد (انظر إلى الشكل 1-11).



شكل (1-12): حاسب آلي

◀ الحاسب الآليّ:

يُعدّ ظهور الحاسوب ثورةً في عالم الاتّصال والتّواصل، حيث ظهر أوّل حاسبٍ شخصيّ عام 1981، وهو من إنتاج شركة (IBM) الأمريكيّة، ثمّ تطوّر شيئاً فشيئاً إلى أن وصل إلى ما وصل إليه اليوم من سيطرةٍ على أدقّ تفاصيل الحياة؛ حيث استطاع بفضل الحاسوب التّوصّل إلى اختراع شبكة الإنترنت التي تهيمن على عالم الاتّصالات الآن (انظر إلى الشكل 1-12).

◀ شبكة الإنترنت (World Wide Web):



شكل (1-13): شبكة الانترنت

(World Wide Web): تمّ تقديمها للناس عام 1989، من قِبَل عالمٍ بريطانيّ يُدعى (تيم بيرنرزلي)، حيث طوّر شبكة الويب؛ لتلبية الطلب على مشاركة المعلومات بين العلماء في الجامعات والمعاهد حول العالم، وقد أصبحت بعد ذلك شبكةً عالميّةً تُستخدم من قبل جميع الأشخاص في شتّى المجالات. (انظر إلى الشكل 1-13)

◀ الهاتف الذكيّ:



شكل (1-14): هاتف ذكي

ظهر أوّل هاتفٍ ذكيّ عام 1994، حيث تمّ إنتاجه من قِبَل شركة (IBM) الأمريكيّة، وبذلك وضعت أساساً للهواتف التي تجمع بين وظائف الهاتف المحمول والكمبيوتر الشخصي. (انظر إلى الشكل 1-14)

نشاط نظري (6): أبحث وأناقش



ما ميّزات الجيل الثالث 3G من الهواتف النّقالة؟

نشاط نظري (7):



أناقش في مجموعات أهمّ الفروقات بين الهاتف السّلكيّ والهاتف المحمول الذكيّ.

نشاط للاطلاع (8):

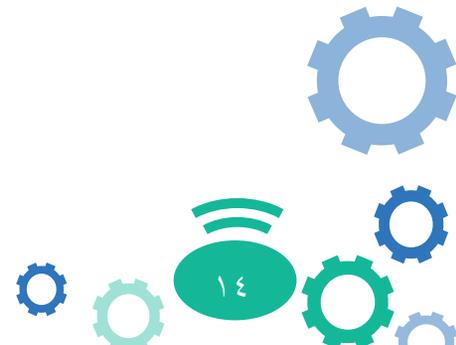


من وسائل الاتّصال المستخدمة في أوائل عصر ثورة الاتّصالات التّلفراف، أبحثُ ضمن مجموعات، وباستخدام الشبكة العنكبوتية عن مبدأ عملها، وأدوّن ذلك في تقرير.

الأسئلة



1. أذكرُ محطات تاريخية في عالم الاتّصال والتّواصل.
2. أذكرُ وسائل الاتّصال المستخدمة في حياتي اليومية.



الموقف التعليمي التعلّمي (1 - 2):

معرفة العناصر الرئيسية التي تحدّد كفاءة الأجهزة النقالة الذكيّة

وصف الموقف: قام طلبة إحدى المدارس الأكاديميّة برحلة علميّة إلى ورشة فنيّة خاصّة بالهواتف النقالة في مدرسة الشهيد سعد صايل المهنّيّة؛ بهدف معرفة أنواع الأجهزة الذكيّة والإمكانيّات الأدائيّة لكلّ منها، من أجل المشاركة في المجلة العلميّة في مدرستهم.

العمل الكامل			
خطوات العمل	وصف الموقف الصّفيّ	المنهجية (استراتيجيةّ التّعلم)	الموارد وفق الموقف الصّفيّ
أجمع البيانات وأحلّها	<p>جمع البيانات من طلبة المدرسة الأكاديميّة حول:</p> <ul style="list-style-type: none">• معرفتهم السابقة بأنواع الأجهزة الذكيّة، والإمكانيّات الأدائيّة لها . <p>جمع البيانات عن:</p> <ul style="list-style-type: none">• أنواع الأجهزة المحمولة الذكيّة، والإمكانيّات الأدائيّة لكلّ منها.• الشركات المنتجة لها.• العناصر التي تُحدّد كفاءة أداء الأجهزة المحمولة الذكيّة.	<ul style="list-style-type: none">• حوار ومناقشة.• التّعلم التعاونيّ /مجموعات.• البحث العلميّ / زيارة ميدانيّة.	<ul style="list-style-type: none">• وثائق (طلب المدرسة الزائرة، مكتبة المدرسة، نشرات، مجلات، كتالوجات عن خصائص الأجهزة المحمولة ومعايير كفاءتها)• الورشة الفنيّة وأجهزة الهواتف المختلفة الموجودة فيها.• التكنولوجيا (روابط إلكترونية متنوعة من مصادر موثوقة عن الشركات المُصنّعة للأجهزة المحمولة ومميّزات منتجاتها، حاسوب، شاشة عرض، إنترنت)
أخطّط وأقرّر	<ul style="list-style-type: none">• تصنيف البيانات لتحديد أنواع الأجهزة المحمولة الذكيّة.• تحديد كفاءة الأجهزة الذكيّة وفق عناصرها (سرعة المعالج، وحجم الذاكرة العشوائيّة RAM، وحجم الذاكرة الداخليّة والخارجيّة، وحجم الشاشة ودقّتها، ودقّة الكاميرا، وقدرة البطاريّة).• المقارنة بين عناصر الهواتف المختلفة.• إعداد خطة العمل التي تتضمّن موعد تسليم البيانات اللازمة للزائرين.	<ul style="list-style-type: none">• تعلم تعاونيّ/ مجموعات.• حوار ومناقشة.	<ul style="list-style-type: none">• وثائق (طلب المدرسة الزائرة)، مكتبة المدرسة، نشرات، مجلات، كتالوجات عن خصائص الأجهزة النقالة.

<ul style="list-style-type: none"> • الأجهزة الخليوية. • البيانات التي تم جمعها. • الوثائق الخاصة بكتالوجات الأجهزة النقالة. • قرطاسية. 	<ul style="list-style-type: none"> • التعلم التعاوني. • حوار ومناقشة. 	<ul style="list-style-type: none"> • تفحصُ الأجهزة الذكية لتحديد مواصفاتها من حيث: الشركة المصنّعة، وسرعة المعالج، وحجم الذاكرة العشوائية RAM، وحجم الذاكرة الداخلية والخارجية، وحجم الشاشة ودقتها، ودقة الكاميرا، وقدرة البطارية...). • كتابة أنواع الأجهزة الذكية المحمولة، ومميزات كل نوع. • تسجيل بيانات الأجهزة الذكية في جداول تتضمن العناصر المُحدّدة للكفاءة. • مقارنة بين أفضل عنصر على مستوى الشركة الواحدة والشركات الأخرى. • مقارنة بين أفضل المواصفات لأفضل عنصر بين الشركات المختلفة. • اختيار الهاتف المحمول الذكي الأفضل بناءً على مواصفات الجهاز واحتياجات الزبون. 	<p>أنفذ</p>
<ul style="list-style-type: none"> • نشرات، مجلات، كتالوجات عن خصائص الأجهزة النقالة. • الخطة المعدة سابقاً. 	<ul style="list-style-type: none"> • حوار ومناقشة. 	<ul style="list-style-type: none"> • التحقّق من تحديد أنواع الأجهزة المحمولة الذكية. • التأكد من تصنيف عناصر الأجهزة الذكية وفق كفاءتها. • التحقّق من معرفة الشركات المصنّعة للهواتف النقالة الذكية. • التحقّق من تحديد كفاءة الأجهزة المحمولة الذكية. 	<p>أتحقّق</p>
<ul style="list-style-type: none"> • نشرات، مجلات، كتالوجات عن خصائص الأجهزة النقالة. • حاسوب. • شاشة عرض. • قرطاسية. 	<ul style="list-style-type: none"> • حوار ومناقشة. • تعلم تعاوني/ مجموعات عمل. 	<p>أوثق البيانات التي تتعلق بـ:</p> <ul style="list-style-type: none"> - أنواع الأجهزة المحمولة الذكية. - الشركات المنتجة لها. - العناصر التي تُحدّد كفاءة أداء الأجهزة المحمولة الذكية. • إعداد الملخص الذي يتضمن أنواع الأجهزة المحمولة الذكية، والشركات المنتجة لها، والعناصر التي تُحدّد كفاءتها. • إعداد العروض التقديمية. • إعداد ملفّ بالحالة (معرفة العناصر الرئيسية التي تحدّد كفاءة الأجهزة النقالة الذكية). 	<p>أوثق وأقدّم</p>
<ul style="list-style-type: none"> • نماذج للتقويم معدّة سابقاً. 	<ul style="list-style-type: none"> • حوار ومناقشة. 	<ul style="list-style-type: none"> • مقارنة معرفة الطلبة الزائرين عن أنواع الأجهزة المحمولة، وعناصر أداء الكفاءة، والشركات المصنّعة قبل الزيارة وبعدها. 	<p>أقوم</p>

أذكرُ المواصفات التي أفضلها في الهاتف الذكي.



معرفة العناصر الرئيسية التي تحدّد كفاءة الأجهزة النقالة الذكية

نشاط نظري (1):

أذكرُ أمثلة لشركات تنتج ما يلي :

- ساعة ذكيّة
- نظارة ذكيّة
- سبورة ذكيّة
- تابلت (Tablet).

تتميّز الأجهزة الذكية (SMART DEVICES) بكونها توفر مجموعة من التطبيقات والحلول التي تُلبّي احتياجات مختلف أنواع المستخدمين، وفئاتهم العمرية المختلفة. ولم يعد حملُ تلك الأجهزة لإجراء المكالمات فقط، بل أصبحت بوجود التطبيقات المتعدّدة أجهزة لتصفّح الإنترنت، والتقاط الصّور، وتشغيل ملفات الصوت والفيديو، كما تُستخدم أيضاً منصّات ألعاب وغيرها.

ونظراً لحدائثة هذه الأجهزة، والتطوّر السريع الذي تحظى به، وتطبيقاتها كذلك فإنّه لا يوجد تعريف موحد متفق عليه للأجهزة الذكية بين الشركات المصنّعة، أو حتى على مستوى مراكز الأبحاث العلميّة، إضافة الى أنّ كلمة "ذكيّة" تستخدمها الكثير من الشركات لأغراض تسويقية؛ كالسيارة الذكيّة، والمنزل الذكيّ، والمدرسة الذكيّة، ومع ذلك يمكننا تعريفها بأنّها:

الأجهزة الذكيّة:



أجهزة إلكترونية تفاعلية تعمل بواسطة نظام تشغيل، تستجيب لما يوجّه إليها مستخدموها من أوامر بسيطة، وتساعد على القيام بالأنشطة اليومية، كما أنّ لديها القدرة على الاتّصال بخدمة الإنترنت، وتصفّح المواقع والبريد الإلكترونيّ، ومواقع التّواصل الاجتماعيّ وذلك باستخدام الشبكات اللاسلكية، أو خدمات شركات الاتّصالات. (انظر إلى الشكل 1-15).

◀ أنواع الأجهزة الذكيّة:

انظرُ إلى متاجر الأجهزة الذكيّة فأرى مدى التنوّع الكبير في أحجامها وأشكالها وأنواعها، وكذلك الاختلاف في خصائصها ومميزاتها ومهمّاتها. ورغم هذا التنوّع، إلّا أنّه يمكن تصنيفها بناءً على بعض الخصائص الرئيسيّة التي تشترك أو تختلف بها، من حيث الأحجام أو غرض الاستعمال:



شكل (1 - 16): جهاز متحوّل

• الأجهزة المحمولة صغيرة الحجم، وتشمل الهواتف الذكيّة، والبطاقات الذكيّة، والأجهزة اللوحيّة.

• الأجهزة المحمولة متوسطة الحجم، وتشمل الحواسيب المحمولة، والمتحوّلة التي تعمل باللمس (الشكل 1-16).

• الأجهزة كبيرة الحجم كالسبّورة الذكيّة (الشكل 1-17).



شكل (1 - 17): سبورة ذكية

◀ الهواتف الذكيّة:

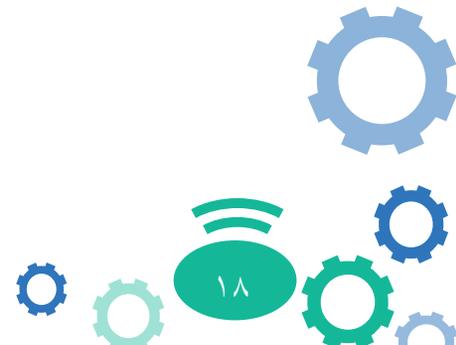
مع بداية انتشار الهواتف النّقالة، ظهرت خدمات الاتّصال والرسائل النصّيّة، ومع التطوّر التقنيّ السريع تحوّلت الهواتف إلى هواتف أكثر ذكاءً، وذلك لاحتوائها على مميّزات عديدة جعلت منها حواسيب مصغّرة لها معالجاتها الخاصّة، وذاكرتها المؤقّته والدائمة، كما تمّ تزويدها بمساحات تخزين ثابتة وأخرى قابلة للتوسّع وفق حاجة المستخدم، كما أنّ أهمّ ما يميّزها هو وجود أنظمة تشغيل خاصّة بها كنظام التشغيل Android ونظام iOS. ولا تقتصر هذه الهواتف على عمليّات استقبال وإرسال المكالمات والرسائل، كما في الهواتف النّقالة القديمة، القيام بعمليّات التصفّح على شبكة الإنترنت، وتحميل وتشغيل العديد من التطبيقات المفيدة.

نشاط نظري (2):



ابحث في مجموعات عن وحدة قياس أداء كلّ من مكوّنات الهاتف الآتية:

وحدة قياس الأداء	المكوّن
	المعالج
	حجم الذاكرة العشوائيّة RAM
	حجم الذاكرة الداخليّة
	الكاميرا
	دقّة الشّاشة
	دقّة الكاميرا
	قدرة البطارية



نشاط (3) للاطلاع:



بالاستعانة بالشبكة العنكبوتية، أملأ الجدول الآتي بمواصفات أجهزة هواتف محمولة مختلفة:

الرقم	الشركة المُصنِّعة	اسم الهاتف	سرعة المعالج	حجم الذاكرة العشوائية RAM	حجم الذاكرة الداخلية	دقَّة الكاميرا	دقَّة الشاشة	قدرة البطارية	السَّعر
1									
2									
3									
4									
5									

نشاط للاطلاع (4):



اعتماداً على بيانات الجدول في نشاط (3)، أجب عما يأتي:

الهاتف المحمول صاحب أسرع معالج.
الهاتف المحمول صاحب أكبر ذاكرة عشوائية (RAM)
الهاتف المحمول صاحب أكبر حجم ذاكرة داخلية.
الهاتف المحمول صاحب أفضل دقَّة كاميرا.
الهاتف المحمول صاحب أفضل دقَّة شاشة.
الهاتف المحمول صاحب أفضل قدرة بطارية.

أمور علينا مراعاتها عند شراء هاتف ذكي:



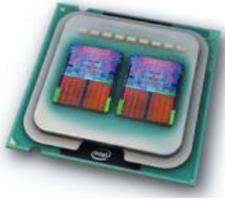
◀ نظام التشغيل:

تختلف أنظمة التشغيل وفق الشركة المصنِّعة، ولكلّ نظام مزايا تختلف عن الآخر، ولكلّ مُستخدم ميول يختلف عن ميول الآخرين؛ لذلك عند شرائي الهاتف الذكيّ أختار نظام التشغيل الذي أراه مناسباً لي.

◀ الشاشة:

بعض المستخدمين يفضّلون الشاشات ذات الحجم الكبير، والبعض الآخر يميل إلى الأحجام الأصغر حجماً، تبعاً لأذواقهم وحاجاتهم وإمكاناتهم (شكل 1-18): صورة تعبيرية لشاشة هاتف ذكيّ

◀ المعالج:



الشكل (19-1): معالجات الهواتف الذكية

يُعدُّ المُعالِجُ النَّواةُ الرَّئيسيَّةُ لأيِّ جِهَازٍ إلكترونيٍّ رقميٍّ، ويكاد لا يخلو أيُّ جِهَازٍ رقميٍّ مهما صغر أو كبر من المُعالِجِ. في السَّناتِ الأَخيرةِ، انتشرت معالجات ذات أنوية متعدِّدة (ثنائيَّة النواة، رباعيَّة النواة، ...). والفائدة الحقيقيَّة من ذلك تكمن في قدرتها، وتمكُّننا من تشغيل عدَّة برامج، والعمل عليها بالوقت نفسه دون تأثُّر أحدهما بالآخر؛ لأنَّ كلَّ نواة ستنفِّذ برنامجاً؛ لذلك عند شراء الهواتف الذكية يتوجَّب علينا أيضاً مراعاة عدد الأنوية في المُعالِجِ (شكل 19-1).

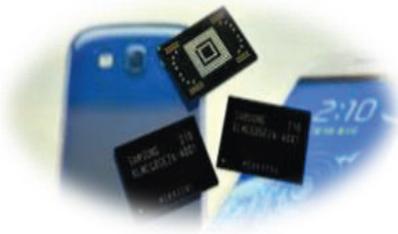
◀ الكاميرا:



الشكل (21-1): كاميرا هاتف ذكيّ

أصبحت الكاميرا في الهواتف الذكية عاملاً مهمّاً؛ لأنَّ غالبيَّة المستخدمين يستخدمون كاميرات الهاتف لتوثيق لحظات مهمّة في حياتهم بشكلٍ عفويٍّ، ولا يفكِّرون في استخدام الكاميرات الرقميَّة؛ لأنَّهم ببساطة يملكون هاتفاً ذا كاميرا عالية الدقَّة، فعند شرائي الهاتف أنتبه جيِّداً لدقَّة الكاميرا ووضوحها (شكل 21-1).

◀ الذاكرة العشوائيّة RAM:



الشكل (20-1): ذاكرة عشوائيّة RAM

مع تطوُّر أنظمة التشغيل والتطبيقات، أصبحت الأنظمة الحديثة والتطبيقات تستهلك نسبة كبيرة من الذاكرة العشوائيّة RAM؛ لذلك ومع مرور الوقت على استخدام الهاتف، فإنَّ أداء الجهاز يضعف، وقد يتوقَّف أحياناً عن العمل بشكلٍ مفاجئ. وبالتالي، فقبل شراء الهاتف الذكيّ، يجب الاطِّلاع على حجم الذاكرة العشوائيّة RAM بعناية، بحيث لا يقلُّ حجمها عن 3 جيجابايت (شكل 20-1).

◀ الذاكرة الداخليَّة والخارجيّة:



الشكل (22-1): ذاكرة خارجيّة

يُنصَحُ بالتأمُّن والتفكير جيِّداً عند شراء هاتف ذكيٍّ، والانتباه إلى حجم الذاكرة الداخليَّة، خاصَّة في الأجهزة التي لا تسمح بإضافة ذاكرة خارجيّة، مع مراعاة أنَّ الأنظمة الحديثة تستهلك حجماً كبيراً من المساحة التخزينيّة، وأنني قد أقوم بتشغيل تطبيقات متنوّعة على هاتفي، إضافة إلى استخدام الكاميرا لتصوير مقاطع فيديو التي قد تكون كبيرة نسبياً، وتحتاج إلى مساحة تخزين عالية؛ لذلك يُنصَحُ بالاعتماد على ذاكرة التخزين عن 16 جيجابايت للأجهزة التي تدعم أو تسمح باستخدام الذاكرة الخارجيَّة، وأن لا تقلَّ عن 32 جيجابايت للأجهزة التي لا تدعم، أو لا تسمح باستخدام الذاكرة الخارجيَّة (شكل 22-1).



البطارية: ◀



الشكل (1-23): بطارية هاتف ذكي

من الضروري ألا تقل قدرة البطارية عن ٣٠٠٠ ميلي امبير في الساعة نظراً إلى أن شاشة الهاتف تستهلك كمية كبيرة من الطاقة خاصة إذا كانت دقة الشاشة عالية.

أنتبه واختار جهازي بعناية، وأفكر جيداً ولا أتسرع. ⚠️

نشاط نظري (5):



أبحث عن المواصفات الأمثل المفروض توفرها في الهاتف المحمول لكل من الأشخاص الآتية: رجل مسن، شاب هوايته ألعاب إلكترونية، طالب مدرسة، شاب هوايته التصوير.

نشاط للاطلاع (6):



أناقش مع زملائي، لماذا تقوم بعض الشركات بطرح أجهزتها بنواة واحدة أو نواتين فقط، رغم توفر القدرة والإمكانيات لتصنيع معالجاتها بأنوية أكثر؟

نشاط نظري (7):



أبحث عن الفرق بين الذاكرة الداخلية والخارجية والعشوائية (RAM).

الأسئلة



1. أذكر أهم العناصر التي تُحدّد كفاءة الهواتف النقالة الذكية.
2. أذكر مواصفات هواتف تم إصدارها مؤخراً، واسم الشركة المصنعة لها.

الموقف التعليمي التعلّمي (3 - 1):

معرفة أنظمة التشغيل المختلفة المستخدمة في الهواتف النقّالة الذكيّة وتطبيقاتها.

وصف الموقف: استضاف طلبة إحدى المدارس الأكاديمية مجموعة من طلبة مدرسة مهنيّة؛ بهدف التعرّف إلى أنظمة التشغيل المختلفة المستخدمة في الهواتف النقّالة الذكيّة وتطبيقاتها.

العمل الكامل			
خطوات العمل	وصف الموقف الصّفيّ	المنهجية الاستراتيجية (التّعلم)	الموارد وفق الموقف الصّفيّ
أجمع البيانات وأحلّها	<p>جمع بيانات من طلبة المدرسة الأكاديمية عن:</p> <ul style="list-style-type: none"> • معرفتهم السابقة بأنظمة التشغيل المختلفة، المستخدمة في الهواتف النقّالة الذكيّة وتطبيقاتها. <p>جمع البيانات عن:</p> <ul style="list-style-type: none"> • أنظمة التشغيل الخاصّة بالهواتف الذكيّة. • مميّزات كلّ نوع من أنظمة التشغيل. • متاجر التطبيقات الخاصّة بكلّ نظام تشغيل. • التطبيقات المتاحة في كلّ متجر. 	<ul style="list-style-type: none"> • حوار ومناقشة. • التّعلم التعاونيّ / مجموعات. • البحث العلميّ • زيارة ميدانيّة. 	<ul style="list-style-type: none"> • وثائق (طلب المدرسة الزائرة، مكتبة الجامعة، نشرات، مجلات، كتالوجات عن أنظمة تشغيل الأجهزة المحمولة ومميزاتها). • هواتف محمولة لأنظمة تشغيل مختلفة. • التكنولوجيا (روابط إلكترونية متنوعة من مصادر موثوقة حول أنظمة تشغيل الهواتف المحمولة ومميزاتها، وطريقة التعامل معها، حاسوب، شاشة عرض، إنترنت).
أخطّط وأقرّر	<ul style="list-style-type: none"> • تصنيف الهواتف المحمولة حسب أنظمة تشغيلها. • تحديد مميّزات أنظمة التشغيل المختلفة. • تحديد التطبيقات المتوفرة لكلّ نظام تشغيل. • تحديد متجر التطبيقات الخاصّ بكلّ نظام تشغيل. • إعداد خطة العمل التي تتضمّن موعد تسليم البيانات اللازمة للزائرين. 	<ul style="list-style-type: none"> • تعلم تعاونيّ / مجموعات. • حوار ومناقشة. 	<ul style="list-style-type: none"> • وثائق (طلب المدرسة الزائرة، مكتبة الجامعة، نشرات، مجلات، كتالوجات عن أنظمة تشغيل الأجهزة المحمولة ومميزاتها).



<ul style="list-style-type: none"> • هواتف محمولة . • البيانات التي تمّ جمعها . • الوثائق الخاصة بكتالوجات أنظمة التشغيل . • إنترنت . • قرطاسية . 	<ul style="list-style-type: none"> • تعلّم تعاوني/ مجموعات • حوار • ومناقشة . 	<ul style="list-style-type: none"> • تفحص الهواتف الخليوية لتحديد نظام التشغيل الذي يعمل عليه، ورقم الإصدار . • التعامل مع بيئة نظم التشغيل المختلفة (ضبط الوقت، والتاريخ، واللغة... الخ) • كتابة أسماء التطبيقات المتوفرة حالياً على الأجهزة الخليوية . • حذف وإضافة تطبيقات من المتجر الخاص بنظام التشغيل . • المقارنة بين أنظمة التشغيل المختلفة . 	<p>أنفذ</p>
<ul style="list-style-type: none"> • هواتف محمولة . • نشرات، مجلات، كتالوجات • عن أنظمة التشغيل الخاصة بالهواتف المحمولة . • إنترنت . • الخطة المعدة سابقاً . 	<ul style="list-style-type: none"> • تعلم تعاوني/ مجموعات • حوار • ومناقشة . 	<ul style="list-style-type: none"> • التحقّق من تحديد أنظمة التشغيل للأجهزة المختلفة . • التأكد من مقدرة الطلبة الزائرين على التعامل مع بيئة نظام التشغيل . • التحقّق من معرفة التطبيقات المتوفرة في الهاتف المحمول واستخدامها . • التحقّق من حذف وإضافة تطبيقات من المتجر الخاصّ بكلّ نظام التشغيل . • التحقّق من تحديد مميّزات كلّ نظام تشغيل . تحديد مميّزات كلّ نظام تشغيل . 	<p>أتحقّق</p>
<ul style="list-style-type: none"> • نشرات، مجلات، كتالوجات • عن انظمة التشغيل الخاصة بالهواتف المحمولة . • حاسوب . • شاشة عرض . • قرطاسية . 	<ul style="list-style-type: none"> • حوار • ومناقشة . • تعلم تعاوني/ مجموعات • عمل 	<p>توثيق البيانات التي تتعلق بـ:</p> <ul style="list-style-type: none"> • أنظمة التشغيل الخاصة بالهواتف الذكية . • مميزات كلّ نوع من أنظمة التشغيل . • متاجر التطبيقات الخاصة بكلّ نظام تشغيل . • التطبيقات المتاحة في كلّ متجر . - إعداد الملخص الذي يتضمّن أنواع أنظمة تشغيل الهواتف المحمولة، ومميّزات كلّ منها والتطبيقات التي توفّرها في متاجرها . - إعداد العروض التقديمية . - إعداد ملفّ بالحالة (معرفة أنظمة التشغيل المختلفة المستخدمة في الهواتف النقالة الذكية وتطبيقاتها) . 	<p>أوثق وأقدم</p>
<ul style="list-style-type: none"> • نماذج للتقويم مُعدّة سابقاً . 	<ul style="list-style-type: none"> • حوار • ومناقشة . 	<ul style="list-style-type: none"> • مقارنة معرفة الطلبة الزائرين عن أنواع أنظمة التشغيل، مميّزات كلّ نوع والشركات المستخدمة لها، وتطبيقاتها المختلفة قبل الزيارة وبعدها . 	<p>أقوم</p>

◀ ما رأيك في المقولة الآتية: " يُعدُّ نظام التشغيل iOS هو الأفضل على الإطلاق؟"

أنظمة التشغيل

نشاط نظري (1):

أعرف أنظمة تشغيل الهواتف الذكية



الشكل (24-1)

نظام تشغيل الهواتف المحمولة (Mobile Operating System)

هو مجموعة من البرمجيات المسؤولة عن إدارة المكونات المادية والتطبيقات في الهاتف الذكي، ويُمثّل وسيطاً بين المستخدم والمكونات المادية (الذاكرة العشوائية RAM، ذاكرة التخزين، الكاميرا،...)، كما يوضّحه (الشكل 24-1)

أهم مهام نظام التشغيل:

- ◀ إدارة الذاكرة.
- ◀ إدارة المعالج.
- ◀ إدارة الملفات.
- ◀ إدارة الجهاز.
- ◀ المساعدة في معالجة الأخطاء.
- ◀ تشغيل وإيقاف الهاتف المحمول.
- ◀ تشغيل البرامج التطبيقية.
- ◀ التنسيق بين التطبيقات والمستخدم.
- ◀ إدارة العمليات التي تقوم بها وحدات الهاتف المحمول.



نشاط نظري (2):



اعتماداً على الأجهزة الخلوية المتوفرة لدى المدرسة، أقوم بتسجيل البيانات الآتية:

الإصدار	نظام التشغيل	اسم الهاتف المحمول

أبرز أنظمة التشغيل الخاصة بالهواتف الذكية:

1. نظام التشغيل أندرويد (Android):



Play Store

الشكل (1-25): ايقونة PLAY STORE

صُممَ نظامُ التشغيل أندرويد في الولايات المتحدة الأمريكية، ثم قامت شركة جوجل بشرائه عام 2005، والعمل على إدارة تطويره بالمشاركة مع 45 شركة مصنعة للأجهزة الذكية، وجعلته مفتوح المصدر، مبني على نواة لينكس، وهو الأكثر انتشاراً بين الأجهزة، حيث يحتل نسبة ٧٥٪ من المبيعات.

يُطلقُ على المتجر الخاص بنظام التشغيل Android: **Google Play Store**. انظر إلى الشكل (1-25).

2. نظام التشغيل (iOS):



App Store

الشكل (1-26): ايقونة APP STORE

نظامُ تشغيلٍ تجاريٍّ للأجهزة الذكية من شركة أبل (Apple)، وأول إصدارٍ له كان عام 2007، وأطلقَ عليه اسم OS X، ثم أعيدَ تسميته في عام 2009 إلى iOS، ويُستخدمُ فقط على أجهزة Apple الذكية، مثل: iPhone، iPad، iPod، ويُعدُّ من أسرة نظام الماك، ويتميزُ بنسبة أمان عالية جداً. يُطلق على المتجر الخاص بنظام التشغيل **iOS App Store**: (انظر إلى الشكل 1-26).

3. نظام التشغيل Windows Phone:



Windows Store

الشكل (1-27): ايقونة WINDOWS STORE

نظامُ تشغيلٍ تجاريٍّ قامت بإنتاجه شركة ميكروسوفت، وهو نظامٌ شبيهٌ بنظام التشغيل ويندوز للحواسيب. النظام تم إنشاؤه في عام ٢٠١١ على أجهزة لوميا، أنشئ بعد نظام الأندرويد و الـ iOS. يُطلقُ على المتجر الخاص بنظام التشغيل ويندوز فون: **Windows Store**. (انظر إلى الشكل 1-27).

نشاط للاطلاع (3):



أضيف اللغات الآتية: العربية، والإنجليزية، والصينية للأجهزة التي تحمل نظام التشغيل الآتي:

Android, iOS, Windows phone

أدوّن ملحوظاتي حول مدى سهولة استخدام كلّ نظام.

نشاط للاطلاع (4):



أحمّل تطبيقات تخدمني في المجالات الآتية:

اسم التطبيق متجر Play Store	اسم التطبيق متجر App Store	مهمّة التطبيق
		إجراء العمليّات الحسابيّة.
		تدوين ملحوظات.
		ترجمة عربي- إنجليزي.
		حجز تذكرة طيران.
		معرفة مواقيت الصلاة.

ما هي تطبيقات الأجهزة الذكيّة:

برامج تعمل على الأجهزة الذكيّة مستغلّة عدداً من إمكانيّاتها، لتقديم خدمة معيّنة للمستخدم.

من مزايا تطبيقات الأجهزة الذكيّة:

- متوفّر على جهاز المستخدم أينما كان.
- التنوّع (تشمل خدمات كثيرة ومتجدّدة تفيد المستخدم).



أهمّ مجالاتها:

- ▶ تطبيقات تعليمية.
- ▶ تطبيقات الكتب.
- ▶ تطبيقات السفر والسياحة.
- ▶ تطبيقات المطاعم.
- ▶ تطبيقات الصحف والإعلام.
- ▶ تطبيقات المال والأعمال.
- ▶ تطبيقات الترفيه.
- ▶ تطبيقات خدماتية.

الأسئلة

1. من مهام نظام التشغيل إدارة الملفات، أوضّح.
2. أذكرُ مزايا التطبيقات الذكية.



أسئلة الوحدة

السؤال الأول :

أختار رمز الإجابة الصحيحة فيما يأتي :

- ١- ما الشركة التي تُطوّر نظام التشغيل أندرويد؟
1. Google .2. Apple .3. Microsoft .4. Sony
- ٢- على أيّ نظام تشغيل تعمل أجهزة الـ iPhone؟
1. Android .2. iOS .3. Windows Phone .4. Dos
- ٣- ماذا يُطلقُ على المتجر الخاص بتطبيقات جوجل؟
1. Google shop .2. App store .3. Google Play Store .4. Google store
- ٤- أيّ من أنظمة التشغيل الآتية تتميز بكثرة شركات الهواتف النقالة التي تدعمها؟
1. Windows phone .2. iOS .3. Blackberry .4. Android
- ٥- ما العنصر الإلكتروني الأكثر أهميّة في تحديد كفاءة الهاتف المحمول؟
1. الذاكرة العشوائيّة .2. المعالج .3. الكاميرا .4. الشاشة
- ٦- أيّ من العبارات الآتية ليست صحيحة؟
 1. نظام التشغيل هو حلقة وصل بين التطبيقات والعتاد.
 2. الهواتف المحمول الواحد يُمكنُ أن يحتوي على أكثر من نظام تشغيل.
 3. بعض الهواتف المحمولة لا تتوافق بشكل كامل مع نظام تشغيلها.
 4. كلُّ نظام تشغيل له تطبيقاته الخاصّة.

السؤال الثاني :

أعرّف كلاً من :

- ◀ عمليّة الاتصال .
- ◀ تطبيقات الأجهزة الذكيّة .
- ◀ الأجهزة الذكيّة .
- ◀ نظام تشغيل الهواتف الذكيّة .

السؤال الرابع :

اذكر الفرق بين كل من الذاكرة العشوائية RAM والذاكرة التخزينية من حيث :

- ◀ حجم الذاكرة
- ◀ زمن الوصول
- ◀ الاحتفاظ بالبيانات
- ◀ الوظيفة



السؤال الخامس:

أرتب وسائل الاتصال الآتية وفق تاريخ ظهورها من الأقدم إلى الأحدث:
(التلفاز، الإنترنت، الحاسوب، الحمام الزاجل، الكلام الشفوي، الهاتف اللاسلكي، الهاتف السلكي).

السؤال السادس:

أوضح الأمور التي علينا مراعاتها عند شراء هاتف ذكي.

مشروع الوحدة



مقارنة بين كل من أنظمة التشغيل الثلاثة: iOS، Android، Windows Phone من حيث:

- الشركة المطورة.
- متجر التطبيقات الرسمي.
- القدرة على عمل نسخ احتياطي كامل محلي.
- القدرة على أخذ صورة للشاشة، كيف؟
- المتصفحات الرئيسية المتوفرة.
- دعم عمليات قص ونسخ ولصق.
- دعم الطابعات.
- تطبيقات تحرير وكتابة النصوص.



أناقش:

التفكير المنطقي
أساس حل المسائل



يُتوقَّع من الطلبة بعد دراسة هذه الوحدة، والتَّفاعُلِ مع أنشطتها أن يكونوا قادرين على
بناء برنامج تطبيقي من خلال الآتي:

- ✓ توظيف الخوارزميات لإيجاد حلول برمجية لمشاكل حياتية.
- ✓ توظيف مخطّط سير العمليّات لإيجاد حلول برمجية لمشاكل حياتية.



الكفايات المهنية

الكفايات المهنية المتوقعة من الطلبة امتلاكها بعد انتهاء دراسة هذه الوحدة، والتفاعل مع أنشطتها:
أولاً: الكفايات الحرفية.

- ▶ امتلاك أسس التفكير المنطقي.
- ▶ القدرة على إيجاد حلول بمرجعية لمشاكل حياتية من خلال توظيف الخوارزميات ومخططات سير العمليات.

ثانياً: الكفايات الاجتماعية والشخصية.

- ▶ العمل ضمن فريق.
- ▶ تقبل التغذية الراجعة.
- ▶ احترام رأي الآخرين.
- ▶ مصداقية التعامل مع الزبون.
- ▶ حفظ خصوصية الزبون.
- ▶ الاستعداد للاستجابة لطلبات الزبون.
- ▶ التحلي بأخلاقيات المهنة أثناء العمل.
- ▶ التمتع بالفكر الريادي أثناء العمل.
- ▶ استخلاص النتائج ودقة الملاحظة.
- ▶ الاتصال والتواصل الفعال.
- ▶ الاستعداد للاستفسار والاستفادة من ذوي الخبرة.
- ▶ الإفادة من مقترحات الآخرين.
- ▶ امتلاك مهارة التأمل الذاتي.
- ▶ الاستعداد التام لتقبل ملحوظات الزبون وانتقاداتهم.
- ▶ القدرة على الإقناع.

ثالثاً: الكفايات المنهجية.

- ▶ توظيف التعلم التعاوني.
- ▶ القدرة على إدارة الحوار وتنظيم النقاش.
- ▶ امتلاك مهارة البحث العلمي، والقدرة على توظيف أساليبه.
- ▶ القدرة على التخطيط.
- ▶ القدرة على استمطار الأفكار.
- ▶ استخدام المصادر والمراجع المختلفة.
- ▶ توثيق نتائج العمل وعرضها.
- ▶ تقويم النتائج وعرضها.
- ▶ توظيف التكنولوجيا والبرامج الإحصائية وتمثيل البيانات.
- ▶ جمع البيانات وتحليلها وتبويبها.



الموقف التعليمي التعلّمي (2 - 1):

توظيف الخوارزميات لإيجاد حلول برمجية لمشاكل حياتية.

وصف الموقف التعليمي: زار موظف من وزارة المالية الفلسطينية مشغلاً لصناعة البرمجيات لمساعدته في بناء خوارزمية لاحتساب رواتب الموظفين.

العمل الكامل			
خطوات العمل	وصف الموقف الصّفي	المنهجية (استراتيجية التّعلم)	الموارد
أجمع البيانات وأحلّها	<p>جمع البيانات من الموظف عن:</p> <ul style="list-style-type: none"> المعطيات اللازمة لاحتساب راتب الموظف. المعادلة الحسابية الخاصة باحتساب صافي الراتب. الشروط المفروضة على العلاوات والخصومات. استثناءات إن وُجدت. <p>جمع البيانات عن:</p> <ul style="list-style-type: none"> مفهوم الخوارزميات. أنواع الخوارزميات وطريقة كتابتها. 	<ul style="list-style-type: none"> حوار ومناقشة. التعلم التعاوني/ مجموعات. البحث العلمي. 	<ul style="list-style-type: none"> وثائق(طلب الموظف، نشرات، كتب ومصادر متنوعة عن الخوارزميات وأنواعها). الشبكة العنكبوتية .
أخطّط وأقرّر	<ul style="list-style-type: none"> فهم وتحليل المسألة. تحديد المعطيات (اسم الموظف، الشهادة، سنوات الخدمة، المواصلات، عدد الأولاد، الدرجة...). تحديد المخرجات (صافي الراتب). تحديد عمليّات المعالجة اللازمة لاحتساب صافي الراتب (الراتب الأساسي، العلاوات، الخصومات، الضريبة...). تحديد نوع الخوارزمية المطلوبة لتمثيل الحلّ(الخوارزمية التكرارية). إعداد خطة التنفيذ والاتفاق على موعد تسليم خوارزمية الحلّ للوزارة. 	<ul style="list-style-type: none"> تعلم تعاوني/مجموعات. حوار ومناقشة . عصف ذهني/استمطار الأفكار. 	<ul style="list-style-type: none"> وثائق(طلب الموظف، نشرات، كتب ومصادر متنوعة). الشبكة العنكبوتية .

<ul style="list-style-type: none"> • قرطاسية . • الطلب الخاص بالموظف . 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات . • حوار ومناقشة . • عصف ذهني/ استمطار الأفكار . 	<ul style="list-style-type: none"> • استخدام لغة مناسبة للتعبير عن التفكير المنطقي . • بناء آلية الاحتساب (خوارزمية الحل) باستخدام المعطيات التي تمّ تحديدها (اسم الموظف، الشهادة، سنوات الخدمة، المواصلات، عدد الأولاد، الدرجة...) لإيجاد صافي الراتب . • تطبيق الخوارزمية التكرارية لإيجاد صافي الراتب لموظفي السلطة . 	<p>أُنفَذ</p>
<ul style="list-style-type: none"> • الطلب الخاص بالموظف . • الخطة المعدة سابقاً . • الشبكة العنكبوتية . • قرطاسية . 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات . • حوار ومناقشة . 	<ul style="list-style-type: none"> • التحقُّق من استخدام لغة تعبيرية واضحة، متسلسلة، منطقية في بناء الخوارزمية . • التحقُّق من تحديد المعطيات، وعمليات المعالجة والمخرجات . • التحقُّق من تسلسل خطوات خوارزمية الحل . • التحقُّق من صحّة خوارزمية الحل . • اكتشاف الأخطاء وتصويبها واقتراح البدائل . 	<p>أُتَحَقَّقُ</p>
<ul style="list-style-type: none"> • قرطاسية . • حاسوب . • شاشة عرض . 	<ul style="list-style-type: none"> • حوار ومناقشة . • تعلم تعاوني/ مجموعات عمل . 	<p>أوثق البيانات التي تتعلق بـ:</p> <ul style="list-style-type: none"> • مفهوم الخوارزميات . • أنواع الخوارزميات : • الخوارزميات التسلسلية . • الخوارزميات ذات الفرع . • الخوارزميات التكرارية . - إعداد العروض التقديمية . - توثيق آلية بناء خوارزمية الحل بناءً على التفكير المنطقي . - إعداد ملفّ بالحالة (احتساب رواتب الموظفين بالاعتماد على عناصر محددة) 	<p>أوثق وأقدم</p>
<ul style="list-style-type: none"> • نماذج للتقويم معدة سابقاً . • طلب الوزارة . 	<ul style="list-style-type: none"> • حوار ومناقشة . 	<ul style="list-style-type: none"> • نجاعة الخوارزمية في احتساب رواتب الموظفين، وإمكانية تطبيقها في مواقف جديدة للغرض نفسه . • رضا الوزارة عن الخوارزمية . 	<p>أُقَوِّمُ</p>



أذكر اللغات المستخدمة في صياغة الخوارزمية.



توظيف الخوارزميات لإيجاد حلول برمجية لمشاكل حياتية

نشاط نظري (1):

أزمة عائلة

تعليمات :



شكل (2 - 1) : أزمة عائلة

هنالك عائلة مكونة من 5 أشخاص، بحوزتهم مصباح واحد فقط، وتريد هذه العائلة العبور للجهة الأخرى في ظلام الليل (انظر إلى شكل 1-2).

الشروط والمعوقات:

- المصباح يعمل لمدة 29 ثانية فقط.
- الجسر لا يحمل إلا شخصين على الأكثر، بشرط أن يعود شخص منهما مع المصباح، لإيصاله لباقي أفراد العائلة.
- يستغرق كل شخص زمناً معيناً (1 ثانية، 3 ثواني، 6 ثواني، 8 ثواني، 12 ثانية)، ليقطع المسافة وفق سرعته، كما هو مفصّل في شكل 1-2.

توجيهات للحل:

1. أنقل الشخصين الأسرعين إلى المنطقة B مع المصباح.
2. أنقل المصباح مع أسرع الموجودين في المنطقة B إلى المنطقة A
3. أنقل الشخصين الأبطأ سرعةً من المنطقة A إلى المنطقة B مع المصباح.
4. أنقل المصباح مع أسرع الموجودين في المنطقة B إلى المنطقة A
5. أنقل الشخصين الأسرعين إلى المنطقة B مع المصباح.
6. أنقل المصباح مع أسرع الموجودين في المنطقة B إلى المنطقة A
7. أنقل الشخصين المتبقيين من المنطقة A إلى المنطقة B

اعتماداً على التوجيهات السابقة، أكتب الخطوات اللازمة لنقل العائلة من المنطقة A إلى المنطقة B:

1. أنقل الشخصين الأسرعين إلى المنطقة B : صاحب الزمن ١ وصاحب الزمن ٣.
2. أنقل أسرع الموجودين في المنطقة B إلى المنطقة A ،ومعه المصباح: صاحب الزمن ١.
3. _____
4. _____
5. _____
6. _____
7. _____

الأسئلة

١. ما الغاية من نقل الأشخاص الأسرع، ثم الأبطأ؟
٢. لماذا أنقل المصباح من المنطقة B إلى المنطقة A مع أسرع الموجودين في المنطقة B؟
٣. هل ينجح هذا الإجراء مع عائلة أخرى مكوّنة من خمسة أشخاص (سرعاتهم: 1، 2، 3، 5، 8) للوصول خلال فترة زمنية لا تتجاوز 91 ثانية؟ كيف؟

التفكير المنطقي :

عندما تواجهنا مسائل أو مشكلات نحتاج إلى حلّها في حياتنا اليومية فإننا نحتاج إلى فهم المشكلة وتحديد تفاصيلها، ومن ثمّ جمع المعلومات عنها، واختيار أفضل طرق الحلّ لاتباعها، فالتخطيط والتفكير لحلّ المشكلات، والتعامل معها يحقق لنا النجاح والتغلب على المشكلات في حياتنا اليومية.

إنّ الحاسوب لا يستطيع حلّ جميع المسائل أو المشكلات التي تواجهنا في حياتنا اليومية، وإن كانت يسيرة في نظرنا، لتحديد الزيّ المفضّل لديك أو لدى زميلك، أو اتخاذ قرارٍ ما، فمثل هذه المسائل لا يمكن للحاسب أن يحلّها، ولكنه عندما يقوم بحلّ المسائل المتعلقة بالأرقام ومعالجتها، أو تحرير النصوص، أو معالجة الصور والأصوات، فهذا يُعدّ مجالاً خصباً للحاسب لا تُجاربه في ذلك أيّة آلة أخرى.

فالتفكير المنطقيّ هو التفكير الذي يُمارَس عند محاولة معرفة الأسباب والعِلل والمشاكل التي تواجه الإنسان، وطرق التخلُّص منها باختيار أفضل البدائل.



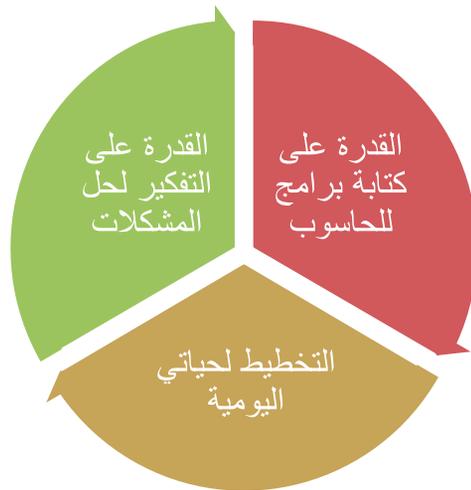
◀ هناك نوعان رئيسيان من التفكير المنطقي:

1. التفكير الاستنباطي (الاستدلالي): هو استنتاج محدد من قاعدة عامّة، يكون هذا الاستنتاج صحيحاً إذا كانت القاعدة العامة صحيحة، مثال: عندما تمطر السماء، تتعرض الأشياء الموجودة في الهواء الطلق للبلل. والعشب موجود في الهواء الطلق، وبالتالي: عندما تمطر السماء، سيتعرض العشب للبلل. ويمكن استنباط أكثر من نتيجة من القاعدة العامّة. مثال آخر: إذا كانت $s = v$ ، حيث $v = c$ ، فإن: $s = c$.

2. التفكير الاستقرائي: وهو الوصول الى قاعدة عامة من خلال استنتاجات محدّدة، وليس بالضرورة أن تكون صحيحة. مثلاً: ألبرت آينشتاين وهو في الخامسة عشرة من عمره، عندما لاحظ تحرك البوصلة أصبح مفتوناً بفكرة أن شيئاً غير مرئي في الفضاء حول إبرة البوصلة يتسبب في تحريكها، هنا اكتشف أن هناك كائناً غير مرئي يحرك إبرة البوصلة، وهذه تُعدّ قاعدة عامّة، في حين اكتشفها من ملاحظته تحركها دون أن يحركها أحد. هذه القاعدة العامّة التي توصل لها ليس لها أساس من الصحة. مثال آخر: أردت شراء هاتف ذكي، وبعد تنقلي بين أكثر من محلّ خاصّ ببيع الهواتف، وجدت أن أكثر الاجهزة المعروضة للبيع من نوع A و B، فاكشفت من ذلك أن أفضل الأجهزة الموجودة في السوق هي A و B، فالقاعدة العامّة التي توصلت إليها هي: " أن أفضل الأجهزة الموجودة في السوق هي A و B"، وهذه القاعدة تتحمّل الصواب أو الخطأ.

الهدف من تعلّم صياغة حلّ المسائل:

عندما أتعلّم وأتدرّب على صياغة حلّ المسائل بواسطة الحاسوب، فإنّ هذا لا يعني أن الفائدة تقتصر على المسائل الحسائيّة والمنطقيّة فحسب، بل أهداف من ذلك أن أتعلّم ما يأتي (شكل 2-2):



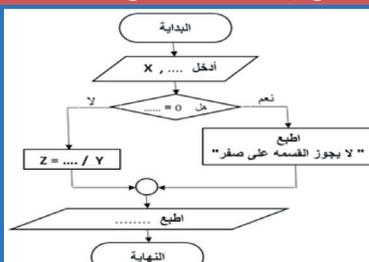
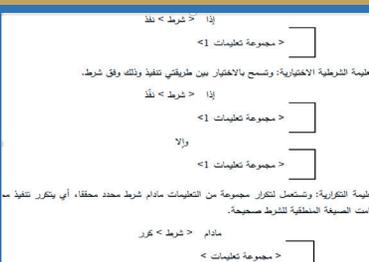
شكل (2 - 2)

خطوات حلّ المسائل

حلّ المسائل مهارة تُكتسب كغيرها من المهارات الأخرى بالتدرّب عليها. ولحلّ المسائل أو المشكلات بواسطة الحاسوب (لتكتمل في النهاية على شكل برنامج يستطيع الحاسب فهمه والتعامل معه)، فإنّ هناك خطوات ومراحل يجب اتّباعها في حلّ تلك المسائل، وهي:

◀ أولاً: صياغة حلّ المسألة.

المقصود بصياغة الحلّ هو تحديد الخطوات المتّبعة للوصول إلى الحلّ لضمان صحّة الحلّ، وتتكوّن هذه الصياغة من ثلاث خطوات أساسية، هي:

<p>• التمثيل البياني للخوارزمية عن طريق مخطط سير العمليات</p> 	<p>• كتابة الخوارزمية والخطوات المنطقية للحل</p> 	<p>فهم المسألة وتحديد عناصرها</p> 
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------

شكل (2 - 3)

◀ ثانياً: كتابة البرنامج وتنفيذه.

تتكوّن هذه المرحلة من ثلاث خطوات أساسية، هي:

<p>• كتابة البرنامج بواسطة إحدى لغات البرمجة من قبل المبرمج</p> <pre>class Person : Eatble { override fun vegetables(name: String): String { return "eat \$name" } override fun fruits(name: String): String { return "eat \$name" } override fun animal(name: String): String { return "eat \$name" } } fun main(args: Array<String>) { val Saed = Person() Saed.vegetables(name: "Tomato") Saed.fruits(name: "Banana") Saed.animal(name: "Chicken") }</pre>	<p>• ترجمة البرنامج إلى لغة الآلة وتنفيذه، وهذا هو دور الحاسب الآلي</p> 	<p>• اختبار البرنامج وتصحيح الأخطاء</p> 
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------

شكل (2 - 4)



◀ ثالثاً: فهم المسألة وتحديد عناصرها:

لا يمكن للمرء حلّ مسألة ما لم يكن متأكّداً من فهمها بشكل كامل، وكما يقال: « فهم السؤال نصف الجواب ». المقصود هنا بفهم المسألة وتحليل عناصرها، أن نعرف ماذا نريد بالضبط من البرنامج، وأن نحدّد العناصر الأساسيّة لحلّ المسألة، كما في الشكل 2-5:



شكل(2-5): عناصر حلّ المسألة

- مخرجات البرنامج: النتائج والمعلومات المراد التوصل إليها عند حلّ المسألة.
- مدخلات البرنامج: المدخلات والبيانات اللازم الحصول عليها لمعرفة النتائج والمخرجات.
- عمليات المعالجة: العمليّات الحسابيّة والخطوات المنطقيّة التي نقوم بإجرائها.

ملحوظة:

تتمّ صياغة نتاج التفكير المنطقيّ لمسألةٍ ما على شكل خوارزميّة، توضّح طريقة الحلّ؛ وذلك لمساعدتي في حلّ المواقف المشابهة.

الخوارزمية هي:

مجموعة من الخطوات الرياضيّة والمنطقيّة والمتسلسلة اللازمة لحلّ مشكلةٍ ما. سُمّيت الخوارزميّة بهذا الاسم نسبةً إلى العالم المسلم (أبو جعفر محمد بن موسى الخوارزمي)، حيث استخدم المفهوم في الرياضيات للدلالة على مجموعة من الخطوات الرياضيّة لحلّ مسألة معيّنة.

أنواع الخوارزميات:

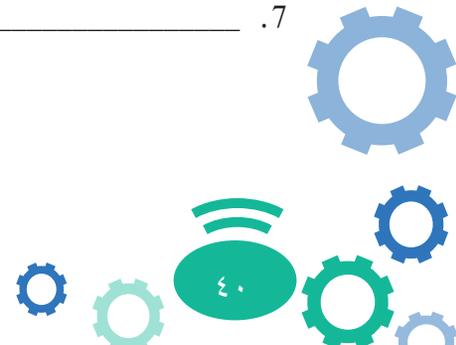
1. الخوارزميات المتسلسلة Sequential Algorithms
2. الخوارزميات ذات التفريع Branched Algorithms
3. الخوارزميات التكراريّة Loop Algorithms

- مجموعة من التعليمات المتسلسلة من بداية الخوارزمية حتى نهايتها؛ (أي خالية من التفرعات والتكرارات).
- مثال: خوارزمية الحل لإيجاد متوسط علامات ثلاث مواد دراسية:
- المعطيات: العلامات الثلاث (ع1، ع2، ع3)
 - المخرجات: متوسط العلامات.
 - المعالجة: متوسط العلامات = مجموع العلامات ÷ عدد العلامات
 - خوارزمية الحل:
1. البداية.
 2. اقرأ الدرجات الثلاث: ع1، ع2، ع3
 3. احسب متوسط الدرجات الثلاث من المعادلة: $م = (ع1 + ع2 + ع3) ÷ 3$
 4. اطبع نتيجة المتوسط م
 5. النهاية

نشاط نظري (2):



- أكمل خوارزمية الحل الآتية لصنع كوب من القهوة، ثم أقرن خوارزمية الحل مع باقي زملاءك.
- المعطيات: ماء، قهوة، سكر، كوب، _____
 - المخرجات: فنجان من القهوة.
 - المعالجة: عملية صنع القهوة.
 - خوارزمية الحل:
1. بداية
 2. _____
 3. _____
 4. _____
 5. _____
 6. _____
 7. _____



نشاط نظري (3):



أكمل خوارزمية الحل الآتية بحيث تستقبل درجة الحرارة بالنظام المئوي (Celsius) ، وتحولها إلى النظام الفهرنهايتي (Fahrenheit) ، علماً بأن معادلة التحويل هي: $Fahrenheit = 9/5 \times Celsius - 32$

المعطيات: درجة الحرارة بالنظام المئوي

المخرجات: _____

المعالجة: قانون التحويل

(Fahrenheit= 9/5 × Celsius -32)

خوارزمية الحل:

1. بداية

2. _____

3. _____

4. _____

5. _____

6. _____

7. _____

الخوارزمية ذات التفريع:

2

هي خوارزمية تختبر بعض الشروط، فإذا كانت النتيجة تحقق الشرط تتبع مساراً يحوي مجموعة من التعليمات، وإذا لم تحقق الشرط تتبع مساراً آخر مختلفاً من التعليمات.

مثال: خوارزمية الحل لإيجاد الرقم الأصغر بين رقمين.

المعطيات: الرقم الأول والرقم الثاني (N1, N2)

المخرجات: الرقم الأصغر.

المعالجة: مقارنة الرقمين.

خوارزمية الحل:

1. البداية

2. اقرأ الرقمين الأول والثاني N1, N2

3. اجعل قيمة MIN مساوياً لـ N1

4. إذا كان الرقم N2 أكبر من الرقم MIN ، اذهب إلى الخطوة (6)

5. اجعل قيمة MIN مساوياً لـ N2

6. اطبع MIN

7. النهاية



نشاط نظري (4):



أكمل خوارزمية الحل الآتية لتحديد الرقم المُدخل إذا كان موجباً، أو سالباً، أو صفراً.

المعطيات: الرقم المدخل

المخرجات: موجب أو _____ أو _____

المعالجة: _____

خوارزمية الحل:

1. بداية

2. اقرأ الرقم N

3. _____

4. _____

5. _____

6. _____

7. _____

نشاط نظري (5):



أكمل خوارزمية الحل لإيجاد القيمة المطلقة لـ x بناء على الاقتران الآتي:

$$f(x) = \begin{cases} x, & x \geq 0 \\ -x, & x < 0 \end{cases}$$

المعطيات: X

المخرجات: _____

المعالجة: اقتران القيمة المطلقة

خوارزمية الحل:

1. بداية

2. اقرأ X

3. _____

4. _____

5. _____

6. _____

7. _____





أكمل خوارزمية الحلّ، لإدخال معدّل طالب وإظهار التقدير بناءً على المعطيات في الجدول الآتي:

التقدير	المعدّل
جيد جداً	$16 \leq$ المعدّل
جيد	$14 < \text{المعدّل} \leq 16$
حسن	$12 < \text{المعدّل} \leq 14$
مقبول	$10 < \text{المعدّل} \leq 12$
غير ناجح	$10 > \text{المعدّل}$

المعطيات: معدّل الطالب

المخرجات: _____

المعالجة: عمل مقارنة لمعدّل الطالب

خوارزمية الحلّ:

1. بداية

2. اقرأ المعدّل م

3. _____

4. _____

5. _____

6. _____

7. _____

أسجل أحد الحلول الصحيحة الأخرى التي تختلف عن حلّي من أحد زملائي.

هي الخوارزمية التي تقوم بإعادة تسلسل الخطوات نفسها عدداً من المرات.

مثال: خوارزمية الحل لطباعة الأرقام المحصورة بين 1 إلى 20 مع مربعاتها.

المعطيات: الأرقام من 1 إلى 20

المخرجات: طباعة هذه الأرقام ومربعاتها بشكل تصاعدي.

المعالجة: زيادة بمقدار واحد في كل مرة.

خوارزمية الحل:

1. البداية

2. اجعل قيمة I مساوياً لـ 1

3. $X=I \times I$

4. اطبع قيمة X, I

5. $I=I+1$

6. إذا كان I أقل أو يساوي 20 اذهب إلى الخطوة (3)

7. النهاية

نشاط نظري (7):

أكمل خوارزمية الحل لحساب وطباعة مجموع المتسلسلة الآتية:

$$\frac{1}{10} + \frac{2}{9} + \frac{3}{8} + \frac{4}{7} + \frac{5}{6} + \frac{6}{5} + \frac{7}{4} + \frac{8}{3} + \frac{9}{2} + \frac{10}{1}$$

المعطيات: المتسلسلة

المخرجات: مجموع المتسلسلة

المعالجة: زيادة بمقدار واحد على البسط وطرح بمقدار واحد على المقام في كل مرة.

خوارزمية الحل:

- | | |
|----------|----------|
| _____ .5 | _____ .1 |
| _____ .6 | _____ .2 |
| _____ .7 | _____ .3 |
| _____ .8 | _____ .4 |
| _____ .9 | |



نشاط نظري (8):



أكمل خوارزمية الحل الآتية لطباعة متوسط علامات عشرين طالباً وأعلى علامة وأقل علامة:

المعطيات: علامات 20 طالب

المخرجات: متوسط العلامات، _____، _____

المعالجة: _____، _____

خوارزمية الحل:

1. _____

2. _____

3. _____

4. _____

5. _____

6. _____

7. _____

8. _____

9. _____

الأسئلة



1. اكتب خوارزمية لطباعة عدد الأعداد الموجبة، وعدد الأعداد السالبة من بين 10 أرقام مدخلة.

2. اكتب خوارزمية لإيجاد حاصل ضرب ن من الأرقام المدخلة.



الموقف التعليمي التعلّمي (2 - 2):

توظيف مخطّط سير العمليّات (Flowchart) لإيجاد حلول برمجية لمشاكل حياتية.

وصفُ الموقف التعليمي: حضر مندوبٌ من مصلحة مياه محافظة القدس مشغلاً لتطبيقات الحاسوب في مدرسة الأنوار المهنيّة للبنات، لمساعدته في بناء مخطّط سير العمليّات لاحتساب المبلغ المُستحقّ على المشتركين شهرياً .

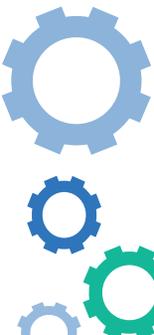
العمل الكامل			
خطوات العمل	وصف الموقف الصّفيّ	المنهجية (استراتيجية التّعلم)	الموارد
أجمع البيانات وأحلّها	<p>جمع البيانات من المندوب عن:</p> <ul style="list-style-type: none"> المعطيات اللازمة لاحتساب المبلغ المطلوب للدفع. المعادلات الحسابية اللازمة لاحتساب المبلغ المستحقّ. استثناءات إن وُجدت. <p>جمع البيانات عن :</p> <ul style="list-style-type: none"> مفهوم مخطّط سير العمليّات. عناصر مخطّط سير العمليّات. أنواع المخطّطات وطريقة تمثيلها. 	<p>حوار ومناقشة .</p> <p>التعلم التعاوني/ مجموعات .</p> <p>البحث العلميّ</p>	<ul style="list-style-type: none"> وثائق (طلب المندوب، نشرات، كتب ومصادر متنوعة). الشبكة العنكبوتية .
أخطّط وأقرّر	<ul style="list-style-type: none"> فهم وتحليل المسألة. تحديد المعطيات (اسم المشترك، القراءة السابقة، القراءة الحاليّة، سعر الوحدة، ديون سابقة...). تحديد المخرجات (المبلغ المستحق). تحديد عمليات المعالجة اللازمة لاحتساب المبلغ المستحقّ. تحديد نوع مخطّط سير العمليّات المطلوب لتمثيل الحلّ (التكراري). إعداد خطة التنفيذ والاتفاق على موعد تسليم مخطّط سير العمليّات لمصلحة المياه. 	<ul style="list-style-type: none"> تعلم تعاوني/مجموعات . حوار ومناقشة . عصف ذهنيّ/استمطار الأفكار. 	<ul style="list-style-type: none"> وثائق (طلب المندوب، نشرات، كتب ومصادر متنوعة). الشبكة العنكبوتية .



<ul style="list-style-type: none"> • قرطاسية. • الطلب الخاص بالمندوب. 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات. • حوار ومناقشة. • عصف ذهني/ استمطار الأفكار. 	<ul style="list-style-type: none"> • بناء آلية الاحتساب (مخطط سير العمليات) باستخدام المعطيات التي تم تحديدها (اسم المشترك، القراءة السابقة، القراءة الحالية، سعر الوحدة، ديون سابقة...) لإيجاد المبلغ المستحق. • أطبق المخطط التكراري لإيجاد المبالغ المستحقة على المشتركين. 	<p>أنفذ</p>
<ul style="list-style-type: none"> • الطلب الخاص بالمندوب. • الخطة المعدة سابقاً. • الشبكة العنكبوتية. • قرطاسية. 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات. • حوار ومناقشة. 	<ul style="list-style-type: none"> • التحقق من تحديد المعطيات، عمليات المعالجة والمخرجات. • التحقق من استخدام عناصر مخطط سير العمليات بشكل صحيح. • التحقق من تسلسل خطوات الحل. • التحقق من صحة الحل. • اكتشاف الأخطاء وتصويبها واقتراح البدائل. 	<p>أتحقق</p>
<ul style="list-style-type: none"> • قرطاسية. • حاسوب. • شاشة عرض. 	<ul style="list-style-type: none"> • حوار ومناقشة. • تعلم تعاوني/ مجموعات عمل. 	<p>توثيق البيانات التي تتعلق بـ:</p> <ul style="list-style-type: none"> • مفهوم مخطط سير العمليات. • عناصر مخطط سير العمليات. • أنواع مخططات سير العمليات. • « مخطط سير العمليات التسلسلي. • « مخطط سير العمليات ذات التفرع. • « مخطط سير العمليات التكراري. • إعداد العروض التقديمية. • توثيق آلية بناء مخطط سير العمليات بناءً على التفكير المنطقي. • إعداد ملف بالحالة (احتساب المبلغ المستحق على المشتركين بالاعتماد على عناصر محددة). 	<p>أوثق وأقدم</p>
<ul style="list-style-type: none"> • نماذج للتقويم معدة سابقاً. • طلب مصلحة المياه. 	<ul style="list-style-type: none"> • حوار ومناقشة. 	<ul style="list-style-type: none"> • نجاعة المخطط في احتساب المبلغ المستحق على المشتركين وإمكانية تطبيقها في مواقف جديدة للغرض نفسه. • رضا مصلحة المياه عن المخطط. 	<p>أقوم</p>



أوضح عناصر المخطط الانسيابي مع ذكر أمثلة على كل منها.



مخطّط سير العمليّات (Flowchart)

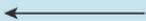
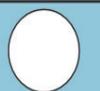
بعد أن نتأكّد من أنّ خطوات الخوارزمية سليمة، وقبل ترجمة الخطوات إلى إحدى لغات البرمجة، علينا أن نرسم مخطّطاً انسيابياً لسير خطوات هذه الخوارزمية، فما هو مخطّط سير العمليّات؟

مخطّط سير العمليّات (Flowchart):

هو تمثيلٌ بيانيّ للخوارزمية نوضّح به الخطوات المتّبعة لحلّ المسألة من البداية إلى النهاية، مع إخفاء التفاصيل الدقيقة لإعطاء الصورة العامّة عن الحلّ.

رموز تمثيل مخطّط سير العمليّات:

لتمثيل المخطّطات بيانيّاً نحتاج إلى مجموعة من الرموز ذات الأشكال المختلفة لتدلّ على وصف معيّن، كما يوضّح الشكل (2-6)، وفي داخل هذه الأشكال نكتب عناصر الإدخال والإخراج، وعمليات المعالجة للدلالة على العمليّة المطلوبة.

الرمز	الاسم	المعنى
	بداية / نهاية	يمثل بداية أو نهاية البرنامج.
	إدخال / إخراج	يمثل إدخال البيانات أثناء البرنامج أو إخراجها.
	عملية	يمثل عملية معالجة للبيانات.
	قرار	يمثل اتخاذ قرار أو تعبير منطقي يحتاج إلى جواب.
	خط انسياب	يمثل اتجاه الانسياب المنطقي للبرنامج.
	توصيلة	لتوصيل الأجزاء المختلفة في المخطط.

شكل (2-6) رموز تمثيل مخطّطات سير العمليّات ومعانيها



فوائد مخطّط سير العمليّات:

- ◀ تسهيل عمليّة دراسة البرنامج من قبل المستخدمين.
- ◀ تسهيل عمليّة مراجعة البرنامج لتعديله، أو لاكتشاف الأخطاء الموجودة فيه.
- ◀ يُستخدم وسيلةً لتوثيق البرامج، حيث يعكس المخطّط العمليّات كافّةً، من إدخالٍ وإخراجٍ ومعالجة.

أنواع مخطّطات سير العمليّات:

- ◀ المخطّط المتسلسل Sequential Flowcharts
- ◀ المخطّط ذو التفرّع Branched Flowcharts
- ◀ المخطّط التكراري Simple Loop Flowcharts

المخطّط المتسلسل:

هو مجموعة من التعليمات مرتّبة بشكل سلسلّةٍ مستقيمةٍ من بداية البرنامج حتى نهايته، بحيث تنعدم فيها أيّة تفرّعات على الطريق.

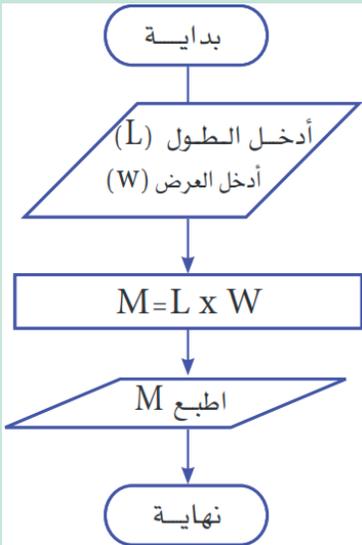
◀ مثال: ما صياغة الحلّ لإيجاد مساحة المستطيل ومحيطه، علماً بأنّ:

◀ مساحة المستطيل = الطول × العرض

◀ محيط المستطيل = $2 \times (\text{الطول} + \text{العرض})$

طريقة حلّ المسألة:

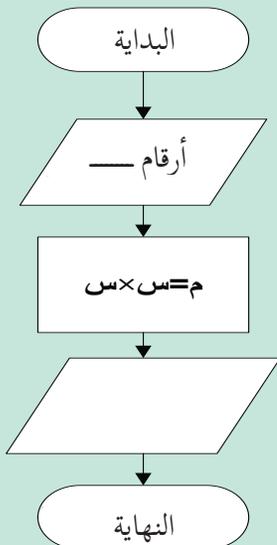
- أولاً: فهم المسألة وتحليل عناصرها.
- ثانياً: كتابة خطوات الخوارزمية.
- ثالثاً: رسم مخطّط سير العمليّات.

مخطّط سير العمليّات	الخوارزمية	عناصر المسألة
 <pre> graph TD Start([بداية]) --> Input[/أدخل الطول (L) أدخل العرض (W)/] Input --> Process[M=L x W] Process --> Output[/اطبع M/] Output --> End([نهاية]) </pre>	<p>خوارزمية الحل:</p> <ul style="list-style-type: none"> • اقرأ الطول والعرض (L, W) • احسب مساحة المستطيل من المعادلة: $M=L \times W$ • اطبع المساحة (M) 	<ul style="list-style-type: none"> • المعطيات: <ul style="list-style-type: none"> • الطول L • العرض W • المخرجات: مساحة المستطيل • المعالجة: قانون مساحة المستطيل

نشاط نظري (1):



أكمل الجدول الآتي لصياغة حلّ المسألة بالتعليمات اللازمة لإيجاد مربع أي رقم:

مخطّط سير العمليّات	الخوارزمية	عناصر المسألة
 <pre> graph TD Start([البداية]) --> Input[/أرقام —/] Input --> Process[م=س×س] Process --> Output[/ /] Output --> End([النهاية]) </pre>	<p>خوارزمية الحل:</p>	<ul style="list-style-type: none"> • المعطيات: • المخرجات: • المعالجة:



نشاط نظري (2):



أكمل الجدول الآتي لصياغة حلّ المسألة بالتعليمات اللازمة لإيجاد حاصل جمع أي رقمين:

مخطط سير العمليات	الخوارزمية	عناصر المسألة
	خوارزمية الحل:	<ul style="list-style-type: none"> المعطيات: المخرجات: المعالجة:

2- المخطط ذو التفرع

هو مخطط يختبر بعض الشروط ، فإذا كانت النتيجة تحقق الشرط يتبع مساراً يحوي مجموعة معينة من التعليمات ، وإذا لم تحقق الشرط يتبع مساراً آخر مختلفاً من التعليمات .

نشاط نظري (3):



أكمل الجدول الآتي لصياغة حلّ المسألة بالتعليمات اللازمة لإيجاد القيمة العظمى من بين ثلاث قيم: A, B, C

مخطط سير العمليات	الخوارزمية	عناصر المسألة
	خوارزمية الحل:	<ul style="list-style-type: none"> المعطيات: المخرجات: المعالجة:

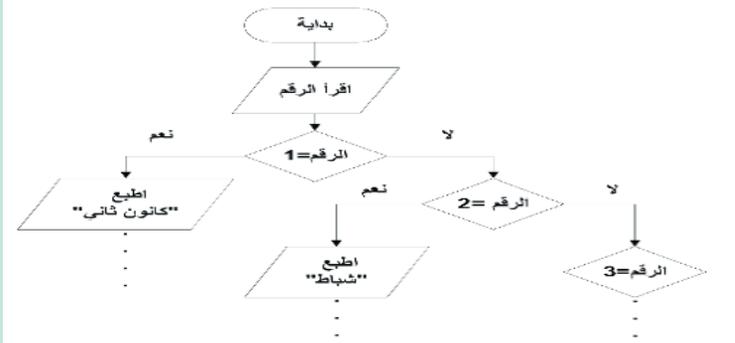


نشاط نظري (4):

ابحث عن طرق أخرى لحلّ المثال السابق باستخدام مخطّط سير العمليّات.

نشاط نظري (5):

أكمل الجدول الآتي الذي يقوم بقراءة رقم الأشهر (1-12)، وطباعة اسم الشهر المقابل للرقم المدخل:

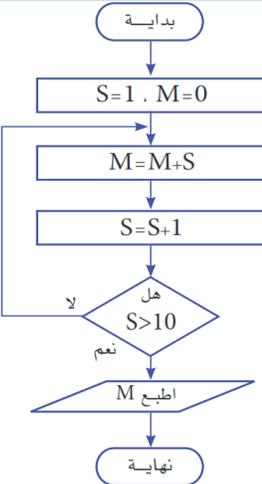
مخطّط سير العمليّات	الخوارزمية	عناصر المسألة
	خوارزمية الحل:	المعطيات: المخرجات: المعالجة:

3- المخطّطات التكرارية

تقوم المخطّطات التكرارية بإعادة تسلسل الخطوات نفسها عدداً من المرات.

نشاط نظري (6):

أكمل الجدول الآتي لصياغة حلّ المسألة بالتعليمات اللازمة لطباعة مجموع الأعداد من 1 إلى 10.

مخطّط سير العمليّات	الخوارزمية	عناصر المسألة
	خوارزمية الحل:	المعطيات: المخرجات: المعالجة:



نشاط نظري (7):



أكمل الجدول الآتي بالتعليمات اللازمة لحلّ المسألة الآتية: في نهاية كلّ شهر يتسلّم العاملون في مصنع للأجهزة الذكيّة أجوراً عن الساعات الإضافيّة التي يعملون بها، ويكون المبلغ مساوياً لعدد الساعات الإضافيّة مضروباً في أجر الساعة، فإذا كان عدد العاملين 05. احسب المبلغ الإضافي لكلّ عامل.

مخطط سير العمليات	الخوارزمية	عناصر المسألة
	خوارزمية الحل:	المعطيات: المخرجات: المعالجة:

نشاط نظري (8):



أكمل الجدول الآتي لصياغة حلّ المسألة بالتعليمات اللازمة لإيجاد حاصل جمع المتسلسلة الآتية:

$$\frac{1}{3} + \frac{2}{3} + 1 + \frac{4}{3} + \frac{5}{3} + 2 + \frac{7}{3} + \dots + \frac{N}{3}$$

مخطط سير العمليات	الخوارزمية	عناصر المسألة
	خوارزمية الحل:	المعطيات: المخرجات: المعالجة:



نشاط نظري (9):



اشترى شخص ثلاث قطع من الأرض مختلفة الأشكال (Shapes)، دائرية (Circle)، ومستطيلة (Rectangle)، ومثلثة (Triangle)

المطلوب: إكمال مخطط سير العمليات لإيجاد مساحة كل قطعة (Area) بحيث يتم إدخال رمز شكل القطعة (ShapeCode) من لوحة المفاتيح. تم ترميز كل شكل بالرموز الآتية على التوالي (T, R, C). في حال تم إدخال

رمز مختلف يسمح له بإعادة المحاولة. علماً بأن مساحة الدائرة = $\pi \times (\text{نصف القطر})^2$

مساحة المستطيل = الطول \times العرض ومساحة المثلث = نصف طول القاعدة \times الارتفاع.

أكمل الجدول الآتي لصياغة حل المسألة بالتعليمات اللازمة لإيجاد مساحة الشكل الهندسي:

مخطط سير العمليات	الخوارزمية	عناصر المسألة
<pre> graph TD Start([Start]) --> ReadShapeCode[/Read ShapeCode/] ReadShapeCode --> IsT{ShapeCode = "T"} IsT -- Yes --> ReadBH[/Read B, H/] ReadBH --> AreaCalc[Area = 0.5 x B x H] AreaCalc --> PrintArea[/Print Area/] IsT -- No --> IsOther{ } IsOther -- Yes --> PrintArea IsOther -- No --> IsLoop{ } IsLoop -- Yes --> PrintArea IsLoop -- No --> IsT PrintArea --> End([End]) </pre>	<p>خوارزمية</p> <p>الحل:</p>	<p>المعطيات:</p> <p>المخرجات:</p> <p>المعالجة:</p>



التأكد من صحة خوارزمية الحل

أية خوارزمية تتم كتابتها يتم التأكد من مدى صحتها وقدرتها على حل المسألة، بتتبع قيم متغيراتها المختلفة والنتائج الناجمة عنها (المخرجات) أثناء تنفيذ الخوارزمية، ومقارنة مدى توافقها مع المخرجات المطلوبة.

◀ مثال توضيحي: إيجاد حاصل ضرب الأعداد من 1 إلى 10

مخطط سير العمليات	الخوارزمية	عناصر المسألة
<pre> graph TD Start([البداية]) --> Init[Acc = 1 c = 1] Init --> Loop[Acc=Acc*c C=C+1] Loop --> Decision{C > 10} Decision -- No --> Loop Decision -- Yes --> Print[/اطبع Acc/] Print --> End([النهاية]) </pre>	<p>اجعل قيمة العداد $C=1$ المجمع $Acc=1$. اجعل قيمة المجمع $Acc=Acc * C$. اجعل قيمة العداد $C=C+1$. إذا كانت قيمة العداد أكبر من 10، اذهب للخطوة (7)، غير ذلك اذهب للخطوة (4). اطبع قيمة المجمع Acc</p>	<p>المعطيات: الأعداد من 1 إلى 10 المخرجات: القيمة 3628800 المعالجة: عملية الضرب</p>

المخرجات
3628800

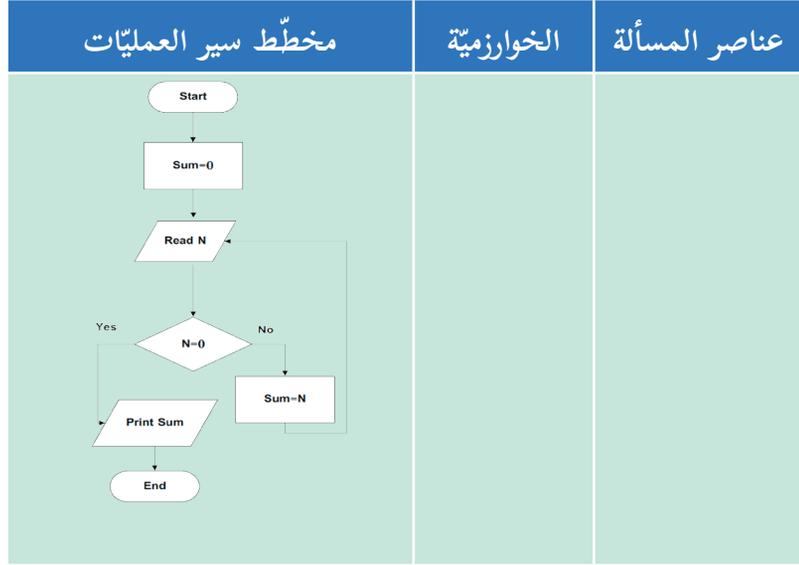
جدول التتبع	
Acc	C
1	1
1	2
2	3
6	4
24	5
120	6
720	7
5040	8
40320	9
362880	10
3628800	11



نشاط نظري (10):

أكمل الجدول الآتي لصياغة حلّ المسألة لحساب مجموع الأرقام المدخلة، والتوقّف عند إدخال الرقم (0)، علماً بأنّ المدخلات كانت على التوالي: 4، 2، 6، 3، 4، 5، 1، 0.

المخرجات	جدول التتبع	
	Sum	N
	0	4



أكمل جدول التتبع وعمود المخرجات الناتج من عملية تنفيذ المخطّط الحالي.

- هل مخطّط سير العمليّات يعمل بشكل صحيح؟ (نعم/لا)
- في حال كانت الإجابة لا، أصحّح الخطأ وتأكد من الحلّ مرّة أخرى.

الأسئلة

1. اكتب مخطّط سير العمليّات الذي يقوم بإدخال علامات الطلاب في الصفّ، وطباعة عدد الناجحين وعدد الراسبين، علماً بأنّ علامة النجاح يجب أن تكون أكبر أو تساوي ٥٠.
2. اكتب مخطّط سير العمليّات لإيجاد الرقم الأكبر من بين ٥١ رقم مدخل.



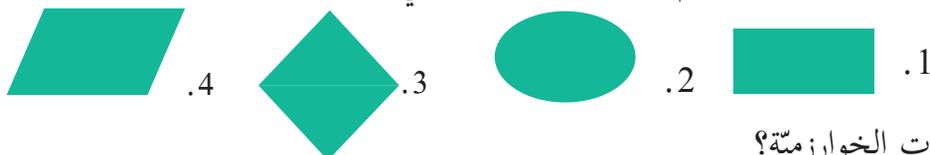
أسئلة الوحدة

السؤال الأول :

بناء على الخوارزمية الآتية اختر رمز الإجابة الصحيحة فيما يأتي:

1. start
2. $x=9$
3. $sum=0$
4. $sum=sum + x$
5. $x=x-3$
6. If $x > 0$ goto step 4
7. print sum
8. end

1. أيّ من الأشكال الآتية تُستخدم لكتابة الخطوة (2) في مخطط سير العمليات؟



2. ما مخرجات الخوارزمية؟

1. 3 2. 6 3. 9 4. 18

3. ما القيمة النهائية لـ x ؟

1. 6 2. 0 3. 9 4. 3

4. ما عدد مرّات دخول الحلقة؟

1. 1 2. 2 3. 3 4. 4

5. ما وظيفة الخوارزمية؟

1. طباعة مضاعفات العدد 3 تنازلياً بين 3 و 9
2. طباعة مجموع مضاعفات العدد 3 بين 1 و 10
3. طباعة مضاعفات العدد 3 تصاعدياً بين 1 و 10
4. طباعة حاصل ضرب مضاعفات 3 بين 1 و 10



السؤال الثاني:

عرف كلاً ممّا يأتي:

1. التفكير المنطقيّ. 2. الخوارزمية. 3. مخطّط سير العمليّات.

السؤال الثالث:

أنشئ صياغة الحلّ لكلّ من المسائل الآتية:

1. تحويل مبلغ يدخله المستخدم بالشيكل إلى ما يعادله بالدولار الأمريكي ، إذا علمت أنّ الدولار = 3.75 شيكل.

2. لإدخال قيمة س وإيجاد قيمة ص حسب المعادلة الآتية: $ص = س^3 + س^2$

3. قراءة عدد ثمّ إظهار العشرة أعداد الموالية، مثال العدد 21 ، يظهر لنا الأعداد من 22 إلى 31.

4. قراءة عدد ثم حساب مجموع الأعداد الطبيعيّة إلى غاية هذا العدد، مثال لو أدخلنا العدد 5 فإنّه يتمّ حساب $1+2+3+4+5 = 15$.

5. حساب المضروب لأيّ رقم مدخل (ن! = $1 \times 2 \times 3 \times \dots \times (ن-1) \times (ن-2) \times \dots \times 1$)

مثال : إذا كانت القيمة المدخلة = 5 فإن $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

استثناء : $1! = 1$

6. حساب مربعات الأرقام المدخلة حتى أول عدد سالب.

7. حساب مجموع ن من العلامات.

8. حساب معدل 4 علامات لمئة طالب

9. طباعة المتتالية الآتية: $1, \frac{2}{4}, \frac{3}{9}, \frac{4}{16}, \frac{5}{25}, \dots, \frac{10}{100}$

10. إيجاد مجموع المتسلسلة الآتية:

$$\frac{1}{m+1} + \frac{1}{m+2} + \frac{1}{m+3} + \dots + \frac{1}{m+n}$$

حيث إنّ $m < n$

11. حساب حاصل ضرب الأعداد الزوجيّة من 1 إلى 100

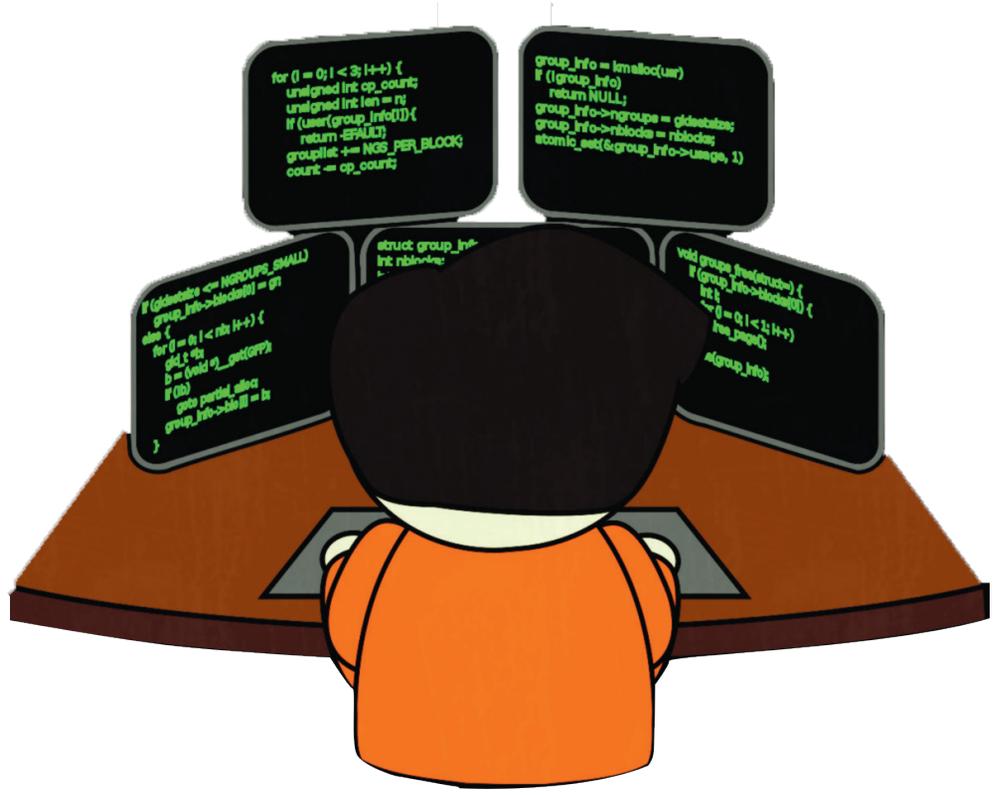
12. قراءة 10 أرقام، ومن ثمّ طباعة متوسط الأرقام الموجبة ومتوسط الأرقام السالبة.

مشروع الوحدة

صياغة الحل لإيجاد المبلغ المستحق لمشتريات زبون من سوبر ماركت.



أساسيات البرمجة



أناقش:

لغة البرمجة لغة التخاطب
بين الآلة والإنسان.



يُتَوَقَّعُ من الطلبة بعد دراسة هذه الوَحْدَةِ، والتَّفَاعُلِ مَعَ أنشطتها أن يكونوا قادرين على
بناء برنامجٍ تطبيقيٍّ من خلال الآتي:

- ✓ توظيف مبادئ (أساسيات البرمجة) في بناء برنامج.
- ✓ توظيف مبادئ (برمجة الكيانات) في بناء برنامج.



الكفايات المهنية

الكفايات المهنية المتوقع من الطلبة امتلاكها بعد الانتهاء من دراسة هذه الوحدة، والتفاعل مع أنشطتها:
أولاً: الكفايات الحرفية.

- ◀ القدرة على بناء برنامج باستخدام (أساسيات البرمجة).
- ◀ القدرة على بناء برنامج موظفاً (برمجة الكيانات).

ثانياً: الكفايات الاجتماعية والشخصية.

- ◀ العمل ضمن فريق .
- ◀ تقبل التغذية الراجعة .
- ◀ احترام رأي الآخرين .
- ◀ مصداقية التعامل مع الزبون .
- ◀ حفظ خصوصية الزبون .
- ◀ الاستعداد للاستجابة لطلبات الزبون .
- ◀ التحلي بأخلاقيات المهنة
- ◀ أثناء العمل .
- ◀ التمتع بالفكر الريادي أثناء العمل .
- ◀ استخلاص النتائج ودقة الملاحظة .
- ◀ الاتصال والتواصل الفعال .
- ◀ الاستعداد للاستفسار والاستفادة من ذوي الخبرة .
- ◀ الإفادة من مقترحات الآخرين .
- ◀ امتلاك مهارة التأمل الذاتي .
- ◀ الاستعداد التام لتقبل ملحوظات الزبون وانتقاداتهم .
- ◀ القدرة على الإقناع .

ثالثاً: الكفايات المنهجية:

- ◀ توظيف التعلم التعاوني .
- ◀ القدرة على إدارة الحوار وتنظيم النقاش .
- ◀ امتلاك مهارة البحث العلمي، والقدرة على توظيف أساليبه .
- ◀ القدرة على التخطيط .
- ◀ القدرة على استمطار الأفكار .
- ◀ استخدام المصادر والمراجع المختلفة .
- ◀ توثيق نتائج العمل وعرضها .
- ◀ تقويم النتائج وعرضها .
- ◀ جمع البيانات وتحليلها وتبويبها .

قواعد الأمان والسلامة:

- ✓ الالتزام بالجلسة الصحيّة أمام الحاسوب.
- ✓ إراحة العينين كلّ ربع ساعة.
- ✓ الجلوس في غرفة فيها إضاءة مناسبة وتهوية جيّدة.
- ✓ الابتعاد عن الحاسوب كلّ ساعة أو القيام ببعض التمارين الرياضيّة.



الموقف التعليمي التعلّمي (3 - 1):

توظيف مبادئ أساسيات البرمجة في بناء برنامج 1.

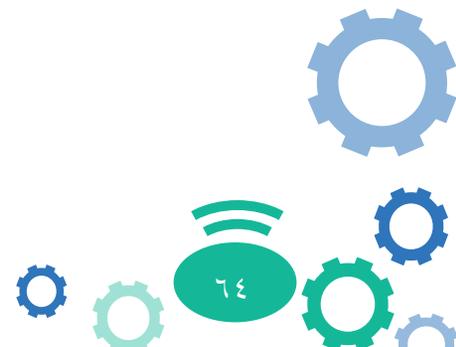
وصف الموقف التعليمي: زار معلّم الرياضيات مشغلاً لتطبيقات المحمول في مدرسة يافا المهنيّة، لمساعدته في بناء برنامج لإيجاد حلّ معادلةٍ خطيّة.

العمل الكامل			
الموارد	المنهجية (استراتيجية التعلّم)	وصف الموقف الصّفي	خطوات العمل
<ul style="list-style-type: none"> • وثائق (طلب معلّم الرياضيات، نشرات، كتب ومصادر متنوعة). • التكنولوجيا (روابط إلكترونية متنوعة من مصادر موثوقة، حاسوب، شاشة عرض، إنترنت) 	<ul style="list-style-type: none"> • حوار ومناقشة. • التعلّم التعاوني/ مجموعات. • البحث العلميّ. 	<p>جمع البيانات من معلّم الرياضيات عن:</p> <ul style="list-style-type: none"> • عناصر المعادلة الخطيّة. • طريقة حلّ المعادلة الخطيّة. <p>جمع البيانات عن:</p> <ul style="list-style-type: none"> • لغات البرمجة. • المتغيّرات البرمجية. • العمليات الحسابية، وألويّات تنفيذها. • جمل الإدخال والإخراج. 	أجمع البيانات وأحلّها
<ul style="list-style-type: none"> • وثائق (طلب معلّم الرياضيات، نشرات، كتب ومصادر متنوعة، كتب). • الشبكة العنكبوتية. 	<ul style="list-style-type: none"> • تعلّم تعاوني/ مجموعات حوار ومناقشة . 	<p>أصنّف البيانات الخاصّة بـ:</p> <ul style="list-style-type: none"> • لغات البرمجة. • المتغيّرات البرمجية. • العمليات الحسابية، وألويّات تنفيذها. • جمل الإدخال والإخراج. 	أخطّط وأقرّر
<ul style="list-style-type: none"> • قرطاسية • الطلب الخاصّ بمعلّم الرياضيات • حاسوب، برنامج خاصّ (JDK, IntelliJ IDEA) 	<ul style="list-style-type: none"> • تعلّم تعاوني/ مجموعات حوار ومناقشة. • عصف ذهنيّ/ استمطار الأفكار. 	<ul style="list-style-type: none"> • اختيار لغة البرمجة كوتلن (Kotlin) وتنصيبها على جهاز الحاسوب. • تحديد المدخلات (معاملات المعادلة الخطيّة). • اختيار العمليات الحسابية لحلّ المعادلة الخطيّة. • تحديد المخرجات (حلّ المعادلة). • أكتب خوارزمية الحلّ. • تحديد المتغيّرات البرمجية الرّقميّة وتسميتها. • أكتب البرنامج. 	أنفّذ

<ul style="list-style-type: none"> • الخطة المُعدَّة سابقاً • قرطاسية • الطلب الخاصّ بمعلّم الرياضيات • حاسوب، إنترنت، برنامج خاصّ (JDK, intelliJ, IDEA) 	<ul style="list-style-type: none"> • تعلّم تعاوني/مجموعات • حوار ومناقشة. 	<ul style="list-style-type: none"> • التحقُّق من ملاءمة لغة البرمجة. • التحقُّق من المتغيّرات الرقمية. • التحقُّق من صحّة العمليات الحسابية . • التحقُّق من مدى ملاءمة المدخلات والمخرجات والعمليات . • التحقُّق من تشغيل البرنامج وصحّة النتائج. • اكتشاف الأخطاء وتصويبها واقتراح البدائل. 	<p>أُتحقَّقُ</p>
<ul style="list-style-type: none"> • حاسوب • شاشة عرض • قرطاسية 	<ul style="list-style-type: none"> • حوار ومناقشة. • تعلّم تعاوني/مجموعات. 	<p>توثيق البيانات التي تتعلق بـ:</p> <ul style="list-style-type: none"> • لغات البرمجة. • المتغيّرات البرمجية. • العمليات الحسابية، وأوليات تنفيذها. • جمل الإدخال والإخراج. - إعداد العروض التقديمية - إعداد ملف بالحالة (بناء برنامج لإيجاد حلّ معادلة خطية) 	<p>أوثقُ وأقدّم</p>
<ul style="list-style-type: none"> • نماذج للتقويم • مُعدَّة سابقاً. 	<ul style="list-style-type: none"> • حوار ومناقشة. 	<ul style="list-style-type: none"> • إيجاد حلّ أكثر من معادلة خطية بالبرنامج نفسه. • رضا المعلّم عن البرنامج. 	<p>أقومُ</p>



أفسر مفهوم لغة البرمجة و أذكر أمثلة على لغات البرمجة القديمة و الحديثة



توظيف مبادئ أساسيات البرمجة في بناء برنامج 1

نشاط نظري (1):

أبحث عن لغات البرمجة المختلفة ومجال استخدام كل لغة، وأصنّفها وفق الجدول الآتي:

لغة البرمجة	مجال الاستخدام
Java، Kotlin	برمجة تطبيقات الهواتف المحمولة الخاصة بنظام Android
_____	برمجة تطبيقات الهواتف المحمولة الخاصة بنظام iOS
_____	تصميم وبرمجة مواقع الإنترنت.
_____	برمجة تطبيقات سطح المكتب.

لغة البرمجة

تعلمتُ في الوَحْدَة السابقة طُرُق كتابة الخوارزميات بلغتي المبسطة. سأتعلم الآن باستخدام لغة البرمجة «كوتلن»، كيفية كتابة الخوارزمية على هيئة سلسلة من الأوامر والتعليمات بما يُعرف بِ (البرنامج). هذه البرامج تُكتب ضمن شروط لغة البرمجة وقواعدها، وبعدها تُعطى للحاسوب الذي يقوم بدوره بترجمتها، وتنفيذ سلسلة الأوامر التي تحتويها.

تصنيف لغات البرمجة:

◀ وفق المستوى:

يمكن تصنيف لغات البرمجة وفق سهولة فهمنا نحن البشر لمفرداتها (الأوامر والمتغيرات) إلى:

◀ منخفضة المستوى:

1. تُعدّ لغة الآلة من اللغات المنخفضة المستوى؛ لأنّ مفرداتها تتكوّن فقط من النظام الثنائي (0 و1).
2. لغة التجميع كذلك، فهي لغة منخفضة المستوى وتتضمّن مفردات تتعامل مع عناوين الذاكرة، ويمكن للحاسوب فهمها.

◀ عالية المستوى:

1. اللغات الراقية مثل لغة البرمجة جافا، فهي لغات عالية المستوى حيث تكتب مفرداتها بلغة قريبة من لغة الانسان.
2. وفق الاستخدام: يمكن تصنيف لغات البرمجة وفق بيئة العمل المستهدفة إلى:
 - لغات لبرمجة مواقع الانترنت.
 - لغات لبرمجة تطبيقات تعمل في بيئة الحاسوب الشخصي.
 - لغات لبرمجة تطبيقات الهواتف، مثل تطبيقات الأندرويد أو iOS.

نشاط نظري (2):

يمكن استخدام لغة كوتلن ولغة جافا في برمجة هواتف الأندرويد، أفرارً بين لغة اللغتين من حيث المعايير الموضحة في الجدول الآتي:

المعيار	جافا	كوتلن
سهولة التعلُّم		
المرونة		
حجم البرنامج		
استخدام الذاكرة		

ملحوظة:

لقد تمَّ اعتماد لغة Kotlin من قبل شركة جوجل عام 2017 لغةً رسميَّةً لتطبيقات الهواتف المحمولة التي تعمل بنظام الأندرويد؛ لذلك سنستخدم لغة كوتلن في هذا الكتاب لننطلق إلى عالم ترجمة تطبيقات الهواتف.

نشاط عملي (3):

أهيبُ بيئة العمل الخاصَّة بلغة كوتلن في حاسوبي الخاصِّ.

بيئة العمل الخاصَّة بلغة كوتلن:

- لبدء التعامل مع لغة Kotlin لا بدَّ من تهيئة بيئة العمل، ويلزمنا لذلك تثبيت كلِّ من:
- ◀ JDK Java: وهي حزمة برمجيات لتطوير البرامج.
 - ◀ IntelliJ IDEA: وهي عبارة عن بيئة التطوير المتكاملة (IDE) للغة كوتلن.



تثبيت JAVA JDK:

باستخدام محرك البحث Google، أبحث عن Java JDK، وأختار رابط الموقع الذي يحوي على الحزمة (موقع شركة أوراكل)، وأضغط على أيقونة التنزيل، كما في الشكل 3 - 1.



شكل (3-1): الأيقونة الخاصة بتحميل Java JDK

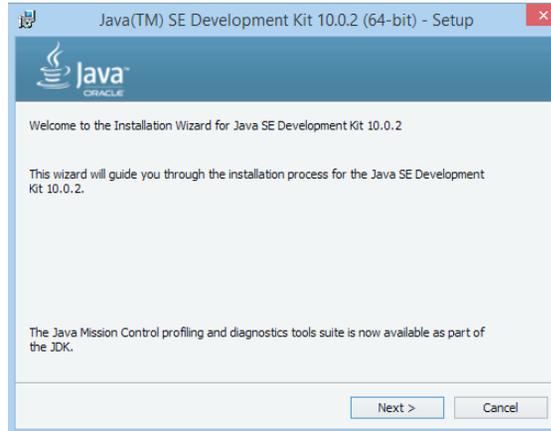
قبل بدء التنزيل يجب الموافقة على اتفاقية الترخيص وذلك باختيار Accept License Agreement، ثم أنقر على رابط برنامج وفقاً لنظام التشغيل المناسب (انظر إلى شكل 3-2).

Java SE Development Kit 10.0.2		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
<input type="radio"/> Accept License Agreement <input checked="" type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux	306 MB	jdk-10.0.2_linux-x64_bin.rpm
Linux	338.43 MB	jdk-10.0.2_linux-x64_bin.tar.gz
macOS	395.46 MB	jdk-10.0.2_osx-x64_bin.dmg
Solaris SPARC	207.07 MB	jdk-10.0.2_solaris-sparcv9_bin.tar.gz
Windows	390.25 MB	jdk-10.0.2_windows-x64_bin.exe

شكل (3-2): شاشة الموافقة على شروط اتفاقية الترخيص واختيار نظام التشغيل المناسب

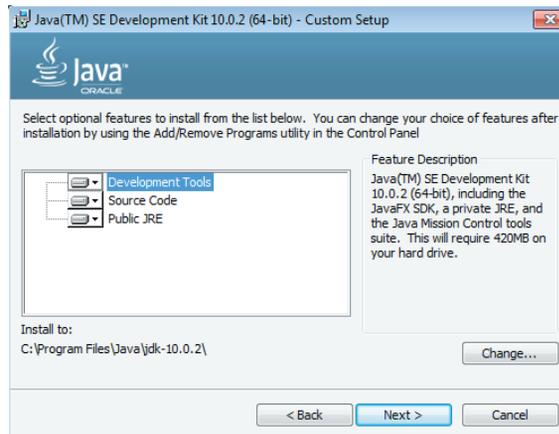


- في حالتنا تمّ اختيار نظام التشغيل Windows وبعد تنزيل الملفّ يتمّ تشغيله، فتظهر النافذة الترحيبية الآتية، (انظر إلى شكل 3-3):



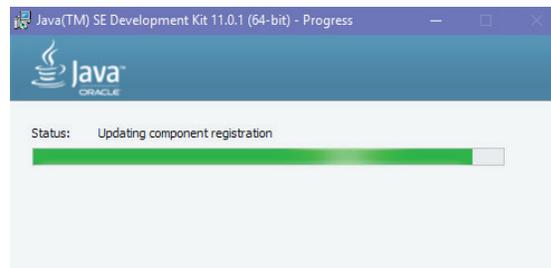
شكل(3-3): شاشة ترحيبية للبدء في تثبيت JDK

- وعند الضغط على (Next) تتمّ عملية تثبيت البرنامج (انظر إلى شكل 4-3).



شكل (4-3): شاشة لاختيار الخصائص الاختيارية المراد تثبيتها

- وعند الضغط على (Next) تتمّ عملية تثبيت البرنامج (انظر إلى شكل 5-3).



شكل (5-3): شاشة تثبيت البرنامج



حتى يتم تثبيت البرنامج كاملاً ويعطي شاشة الإنهاء الآتية، (انظر إلى شكل 6-3):



شكل (6-3): شاشة تنفيذ بانتهاء تثبيت برنامج JDK بنجاح

تثبيت IntelliJ IDEA: يتم البحث باستخدام محرك البحث Google عن جملة 'download IntelliJ IDEA', ثم تنزيل البرنامج بالضغط على (download) أسفل كلمة (Community):

Community

Java, Groovy, Scala and
Android development

DOWNLOAD

.EXE
↓

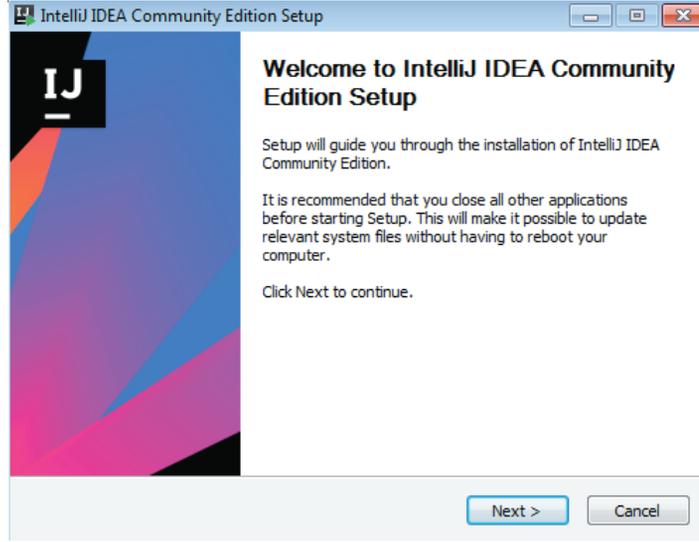
Free, open-source

الشكل (7-3)



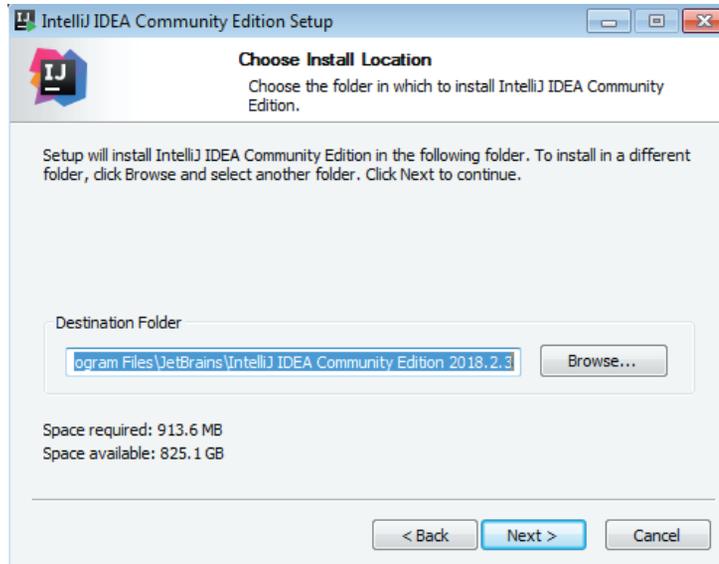
تثبيت IntelliJ IDEA:

باستخدام محرك البحث Google، أبحث عن IntelliJ IDEA، وأختار رابط الموقع الذي يحوي على البرنامج (موقع شركة JetBrains)، وأضغط على أيقونة التنزيل، كما في الشكل (7-3).
١. عند تشغيل ملف برنامج IntelliJ IDEA الذي تمّ تحميله تظهر شاشة الترحيب الآتية، انظر إلى شكل (8-3):

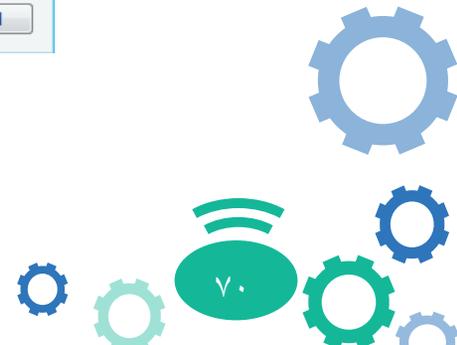


شكل (8-3): شاشة ترحيبية تمهيدية لتثبيت IntelliJ IDEA

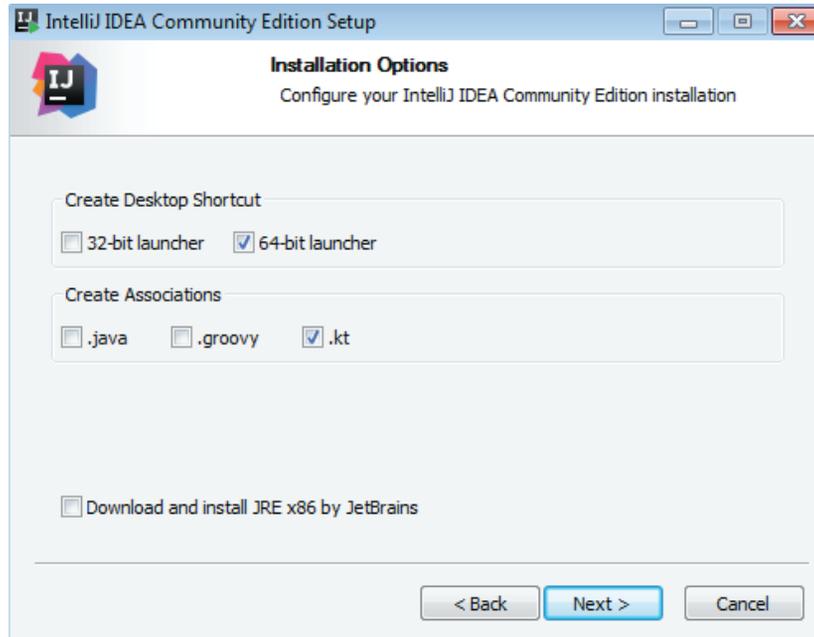
٢. عند الضغط على زر (Next) تظهر شاشة اختيار مكان تثبيت البرنامج (انظر إلى شكل 9-3):



شكل (9-3): شاشة لاختيار مكان تثبيت البرنامج

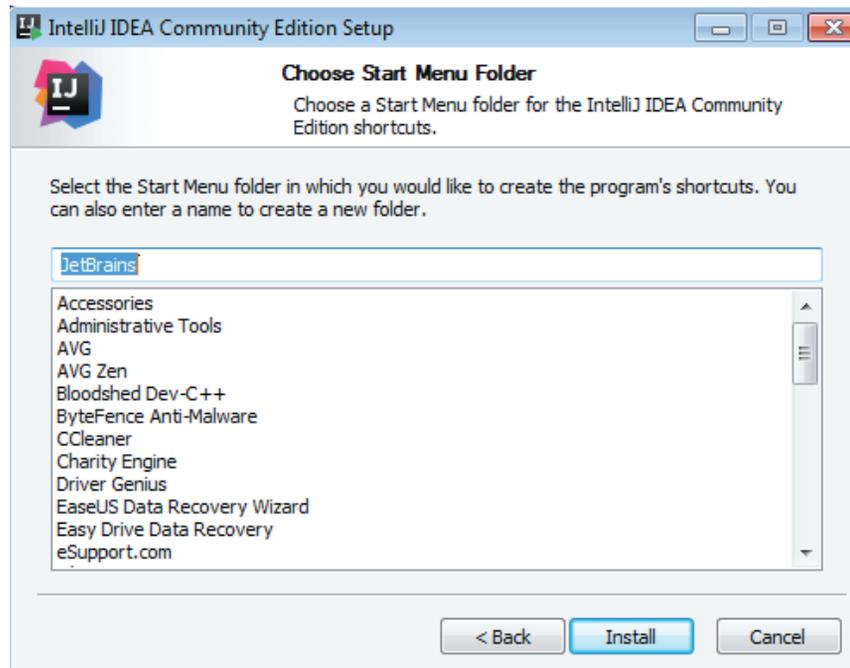


٣. عند الضغط على زر (Next) تظهر شاشة اختيار إعدادات للبرنامج، أختار (كم بت) يدعم نظام التشغيل الخاص بحاسوبي، وربط ملفات كوتلن (.kt). بالبرنامج كما هو موضَّح بالشكل الآتي، (انظر إلى شكل 10-3):



شكل (10-3): شاشة اختيار إعدادات البرنامج

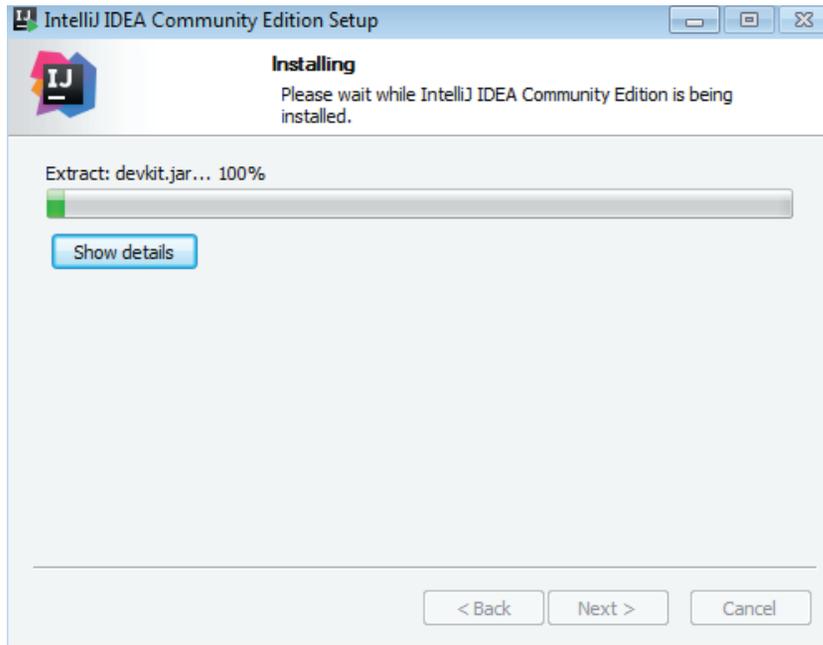
٤. عند الضغط على زر (Next) تظهر شاشة اختيار اسم المجلد (Folder) الذي سيتم تخزين اختصارات البرنامج بداخله في قائمة ابدأ (start) (انظر إلى شكل 11-3):



شكل (11-3): شاشة اختيار اسم المجلد الذي سيتم تخزين اختصارات البرنامج بداخله في قائمة ابدأ

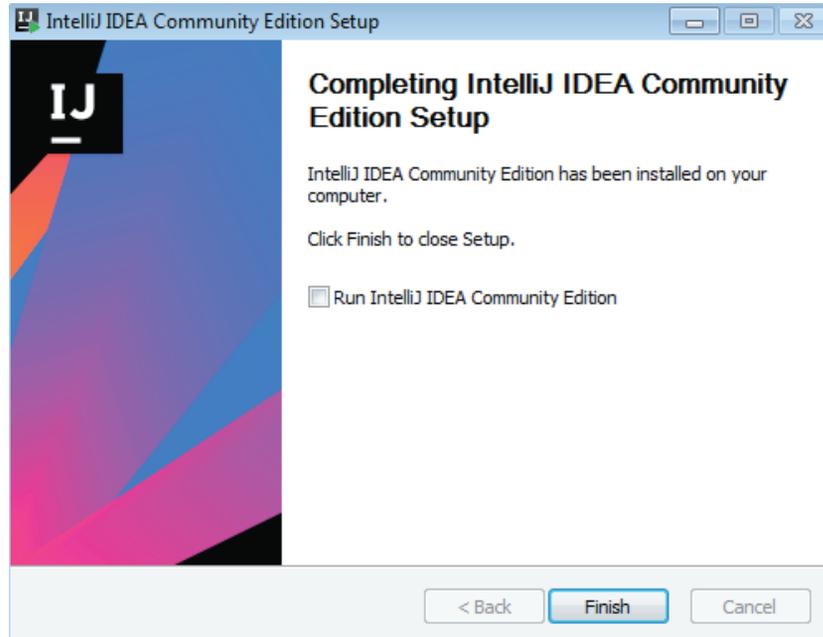


٥. عند الضغط على زر (Install) تبدأ عملية تثبيت البرنامج (انظر إلى شكل 12-3):



شكل (12-3): شاشة تثبيت البرنامج

٦. أخيراً تظهر شاشة تفيد انتهاء تثبيت البرنامج، (انظر إلى شكل 3 - 13).



شكل (13-3): شاشة انتهاء تثبيت البرنامج بنجاح





أشغل برنامج IntelliJ IDEA، وإعداد مشروع الأول (Hello World).

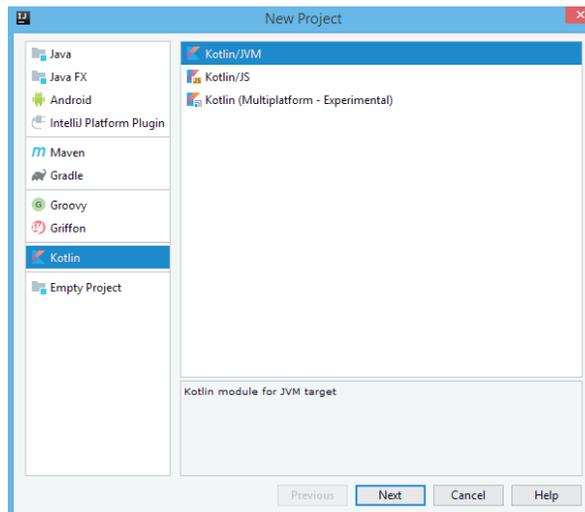
إنشاء المشروع الأول "Hello World"

عند تشغيل البرنامج تظهر شاشة اختيار المشروع، ويتم اختيار (Create New Project) لإنشاء مشروع جديد (كما في الشكل 14-3):

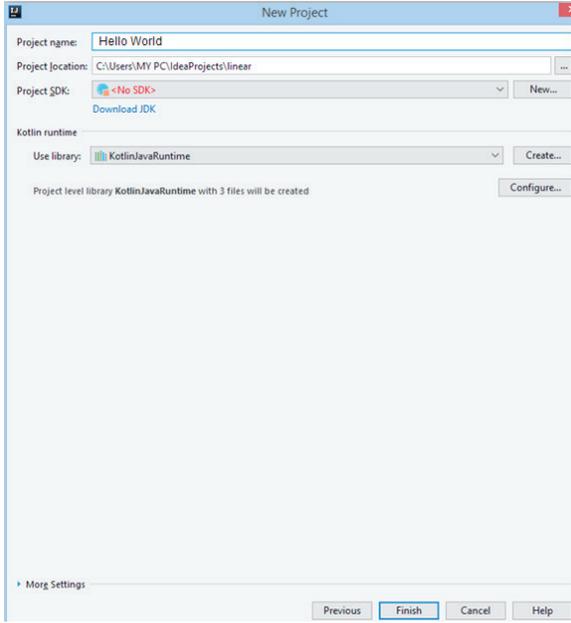


شكل (14-3): شاشة البداية لبرنامج IntelliJ IDEA

1. تظهر نافذة اختيار نوع المشروع، فيتم اختيار Kotlin من القائمة اليسرى، ثم اختيار Kotlin/JVM من القائمة اليمنى، ثم الضغط على زر Next (انظر الى شكل 15-3):



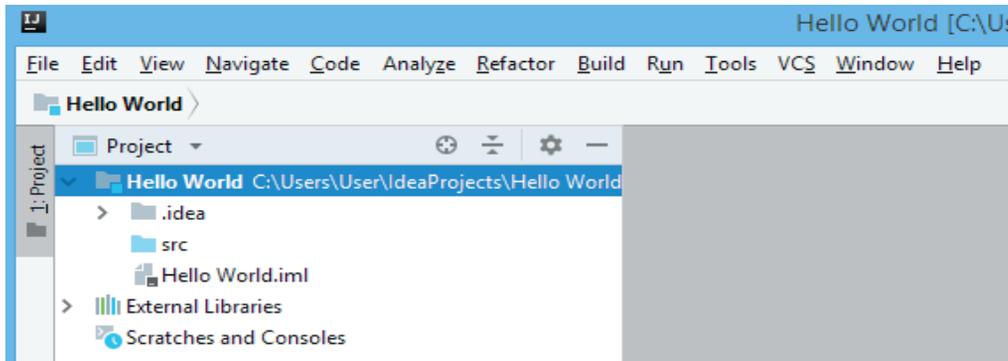
شكل (15-3): شاشة اختيار نوع المشروع



2. ثم تظهر شاشة إعداد للمشروع (خانة لإدخال اسم المشروع)، سأختار اسم (Hello World) في مثلنا الحالي، ثم يتم تحديد مكان تخزين المشروع، ثم اختيار Project SDK والمقصود به الـ JDK، ويتم اختياره بالضغط على زر New، واختيار المكان الذي تم فيه تثبيت الـ JDK سابقاً، وفي العادة يكون في المجلد (C:\ProgramFiles\Java\jdk-10.0.2) أو حسب إصدار نسخة الـ JDK (تتم هذه العملية فقط عند إنشاء المشروع الأول)، ثم الضغط على الزر Finish. (انظر إلى الشكل 3-16).

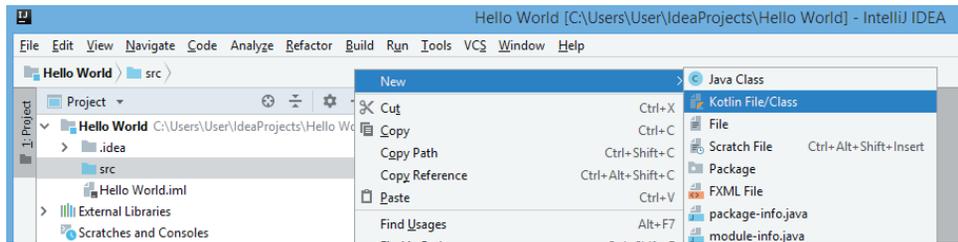
شكل (3-16): شاشة اختيار اسم المشروع ومكان تخزينه

3. ثم تأتي شاشة البرنامج ويظهر على اليسار مجلد المشروع (Hello World)، وبالضغط على هذا المجلد تظهر المجلدات

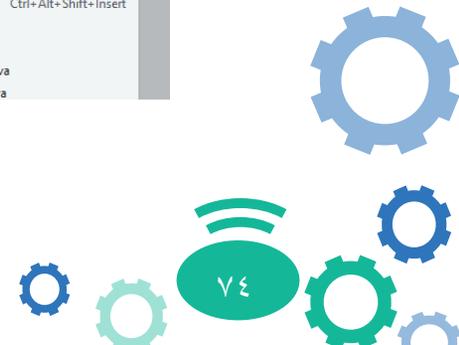


شكل (3-17): شاشة البرنامج

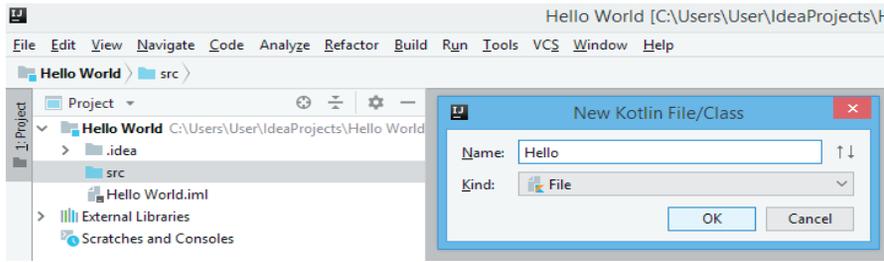
4. أضغط على مجلد src بالزر الأيمن للفأرة وأختار New، ثم Kotlin file/Class. (انظر شكل 3-18).



شكل (3-18): لإدراج ملف جديد

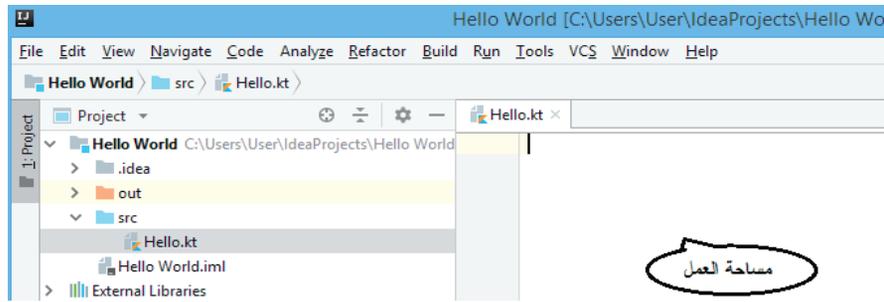


5. ثم أُحَدِّدُ اسماً مناسباً للملف كما في الشكل الآتي، (انظر شكل 3-19):



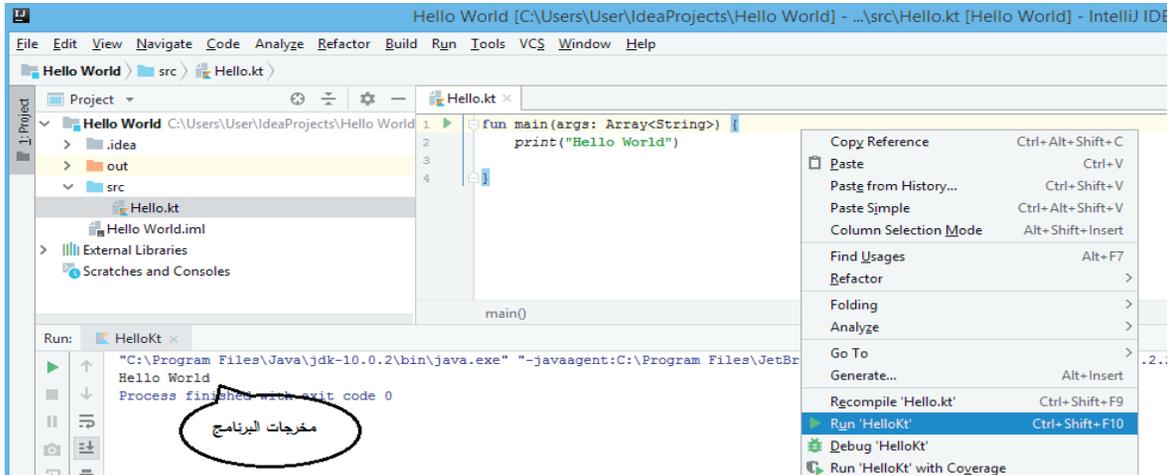
شكل (3-19): لتحديد اسم مناسب للملف

6. تأتي مساحة العمل المخصصة لكتابة الترميز البرمجيّ Source Code (انظر شكل 3-20).



شكل (3-20): مساحة العمل المخصصة لكتابة الترميز البرمجيّ

7. أكتب كلمة main في مساحة العمل، ثم أضغط على مفتاح Enter لبناء الدالّة الرئيسية للمشروع، بشكلٍ تلقائيّ، ثم أكتب في داخلها الأمر ("Hello World") print (انظر الى شكل 3-21):



شكل (3-21): برنامج Hello World

8. لتنفيذ البرنامج السابق أضغط على زر الفأرة الأيمن، وأختار Run Hello.kt ومخرجات البرنامج ستظهر كما هو موضّح في شكل (3-21).



شرح الكود البرمجي :

عند كتابة برنامج بلغة كوتلن، يسمح باستخدام الدالة الرئيسية main لمرة واحدة فقط، والتي يبدأ منها تنفيذ البرنامج بطريقة كتابة الدالة main

```
fun main(args: Array<String>){  
print ("Hello World")  
}
```

الدالة print وتُستعمل في الطباعة سواء كان نصوصاً أو أرقاماً، ويتم تمييز النصوص من طرف البرنامج بوضعها بين علامتي التنصيص "". في هذه الحالة ستم طباعة Hello World

المتغيرات في كوتلن:

المتغير: هو اسم لموقع في ذاكرة الحاسوب لتخزين بيانات فيه، ويتم الإعلان عنه من خلال تعريفه في البرنامج.
شروط تسمية المتغيرات:

1. يتكوّن من حروف وأرقام اللغة الانجليزية ويبدأ بحرف.
 2. لا يحتوي على فراغات أو رموز خاصّة مثل: %، *، >، <، :، !، @، #، \$، %، ^، &، *... الخ.
- أنواع المتغيرات:**

المتغيرات	النوع	الحجم في الذاكرة بالبايت
الرّقميّة	Long	8
الصحيحة	Int	4
	Short	2
	Byte	1
الرّقميّة	Float	4
العشريّة	Double	8
النصيّة	Char	2
أي شيء	String	حسب المدخل
	Any	حسب المدخل
المنطقيّة	Boolean	1

أنواع المتغيرات هي كلمات محجوزة ولا يجوز استخدامها إلا للتعريف بنوع المتغير، ويجب أن تبدأ بحرف كبير (Capital letter) دائماً.

ملحوظة:

نشاط نظري (5): أبحث عن الفرق بين ال Char وال char وأناقش ما توصلتُ إليه مع زملائي.



تعريف المتغيرات:

يتمّ الإعلان والتعريف بالمتغيرات التي سيتم استخدامها في البرنامج عن طريق الكلمة المحجوزة var أو val. الجدول الآتي يوضّح الطرق المختلفة لتعريف المتغيرات:

الشكل العام	مثال
var variableName:variableType=value القيمة الابتدائية = نوع المتغير: اسم المتغير	var salary:Double=2000 val name:String="Ahmad"
var variableName:variableType نوع المتغير: اسم المتغير	var salary:Double val name:String
var variableName = value القيمة الابتدائية = اسم المتغير	var salary=2000 val name="Ahmad"
ملحوظة: إذا لم يحدّد نوع المتغير عند التعريف، فإنّ كوتلن يتعرّف تلقائياً على نوع المتغير من القيمة الابتدائية المسندة له.	

في حال عدم تحديد نوع المتغير فإنه يمكننا:

- ◀ تمييز القيم العشرية Float عن Double باستخدام حرف F مع العدد الحقيقي من نوع Float
- ◀ تمييز القيم الصحيحة Long عن Int باستخدام حرف L مع العدد الصحيح الطويل.

نشاط نظري (6):

أكمل الجدول الآتي بذكر نوع المتغير في كلّ مثال:

نوع المتغير	تعريف المتغير
Double	val distance= 50.5
Char	val symbol= 'a'
	var x="xyz"
	val avg=90F
	var width=200
	var length=150L



الفرق بين val و var:

var	val
يمكن إسناد قيم مختلفة للمتغير خلال البرنامج.	يُستخدَم المتغير كثابت حيث يُسند إليه قيمة واحدة فقط خلال البرنامج.

◀ مثال: يوضح الفرق بين val و var

صواب: يمكن إسناد أكثر من قيمة للمتغيرات من نوع var	var lang:String="English" lang="Arabic"
خطأ: لا يجوز إسناد قيمة ثانية للمتغير من نوع val	val lang:String="English" lang="Arabic"

نشاط نظري (7): أكمل الجدول بجمل التعريف الممكنة في كل حالة:



الحلّ	السؤال
var score:Int = 12 أو var score=12	تعريف متغير باسم score من نوع Int وقيمه الابتدائية 12
val avg:Float=80.9 أو val avg=80.9F	تعريف ثابت حقيقي باسم avg من نوع Float وإعطاؤه القيمة 80.9
	تعريف متغير نصي باسم day وإعطاؤه قيمة ابتدائية saturday
	تعريف ثابت باسم count من نوع Int
	تعريف متغير رمزي (Char) باسم opr وإعطاؤه قيمة ابتدائية @
	تعريف ثابت باسم temp من نوع Double
	تعريف متغير باسم id من نوع Long وإعطاؤه القيمة 123456789

القيم النصية (String) في البرنامج دائماً توضع بين علامتي التنصيص المزدوجة، مثال: "GoodMorning"
بينما القيم الرمزية (char) توضع بين علامتي التنصيص المفردة، مثال: 'b' ، 'y' ، '!'

ملحوظة:



جُمْل الطباعة(الإخراج) في كوتلن

print ()	طباعة النصّ الذي يحتويه
println ()	طباعة النصّ الحالي ثم الانتقال إلى سطر جديد

◀ مثال:

```
fun main(args: Array<String>) {  
    println("first");  
    println("second");  
    print("third\n");  
    print("fourth");  
    print("fifth");  
}
```

◀ المخرجات الناتجة عن تنفيذ البرنامج:

```
first  
second  
third  
fourth fifth
```

ملحوظة:

الرمز '\n' في جملة الطباعة يعني رمز السطر الجديد.

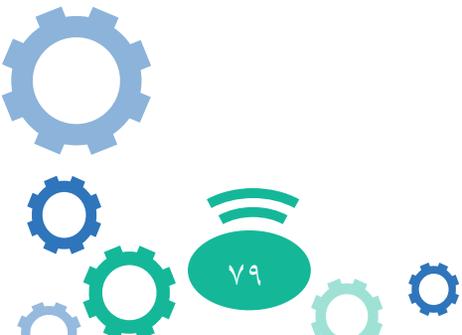
◀ لطباعة قِيَم المتغيّرات يُستخدم الأمر:

```
print(variableName)
```

◀ مثال:

```
val age=18
```

```
print("my age is $age")
```



◀ لتنسيق المخرجات تتم طباعة النصوص والمتغيرات معاً، ولتمييز المتغيرات عن النص المكتوب تتم إضافة إشارة \$ قبلها.
 مثال:

```
val age=18
print ("my age is $age")
```

◀ أو إضافة إشارة "+" لدمج العناصر المطبوعة مع بعضها

```
print("my age is"+age)
```

◀ مثال: برنامج يوضح طرق تنسيق جمل الإخراج

```
fun main(args : Array<String>) {
    val score = 12.3
    println("score")
    println("$score")
    println("score = $score")
    println("${score} + score")
    println(12.3)
    println("score="+score+" point")
}
```

◀ المخرجات الناتجة عن تنفيذ البرنامج:

```
score
12.3
score = 12.3
24.6
12.3
score = 12.3 point
```

نشاط عملي (8): أكمل الجدول الآتي بجمل الطباعة اللازمة لحلّ كلّ مسألة:



المسألة	الحلّ
طباعة الشكل الآتي:	<pre>print ("1\n 2\n 3\n 4\n 5")</pre>
طباعة الشكل الآتي:	<pre> x xxx xxxxx</pre>
طباعة اسمي الرباعي، بحيث يكون كلّ اسم على سطر منفصل (باستخدام جملة طباعة واحدة فقط).	



جمل القراءة (الإدخال) في كوتلن

الدالة () readLine تقوم بقراءة النص المدخل كسلسلة من الحروف (String)

◀ مثال: برنامج لقراءة نص وطباعته

```
fun main(args: Array<String>) {  
    print("Enter text:")  
    val stringInput = readLine()  
    println("You entered: $stringInput")  
}
```

التعليقات (Comments):

هي جمل توضيحية أضيفها إلى أي مكان في الكود البرمجي أثناء كتابته، والهدف منها تسهيل قراءة الكود البرمجي سواءً للشخص الذي قام ببنائه، أو حتى لأي شخص آخر يحاول فهمه أو التعديل عليه.

التعليقات يتم تجاهلها في البرنامج أثناء تنفيذ الكود.

ملحوظة:

فائدة التعليقات:

- تسهيل فهم الكود البرمجي من قبل المبرمج أو اشخاص آخرين.
- توفير الوقت والجهد إذا ما تمّ عمل إصلاح في الكود البرمجي.

◀ مثال: توضيح فائدة التعليقات

لأفترض أنني قد كتبت برنامجاً يحتوي على ٥٠ سطراً برمجياً، وأردت بعد فترة من الزمن استخدام جزء منه أو إصلاحه، فإذا لم أكتب من البداية التعليقات أجد صعوبة في فهم الكود، وأستغرق وقتاً أطول، ولتجنّب ذلك أكتب التعليقات لتوضيح عمل الأمور التي قد تواجهني مشاكل في فهمها مستقبلاً، أو إذا كنتُ أعمل ضمن فريق، وأخذ زملائي الكود البرمجي يستطيعون فهمه بسهولة وبوقت قصير.

أنواع التعليقات:

١ . تعليق في سطر واحد فقط: يجب أن يبدأ التعليق ب (//) ، وبعدها أكتب التعليق بلغة مناسبة، ويمكن استخدام اللغة العربيّة، أو الإنجليزيّة، أو غيرها...

```
fun main(args: Array<String>) {  
    // Hello السطر الآتي يقوم بطباعة كلمة Hello  
    println("Hello")  
  
    println(5) // this line print number 5  
  
    // val x= 10  
}
```

لاحظ من المثال السابق أنني أستطيع أن أكتب في التعليق أيّ شيء، وبأيّ لغة كتابة وفي أيّ مكان، وأيضاً أستطيع تعليق سطر برمجيّ لا أريد استخدامه.

٢ . التعليق في أكثر من سطر: لا يختلف عن التعليق بسطر واحد من حيث الاستخدام، ولكن يختلف بطريقة العمل، فإذا أردت التعليق بأكثر من سطر أضعه بين هذه الإشارات (/* ... */)، كما يأتي:

```
/*  
1-  
2-  
var x = 7  
*/
```

استخدم طريقة التعليق بأسطر متعدّدة غالباً في بداية الكود، لشرح عمل الكود بخطوات مفهومة، أو لتعليق عدة أسطر برمجيّة لا أريد استخدامها.



دوال للتحويل بين أنواع البيانات

toShort () toInt () toLong () toFloat () toDouble () toChar ()

يمكن تحويل Int الى String، مثلاً يمكن تحويل 3 إلى "3" وقد يتساءل البعض ما الفرق بينها، الأول قابل لإنجاز العمليات الحسابية، كإضافة رقم أمّا الثاني فلا، لكن لا يمكن تحويل حرف مثلاً "Z" إلى رقم لأنه ليس رقماً .

```
fun main(args: Array<String>) {
    val number1="3"
    println(number1.toDouble())
    println(number1.toInt()+number1.toInt())
}
```

المخرجات: ◀

3.0

6

يمكن استخدام دوال التحويل مع جمل القراءة أيضاً، ولكن في هذه الحالة يجب استخدام الإشارة !! وهي تُستخدم للتأكيد على أنّ القيمة المدخلة ليست null حتى يجري عملية التحويل، غير ذلك يخرج من البرنامج.

```
readLine ( )!!.toLong ( )
readLine ( )!!.toFloat ( )
readLine ( )!!.toDouble ( ) .....
```

◀ مثال: برنامج لقراءة العمر الحالي وطباعته

```
fun main(args: Array<String>)
{
    print("Enter your age:")
    val age:Int = readLine()!!.toInt()
    println("Your age: $age")
}
```

الرموز والعمليات الحسابية (Arithmetic operators):

أولوية التنفيذ	مثال	الوظيفة	رمز العملية الحسابية في كوتلن
1	++a --a	الزيادة بمقدار واحد النقصان بمقدار واحد	++ (القبلي) -- (القبلي)
2	a*b a/b a%b	الضرب القسمة باقي القسمة	× / %
3	a+b a-b	الجمع الطرح	+ -
4	a++ a--	الزيادة بمقدار واحد النقصان بمقدار واحد	++ (البعدي) -- (البعدي)

- إن وُجِدَت الأقواس () في العملية الحسابية فإن أولوية التنفيذ تبدأ منها.
- في حال تساوت الأولوية يكون اتجاه تنفيذ العملية الحسابية من اليسار إلى اليمين.

◀ مثال: توضيح أولويات العمليات الحسابية

$$x = 5 + \underline{4 \times 2} - 15 + 30 / 2 \times 3$$

$$x = 5 + 8 - 15 + \underline{30 / 2} \times 3$$

$$x = 5 + 8 - 15 + \underline{15 \times 3}$$

$$x = \underline{5 + 8} - 15 + 45$$

$$x = \underline{13} - 15 + 45$$

$$x = \underline{-2} + 45$$

$$x = 43$$



إسناد القيم (Assignment operators)

رمز العملية	المعنى	تمثيله برمجياً	المكافئ (رياضياً وبرمجياً)
+=	الجمع	a+=b	a=a+b
-=	الطرح	a-=b	a=a-b
×=	الضرب	a×=b	a=a×b
%=	باقي القسمة	=%a	a=a%b

نشاط نظري + عملي (9):

أجري العمليات الحسابية الآتية لإيجاد قيمة المتغير a في كل حالة، علماً بأن a=10 و b=5.

المعادلة الحسابية	النتيجة
a=10-b×2	
a%=b-2	a=a%(b-2) a=10%3 a=1
a/=6-3×2+8	
a=(2+b)/4+2	

عمليات الزيادة والنقصان

رمز العملية	تمثيله برمجياً	المكافئ (رياضياً وبرمجياً)
++	x++ ++x	x=x+1
--	x-- --x	x=x-1

الفرق بين (x++, ++x)

++x : عند احتواء جملة برمجية على قيمة ++x، يقوم البرنامج بتنفيذ الجملة باستخدام آخر قيمة لـ x قبل الزيادة، وبعد التنفيذ، يتم زيادة الرقم 1 لـ x

◀ مثال توضيحيّ

```
var a: Int
a=2
println(a++)
println(a)
```

◀ المخرجات:

2
3

++x : عند احتواء جملة برمجية على قيمة ++x، يقوم البرنامج بتنفيذ الجملة باستخدام القيمة الجديدة لـ x (أي) بعد زيادة الرقم 1 لـ X).

◀ مثال توضيحيّ

```
var a: Int
a=2
println(++a)
println(a)
```

◀ المخرجات:

3
3

◀ مثال: برنامج يوضّح كيفية التّأثر بمقدار الزيادة

```
fun main(args: Array<String>) {
var a=2
var b: Int
b=3+a++
println("a=$a b=$b")
b= ++a + b
println("a=$a b=$b")
}
```

a=3 b=5

◀ المخرجات:



الفرق بين (--x, x--)

X--: عند احتواء جملة برمجية على قيمة X--، يقوم البرنامج بتنفيذ الجملة باستخدام آخر قيمة لـ X قبل الطرح، وبعد التنفيذ، يتم طرح الرقم 1 من X

◀ مثال توضيحي

```
var a: Int
a=2
println(a--)
println(a)
```

◀ المخرجات:

```
2
1
```

عند احتواء جملة برمجية على قيمة --X، يقوم البرنامج بتنفيذ الجملة باستخدام القيمة الجديدة لـ X قبل الطرح، (أي بعد طرح الرقم 1 من X).

◀ مثال توضيحي

```
var a: Int
a=2
println(--a)
println(a)
```

◀ المخرجات:

```
1
1
```



◀ مثال: برنامج يوضّح كيفيّة التّأثّر بمقدار النقصان

```
fun main(args: Array<String>) {  
    var a=2  
    var b:Int  
    b=3+a--  
    println("a=$a b=$b")  
    b= --a + b  
    println("a=$a b=$b")  
}
```

◀ المخرجات:

a=1 b=5

a=0 b=5

نشاط نظري + عملي (10):



إيجاد قيم المتغيّرات في لحظة تنفيذ كلّ من التعبيرات الرياضيّة الآتية، علماً بأنّ $a=5, b=7, c=3$

قيم المتغيّرات	التعبير الرياضي
$d=8-5=3, b=8, a=5$	$d=++b - a++$
$d=6 \times 7 + 7 = 49, b=7, a=7$	$d= ++a \times b + ++a$
$d=, c=, a=$	$d=a-- - --c$
$d=, c=, a=$	$d=--a +--c$
$d=$	$d+= ++d$
$d=, b=$	$--d= b-- - b$



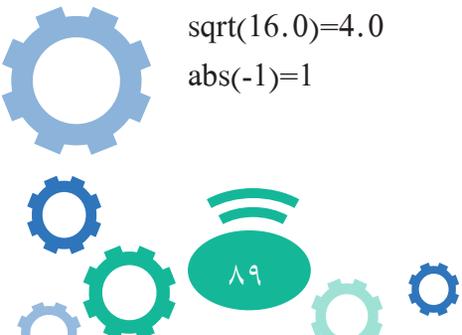
وظيفتها	الدالة
حساب القيمة المطلقة	Math.abs ()
إرجاع القيمة الأكبر بين عددين	Math.max ()
إرجاع القيمة الأصغر بين عددين	Math.min ()
إرجاع عدد عشوائي محصور في الفترة [0,1]	Math.random ()
حساب الجذر التربيعي.	Math.sqrt ()

◀ مثال: برنامج يستخدم بعض الدوال الرياضية الجاهزة

```
fun main(args: Array<String>) {
    val a:Int
    val b:Int
    a=16
    b=9
    var x:Double = Math.max(a,b).toDouble()
    println("max($a,$b)=" + x)
    x=Math.min(a,b).toDouble()
    println("min($a,$b)="+x)
    x=Math.sqrt(a.toDouble())
    println("sqrt(${a.toDouble()})="+x)
    println("abs(-1)="+Math.abs(-1))
}
```

◀ المخرجات الناتجة عن تنفيذ البرنامج

```
max(16,9)=16.0
min(16,9)=9.0
sqrt(16.0)=4.0
abs(-1)=1
```

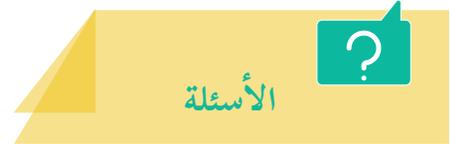




أكمل الجدول الآتي لبناء برنامج يقوم بقراءة اسم الطالب، وعلاماته الثلاث، وطباعة اسمه ومعدله.

fun main(args : Array<String>) {	فتح الدالة الرئيسية تمهيداً لكتابة أكواد البرنامج داخلها
var name: String var n1: Double var n2: Double var n3: Double var avg: Double	تعريف المتغيرات اللازمة
	جمل القراءة اللازمة لإدخال اسم الطالب وعلاماته الثلاث
	حساب المعدل بناء على القيم المدخلة مسبقاً المعدل = مجموع العلامات/عددتها
println("Name=\$name\ nAverage=\$avg")	طباعة الاسم والمعدل
}	إغلاق الدالة الرئيسية

◀ أنفذ البرنامج باستخدام الحاسوب وتأكد من صحة النتائج.



1- أكتب برنامجاً يقوم بقراءة اسمي الرباعي، بحيث يتم إدخال كل جزء من الاسم بمتغير، ومن ثم طباعة اسمي الأول والأخير كاسم واحد (معاً في سطر واحد).

2- أكتب برنامجاً يقوم بحساب الجذور للمعادلة التربيعية $ax^2 + bx + c = 0$ ، علماً بأن الجذور تُحسب بالمعادلة الآتية:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$



الموقف التعليمي التعلّمي (3 - 2):

توظيف مبادئ أساسيات البرمجة في بناء برنامج 2.

وصف الموقف: طلب معلّم العلوم من مشغلٍ لتطبيقات الهواتف المحمولة مساعدته في تطوير برنامج لتحليل نتائج طلبته في امتحان نصف الفصل.

العمل الكامل			
خطوات العمل	وصف الموقف الصّفيّ	المنهجية (استراتيجية التعلّم)	الموارد
أجمع البيانات وأحلّها	<p>جمع البيانات من معلّم العلوم عن:</p> <ul style="list-style-type: none"> • أسس النجاح والرسوب. • علامة النجاح، عدد الطلبة • فئات الطلبة (عالٍ، متوسط، ..)، علامة الحدّ الأدنى والحدّ الأعلى لفئات الطلبة. <p>جمع البيانات عن:</p> <ul style="list-style-type: none"> • المتغيّرات البرمجية. • العمليات الحسابية، وأوليات تنفيذها. • الجمل المنطقية وطريقة صياغتها. • جمل الإدخال والإخراج. • جمل التحكم، جمل التكرار. • المصفوفات، الدوال. 	<ul style="list-style-type: none"> • حوار ومناقشة. • التعلّم التعاوني/ مجموعات. • البحث العلميّ. 	<ul style="list-style-type: none"> • وثائق (طلب معلّم العلوم، نشرات، كتب ومصادر متنوعة). • التكنولوجيا. • روابط إلكترونية متنوعة من مصادر موثوقة، حاسوب، شاشة عرض، إنترنت).
أخطّط وأقرّر	<ul style="list-style-type: none"> • تحديد العمليات التي تلزمها جمل التحكم في البرنامج (جملة if, when, ..). • تحديد الجمل المنطقية اللازمة. • الاتفاق على جمل التكرار (for, do..while). • تحديد المصفوفات الأحادية ونوعها وسعتها. • تعيين دوال مساعدة في تنفيذ الحسابات المتكررة. 	<ul style="list-style-type: none"> • تعلّم تعاوني/ مجموعات • حوار ومناقشة . • عصف ذهنيّ/استمطار الأفكار. 	<ul style="list-style-type: none"> • وثائق (طلب معلّم العلوم، نشرات، كتب ومصادر متنوعة). • الشبكة • العنكبوتية.

<ul style="list-style-type: none"> • قرطاسية • الطلب الخاص بمعلم العلوم • حاسوب، برامج خاصة بلغة Kotlin • (JDK، intelliJ) 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات • حوار ومناقشة. 	<ul style="list-style-type: none"> • اختيار جمل التحكم والتكرار المناسبة. • توظيف المصفوفات في تخزين أسماء الطلاب وعلاماتهم. • كتابة (الكود) لتعريف الدوال المناسبة. • توضيح المدخلات (علامات الطلبة، وعددهم). • إجراء عمليات جمع العلامات ومعدل الطلبة، وتصنيف كل طالب وفق مستويات التحصيل في كل فئة. • توضيح المخرجات (عدد الناجحين، عدد الراسبين، عدد الطلبة في كل فئة، وسط حسابي للطلبة، انحراف معياري للعلامات).. 	أنفذ
<ul style="list-style-type: none"> • إنترنت ، حاسوب • قرطاسية • الطلب الخاص بمعلم العلوم • برنامج خاص باللغة (intelliJ، JDK) 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات • حوار ومناقشة. 	<p>التحقق من صحة استخدام كل من:</p> <ul style="list-style-type: none"> - المتغيرات الرقمية والعمليات الحسابية. - جمل التكرار والتحكم، العبارات المنطقية، الدوال والمصفوفات. • تنفيذ البرنامج للتحقق من دقة النتائج. • اكتشاف الأخطاء وتصويبها واقتراح البدائل. 	أتحقق
<ul style="list-style-type: none"> • حاسوب. • شاشة عرض. • قرطاسية 	<ul style="list-style-type: none"> • حوار ومناقشة. • تعلم تعاوني/مجموعات • عمل. 	<p>توثيق البيانات التي تتعلق بـ:</p> <ul style="list-style-type: none"> - تصنيفات جمل التحكم والتكرار والمصفوفات والدوال. - خطوات بناء البرنامج وتحليل النتائج. • إعداد ملف يوضح خطوات البرنامج الخاص بتحليل علامات الطلبة. • إعداد العروض التقديمية الخاصة. • إعداد ملف بالحالة (تطوير برنامج لتحليل نتائج الطلبة). 	أوثق وأقدم
<ul style="list-style-type: none"> • نماذج للتقويم • مُعدة سابقاً. 	<ul style="list-style-type: none"> • حوار ومناقشة. 	<ul style="list-style-type: none"> • مقارنة آلية تحليل علامات الطلبة قبل البرنامج وبعده. • إيجاد تحليل دقيق لنتائج الطلبة لمادة أخرى بالبرنامج نفسه. • رضا المعلم عن البرنامج. 	أقوم

أحدّد جمل برمجية مناسبة لإجراء حسابات تخفيضات على أسعار مجموعة من الكتب.



توظيف مبادئ أساسيات البرمجة في بناء برنامج 2

نشاط نظري + عملي (1):

أبحث عن جمل التحكم الخاصة بلغة كوتلن باستخدام الشبكة العنكبوتية، وأعد عرضاً تقديمياً حولها.

الرموز المنطقية (Logical operators)

العبارات المنطقية هي العبارات التي تكون نتيجتها true أو false

الوصف	الرمز المنطقي
true إذا كانت x تساوي y	$x==y$
true إذا كانت x أكبر أو يساوي y	$x >= y$
true إذا كانت x أقل أو يساوي y	$x <= y$
true إذا كانت x أقل من y	$x < y$
true إذا كانت x أكبر من y	$x > y$
true إذا كانت x لا تساوي y	$x != y$

رموز الربط بين العبارات المنطقية

الجواب	مثال
False	$5 > 3 \ \&\& \ 6 < 5$
True	$5 > 3 \ \ 6 < 5$
False	$!(5 > 3 \ \ 6 < 5)$

الرمز	المعنى
&&	AND
	OR
!	نفي العبارة المنطقية

جدول الصواب للعبارات المنطقية:

b1	b2	b1 && b2	b1 b2	b1!	b2!
True	True	True	True	False	False
True	False	False	True	False	True
False	True	False	True	True	False
False	False	False	False	True	True

- عند استخدام أداة الربط `&&`: ناتج العبارة المنطقية المركبة صحيحة فقط إذا كانت العبارتان المنطقيتان صحيحتين.
- عند استخدام أداة الربط `||`: ناتج العبارة المنطقية المركبة خاطئة فقط إذا كانت العبارتان المنطقيتان خاطئتين.

ملحوظة:

نشاط نظري (2):

أكمل الجدول الآتي:

الجواب	الجملة المنطقية
true	$2 == 2 \ \&\& \ 4 != 5$
	$4 > 5 \ \&\& \ 2 < 7$
	$!(7 > 12 \ \ 14 < 18)$
	$6 > 4 \ \&\& \ 4 > 5$
	$5 == 3 \ \ 6 < 7 \ \&\& \ !(2 > 5)$

المجال (Range)

هو مجموعة القيم المحصورة بين نقطة البداية ونقطة النهاية، علماً بأن قيمتي البداية والنهاية من ضمن قيم المجال.

أنواع المجالات

مجال متباعد		مجال تنازليّ		مجال تصاعديّ	
القيم الممثّلة للمجال	مثال برمجيّ	القيم الممثّلة للمجال	مثال برمجيّ	القيم الممثّلة للمجال	مثال برمجيّ
0, 5, 10, 15, 20	0..20 step 5	10, 9, 8, 7, 6, 5	10 downTo 5	1, 2, 3, 4, 5	1..5
100, 90, 80, 70, 60,50	100 downTo 50 step 10	D, C, B, A	'D' downTo 'A'	w, x , y, z	'x'..'w'

تعريف المتغيّرات من نوع مجال (Range)

المجال نوع خاصّ من أنواع البيانات، ويمكن إسنادها إلى المتغيّرات:
◀ مثال:

```
val odd=1..100 step 2
```

```
val digits= 0..9
```

نشاط نظري (3):

أكمل الجدول الآتي بقيم مجال البيانات التي تمّ تعريفها بلغة كوتلن في العمود الأوّل بما يقابلها في العمود الثاني

قيم المجال	تعريف مجال من البيانات
a, b, c, d, e, f, g, h	val atoz= 'a'..'h'
.....	val oneToTen = 1..10
1,3,5,7,9	val oneToTen2 = 1..10 step 2
.....	val tenToOne= 10 downTo 1
1, 4, 7, 10	val tenToOne2 =1..10 step 3
200,150,100,50,0	val down=200 downTo 0 step 50
.....	val down2= down step 2

جمل التحكم (الجمل الشرطية)

أولاً: (جمل IF الشرطية)

```
if (logical expression)
  stmt
```

الصيغة الأولى (جملة IF)

إذا: if:

عبارة منطقية: logical expr:

جملة بلغة كوتلن: stmt:

◀ في حال تحقق الشرط ينفذ الجملة stmt غير ذلك يكمل تنفيذ البرنامج دون تنفيذها.

◀ مثال: طباعة القيمة الأكبر بين قيمتين:

```
var max = a
```

```
if (a < b)
```

```
  max = b
```

```
print(max)
```

```
if (logical expression)
```

```
  stmt1
```

```
else
```

```
  stmt2
```

الصيغة الثانية (جملة if..else)

• في حال تحققت العبارة الشرطية تُنفذ جملة stmt1 ولا تُنفذ stmt2 ، وفي حال عدم تحقق العبارة الشرطية تُنفذ الجملة stmt2 ولا تُنفذ stmt1 .

• في حال وجود أكثر من جملة تابعة لجملة if أو جملة else ، فيجب وضعها بين الإشارتين { }



◀ مثال: طباعة القيمة الأكبر بين رقمين

```
var max: Int
if (a > b)
{
max = a
print(max)
}
else
{
max = b
print(max)
}
```

```
val varName = if(logical expression)
stmt1 else stmt2
```

الصيغة الثالثة (إسناد ناتج if..else لمتغير)

- إذا تحققت العبارة الشرطية فإن قيمة المتغير (varName) عبارة عن ناتج stmt1 غير ذلك عبارة عن ناتج stmt2

◀ مثال: طباعة الرقم الأكبر بين رقمين

```
val max = if (a > b) a else b
print(max)
```

في حال تحقق الشرط يصبح max=a غير ذلك يصبح max=b
◀ مثال آخر

```
val max = if (a > b) {
println("Choose a")
a
} else {
println("Choose b")
b
}
print(max)
```

في حال تحقق الشرط يطبع "Choose a" وتصبح قيمة max مساوية لقيمة a، غير ذلك فيطبع "Choose b" وتصبح قيمة max مساوية لـ b.



```
when(exp)
{
value1 -> stmt1
value2 -> stmt2
value3-> stmts3
...
else-> stmtn
}
```

الصيغة الاولى (جملة when التقليدية)

في حال تساوت قيمة التعبير `exp` مع أي من القيم الموجودة (`value1, value2, value3,...`) فسيتم تنفيذ `stmt` التابعة لها فقط، غير ذلك يتم تنفيذ `stmtn`.

◀ مثال: آلة حاسبة بسيطة

```
fun main(args: Array<String>) {
val no1:Int
val no2:Int
print("enter number1:")
no1= readLine()!!.toInt()
print("enter number2:")
no2= readLine()!!.toInt()
print("enter operator:")
val op= readLine()
val result:Any
when(op){
"+" -> result= no1+no2
"-" -> result= no1-no2
"×" -> result= no1× no2
"/" -> result= no1/no2
else -> result="invalid operator"
}
print(result)
}
```





اكتب برنامجاً يعمل على إدخال علامة الطالب في إحدى المواد، ويطبع تقديره في هذه المادة باستخدام الجملة السابقة، (جملة when):

نشاط عملي (4):

	فتح الدالة الرئيسية تمهيداً لكتابة أكواد البرنامج داخلها
var mark: Int	يتم تعريف المتغير
	يتم الطلب من المستخدم إدخال العلامة
when(mark) { in 90..100-> println("Excellent") in 80..89-> println("Very Good") }	فحص العلامة وطباعة النتيجة
	إنهاء البرنامج (إغلاق الدالة الرئيسية)
var varName= when(exp) { value1 -> stmt1 value2 -> stmt2 value3-> stmts3 . . . else-> stmtn }	الصيغة الثانية (اسناد ناتج جملة when لمتغير)

في حال تساوت قيمة التعبير exp مع أي من القيم الموجودة (value1, value2, value3,...) يتم اسناد قيمة stmt التابعة لها للمتغير varName غير ذلك يتم إسناد قيمة stmtn إلى المتغير varName

◀ مثال توضيحي:

```
val sol= when(a)
{
  in 1..5 -> "222"
  3 -> "333"
  2,4 -> "xxx"
  else -> "yyy"
}
print(sol)
```

◀ توضيح الكود

- يطبع 222 إذا كانت a في الفترة [1..5]
- يطبع 333 إذا كانت a تساوي 3
- يطبع xxx إذا كانت a تساوي 2 أو 4
- يطبع yyy إذا كانت a غير القيم السابقة

◀ مثال: آلة حاسبة بسيطة

```
fun main(args: Array<String>)
{
    val no1:Int
    val no2:Int
    print("enter number1:")
    no1= readLine()!!.toInt()
    print("enter number2:")
    no2= readLine()!!.toInt()
    print("enter operator:")
    val op= readLine()
    val result=when(op)
    {
        "+" -> no1+no2
        "-" -> no1-no2
        "*" -> no1* no2
        "/" -> no1/no2
        else -> "invalid operator"
    }
    print(result)
}
```

نشاط عملي (5): أكمل الجدول الآتي ببرنامج يقرأ رقم صحيح ويطبع كلمة "even" إذا كان زوجياً، وكلمة "odd" إذا كان فردياً، باستخدام جملة التحكم المذكورة في كل حالة:



باستخدام جملة if else التقليدية	باستناد ناتج if else إلى متغير	باستخدام when التقليدية	باستناد ناتج when إلى متغير
		<pre>fun main(args: Array<String>) { println("Enter Number:") val no:Int=readLine()!!.toInt() val msg:String= when(no%2) { 0-> "Even" else -> "Odd" } print("thnumber(\$no) is \$msg") }</pre>	



جمل الدوران (Loop Statements)

أولاً: (For statement)

الشكل العام:

يتمّ تكرار حلقة الدوران بعدد العناصر الموجودة في المجموعة (collection)، من الممكن أن تكون المجموعة عبارة عن عناصر المدى (range)، أو عناصر المصفوفة (array)، أو عناصر القائمة list... إلخ

```
for (variableName in collection)
{
    stmt1
    stmt2
    ⋮
    stmtN
}
```

◀ مثال: طباعة الأرقام من 20 إلى 50

```
fun main(args: Array<String>)
{
    val range= 20 ..50
    for (k in range)
    println(k)
}
```

المتغيّر k داخل حلقة التكرار لا يحتاج إلى جملة تعريف لتعريفه، ويتمّ استخدامه مباشرة.

◀ مثال: حلقة لطباعة مربّعات الأرقام من 1 إلى 05.

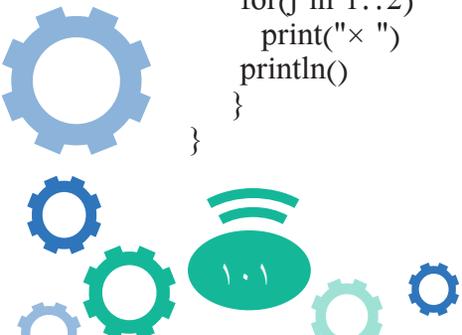
```
fun main(args: Array<String>)
{
    for(k in 1..50){
        println(k×k)
    }
}
```

◀ مثال: طباعة الأعداد الزوجيّة من 2 إلى 05 تنازلياً:

```
fun main(args: Array<String>)
{
    for(i in 50 downTo 1 step 2)
    println(i)
}
```

◀ مثال: طباعة الشكل الآتي باستخدام حلقات التكرار المتداخلة:

```
××
××
××
fun main(args: Array<String>)
{
    for(i in 1..3)
    {
        for(j in 1..2)
        print("× ")
        println()
    }
}
```



ثانياً: (While statement)

الشكل العام:

```
while(expression)
{
statements
}
```

- يتمّ الدخول إلى حلقة الدوران ما دامت العبارة المنطقية (expression) صحيحة.

◀ مثال: برنامج يقوم بطباعة كلمة "Hello" خمس مرات

```
fun main(args: Array<String>) {
var i: Int=1
while(i <=5)
{
println("Hello")
i++
}
}
```

ثالثاً: (Do..While Statement)

الشكل العام:

```
do
{
stmt1
stmt2
...
stmtN
}
while(expression)
```

- يتمّ الدخول إلى حلقة do..while على الأقلّ مرّة واحدة اعتماداً على الشرط المنطقيّ (expression)
- يتمّ الخروج من الحلقة عندما تكون ناتج العبارة المنطقية خاطئة.



◀ مثال: قراءة خمسة أرقام صحيحة وطباعة حاصل جمعها

```
fun main(args: Array<String> )
{
    var i:Int=0
    var sum:Int=0
    do
    {
        println("enter a number:")
        sum += readLine()!!.toInt()
        i++
    }
    while (i<5)
    println(sum)
}
```

نشاط عملي (6):



أكمل الجدول الآتي ببرنامج يقوم بطباعة الأعداد الزوجية من 100..1 بشكلٍ تنازليّ باستخدام حلقة الدوران المذكورة في كلّ حالة.

For statement	Do..while statement	While statement
	<pre>fun main(args: Array<String>) { var i:Int=100 do { println(i) i-=2 } while (i>=1) }</pre>	





أكمل البرنامج الآتي الذي يعمل على إدخال علامات طالب في خمسة مساقات، ثم يطبع المجموع والمعدل.

فتح الدالة الرئيسية تمهيداً لكتابة أكواد البرنامج داخلها	
	يتم تعريف المتغيرات: mark و sum و avg من نوع Double، مع التأكيد على تعيين قيمة (0) للمتغير sum
<pre>for(i in 1..5) { print("Please enter mark \$i :") mark= readLine()!!.toDouble() sum=sum+mark }</pre>	عمل حلقة لإدخال 5 علامات وجمعها مع بعضها بعضاً.
	إيجاد المعدل بقسمة المجموع على 5 وتخزينها في المتغير (avg)
<pre>println("sum = \$sum\n avg = \$avg")</pre>	طباعة الناتج
	إغلاق الدالة الرئيسية

سؤال

عندما تدخل العلامات هل يحتفظ بها برنامجك؟ كيف ستجري عليها حساباتك فيما بعد (طباعة العلامات، أعلى علامة، أقل علامة...)?



المصفوفات

أشكال المصفوفات:

1. سلاسل رقمية ونصية:

◀ أمثلة:

```
val numbers = listOf(1,10,7,4,15,20)
val days=listOf("saterday","Sunday","Monday","Tuesday","Thursday")
val animals= listOf("Dog","Cat","Horse","Lion")
```

2. مصفوفة ثابتة القيم

```
val array = arrayOf("Saturday","Sunday","Monday")
```

سعتها 3 عناصر، وعناوين عناصرها:

```
array[0], array[1], array[2]
```

لطباعة طول المصفوفة:

```
println(array.size)
```

يمكن تجميع عناصر مصفوفتين في مصفوفة واحدة باستخدام العملية + (شرط أن يكون من نوع البيانات نفسها).

```
val array1 = arrayOf("Friday","Thursday")
```

```
val array2 = array+array1
```

عبارة منطقية تحتمل الصواب والخطأ تعني: هل تحتوي المصفوفة على هذا العنصر؟

```
array.contains("Sunday")
```

3. مصفوفة العناصر

```
val arrayList = arrayListOf("saterday","Sunday","Monday")
```

سعتها ثلاثة عناصر وعناوين هذه العناصر:

```
arrayList[0], arrayList[1], arrayList[2]
```

يمكن إضافة عناصر جديدة لهذا الشكل من المصفوفات:

```
arrayList.add("Thursday")
```

يمكن حذف عنصر من هذه المصفوفة:

```
arrayList.remove("Friday")
```

لفحص إذا كانت المصفوفة فارغة أم لا:

```
arraylist.isEmpty()
```



1. المصفوفة العادية

لهذا النوع سعة معيّنة، ونوع عناصر محدد وقيم ابتدائية لهذه العناصر.

```
var arrayName = Array<items data type>(size,{intial value})
```

◀ مثال: مصفوفة باسم y من نوع عدد صحيح بحجم 5 عناصر بقيمة ابتدائية لكل عنصر تساوي 1

```
var z = Array<Int>(4,{1})
for(i in z.indices){
z[i] = i
println(z[i])
}
```

الكلمة المحجوزة `indices` تعني فهارس عناصر المصفوفة، أي مواقعها في المصفوفة وترتيبها وتبدأ من 0-3 في المصفوفة السابقة.

◀ مثال: استخدام المصفوفة في تخزين 4 علامات وجمعها وإيجاد معدلها.

```
fun main(args: Array<String>)
{
var sum:Double =0.0
var j:Int=0
var z = Array<Double>(4,{0.0})
for(i in z.indices){
z[i] = readLine()!!.toDouble()
sum += z[i]
j++
}
println(j)
var avg:Double
avg = sum/j
println(sum)
println(avg)
}
```



نشاط عملي (8):



اكتب برنامجاً يعمل على إدخال علامات طلبة صفك ويعمل على طباعتها، وإيجاد مجموعها ومعدلها، والانحراف المعياري لها.

	تعريف المتغيرات
	إدخال عدد الطلاب
<pre>var marks = Array<Double>(count,{0.0})</pre>	تعريف مصفوفة بعدد الطلاب وتعيين القيمة 0.0 لجميع العناصر
	إدخال علامات الطلاب وجمعها
	إيجاد معدل الطلاب
<pre>var variation:Double=0.0 for(i in 0..count) { variation+=Math.abs(marks[i]-avg) } variation/=count</pre>	إيجاد التباين (معدل انحراف القيم عن وسطها الحسابي)
<pre>var std:Double= Math.sqrt(variation)</pre>	إيجاد الانحراف المعياري وهو الجذر التربيعي للتباين
	طباعة العلامات والمجموع والمعدل والانحراف المعياري
<pre>Please enter the student count : 3 Please enter mark # 1 : 55 Please enter mark # 2 : 66 Please enter mark # 3 : 77 Marks : 556677 Sum= 198 avg = 66.0 std = 8.981462390204987</pre>	النتيجة

نشاط للاطلاع (9):



أنشئ عرضاً تقديمياً يوضح الفروقات بين كلٍّ من المصفوفات الآتية: `arrayOf`، `listOf` وأقوم بعرضه أمام الطلاب.



الدوال (Functions) :

أشكال الدوال: هناك شكلان من الدوال، فبعضها لا يرجع قيمة والبعض الآخر يرجع قيمة عند الانتهاء من التنفيذ.

1. الدوال التي لا ترجع قيمةً.

الشكل العام لتعريف الدالة:

```
fun funName(varName: varType,....)
{
    stmt1
    stmt2
    .....
    .....
}
```

المعاملات الشكلية
المستخدمة للدالة هي
معاملات اختيارية

الشكل العام لاستدعاء الدالة:

```
funName(varName,.....)
```

تحتوي جملة الاستدعاء على معاملات فعلية عددها عدد المعاملات
الشكلية نفسها في تعريف الدالة، ولها نوع البيانات نفسها.

◀ مثال: دالة لطباعة النص "Palestine in the heart"

```
fun printPalestine()
{
    println("Palestine in the heart")
}
```



◀ استدعاء الدالة

```
printPalestine()
```

◀ مثال: دالة لطباعة جملة ترحيبية لشخص ما

```
fun welcome(name:String)
{
println("welcome " + name + " to Palestine")
}
```

◀ استدعاء الدالة

```
Welcome("Ali")
```

◀ مثال: برنامج يوضح كيفية استخدام الدوال واستدعائها في كوتلن

```
fun printPalestine()
{
println("Palestine in the heart")
}
fun welcome(name:String)
{
println("welcome " + name + " to Palestine")
}

fun main(args: Array<String>)
{
printPalestine()
welcome("Ali")
}
```

◀ المخرجات

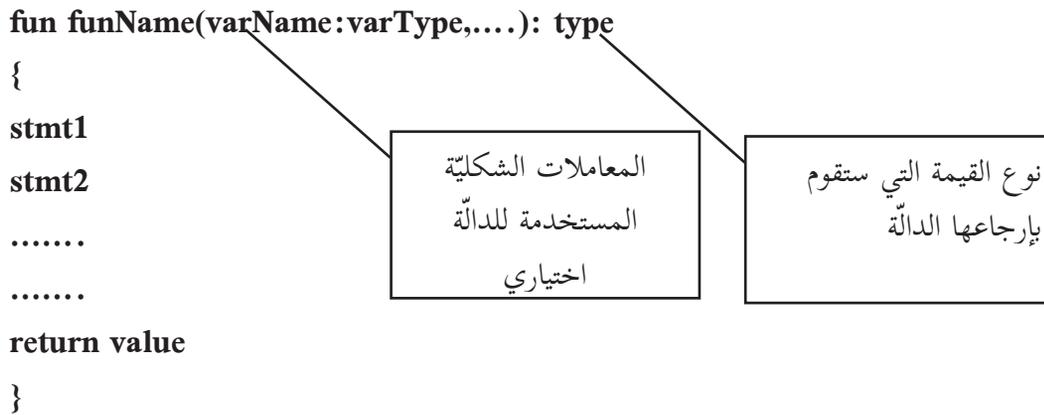
```
Palestine in the heart
welcome Ali to Palestine
```



إذا تمّ تعريف دالة جديدة داخل الدالة الرئيسية main، فلا يمكن استدعاؤها إلا من داخل الدالة الرئيسية فقط. أما إذا تمّ تعريفها خارج الدالة الرئيسية، فعندها يمكننا استدعاؤها من أيّ مكان في البرنامج.

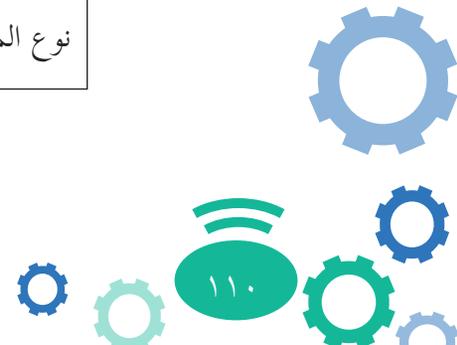
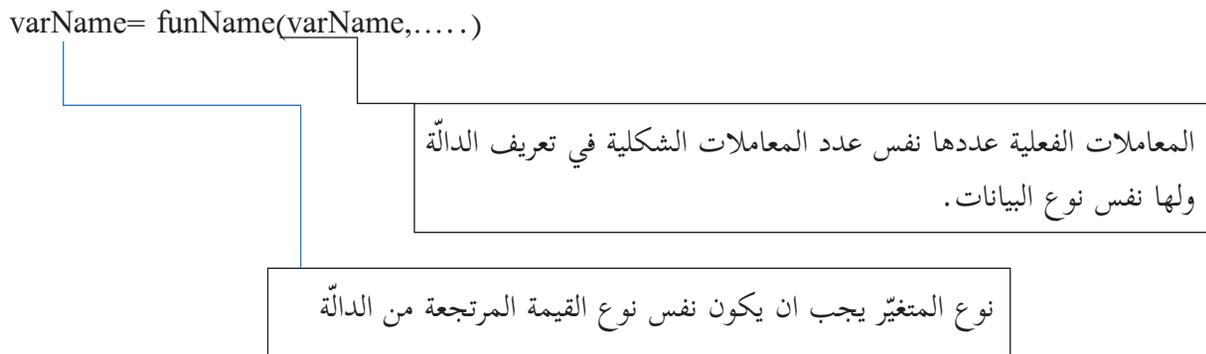
2. الدوال التي ترجع قيماً:

الشكل العام لتعريف الدالة:



يجب استخدام جملة **return value** لتحديد القيمة المرتجعة (value) للدالة، ويجب أن تكون من نوع القيمة نفسها المرتجعة للدالة عند التعريف.

الشكل العام لاستدعاء الدالة:



كما يمكن طباعة القيمة المرتجعة (أي الاستدعاء) داخل جملة الإخراج.
مثال: دالة ترجع مربع رقم

```
fun sqr(i:Int):Int
{
return i×i
}
```

◀ مثال يوضح طرق مختلفة لاستدعاء دالة ترجع مربع رقم:

<pre>fun sqr(x:Int):Int { return x×x } fun main(args: Array<String>) { var k:Int k=sqr(4) print(k) }</pre> <p>المخرجات 16</p>	<pre>fun sqr(x:Int):Int { return x×x } fun main(args: Array<String>) { println(sqr(10)) }</pre> <p>المخرجات 100</p>	<pre>fun main(args: Array<String>) { val a=5 var b=6 val sum= sqr(a)+sqr(b) print(sum) } fun sqr(x:Int):Int { return x×x }</pre> <p>المخرجات 61</p>
-----------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------



نشاط نظري (10):



بالاستعانة بمفهوم الدوال أكمل الجدول الآتي بالدوال المطلوبة:

<pre>fun power(x:Int, y:Int):Int { var r:Int=1 for(i in 1..y) r= r*x return r }</pre>	دالة لحساب قيمة x^y
	دالة لإرجاع الرقم الأكبر بين رقمين حقيقيين.
	دالة تستقبل اسمك الأول والأخير ويطبع اسمك كاملاً.
	دالة يرجع حاصل جمع الأعداد الصحيحة من 1 إلى n
	دالة تطبع الأرقام من 100..1
	دالة تستقبل رقمين صحيحين ويطبع ما يأتي: حاصل الجمع، حاصل الطرح، حاصل الضرب حاصل القسمة

نشاط عملي (11):



أتأكد من صحة الدوال في النشاط السابق، وذلك بتطبيق كل منها على حدة، من خلال كتابتها من ضمن برنامج كامل وتنفيذه باستخدام IntelliJ IDEA

الأسئلة

1. اكتب برنامجاً يقوم بقراءة 10 أرقام صحيحة، وبعد إدخالها يقوم بطباعة ما يأتي:
 - قائمة بالأرقام المدخلة بشكل معاكس.
 - طباعة أول خمسة ارقام تم إدخالها.
 - طباعة آخر خمسة أرقام تم إدخالها.
2. اكتب برنامجاً يقوم بقراءة 3 أرقام صحيحة، وطباعة الرقم الأصغر من بينها، من خلال استدعاء دالة تقوم بإرجاع القيمة الصغرى للقيم الثلاثة.



الموقف التعليمي التعلّمي (3 - 3):

توظيف مبادئ (البرمجة الكائنية) في بناء برنامج 1.

وصف الموقف التعليمي: زار معلّم التربية البدنية (الرياضة) مشغلاً لتطبيقات المحمول في مدرسة يافا المهنية، لمساعدته في بناء برنامج يساعده في احتساب الدرجات والعلامات للطلاب.

العمل الكامل			
الموارد	المنهجية (استراتيجية التعلّم)	وصف الموقف الصّفي	خطوات العمل
<ul style="list-style-type: none"> • وثائق (طلب المعلم، دفتر علامات السنة الماضية، نشرات، كتب ومصادر متنوعة عن أسس البرمجة وآلية بناء البرامج). • التكنولوجيا (روابط إلكترونية متنوعة من مصادر موثوقة، حاسوب، شاشة عرض، إنترنت) 	<ul style="list-style-type: none"> • حوار ومناقشة. • التعلّم التعاوني/ مجموعات • البحث العلمي. 	<p>جمع البيانات من معلّم التربية البدنية عن:</p> <ul style="list-style-type: none"> • الطلاب. • آلية احتساب العلامات. <p>جمع البيانات عن:</p> <ul style="list-style-type: none"> • مفهوم البرمجة الكائنية وأسسها. • خصائص البرمجة الكائنية: - الصنف / النوع (class) - الكائن (object) - الخصائص (المتغيرات) المستخدمة. - الدوال (functions). - إمكانية الوصول . 	أجمع البيانات وأحلّها
<ul style="list-style-type: none"> • وثائق (طلب المعلم، دفتر العلامات، نشرات، كتب ومصادر متنوعة) قرطاسية. • حاسوب. • الشبكة العنكبوتية. 	<ul style="list-style-type: none"> • تعلّم تعاوني/ مجموعات • حوار ومناقشة . • عصف ذهني/ استمطار الأفكار. 	<ul style="list-style-type: none"> • تحديد المعلومات المهمة عن الطلاب. • تحديد العناصر المهمة في احتساب العلامة. • تحديد أوقات وآليات احتساب العلامة. • تحديد طريقة إخراج الناتج المطلوب (العلامة النهائية للطلاب). 	أخطّط وأقرّر
<ul style="list-style-type: none"> • قرطاسية. • الطلب الخاصّ بالمعلّم. • دفتر العلامات • حاسوب، برنامج خاصّ intelliJ 	<ul style="list-style-type: none"> • تعلّم تعاوني/ مجموعات • حوار ومناقشة. • كتابة. 	<ul style="list-style-type: none"> • اختيار الأصناف (classes) (الصفوف، الطلاب). • توضيح الخصائص والعمليات لكلّ صنف. • اختيار إمكانية الوصول المناسبة لكلّ من الخصائص والعمليات (private, protected, public). • كتابة الكود الذي يحقق المطلوب. 	أنفّذ



<ul style="list-style-type: none"> • إنترنت حاسوب. • الخطة المُعدَّة سابقاً. • قرطاسية. • الطلب الخاصّ بالمعلّم • حاسوب، برنامج خاصّ intelliJ 	<ul style="list-style-type: none"> • تعلّم تعاونيّ/ مجموعات • حوار ومناقشة. 	<ul style="list-style-type: none"> • التحقُّق من ملاءمة الأصناف (classes). • التحقُّق من الخصائص والعمليات لكلّ صنف. • التأكد من وضع إمكانية الوصول المناسبة لكلّ من الخصائص والعمليات. • التحقُّق من كتابة الكود بطريقة صحيحة. • التحقُّق من تشغيل البرنامج وصحة النتائج. • اكتشاف الأخطاء وتصويبها، واقتراح البدائل. 	<p>أُتحقَّقُ</p>
<ul style="list-style-type: none"> • حاسوب • شاشة عرض • قرطاسية 	<ul style="list-style-type: none"> • حوار ومناقشة. • تعلّم تعاونيّ/ مجموعات 	<p>توثيق البيانات التي تتعلق بـ:</p> <ul style="list-style-type: none"> • صفات وميزات البرمجة الكائنية. • صفات البرنامج المطلوب. • المدخلات والمخرجات والعمليات التي يقوم بها البرنامج. • الأصناف وخصائصها وعملياتها. - إعداد العروض التقديمية. - اعداد ملفّ بالحالة (برنامج يساعد في احتساب درجات وعلامات الطلاب). - فتح سجلّ خاصّ يُعنى ببرامج احتساب العلامات. 	<p>أوثقُ وأقدِّمُ</p>
<p>نماذج للتقويم (استبانة، ...) طلب المعلّم</p>	<p>- حوار ومناقشة.</p>	<ul style="list-style-type: none"> • نجاعة البرنامج في احتساب العلامة للطلاب، وإمكانية تطبيقه من قِبَل أيّ معلّم تربية رياضية لاحتساب علامة الطلاب. • رضا المعلم عن البرنامج. 	<p>أقوِّمُ</p>



لماذا يُفضَّل استخدام سماحيّة الوصول (Private) للخصائص و(Public) للدوال؟



(الأصناف (Classes) والكائنات (Objects))

البرمجة الكينونية (Object-oriented Programming (OOP)):

في نمط البرمجة الكينونية ، يمكنك تقسيم مشكلة معقدة إلى مجموعات أصغر عن طريق إنشاء كائنات وأصناف.

الكيان (Object):

هو نسخة من صنف (Class) معين تحتوي على دوال (Functions) وبيانات (Data).... ، وكلّ كيان (Object) له ميزتان رئيسيتان:

- الحالة أو الوضع (state): هي حدث يخصّ الكائن، وينتج عن سلوك معين، وله خصائص معينة.
- السلوكيات (behavior): هي الدوال الخاصة بإعطاء أمرٍ ما للوصول إلى حالة، أو وضع محدد.

◀ مثال:

افتراض أنّ الإنارة كائن (object) :

- 1- إما أن تكون الإنارة في وضع تشغيل أو إغلاق، وتُسمّى هذه حالة كائن الإنارة.
- 2- أستطيع تشغيل أو إغلاق الإنارة، وهذه تُسمّى سلوكيات.

◀ مثال آخر:

افتراض أنّ سيّارة من نوع (س) كائن (object):

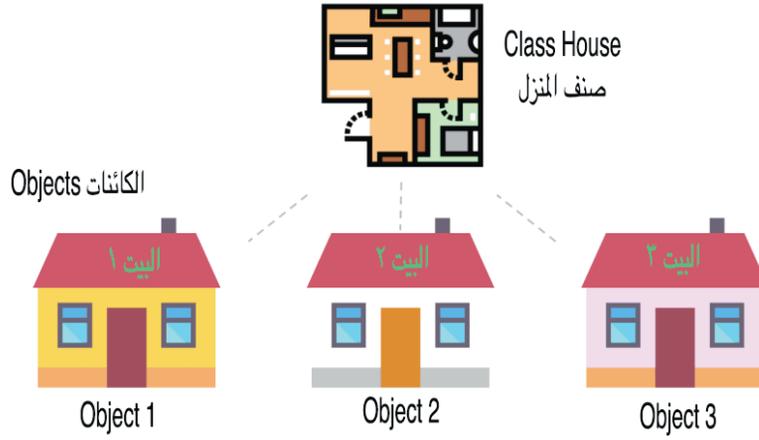
- 1- تحتوي السيّارة على أربعة إطارات، ومقود... الخ (خصائص)، إما أن تكون في وضع تشغيل وتسيير، أو في وضع إيقاف ولا تعمل (الحدث)، هذه هي حالة كائن السيّارة.
- 2- أستطيع تشغيل السيّارة وقيادتها.... الخ، وهذه تُسمّى سلوكيات.



الصَّنْف (Class):

هو القالب أو البناء أو الهيكل الأساسي الذي يتم إنشاء الكائنات (Objects) منه.

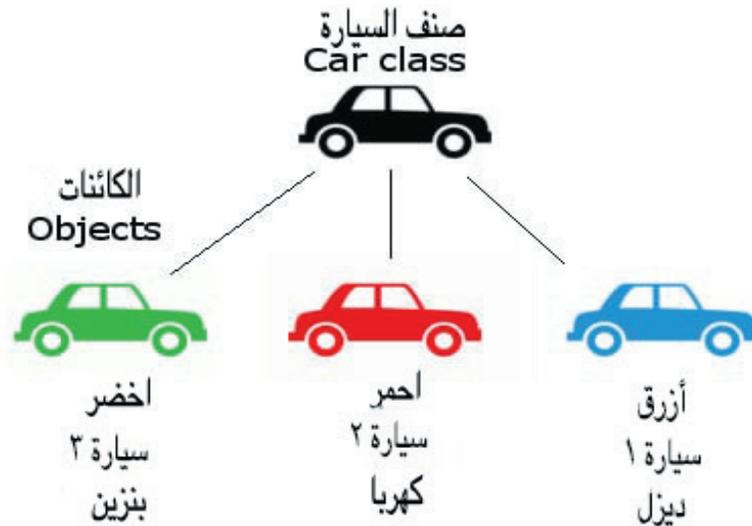
◀ **مثال:** أستطيع عمل صنف كقالب أساسي اسمه منزل، المنزل بشكل عام يتكوّن من: مساحة محدّدة، ونوافذ، وأبواب، وبناءً على هذه الأمور تُبنى المنازل، كلّ بيت تمّ إنشاؤه يُسمّى كائناً (Object)، كما هو موضّح في الشكل (3-28)



الشكل (3-28)

مثال آخر:

أستطيع عمل صنف كقالب رئيسي أسميه سيّارة، مكوّنات السيّارة: اللون، والنوع، وبناءً عليها تُصنّع السيّارات، كلّ سيّارة تمّ تصنيعها تُسمّى كائناً، كما هو موضّح في الشكل 3-29



الشكل (3-29)

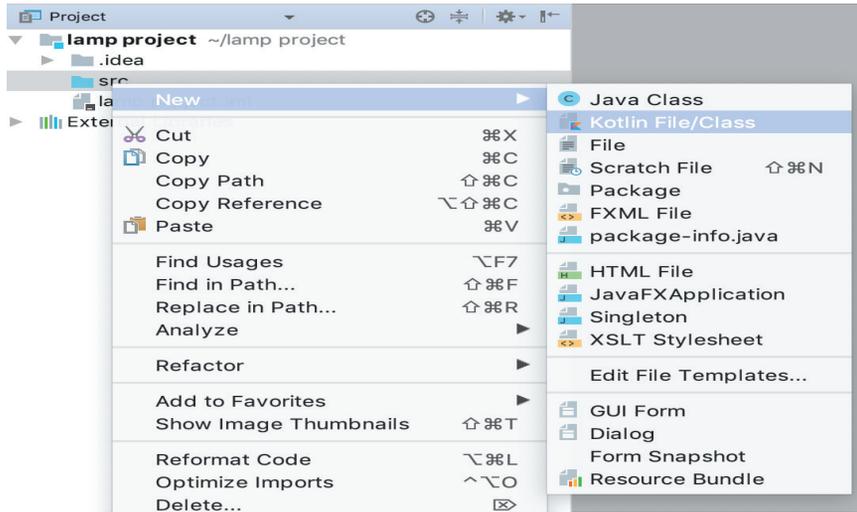


مفهوم البرمجة الكينونية (Object-oriented):

هي نمط برمجة متقدمة، يُستخدم في تمثيل الأشياء في العالم الحقيقي عن طريق الصنف (Class) والكيان (Object)، حيث يُقسم البرنامج إلى وحدات تُسمى الكيانات (Objects)، وكل كيان عبارة عن حزمة من البيانات، والمتغيرات، والثوابت، والدوال، ووحدات التنظيم، وواجهات الاستخدام.

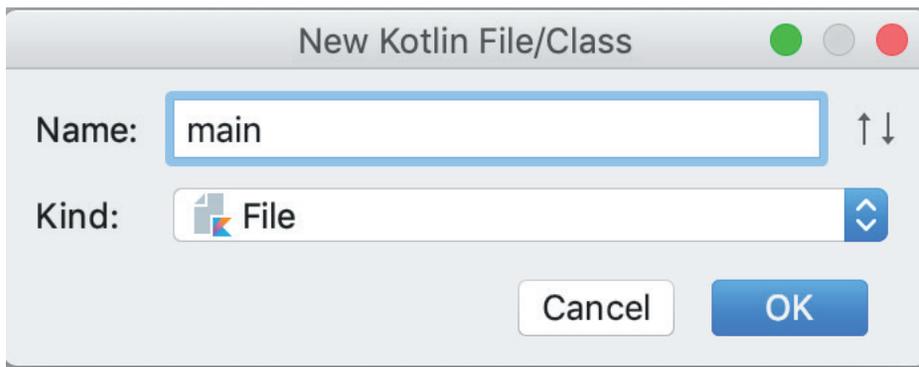
نشاط عملي (1):

أكتب برنامجاً بسيطاً لتشغيل وإغلاق الإنارة باستخدام البرمجة الكينونية.
أفتح برنامج IntelliJ IDEA ، وأنشئ مشروعاً جديداً، كما تعلّمت سابقاً، ثم أتبع الخطوات الآتية:
أولاً: أنشئ ملفّ كوتلن كما في الشكل (30-3).



الشكل (30-3)

ثانياً: أسّميه main، أو أيّ اسم آخر، كما في الشكل (31-3)



الشكل (31-3)



◀ **ثالثاً:** أكتبُ الكود كما يأتي:

```
1  class Lamp {
2
3
4      var isOn: Boolean = false
5
6      // الحالة الاولى الانارة
7      fun turnOn() {
8          isOn = true
9      }
10
11     // الحالة الثانية الاغلاق
12     fun turnOff() {
13         isOn = false
14     }
15
16
17     fun displayLightStatus(lamp: String) {
18         if (isOn == true)
19             println("$lamp الانارة مضيئة ")
20         else
21             println("$lamp الانارة مغلقة ")
22     }
23 }
24
25 fun main(args: Array<String>) {
26
27     val l1 = Lamp() // انشاء كائن l1 من صنف Lamp
28     val l2 = Lamp() // انشاء كائن l2 من صنف Lamp
29
30     l1.turnOn()
31     l2.turnOff()
32
33     l1.displayLightStatus( lamp: "l1")
34     l2.displayLightStatus( lamp: "l2")
35 }
36
```

◀ **أعملُ اختباراً للمشروع، وستظهر النتائج الآتية:**

- الإنارة مضيئة l1
- الإنارة مغلقة l2



شرح الكود البرمجي:

- أنشأتُ صنفاً (class) سمّيته Lamp.
- عرّفتُ متغيّر isOn من نوع Boolean ، ويُعدّ خاصيّة للصف Lamp الذي سيّوضّح حال الإنارة.
- أنشأتُ ثلاث دوال :
- (1) turnOn(): لتمثّل وضع تشغيل الإنارة، وأسندت قيمة true للمتغيّر isOn.
- (2) turnOff(): لتمثّل وضع إغلاق الإنارة، وأسندت قيمة false للمتغيّر isOn.
- (3) displayLightStatus(lamp:String): وتستقبل اسم الإنارة للتمييز بين الإنارات إذا كان هناك أكثر من إنارة.

• في الدالة الرئيسيّة أنشأتُ كائنين (objects) من الصف Lamp أسميتهما 11، 12 (11 تُعبّر عن الإنارة 1 ، 12 تُعبّر عن الإنارة 2)، وأستطيع أن أنشئ عدداً لا نهائياً من الكائنات من الصف نفسه.

• عند إنشاء كائن من صف معين أستطيع الوصول لجميع الدوال والمتغيّرات التي في داخل ذلك الصف، ما دامت لم تأخذ صفة private التي سأشرحها لاحقاً، هنا استدعيت الدالة turnOn() من خلال الكائن 11 لتعطي الإنارة الأولى وضع التشغيل، واستدعيت turnOff() من خلال الكائن 12 لإغلاق الإنارة الثانية، ثم استدعيت الدالة displayLightStatus() من خلال 11 و 12.

خاصيّة الوصول لعناصر الصف (الخصائص "المتغيّرات" والدوال):

هي كلمات توضع قبل التعريف، تفيد في تحديد إمكانيّة الوصول للعنصر، مثل:

1. **public**: وهي السماحيّة الافتراضيّة للعناصر، وتعني أنه يمكن الوصول للعنصر واستخدامه في أيّ مكان، سواء داخل الصف، أو في الأصناف الأبناء، أو حتى أيّ مكان خارج الصف.
2. **protected**: يمكن الوصول للعنصر واستخدامه في داخل الصف، أو في الأصناف الأبناء.
3. **private**: يمكن الوصول للعنصر واستخدامه في داخل الصف فقط.

ملحوظة:

للوصول لأيّ دالة أو متغيّر (خاصيّة) في الصف الرئيسيّ من خلال الكائن الذي تمّ إنشاؤه من ذلك الصف، اكتب اسم الكائن، ثمّ نقطة (.)، كما في النشاط السابق مثلاً 11.turnOn().

الباني (constructor):

يُعدّ الباني من أهمّ الدوال في الصنف، وهو دالة خاصة تُسهّل عمليّة إنشاء الكائن، يتمّ استدعاؤها أثناء إنشاء الكائن؛ لتوليد قيم أوليّة للخصائص الموجودة فيه، ويوجد نوعان من الباني:

- الباني الأساسي (primary constructor).
- الباني الثانوي (secondary constructor).

الباني الأساسي (primary constructor):

يتمّ إنشاؤه بفتح أقواس بعد اسم الصنف، وكتابة المتغيّرات (الخصائص) المطلوبة:

```
class Person (name: String)
```

```
1
2 fun main(args: Array<String>) {
3
4     val person1 = Person( firstName: "Zain", age: 18)
5
6     val firstName: String = person1.firstName
7     val age: Int = person1.age
8
9     println("First Name = $firstName")
10    println("Age = $age")
11 }
12
13 class Person(val firstName: String, var age: Int) {
14
15 }
```

عند عمل تشغيل للبرنامج ينتج:

First Name = Zain

Age = 18

شرح الكود:

1. person1 هو كائن من الصنف Person الذي يحتوي على خاصيتين (firstName, age).
2. عرّفُ متغيّراً من نوع String ، أسميته name لأُسنَدَ له القيمة المخزّنه في firstName، وعرّفُ متغيّراً آخر بالطريقة نفسها، لأُسنَدَ له القيمة المخزّنه في age.
3. كتبتُ جملة الطباعة اللازمة لطباعة القيم.



في المثال السابق إذا أردت كتابة جمل الطباعة داخل الصنف الرئيسي يجب وضع البادئة `init` ، ثم أكتب في داخلها جمل الطباعة، وسوف أحصل على النتائج السابقة نفسها، كما يأتي:

```
1
2 fun main(args: Array<String>) {
3
4     val person1 = Person( firstName: "Zain", age: 18)
5
6 }
7
8 class Person(val firstName: String, var age: Int) {
9
10    init {
11        println("First Name = $firstName")
12        println("Age = $age")
13    }
14 }
```

نشاط عملي (2):



أكمل المثال السابق بإنشاء خمسة كائنات من الصنف `Person`.

الباني الثانوي (secondary constructor)

في الكوتلن أستطيع إنشاء بانٍ ثانوي واحد أو أكثر، باستخدام الكلمة المحجوزة `constructor`. الباني الثانوي غير مستخدم كثيراً في الكوتلن، ويستخدم غالباً في الوراثة التي سأحدث عنها لاحقاً.

الشكل العام للباني الثانوي:

```
class A {
    constructor(data: String) {
    }
    constructor(data: String, numberOfData: Int) {
    }
}
```



مثال توضيحي:

```
1 fun main(args: Array<String>) {
2
3
4- val person1 = Person( firstName: "Zain", age: 18)
   println(person1.Name + "\n" + person1.Age)
5
6
7- val person2 = Person()
   println(person2.Name + "\n" + person2.Age)
8
9 }
10
11 class Person {
12
13- var Name: String? = null
14- var Age: Int? = null
15
16- constructor() {
17-     Name = "Khaled"
18-     Age = 20
19- }
20
21- constructor(firstName: String, age: Int) {
22-     Name = firstName
23-     Age = age
24- }
25
26 }
```

شرح الكود:

1. عرّف متغيرين واحداً من نوع String والأخر من نوع Int وأعطيهما قيمة ابتدائية null.
2. أنشأت بائياً ثانوياً لا يستقبل قيماً، وفي داخله أسندت للمتغيرات التي عرّفتها في الخطوة السابقة قيماً.
3. أنشأت بائياً ثانوياً آخر يستقبل قيمتين، وفي داخله أسندت للمتغيرات التي عرّفتها في الخطوة الأولى القيمتين التي يستقبلهما هذا الباني.
4. في الدالة الرئيسية أنشأت كائناً من الصنف Person، وأعطيته قيمتين، وفي هذه الحالة يُستخدم الباني الذي يستقبل القيم، ثم كتبت جملة طباعة لطباعة قيم المتغيرات الجديدة لـ (Name, Age) الموجودة في الصنف Person.
5. أنشأت كائناً من الصنف Person ولم أعطه قيماً، وفي هذه الحالة يُستخدم الباني الذي لا يستقبل قيماً، ثم كتبت جملة طباعة لطباعة قيم المتغيرات (Name, Age) الموجودة في الصنف Person.

نشاط عملي (3):

في المثال السابق أنشئ كائنين إضافيين: أحدهما يأخذ قيماً، والآخر دون قيم، وأختبر المشروع، وأوضح النتائج.

الأسئلة

1. ما مميزات الكائن (object) الرئيسية؟
2. اذكر أنواع الباني (constructor)، موضحاً كيفية كتابتها؟

الموقف التعليمي التعلّمي (3 - 4):

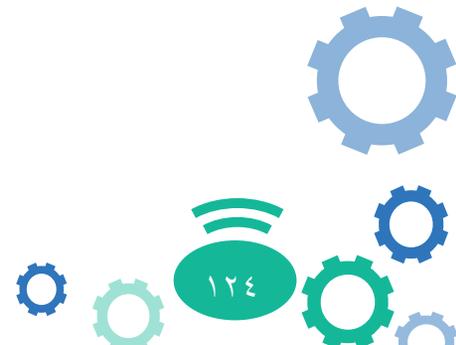
توظيف مبادئ (البرمجة الكائنية) في بناء برنامج 2.

وصف الموقف التعليمي: توجه مدير شركة أدوية إلى مشغل لتطبيقات المحمول في مدرسة يافا المهنية لمساعدته في بناء برنامج يساعده في إدارة شركته، والتعامل مع الزبائن والموظفين، ومخزون الأدوية.

العمل الكامل			
الموارد	المنهجية (استراتيجية التعلّم)	وصف الموقف الصّفي	خطوات العمل
<ul style="list-style-type: none">• وثائق (طلب مدير الشركة، نشرات، كتب ومصادر متنوعة عن أسس البرمجة وآلية بناء البرامج).• التكنولوجيا(روابط إلكترونية متنوعة من مصادر موثوقة، حاسوب، شاشة عرض، إنترنت).	<ul style="list-style-type: none">• حوار ومناقشة.• التعلّم التعاوني/ مجموعات البحث العلمي	<p>جمع البيانات من مدير الشركة عن:</p> <ul style="list-style-type: none">• الأدوية ومواصفاتها.• آلية التعامل مع الزبائن ومطالبتهم بتسديد الديون.• آلية إدارة الموظفين (الوصف الوظيفي، واحتساب الرواتب والمكافآت، وأسس الترقية، ...). <p>جمع البيانات عن:</p> <ul style="list-style-type: none">• خصائص البرمجة الكائنية:- الوراثة (inheritance)- التحميل الزائد (overloading)- إعادة الكتابة (overriding)- الواجهات والتجريد (interface & abstract)- التغليف (packaging)	أجمع البيانات وأحلّها
<ul style="list-style-type: none">• وثائق (طلب مدير الشركة، نشرات، كتب ومصادر متنوعة).• قرطاسية.• حاسوب.• الشبكة العنكبوتية.	<ul style="list-style-type: none">• تعلّم تعاوني/مجموعات حوار ومناقشة .• عصف ذهني/استمطار الأفكار.	<ul style="list-style-type: none">• تحديد خصائص وعمليات عناصر الشركة: (الأدوية، والزبائن، والموظفين).• تحديد الخصائص البرمجية الأنسب لعناصر الشركة.• تحديد المدخلات، والعمليات، والمخرجات لكائنات النظام والكائنات الفرعية: (موظفين: سائقين، مدراء، ...).	أخطّط وأقرّر



<ul style="list-style-type: none"> • قرطاسية • الطلب الخاص • بمدير الشركة • حاسوب، برنامج خاصّ IntelliJ 	<ul style="list-style-type: none"> • تعلّم تعاوني/مجموعات • حوار ومناقشة. • كتابة. 	<ul style="list-style-type: none"> • اختار الأصناف (classes) (الموظفون، الزبائن، الأدوية، ...) • أوضح الخصائص والعمليات لكلّ صنف. • اختار إمكانية الوصول المناسبة لكلّ من الخصائص والعمليات (private, protected, public) • اختار الأصناف الوارثة (الموظفون - < السائقون، مدراء، مراسلون ...) • الاتفاق على الأصناف المجردة (abstract) • اختار العمليات التي سيتمّ إجراء التحميل الزائد (overloading) عليها (عمل عمليتين لاحتساب الإجازة، الأول يخضم يوماً واحداً فقط والثاني يتمّ تمرير عدد الأيام المخصصة، ...) • أهدد العمليات التي سيتمّ إجراء إعادة الكتابة (overriding) عليها (احتساب ساعات العمل للموظفين بشكل عام، واحتسابها للسائقين بشكل خاص، ...) • اكتب الكود الذي يحقق المطلوب. 	<p>أنفذ</p>
<ul style="list-style-type: none"> • إنترنت حاسوب. • الخطة المُعدّة سابقاً. • قرطاسية. • الطلب الخاص • بمدير الشركة. • حاسوب، برنامج خاصّ IntelliJ. 	<ul style="list-style-type: none"> • تعلّم تعاوني/مجموعات. • حوار ومناقشة. 	<ul style="list-style-type: none"> • التحقّق من ملاءمة الأصناف (classes). • التحقّق من الخصائص والعمليات لكلّ صنف. • التأكد من وضع إمكانية الوصول المناسبة لكلّ من الخصائص والعمليات. • التحقّق من ملاءمة الأصناف الوارثة للهدف. • التحقّق من العمليات التي سيتمّ إجراء التحميل الزائد (overloading)، وأو إعادة الكتابة (overriding). • التحقّق من كتابة الكود بطريقة صحيحة. • التحقّق من تشغيل البرنامج وصحة النتائج. • اكتشاف الأخطاء وتصويبها، واقتراح البدائل. 	<p>أتحقق</p>



<ul style="list-style-type: none"> • حاسوب . • شاشة عرض . • قرطاسية . 	<ul style="list-style-type: none"> • حوار ومناقشة . • تعلم تعاوني / مجموعات عمل . 	<p>توثيق البيانات التي تتعلق بـ:</p> <ul style="list-style-type: none"> • صفات البرمجة الكائنية وميزاتها . • صفات البرنامج المطلوب . • المدخلات والمخرجات والعمليات التي يقوم بها البرنامج . • الأصناف وخصائصها وعملياتها . • العلاقة بين الأصناف . • إعداد العروض التقديمية . • إعداد ملفّ بالحالة (برنامج يساعد في إدارة الشركة) . • فتح سجلّ خاصّ يُعنى ببرامج إدارة الشركات والمخزون . 	<p>أوثقُ وأقدمُ</p>
<ul style="list-style-type: none"> • نماذج للتقويم (استبانة، ...) • طلب مدير الشركة . 	<ul style="list-style-type: none"> • حوار ومناقشة . 	<ul style="list-style-type: none"> • نجاعة البرنامج في إدارة الشركة، وإمكانية تطبيقه في شركات أخرى للغرض نفسه . • رضا المدير عن البرنامج . 	<p>أقومُ</p>

أناقشُ:

أناقش في مجموعات أهميّة الوراثة في البرمجة كائنيّة التوجّه من حيث: توفير الوقت والجهد، وإعادة استخدام الكود (code reusing)، وإصلاح الأخطاء، والعمل في مجموعات، والاستفادة من خبرات الآخرين.



توظيف مبادئ (البرمجة الكائنية) في بناء برنامج

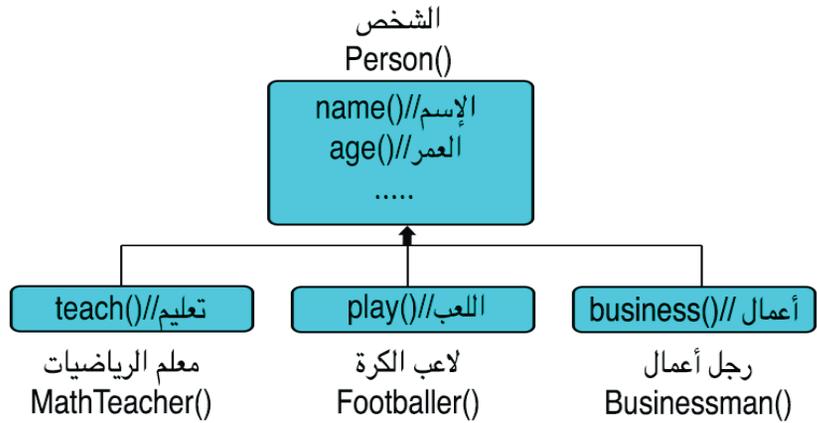
الوراثة (Inheritance):

أفترض في مشروعني ثلاث شخصيات، معلم رياضيات، ولاعب كرة قدم، ورجل أعمال. كل هؤلاء الأشخاص تجمعهم خصائص مشتركة، مثل: الأكل، والكلام، والحركة،... الخ، وكل شخص عنده خاصية فردية، مثلاً: معلم الرياضيات يدرّس الرياضيات، ولاعب كرة القدم يلعب الكرة، ورجل الأعمال عنده مشاريع استثمارية. استطيع عمل صنف (class) لكل شخص خاص به، وأضع فيه عدة أمور، كما في الشكل (32-3):

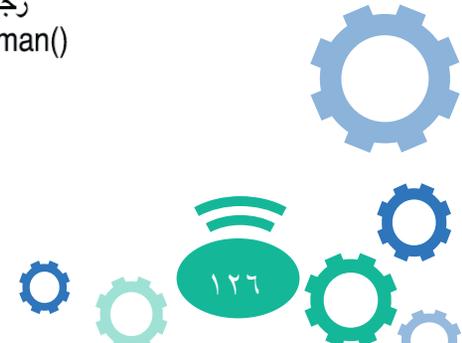


الشكل (32-3)

ألاحظ أنّ هناك أموراً مشتركة، وإذا أردت إنشاء صنف خاص لكل شخص، سأكرّر في كل شخص الأكواد نفسها في كل صنف؛ لذلك حتى أقلل من التكرار أجمع الأمور المشتركة في صنف أسميه Person وأجعل الأصناف الأخرى ترثه، فيسمّى هذا الصنف الأب، وكل صنف يرثه يُسمّى الابن، انظر إلى الشكل (33-3).



الشكل (33-3)



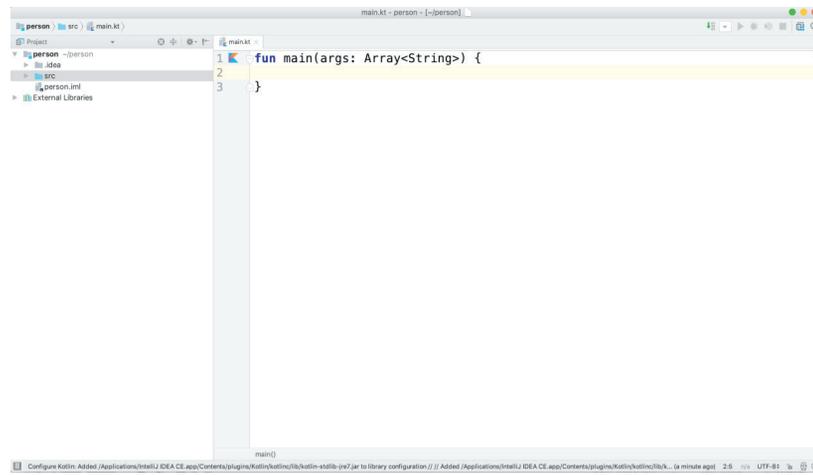
تعريف الوراثة (Inheritance):

هي إمكانية بناء صنف (class) على أساس صنفٍ آخر أو أكثر، ويُسمّى الصنف المُورث صنف الأب والصنف الوارث يُسمّى صنف الابن، يرث الصنف الابن جميع خصائص ودوال صنف الأب.

◀ مثال:

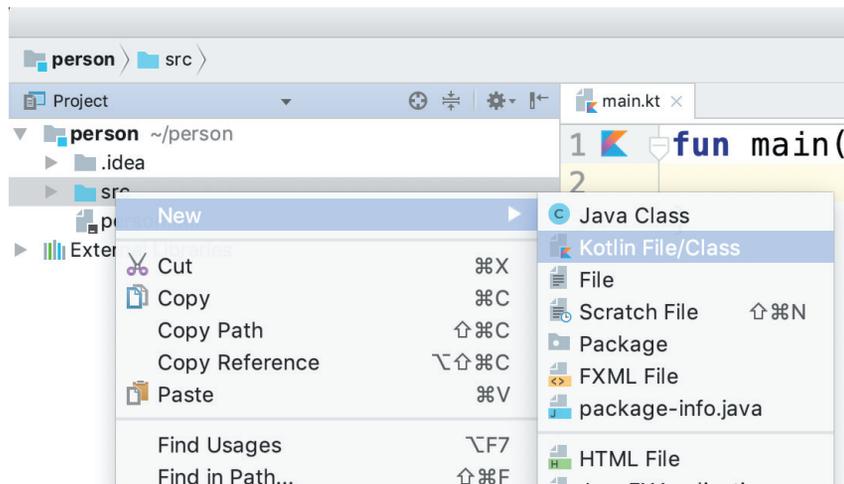
في هذا المثال أريد ترجمة ما كتبته سابقاً إلى كود برمجيّ، وأريد فصل كلِّ صنفٍ أنشئته في مكان مستقلّ، كما يأتي:

◀ أولاً: أنشئ الملفّ الرئيسيّ كما تعلمت سابقاً، ويظهر كما يأتي:



```
main.kt
1 fun main(args: Array<String>) {
2
3 }
```

◀ ثانياً: أنشئ ملفاً جديداً من خلال الضغط بزر الفأرة اليمين على كلمة src ثم new ثم Kotlin file/class كما يأتي:



```
person > src >
Project
person ~/person
  .idea
  src
  External Libraries
New
  Java Class
  Kotlin File/Class
  File
  Scratch File
  Package
  FXML File
  package-info.java
  HTML File
```



◀ **ثالثاً:** عند ظهور الشاشة الآتية أكتب اسم الصنف، وتأكد من اختيار `class` :



◀ **رابعاً:** أكرّر الخطوتين الثانية والثالثة لبقية الأصناف (Teacher, Footballer, Businessman).

◀ **خامساً:** أبداً بكتابة الأكواد البرمجية، أفتح الصنف `Person`، ثم أكتب داخله الخصائص المشتركة، كما يأتي:

```
main.kt x Person.kt x Teacher.kt x Footballer.kt x Businessman.kt x
1  open class Person(name :String, age : Int) {
2
3      init {
4          println("The name is: $name")
5          println("The age is: $age")
6      }
7
8  }
```

◀ **سادساً:** أفتح الصنف `Teacher`، وأكتب ما يأتي :

```
main.kt x Person.kt x Teacher.kt x Footballer.kt x Businessman.kt x
1  class Teacher (name: String, age:Int): Person(name,age) {
2      fun teach(){
3          println("He teach math in school")
4      }
5  }
```

بما أنّ هذا الصنف هو صنف الأب الذي سترثه الأصناف الأخرى كتبت كلمة `open` في البداية، ووضعت في الباني بعض الخصائص المشتركة عند جميع الأشخاص، ثم كتبت جملتي طباعة للاسم والعمر.

```
main.kt x Person.kt x Teacher.kt x Footballer.kt x Businessman.kt x
1  class Footballer(name: String, age:Int): Person(name,age) {
2      fun football(){
3          println("He play football")
4      }
5  }
```



عند إنشاء الصنف Teacher أجعله يرث الصنف Person من خلال وضع نقطتين رأسيين (:), وبما أن الصنف الأب يحتوي على بانٍ أساسي يجب إنشاء بانٍ لصنف الابن، ووضع المتغيرات التي تم إنشاؤها في صنف الأب، ثم أكتب داخل الصنف ما أريد من دوال، أو أي شيء آخر.

◀ **سابعاً:** أكرّر الخطوة السابقة مع الصنفين Businessman، Footballer :

```
main.kt x Person.kt x Teacher.kt x Footballer.kt x Businessman.kt x
1 class Businessman(name: String, age:Int): Person(name,age) {
2     fun business(){
3         println("He is a businessman")
4     }
5 }
```

```
main.kt x Person.kt x Teacher.kt x Footballer.kt x Businessman.kt x
1 class Footballer(name: String, age:Int): Person(name,age) {
2     fun football(){
3         println("He play football")
4     }
5 }
```

◀ **ثامناً:** أفتح الملف الرئيسي، وأكتب ما يأتي:

```
main.kt x Person.kt x Teacher.kt x Footballer.kt x Businessman.kt x
1 fun main(args: Array<String>) {
2     val teacher = Teacher( name: "Hassan", age: 30)
3     teacher.teach()
4
5     val footballer = Footballer( name: "Jod", age: 22)
6     footballer.football()
7
8     val business = Businessman( name: "Mark", age: 35)
9     business.business()
10 }
```

أنشأت ثلاثة كائنات من كل صنف، واستدعيت الدالة التي في داخل كل صنف، وعند عمل تشغيل للبرنامج ظهرت النتائج الآتية:

```
Run MainKt
/Library/Java/JavaVirtualMachines/jdk1.8.0_144.jdk
objc[9712]: Class JavaLaunchHelper is implemented
The name is: Hassan
The age is: 30
He teach math in school
The name is: Jod
The age is: 22
He play football
The name is: Mark
The age is: 35
He is a businessman
```





أناقش مع زملائي فائدة الوراثة من المثال السابق.

إعادة الاستخدام (Overriding):

في الوراثة عندما يرث صنف صنفاً آخر، أحتاج أحياناً لدالة موجودة في صنف الأب، ولكن أريد استخدامها بطريقة مختلفة عن صنف الأب.

مثال:

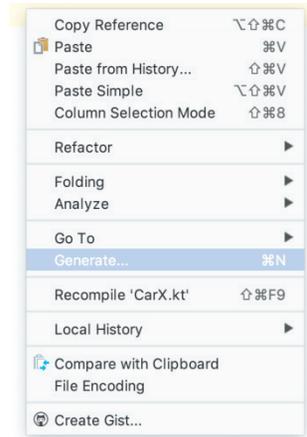
أولاً: أنشئ صنفاً أسميه Car، وأكتب الكود الآتي:

```
1 open class Car {
2   open fun propertis(name:String,color:String,model: Int){
3     println("Name Car $name")
4     println("Color of Car $color")
5     println("Model of Car $model")
6   }
7 }
```

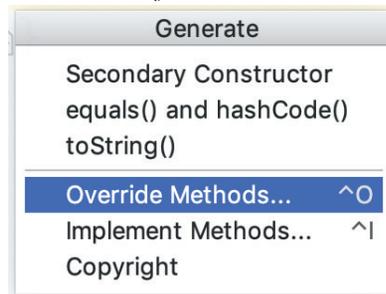
أنشأت داخل الصنف دالة تستقبل ثلاث قيم، وهي: اسم السيارة، ولونها، و طرازها، وكتبت قبلها كلمة open حتى أستطيع إعادة استخدام الدالة في مكان آخر.



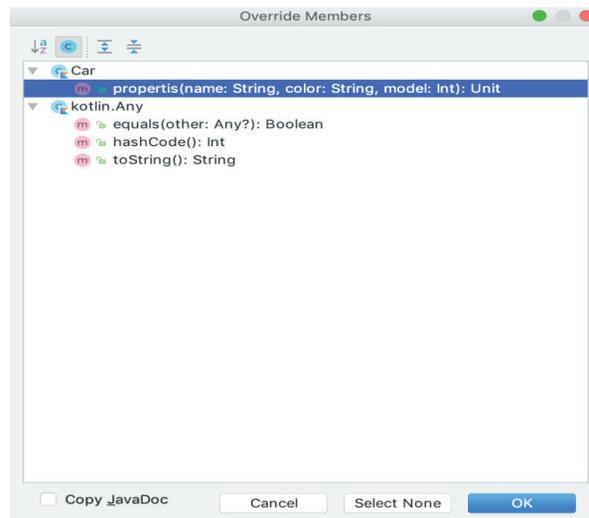
◀ **ثانياً:** أنشئ صنفاً آخر أسميه CarX، وأجعله يرث الصنف Car، ثم أستدعي الدالة الموجودة في الصنف Car لإعادة استخدامها في هذا الصنف، أضغط بزر الفأرة اليمين داخل الصنف CarX، ثم أختار Generate كما يأتي:



ومن القائمة التي تظهر أختار Override Methods كما يأتي:



ثم أختار الدالة التي أنشأتها في صنف Car ثم ok كما يأتي:



التحميل الزائد (Overloading):

وجود أكثر من دالة بالاسم نفسه، ولكن كل دالة تحتوي على خصائص وعمليات مختلفة.

مثال:

حساب مساحة عدة أشكال.

```
1 fun main(args: Array<String>) {
2
3     val a = Area()// إنشاء كائن من الصنف Area()
4     println("Area of Rectangle = ${a.area( length: 3, width: 5)}")
5     println("Area of Triangle = ${a.area( height: 6f, base: 5f)}")
6     println("Area of Square = ${a.area( length: 9)}")
7     println("Area of Circle = ${a.area( radius: 3f)}")
8
9 }
10
11 class Area {
12     // دال حساب مساحة المستطيل
13     fun area(length: Int, width: Int): Int {
14         return length * width
15     }
16
17     // دال حساب مساحة المربع
18     fun area(length: Int): Int {
19         return length * length
20     }
21
22     // دال حساب مساحة الدائرة
23     fun area(radius: Float): Float {
24         return radius * radius * 3.14f
25     }
26
27     // دال حساب مساحة المثلث
28     fun area(height: Float, base: Float): Float {
29         return 0.5f * height * base
30     }
31
32 }
33
```

شرح الكود:

1. أنشأت صنفاً سمّيته Area()، وكتبت بداخله أربع دوال متشابهة بالاسم، ولكنها تختلف بالخصائص (المتغيرات)

التي تستقبلها الدوال:

- دالة حساب مساحة المستطيل: تستقبل قيمتين من نوع Int، وترجع قيمة Int أيضاً وهي حاصل ضرب الطول في العرض.
- دالة حساب مساحة المربع: تستقبل قيمة من نوع Int، وترجع قيمة Int أيضاً وهي حاصل ضرب الطول بنفسه.
- دالة حساب مساحة الدائرة: تستقبل قيمة من نوع Float، وترجع قيمة Float أيضاً وهي حاصل ضرب نصف القطر بنفسه مضروباً بـ 0.5f (وضع حرف f بعد القيمة لتوضيح أن القيمة من نوع Float).



- دالة حساب مساحة المثلث: تستقبل قيمتين من نوع Float، وترجع قيمة Float أيضاً وهي حاصل ضرب القاعدة في الارتفاع في 0.5f.
- 2. في الدالة الرئيسية أنشأت كائناً من صنف Area() لحساب مساحة المستطيل والمربع والدائرة والمثلث، وطباعة النتيجة.
- 3. ألاحظ عند استدعاء الدالة (...) area في كل مرة يعطي عدّة خيارات وفق نوع المتغيّرات المدخلة وعددها أختار من القائمة أيّة دالة أريد استخدامها كما يأتي:

```

${a.}")
m area(length: Int) Int
m area(radius: Float) Float
m area(length: Int, width: Int) Int
m area(height: Float, base: Float) Float

```

- 4. بعد كتابة جمل الطباعة الأربع أشغل البرنامج، وأحصل على النتائج الآتية:

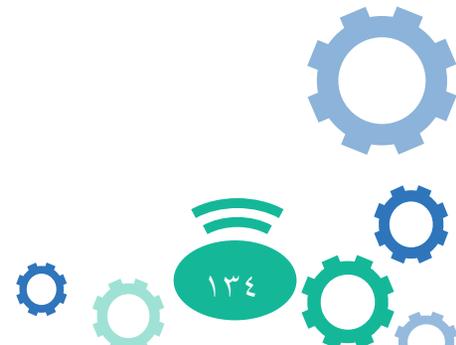
```

Run MainKt
/Library/Java/JavaVirtualMachines/
objc[10368]: Class JavaLaunc
Area of Rectangle = 15
Area of Triangle = 15.0
Area of Square = 81
Area of Circle = 28.26

```

نشاط عملي (2):

اكتب برنامج يعمل على طباعة الاسم الثنائي والثلاثي والرابعي لثلاثة أشخاص.



خاصية الوصول لعناصر الصنف (الخصائص "المتغيرات" والدوال) في الوراثة:

مثال:

```
open class Base() {
    var a = 1
    private var b = 2
    protected open val c = 3
    protected fun d() { }
}

class D: Base() {

    override val c = 9
}

fun main(args: Array<String>) {
    val base = Base()

    base.a

    // base.b, base.c , base.d()
}
```

شرح المثال:

1. أنشأت صنفاً سمّيته **Base** وكتبت بداخله ثلاثة متغيرات ودالة:
 - المتغير **a** أعطيته خاصية **public** ولا يشترط كتابة الكلمة قبل تعريف المتغير.
 - المتغير **b** أعطيته خاصية **private**.
 - المتغير **c** أعطيته خاصية **protected** وسماحية إعادة استخدامه (**override**) في صنف الابن بوضع كلمة **open**.
 - الدالة **d** أعطيتها خاصية **protected**.
2. أنشأت صنفاً سمّيته **D** يرث الصنف **Base**، أستطيع استخدام داخل هذا الصنف المتغيرات والدوال التي تحمل خاصية **public**، **protected** وهي: **a**، **c**، **d**.
3. في الدالة الرئيسية:
 - أنشأت كائناً من الصنف **Base**، أستطيع فقط الوصول إلى المتغير **a** لأنه **public**، أما **b,c,d** لا أستطيع الوصول إليها، لماذا؟
 - أنشأت كائناً من الصنف **D**، لكن لا أستطيع الوصول إلى الدالة **c**؛ لأنها تحمل خاصية **protected** في الصنف الرئيسي.





أعدّل على المثال السابق بحيث أستطيع الوصول إلى الدالة c من الدالة الرئيسية بواسطة الكائن المنشأ من الصنف D.

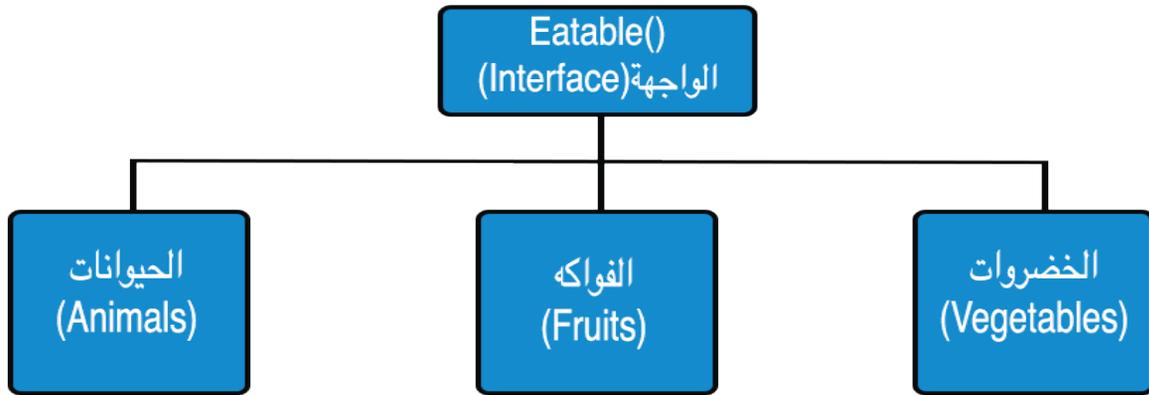


استنتج ممّا سبق الفرق بين `private`، `public`، `protected`.

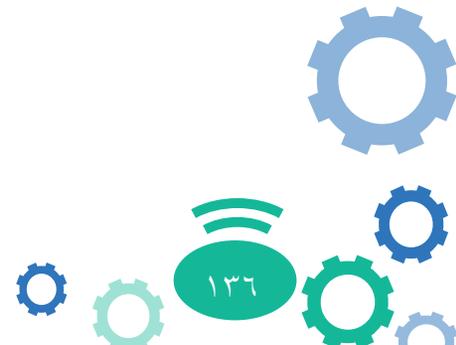
الواجهة (interface):

جاءت فكرة الواجهة (interface) من إمكانية تصنيف الأشياء المختلفة وفق صفة أو خاصية مشتركة بينها، ويمكن تعريف الواجهة أنها الصنف الذي يحتوي على خصائص رئيسية تُمثل برمجياً بدوال (functions) فارغة، حتى يتم إعادة استخدامها تلقائياً في الصنف الذي يرث الواجهة (interface)، وإعطاء الدوال القيم المناسبة. مثلاً: التفاح فاكهة قابلة للأكل، والفلفل خضار قابل للأكل، والدجاج حيوان قابل للأكل، ألاحظ أن خاصية (قابل للأكل) مشتركة بين الحيوانات والنباتات.

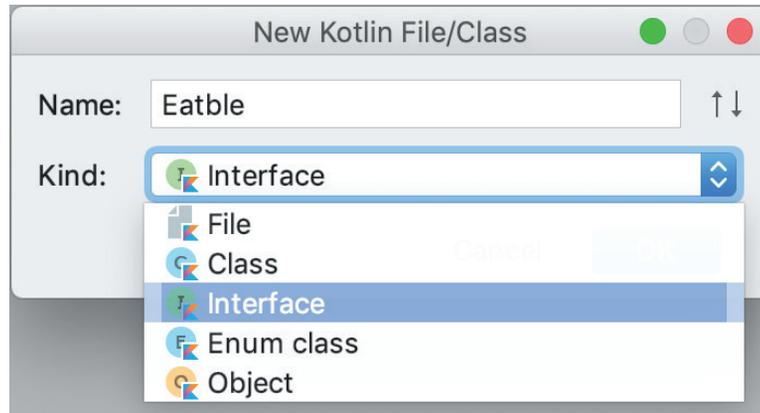
مثال: تحويل الشكل (3-43) إلى كود برمجي.



الشكل 3-34



أولاً: أفتح برنامج IntelliJ IDEA وأنشئ مشروعاً جديداً (New Kotlin File)، ثم أنشئ ملفاً جديداً أسميه Eatble وأختار interface كالاتي:



أكتب ثلاث دوال فارغة كل دالة تستقبل وتعيد قيمة من نوع String كالاتي:

```
1 interface Eatble {  
2     fun vegetables(name:String):String  
3     fun fruits(name:String):String  
4     fun animal(name:String):String  
5 }
```

ثانياً: أنشئ ملفاً جديداً أسميه main وأكتب فيه كالاتي:

```
1 class Person : Eatble {  
2     override fun vegetables(name: String): String {  
3         return "eat $name"  
4     }  
5  
6     override fun fruits(name: String): String {  
7         return "eat $name"  
8     }  
9  
10    override fun animal(name: String): String {  
11        return "eat $name"  
12    }  
13 }  
14  
15 fun main(args: Array<String>) {  
16  
17    val Saed = Person()  
18    println(Saed.vegetables( name: "Tomato"))  
19    println(Saed.fruits( name: "Banana"))  
20    println(Saed.animal( name: "Chicken"))  
21  
22 }
```



شرح الكود:

1. أنشئ صنف باسم Person و يرث Eatble:
- a. يطلب إعادة استخدام(overriding) جميع الدوال الغير معرفة داخل الواجهة Eatble كما يأتي:

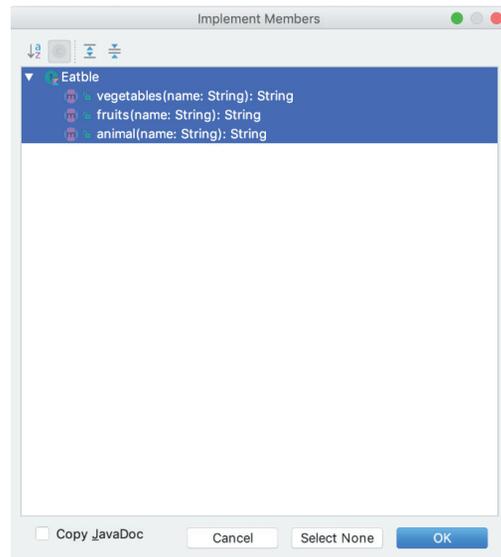
```
1
2 class Person : Eatble{
3
```

Class 'Person' is not abstract and does not implement abstract member **public abstract fun eat(): String** defined in Eatble

- b. أضغط على الإنارة الحمراء وأختار كالاتي:



- c. أختار جميع الدوال التي تظهر في النافذة وأضغط ok كما يأتي:



- d. تظهر جميع الدوال فارغة حتى يتم إعادة استخدامها بالشكل المناسب.
2. كل دالة من الدوال تستقبل وتعيد قيمة؛ لذلك أكتب بداخل كل دالة `return` وأجعلها تعيد نفس الاسم الذي تم استقباله.
3. في الدالة الرئيسية `main` أنشئ كائن من الصنف `Person` ثم أكتب جمل طباعة توضح الوصول للدوال التي تم إعادة استخدامها.
- يتم إعادة استخدام (overriding) الدالة من الواجهة `interface` ما لم تحتوي على أي عملية، وتسمى مجردة (abstract)، فإذا احتوت الدالة على قيمة لا يطلب البرنامج إعادة استخدامها كالتالي:

```
fun animal{
print("lion")
}
```

ولإعادة استخدامها أكتب كلمة `open` كما تعلمت سابقا.

التجريد (abstract)

يشبه التجريد (abstract)، الواجهة (interface) من حيث المبدأ، ويستخدم عادة لحفظ بيانات سرية، مثل رقم الفيزا وغيرها من الأمور الحساسة.

أهم الخصائص المشتركة بين صنف الواجهة (interface) والصنف المجرد (abstract):

1. لا يمكن إنشاء كائن `object` من الواجهة `interface` أو من الصنف المجرد `abstract`.
2. يجب أن تكون الدوال (functions) المراد إعادة استخدامها فارغة لا تحتوي على أي قيم.

أما الاختلاف بينهما فهو كتابة كلمة `abstract` قبل الدالة في الصنف المجرد.

مثال:

```
1  abstract class Person(name: String) {
2
3
4      init {
5          println("My name is $name.")
6      }
7
8      fun displaySSN(ssn: Int) {
9          println("My SSN is $ssn.")
10     }
11
12     abstract fun displayJob(description: String)
13 }
14
15 class Teacher(name: String): Person(name) {
16
17     override fun displayJob(description: String) {
18         println(description)
19     }
20 }
21
22 fun main(args: Array<String>) {
23     val Monthir = Teacher( name: "Monthir")
24     Monthir.displayJob( description: "I'm a mathematics teacher.")
25     Monthir.displaySSN( ssn: 23123)
26 }
```



شرح المثال:

1. أنشئ صنف مجرد أسميه Person يستقبل قيمة من نوع String يتضمن ما يلي:
 - a. جملة طباعة تطبع الاسم.
 - b. أنشئ دالة تستقبل قيمة من نوع int وتحتوي على جملة طباعة تطبع هذا الرقم.
 - c. أنشئ دالة مجردة تستقبل قيمة من نوع String.
2. أنشئ صنف آخر Teacher يستقبل قيمة من نوع String يرث الصنف المجرّد Person، في هذه الخطوة يطلب البرنامج إعادة استخدام الدالة المجرّدة فقط، وبعد استدعائها أكتب جملة طباعة تطبع القيمة التي تستقبلها الدالة.
3. في الدالة الرئيسية main أنشئ كائن Monthir من الصنف Techer الذي يسقبل قيمة من نوع String، ثم من خلال الكائن أصل للدالة المعاد استخدامها داخل الصنف Teacher وإلى الدالة displaySSN الموجودة داخل الصنف Person.

التحريم (packaging)

يمكن تحريم عدّة ملفات كوحدة واحدة) حيث يمكن اعتبار الحزمة كمجلد يحتوي هذه الملفات (باستخدام الكلمة المحجوزة (package) تتبع باسم الحزمة، ويجب كتابتها في السطر الأول في الملف، وعند التحريم يتم التعامل مع عدة ملفات كوحدة واحدة، وعند إنشاء تطبيقات للأندرويد) وهو ما سنقوم به في الوحدة الآتية، (يكون إعطاء حزمة للتطبيق إلزامياً، حيث إنّ من شروط تحميل تطبيق على متجر جوجل - (google play store) وهو أكبر متجر لتطبيقات الأندرويد تشرف عليه شركة جوجل - أن يكون للتطبيق حزمة فريدة، فلا يُقبل أن يشترك تطبيقان باسم الحزمة نفسها، وأيضاً لا يتم قبول تطبيق إذا كان اسم الحزمة له تحتوي على (example) وفي العادة يكون اسم الحزمة اسم نطاق الموقع) عنوان صفحة الويب (الخاصّ بالشركة المنتجة معكوساً يتلوهُ اسم التطبيق، مع مراعاة عدم وجود فراغات. فمثلاً لو أنّ شركة أدوية عنوان موقعها على الويب (www.medicine.com) أنشأت تطبيقاً باسم (aspirin) فالطريقة التقليدية المتّبعة لتسمية الحزمة هي (com.medicine.aspirin) مع إمكانيّة إعطاء تسمية ثانية للحزمة لكنّ هذا قد يُعرّضُ التطبيق لخطر وجود تطبيق باسم الحزمة نفسها في متجر جوجل.

الأسئلة

1. ما هي الوراثة؟
2. ما الفرق بين: أ- إعادة الاستخدام (override) والتحميل الزائد (overload)؟
ب- الواجهة (interface) والتجريد (abstract)؟





أسئلة الوحدة

السؤال الأول: أضع دائرةً حول رمز الإجابة الصحيحة فيما يأتي:

- أ. أيّ من الآتية لا يجوز اتباعها عند تسمية المتغيّر؟
1. أن يبدأ اسم المتغيّر بحرف. 2. أن يبدأ اسم المتغيّر برقم. 3. لا يحتوي على فراغ. 4. لا يحتوي على رموز خاصة.
- ب. كم مرة على الأقلّ يتمّ تنفيذ حلقة الدوران for ؟
1. مرتين. 2. مرة واحدة. 3. لا تُنفَّذ. 4. ٣ مرات.
- ج. ماذا استخدم لطباعة جملة والانتقال لسطر جديد في لغة كوتلين؟
1. writeln 2. write 3. println 4. print
- د. ماذا استخدم لتعريف متغيّر في لغة كوتلين؟
1. var 2. variable 3. store 4. dim
- هـ. أيّ من الجمل الآتية لا تُستخدم في حلقات التكرار في لغة كوتلين؟
1. do 2. while 3. for 4. when
- و. عند توريث صنف، ما الكلمة الواجب كتابتها قبل اسم الصنف الأب؟
1. derived 2. open 3. Parent 4. inherit

السؤال الثاني:

أعرف:

- لغة البرمجة.
- التحميل الزائد (Overloading)
- البرمجة الكينونية (Object Oriented Programing OOP)

السؤال الثالث:

أميّز بين المصطلحات الآتية في لغة كوتلين :

- override & overload
- class & object
- private & protected & public



السؤال الرابع:

باستخدام لغة كوتلن اكتب البرامج الآتية:

1. أكتب برنامجاً بلغة كوتلن يقوم بقراءة ٠٢ رقماً صحيحاً، وتخزين القيم في مصفوفة وطباعة ما يأتي:
 - قائمة بالأرقام الزوجية المدخلة.
 - قائمة بالأرقام الفردية المدخلة.
2. قراءة أسماء طلاب الصف التاسع ومعدلهم الفصلي، وطباعة ما يأتي:
 - معدل علامات الطلاب.
 - عدد الناجحين.
 - عدد الرسوب.
 - اسم الطالب صاحب أعلى علامة ومعدله.
3. قراءة رقمين صحيحين وطباعة ما يأتي (باستخدام الدوال):
 - حاصل جمعهما.
 - حاصل ضربهما.
 - القيمة الأكبر.
 - القيمة الأصغر.
- قراءة عدد ثم طباعة أول عشرة مضاعفات للرقم المدخل، مثال العدد ٥ ، يظهر لنا الأعداد من ١٠ إلى ٥٥.
- حساب مربعات الأرقام المدخلة حتى أول عدد سالب.
- حساب مجموع ن من العلامات.
- حساب معدل ٤ علامات لمئة طالب.

4. حلّ المسألة الآتية:

$$1 + \frac{2}{4} + \frac{3}{9} + \frac{4}{16} + \frac{5}{25} + \dots + \frac{10}{100}$$

5. إيجاد مجموع المتسلسلة الآتية:

$$\frac{1}{m+1} + \frac{1}{m+2} + \frac{1}{m+3} + \dots + \frac{1}{m+n}$$

حيث إن $m < n$

6. حساب حاصل ضرب الأعداد من ١ إلى ١٠.
7. قراءة ١٠ أرقام، ومن ثم طباعة متوسط الأرقام الموجبة ومتوسط الأرقام السالبة.



السؤال الخامس:

باستخدام لغة كوتلن أنشئ الدوال الآتية:

- دالة تستقبل رقمين صحيحين (الحد الأدنى والحد الأعلى) وطباعة الأرقام المحصورة بينهما.
- دالة ترجع القيمة الأكبر من بين ثلاث قيم.
- دالة تحسب مكعب أي رقم
- دالة تقوم بطباعة الشكل الآتي:

*

**

- دالة تقوم بطباعة جدول الضرب للرقم ن.
 - دالة ترجع المضروب لأي رقم مدخل (ن! = ن × (ن-1) × (ن-2) × ... × 1)
- مثال : ن=5 ، 5!=1×2×3×4×5
- ملحوظة: مضرب الصفر يساوي (0!=1)

مشروع الوحدة



لدى مطعم قائمة بالأطباق التي يقدمها، ويود إنشاء برنامج يُمكن الزبائن من عرض ما يأتي:

- قائمة بالأطباق المتوفرة.
- مكونات كل طبق.
- سعر الطبق.

يجب مراعاة ما يأتي أثناء إنشاء البرنامج:

- كل نوع طبق عبارة عن كائن من الصنف أطباق.

الصنف أطباق يحتوي على :

- الخصائص (اسم الطبق، المكونات، السعر).
- الدوال الخاصة بتخصيص قيم الخصائص لكل كائن واسترجاعها.

البرنامج الرئيسي:

- يتم إنشاء كائنات من الأصناف المنشأة سابقاً.
- إسناد المعلومات المتعلقة بكل طبق.
- عرض الأطباق المتوفرة للزبون مع إمكانية اختيار الطبق، وبالتالي عرض مكوناته وسعره.

يُتَوَقَّعُ من الطَّلَبَةِ بعد دراسة هذه الوَحْدَةِ، والتَّفَاعُلِ مع أنشطتها أن يكونوا قادرين على
تأهيل تطبيقٍ بسيطٍ من خلال الآتي:

- ✓ تعرّف كيفية تنزيل البرامج اللازمة وتثبيتها.
- ✓ تعرّف كيفية تصميم التطبيق.
- ✓ توظيف (الأكواد) البرمجية وطريقة دمجها مع التصميم.



الكفايات المهنية

الكفايات المهنية المتوقع من الطلبة امتلاكها بعد انتهاء دراسة هذه الوحدة والتفاعل مع أنشطتها:
أولاً: الكفايات الحرفية.

- ▶ امتلاك مهارة تنزيل البرامج اللازمة وتثبيتها
- ▶ القدرة على تصميم واجهة للتطبيق.
- ▶ القدرة على ربط الواجهة بـ(الكود) البرمجي.
- ▶ القدرة على تأهيل تطبيق بسيط.

ثانياً: الكفايات الاجتماعية والشخصية.

- ▶ العمل ضمن فريق.
- ▶ تقبل التغذية الراجعة.
- ▶ احترام رأي الآخرين.
- ▶ مصداقية التعامل مع الزبون.
- ▶ حفظ خصوصية الزبون.
- ▶ الاستعداد للاستجابة لطلبات الزبون.
- ▶ التحلي بأخلاقيات المهنة أثناء العمل.
- ▶ التمتع بالفكر الريادي أثناء العمل.
- ▶ استخلاص النتائج ودقة الملاحظة.
- ▶ الاتصال والتواصل الفعال.
- ▶ الاستعداد للاستفسار والاستفادة من ذوي الخبرة.
- ▶ الإفادة من مقترحات الآخرين.
- ▶ امتلاك مهارة التأمل الذاتي.
- ▶ الاستعداد التام لتقبل ملحوظات الزبون وانتقاداتهم.
- ▶ القدرة على الإقناع.

ثالثاً: الكفايات المنهجية:

- ▶ توظيف التعلم التعاوني.
- ▶ القدرة على إدارة الحوار وتنظيم النقاش.
- ▶ امتلاك مهارة البحث العلمي، والقدرة على توظيف أساليبه.
- ▶ القدرة على التخطيط.
- ▶ القدرة على استمطار الأفكار.
- ▶ استخدام المصادر والمراجع المختلفة.
- ▶ توثيق نتائج العمل وعرضها.
- ▶ تقييم النتائج وعرضها.
- ▶ توظيف التكنولوجيا والبرامج الإحصائية وتمثيل البيانات.
- ▶ جمع البيانات وتحليلها وتبويبها.



قواعد الأمان والسلامة:



- ✓ الالتزام بالجلسة الصحيّة أمام الحاسوب .
- ✓ إراحة العينين كلّ ربع ساعة .
- ✓ الجلوس في غرفة فيها إضاءة مناسبة وتهوية جيّدة .
- ✓ الابتعاد عن الحاسوب كلّ ساعة، أو القيام ببعض التمارين الرياضيّة .



الموقف التعليمي التعلّمي (4 - 1):

تجهيز بيئة صناعة تطبيقات الأندرويد

وصف الموقف التعليمي: زار خالدٌ مشغلاً لتطبيقات الهاتف المحمول في مدرسة يافا المهنيّة، لمساعدته في تهيئة جهازه للبدء في تطوير تطبيقات الأندرويد.

العمل الكامل			
الموارد	المنهجية (استراتيجية التعلّم)	وصف الموقف الصّفيّ	خطوات العمل
<ul style="list-style-type: none">• وثائق (طلب خالد، نشرات، كتب ومصادر متنوعة).• التكنولوجيا (روابط إلكترونية متنوعة من مصادر موثوقة، حاسوب، إنترنت)	<ul style="list-style-type: none">• حوار ومناقشة.• التعلم التعاوني/ مجموعات.• البحث العلميّ.	<p>أجمع البيانات من خالد عن:</p> <ul style="list-style-type: none">• مواصفات جهاز الحاسوب الذي يستخدمه (ذاكرة التخزين، نظام التشغيل، الذاكرة العشوائية، وحدة المعالجة المركزيّة، ذاكرة الرسوم). <p>أجمع البيانات عن:</p> <ul style="list-style-type: none">• برنامج java jdk .• برنامج الأندرويد ستوديو.• .Sdk• المحاكي (الجهاز الافتراضيّ).	أجمع البيانات وأحلّها
<ul style="list-style-type: none">• وثائق (نشرات، كتب ومصادر متنوعة)• الشبكة العنكبوتية.	<ul style="list-style-type: none">• تعلم تعاونيّ/مجموعات• حوار ومناقشة .	<ul style="list-style-type: none">• أصنّف البيانات (الحاسوب، حجم الذاكرة، ونظام التشغيل، ووحدة المعالجة المركزيّة، والبرامج المستخدمة).• أحدّد المعطيات (البرامج المستخدمة).• أختار برنامج java jdk الملائم مع نظام التشغيل.• أحدّد برنامج الأندرويد ستوديو المتوافق مع نظام التشغيل.• أحدّد sdk المناسبة.• أحدّد المحاكي (الجهاز الافتراضي)• المناسب.• إعداد جدول زمنيّ للتنفيذ.	أخطّط وأقرّر



<ul style="list-style-type: none"> • حاسوب . • إنترنت . 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات • حوار ومناقشة . • عصف ذهني/استمطار الأفكار . 	<ul style="list-style-type: none"> • اتَّفَق مع خالد على تنزيل البرامج المطلوبة وتثبيتها، لتهيئة جهاز الحاسوب الخاص به . • أنزل وأثبت برنامج java jdk . • أنزل وأثبت برنامج الأندرويد ستوديو . • أثبت sdk . • أنشئ المحاكى (الجهاز الافتراضي) . 	<p>أنفذ</p>
<ul style="list-style-type: none"> • الخطة المعدة سابقاً • الطلب الخاص بخالد . • حاسوب . 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات • حوار ومناقشة . 	<ul style="list-style-type: none"> • اتَّحَقَّ من تنزيل البرامج المطلوبة وتثبيتها . • أتأكد من ملائمة البرامج مع الحاسوب . • أتَّحَقَّ من عمل برنامج الأندرويد ستوديو دون وجود أي خلل في تشغيلها . • أختبر المحاكى (الجهاز الافتراضي) . 	<p>أتَّحَقَّ</p>
<ul style="list-style-type: none"> • حاسوب . • شاشة عرض . 	<ul style="list-style-type: none"> • حوار ومناقشة . • تعلم تعاوني/مجموعات . • عمل . 	<p>أوثق البيانات التي تتعلق بـ:</p> <ul style="list-style-type: none"> • برنامج java jdk . • برنامج الأندرويد ستوديو . • Sdk . • المحاكى (الجهاز الافتراضي) . • أقدم عرضاً تقديمياً . • فتح ملف بالحالة (تهيئة جهاز خالد للبدء في تطوير تطبيقات الأندرويد) . 	<p>أوثق وأقدم</p>
<ul style="list-style-type: none"> • نماذج للتقويم معدة سابقاً . • طلب خالد . • شروط تطوير تطبيقات الأندرويد 	<ul style="list-style-type: none"> • حوار ومناقشة . 	<ul style="list-style-type: none"> • مدى فعالية بيئة عمل تطبيقات الأندرويد على جهاز الحاسوب الخاص بخالد، ورضاه عن ذلك . • ملائمة الخدمة للمعايير والمواصفات الخاصة بتطوير تطبيقات الأندرويد . 	<p>أقوم</p>



أناقش: مراحل تطوُّر نظام الأندرويد.



تجهيز بيئة صناعة تطبيقات الأندرويد

أتعلم :



نشاط نظري (1):



أبحث عن المواصفات المفضلة لجهاز الحاسوب القادر على تثبيت برنامج الأندرويد ستوديو عليه.

البرامج والأدوات اللازمة للبدء في عمل تطبيقات الأندرويد:

- Java JDK
- Android Studio
- SDK
- محاكي أندرويد (Emulator)

نشاط للاطلاع (2):



أناقش مع زملائي طريقة تنزيل وتثبيت Java JDK، كما مرّ معنا في الوحدة السابقة.

Android Studio

هو برنامج نقوم من خلاله بتصميم تطبيقات الأندرويد وتثبيتها.

طريقة التنزيل:

أبحث في جوجل عن Android Studio، ثم أختار، كما في الصورة الآتية:

android studio

الكل صور فيديو الكتب الأخبار المزيد الإعدادات الأدوات

حوالي 976,000,000 من النتائج (عدد النواحي: ٠,٢٦)

[Download Android Studio and SDK tools | Android ...](#)
ترجم هذه الصفحة <https://developer.android.com/studio/>
Find opportunities to reduce your Android app size by inspecting the contents of your app APK file, even if it wasn't built with Android Studio. Inspect the manifest ...

[Meet Android Studio](#)
Android Studio is the official Integrated Development ...

[Install Android Studio](#)
To install Android Studio on Windows, proceed as follows:

[Run apps on the Android ...](#)
The Android Emulator simulates Android devices on your ...

[Android Studio release notes](#)
SDK tools - Android emulator - Known issues - SDK platform

[Android Studio features](#)
SDK Tools release notes - Android emulator - SDK platform - ...

[Android Studio download ...](#)
Android Studio download archives. This page provides an archive ...

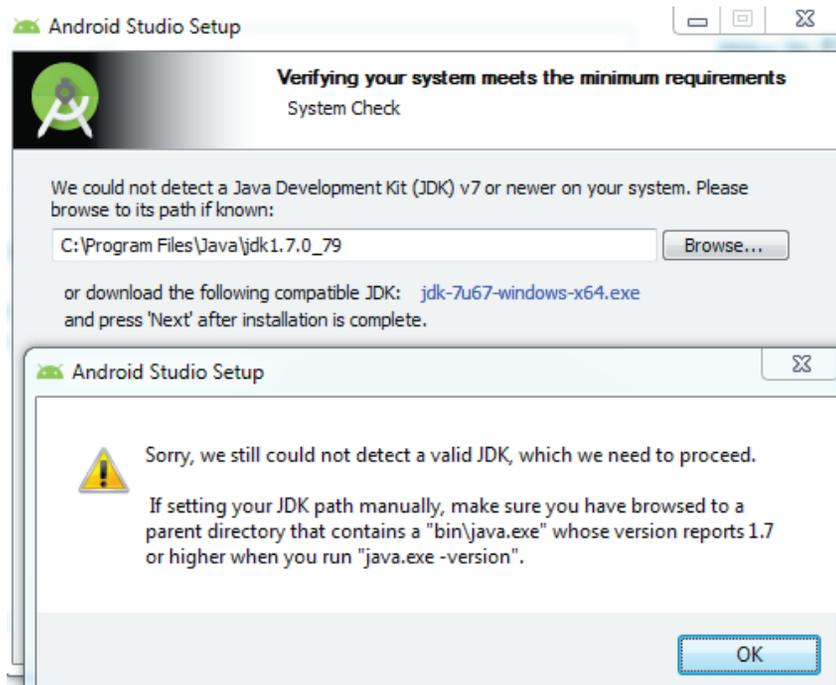


طريقة التثبيت:

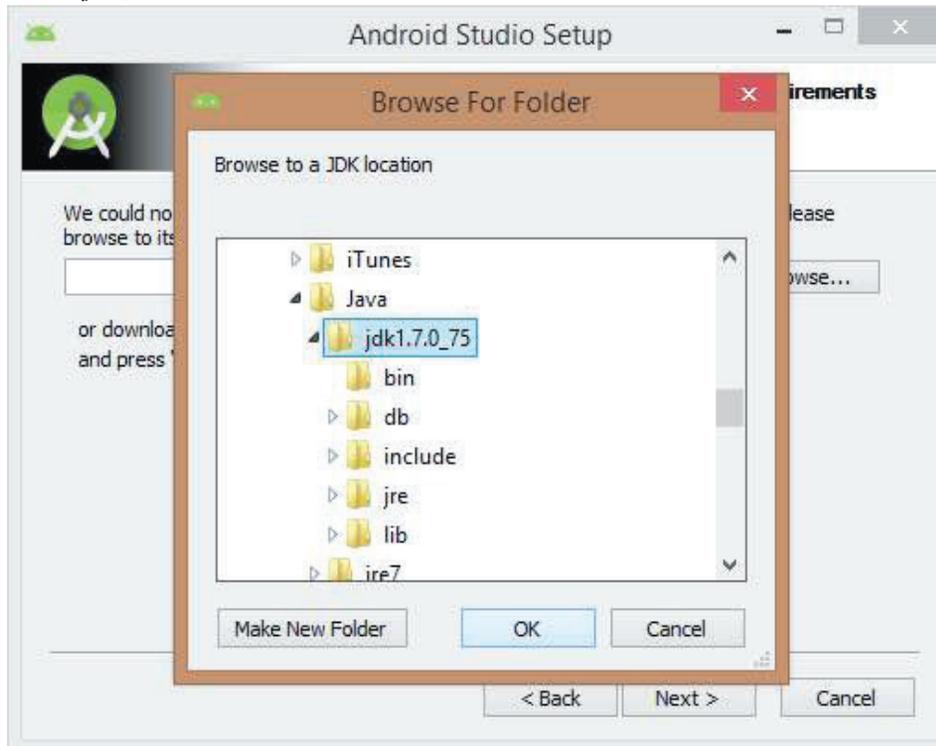
بعد اكتمال تنزيل البرنامج أفتح الملف التنفيذي للبرنامج، وأتبع الخطوات الآتية:



أحياناً في بعض نسخ الويندوز أثناء التثبيت لا يتعرّف البرنامج على مكان ملفات الجافا jdk على الجهاز، وتظهر شاشة كما في الصورة ، ولحلّ هذه المشكلة أتبع الخطوة الآتية:

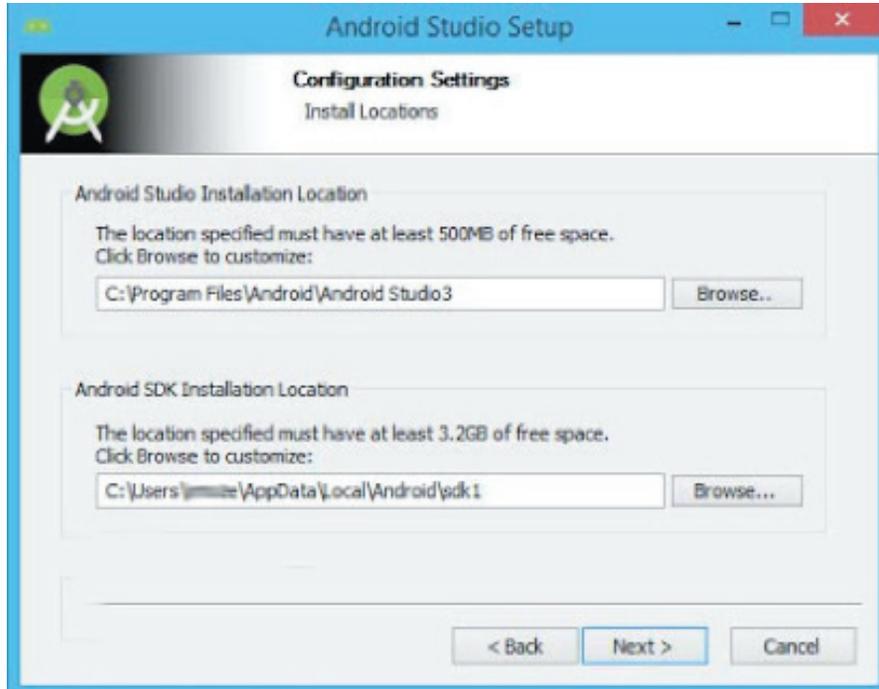


أضيف مسار ملف الجافا يدوياً بعد الضغط على زر استعراض (Browse) ، كما هو موضح في الصورة الآتية:



التأكد من اختيار Android sdk, Android Virtual Device ثم اضغط Next، كالاتي:





بعد الضغط على Finish، يفتح البرنامج لأول مرة يطلب تنزيل ملفات sdk الضرورية لعمل البرنامج.

أناقش:

تعدّ sdk مُهمّة في عمل برنامج أندرويد ستوديو.



ما هي sdk :

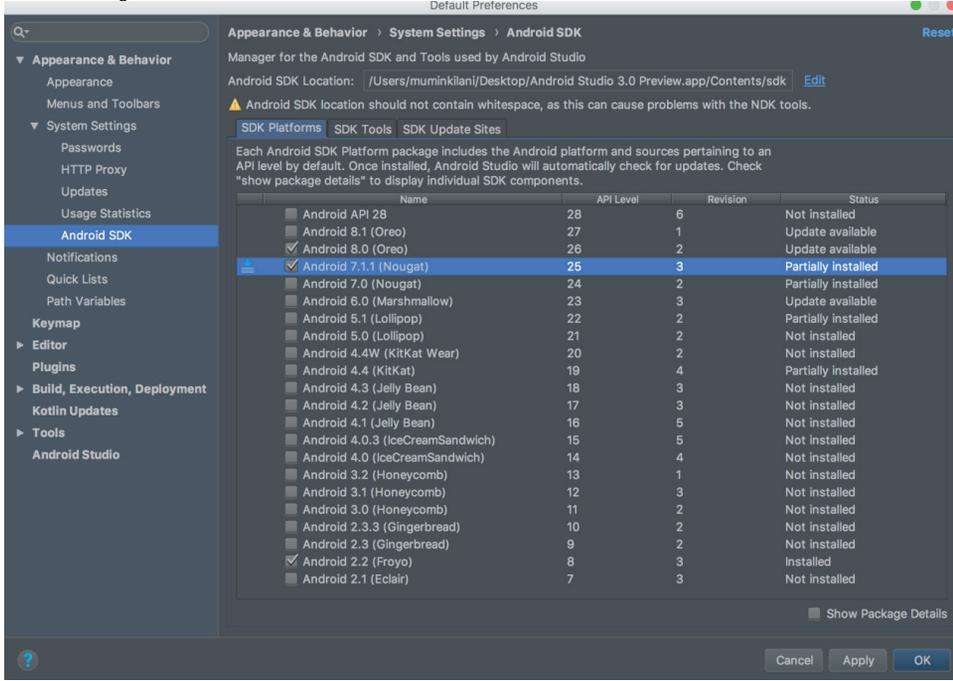
حزمة (Android Software Development Kit) وهي عبارة عن مجموعة أدوات مُتنوّعة تُساعد في تطوير التطبيقات وكتابة الأكواد الأساسية واكتشاف المشاكل وتملك نظاماً مرناً لتطوير التطبيقات وتنفيذ الأوامر بسهولة حتى اتمكّن من إنشاء تطبيق بجودة عالية. على سبيل المثال لتطوير تطبيقات تعمل على نظام التشغيل (Android 7.1.1)، وهو نظام (Nougat) يجب تنزيل (sdk) الخاصّة بهذا النظام.

طريقة تنزيل sdk :



من شريط الأدوات أختار sdk manager

أختار إصدار الأندرويد الذي أريد تنزيل sdk الخاصّة به، ثم أضغط على Apply، كالاتي.



أنتظر حتى يتمّ التنزيل والتثبيت بشكل تلقائيّ، وبعد ذلك يكون البرنامج جاهزاً للاستخدام.

ملحوظة:

تعتمد فترة التنزيل والتثبيت على سرعة الإنترنت وسرعة الجهاز.

نشاط نظري (3):

أبحث عن وظائف محاكي الأندرويد.



محاكي أندرويد:

هو برنامج يعمل بنظام أندرويد على نظام تشغيل آخر مثل ويندوز أو ماكنتوش....، يمتلك خصائص الأجهزة الذكية التي تعمل بنظام الأندرويد، مثل: الهواتف الذكية، والساعة الذكية، وغيرها، ويقوم بوظائف عديدة، منها تشغيل تطبيقات الأندرويد.

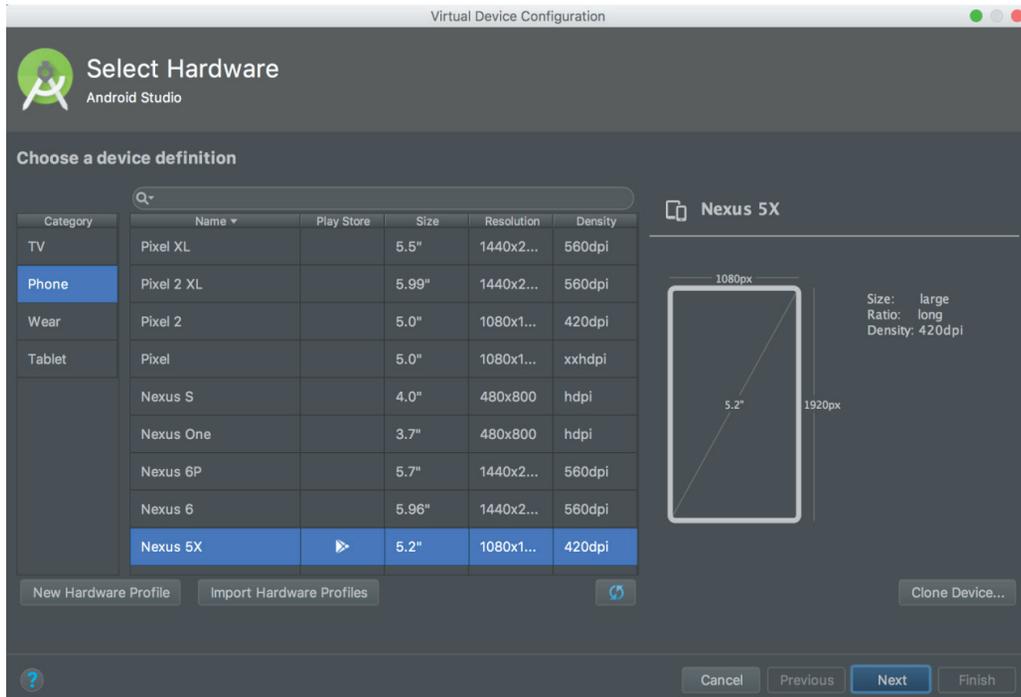
طريقة إعداد المحاكي:

1. عن طريق برنامج الأندرويد ستوديو.
2. عن طريق تنزيل محاكي خارجي من الإنترنت.

محاكي أندرويد ستوديو:

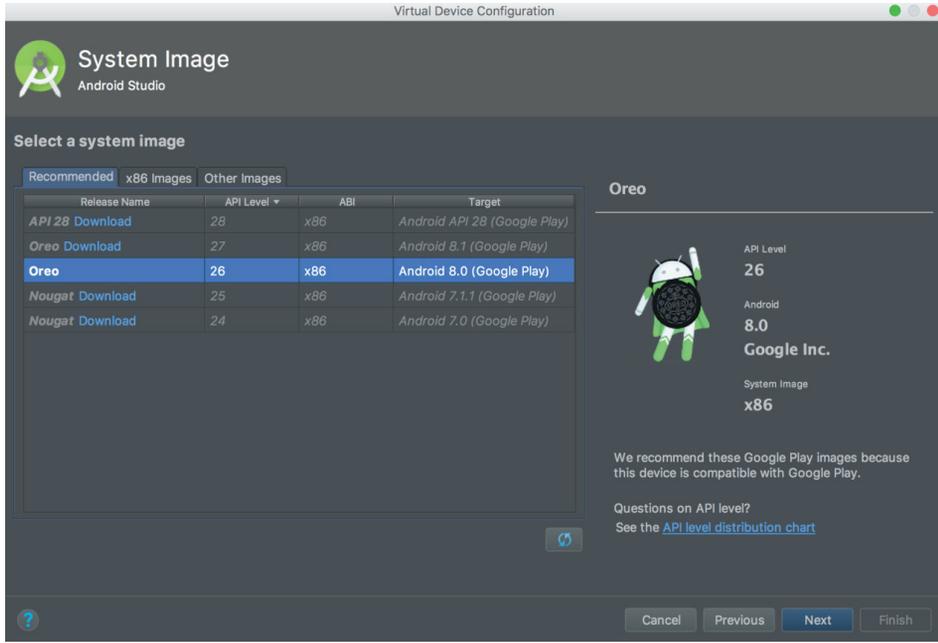


- بعد فتح برنامج أندرويد ستوديو، اضغط على أيقونة AVD manager.
- تظهر نافذة أختار **Create Virtual Device...**
- أختار نوع الجهاز، ثم اضغط **next**، كما هو موضّح في الصورة الآتية:



- أختار إصدار نظام التشغيل وأضغط **next**. (ملحوظة: إذا لم يكن نظام التشغيل متوفراً اضغط على كلمة **Download**، ثم أنتظر حتى يكتمل التنزيل، وأضغط **next**)، كالاتي:

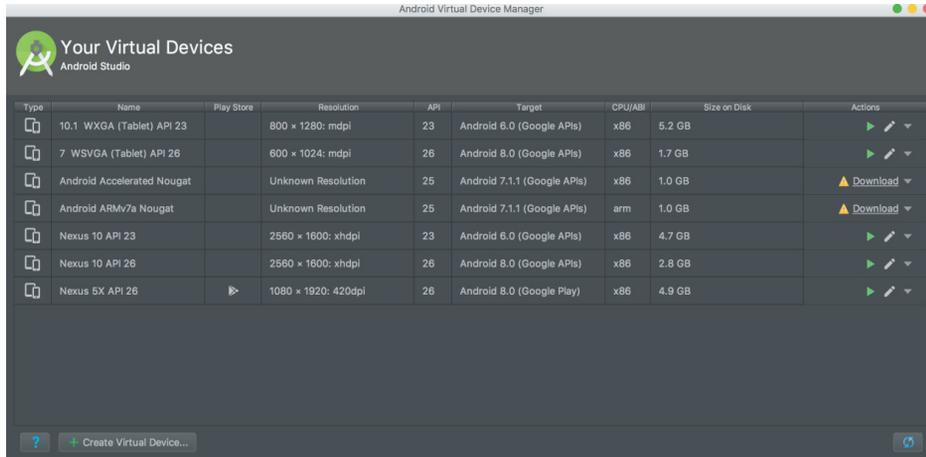




- في النافذة الأخيرة أضغط finish، وانتظر حتى يكتمل إنشاء المحاكي وسوف يعمل تلقائياً فقط في أول مرة.

طريقة تشغيل المحاكي:

أضغط على AVD manager، تظهر نافذة بقائمة الأجهزة التي قمت بإنشائها، أشغل المحاكي الذي أريده بالضغط على زر التشغيل الأخضر المجاور للمحاكي، كما هو موضح في هذه الصورة الآتية:



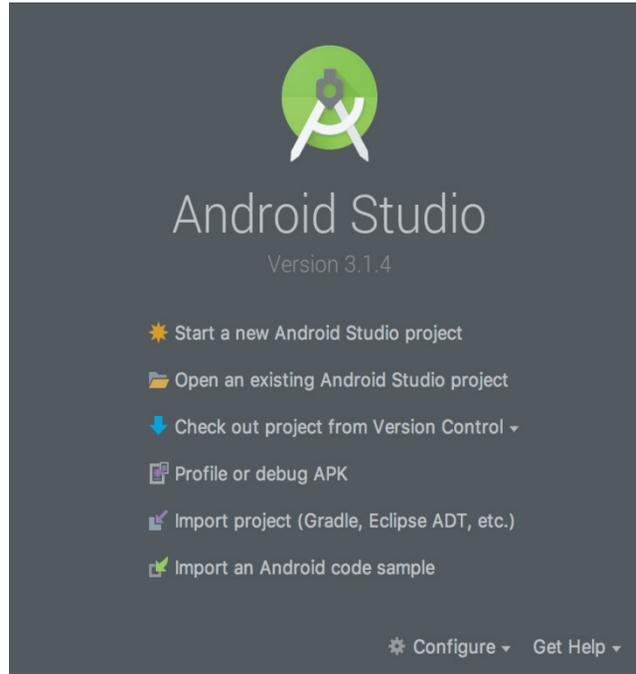


ابحث عن أسماء (محاكيّات) خارجيّة، تثبت بشكل منفصل.
◀ بعض المحاكيات التي تثبت بشكل منفصل:

- Genymotion
- Xamarin

بيئة العمل ونظام الملفات:

◀ خطوات إنشاء مشروعك الأول في عالم الأندرويد، موضّح في الصور الآتية:



أختار الخيار الاول (Start a new Android Studio project)
ثم تأتي النافذة كما في الصورة التالية:

Application name أكتب هنا اسم التطبيق باللغة الانجليزية، يبدأ الاسم بحرف كبير
1- My Application

Company domain يرجى الانتباه أن يكون هذا الاسم لا يحتوي على كلمة example
muminkilani.com

Project location
/Users/muminkilani/Downloads/MyApplication

Package name
com.muminkilani.myapplication Edit

Include C++ support
2- Include Kotlin support للمرجة بلغة الكوتلن، نضغط على هذا الخيار

3- Cancel Previous Next Finish

من الصورة التالية اختار أقل إصدار أندرويد، وهذا الاختيار يعني أن إصدارات الأندرويد الأقل منه لا يعمل عليه هذا التطبيق، وأن الإصدارات المساوية له أو أعلى منه يتوافق معها التطبيق.

Target Android Devices

Select the form factors and minimum SDK
Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.

Phone and Tablet
API 15: Android 4.0.3 (IceCreamSandwich)
By targeting API 15 and later, your app will run on approximately 100% of devices. [Help me choose](#)
 Include Android Instant App support

Wear
API 21: Android 5.0 (Lollipop)

TV
API 21: Android 5.0 (Lollipop)

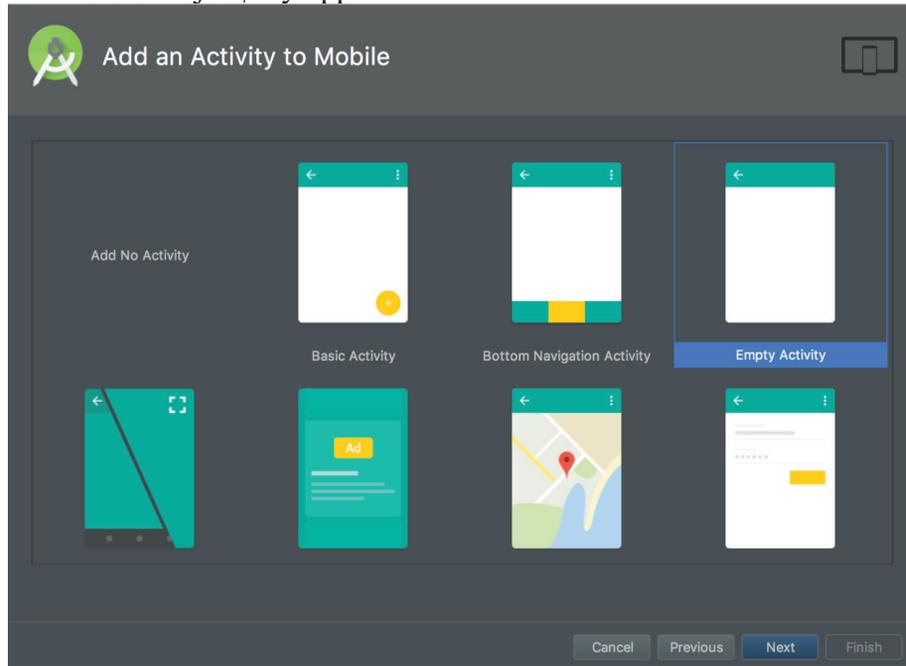
Android Auto

Android Things
API 24: Android 7.0 (Nougat)

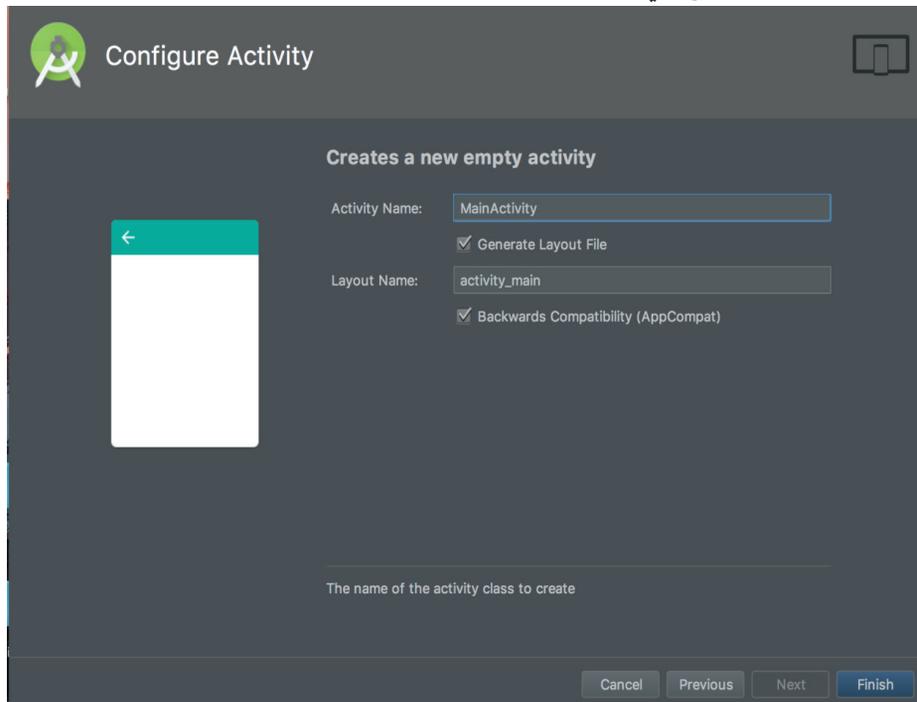
Cancel Previous Next Finish



ملاحظة: مكان تخزين المشروع يجب أن لا يحتوي على أحرف وكلمات عربية مثال: /User/Admin AndroidProject/MyApp



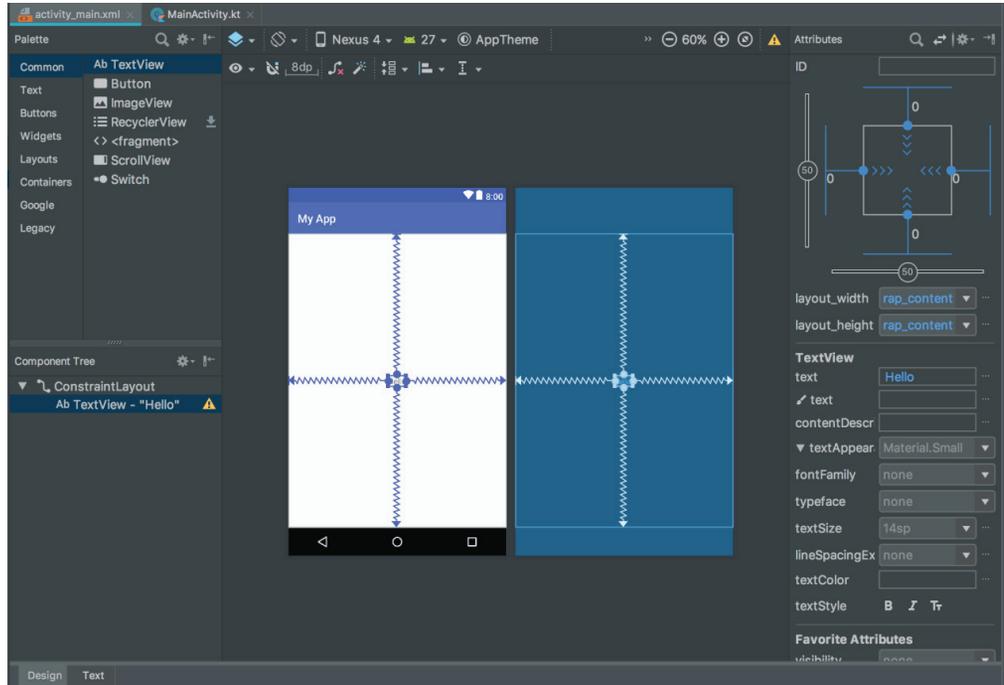
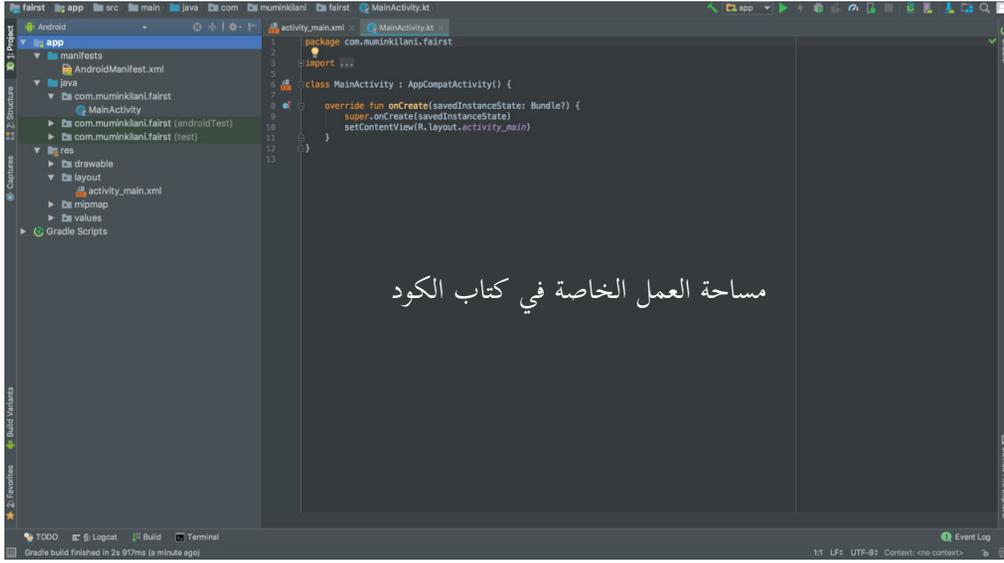
أختار نوع بيئة العمل كما هو موضح في الصورة أعلاه.



أضغط finish ثم يبدأ البرنامج بفتح مشروع، انتظر حتى ينتهي البرنامج من جميع إجراءات التشغيل اللازمة، قد يستغرق بعض الوقت.



التعرف على بيئة العمل :



مساحة العمل الخاصة في تصميم الواجهات.

أناقش:

ملف AndroidManifest.xml يُعدُّ أهمّ ملف في المشروع.



الأنواع الرئيسيّة للملفات في المشروع:

- **Manifests** : الذي يحتوي على ملف `AndroidManifest.xml` ويُعدُّ الإدارة العامّة في المشروع، تُعرّف فيه جميع النشاطات (**Activities**) المستخدمة في المشروع، مكتوب بلغة **xml**، أهمّ المهام التي يقوم بها إعطاء صلاحيّات وصول مكوّنات البرمجة للمكوّنات الماديّة لجهاز الأندرويد.
- مجلد الأكواد (**java**): يحتوي على الكود البرمجيّ الخاصّ بالمشروع.
- مجلد المصادر (**res**): المجلد الذي يحتوي على الملفات الخاصّة في تصميم الواجهات، أهمّها:
 - مجلد **drawable**: مجلد خاص في الصور، التي يجب أن تحمل خصائص معيّنة، أهمّها:
 - « لا يحتوي اسم الصورة على أحرف كبيرة ورموز
 - « تجنّب استخدام الأرقام في التسمية حتى لا يحدث مشاكل أثناء التنفيذ
 - « حجم الصورة.
 - « يفضّل أن تكون الصورة من نوع `png`.
- مجلد **layout**: مجلد تصميم الواجهات والتعامل مع ملفات `xml`، بعض الأمور التي يجب مراعاتها: لا يحتوي اسم ملف `xml` على أحرف كبيرة ورموز، تجنّب استخدام الأرقام في التسمية، واستخدام (-) السفليّة إذا أردتُ الفصل في التسمية بين كلمتين.

الأسئلة

- 1- ما هي البرامج والأدوات اللازمة للبدء في عمل تطبيقات الأندرويد؟
- 2- أشرح باختصار طريقة اعداد محاكي الخاص بالأندرويد ستوديو؟



الموقف التعليمي التعلّمي (4 - 2):

تصميم تطبيق أندرويد (1)

وصف الموقف التعليمي: طلبت دائرة الإحصاء من مشغل لتطبيقات الهاتف المحمول في مدرسة عسقلان المهنيّة، تصميم واجهة تطبيق أندرويد بسيطة لإحصاء عدد السكان.

العمل الكامل			
الموارد	المنهجية (استراتيجية التعلم)	وصف الموقف الصّفيّ	خطوات العمل
<ul style="list-style-type: none"> • وثائق (طلب دائرة الإحصاء ، نشرات، كتب ومصادر متنوعة). • التكنولوجيا(روابط إلكترونية متنوعة من مصادر موثوقة، حاسوب، إنترنت، هاتف يعمل بنظام الأندرويد). 	<ul style="list-style-type: none"> • حوار ومناقشة. • التعلم التعاوني/ مجموعات. • البحث العلميّ 	<p>أجمع البيانات من دائرة الإحصاء عن:</p> <ul style="list-style-type: none"> • البيانات الشخصية المطلوب إدخالها عن كلّ فرد. • إصدار نظام الأندرويد المعتمد استخدامه في الدائرة. <p>أجمع البيانات عن :</p> <ul style="list-style-type: none"> • العناصر (views) المستخدمة للإدخال والإخراج. 	أجمع البيانات وأحلّها
<ul style="list-style-type: none"> • وثائق (نشرات، كتب ومصادر متنوعة) • الشبكة العنكبوتية. 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات. • حوار ومناقشة . 	<ul style="list-style-type: none"> • أصنّف البيانات (الاسم الرباعي، العمر، الحالة الاجتماعية، نظام الأندرويد المستخدم). • أحدّد المعطيات(الاسم الرباعي، العمر، الحالة الاجتماعية). • أحدّد العناصر (views) المستخدمة لإعداد الواجهة. • إعداد جدول زمنيّ للتنفيذ. 	أخطّط وأقرّر
<ul style="list-style-type: none"> • حاسوب. • برنامج Android Studio. • جهاز افتراضي أو جهاز يعمل بنظام أندرويد. 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات. • حوار ومناقشة. • عصف ذهنيّ/استمطار الأفكار 	<ul style="list-style-type: none"> • اختيار عدد من الواجهات المقترحة قابلة للاستخدام في تطبيق الأندرويد. • أتفق مع دائرة الإحصاء على اختيار واجهة التطبيق المناسبة. • أصمّم واجهة التطبيق. • أعمل اختبار للمشروع على جهاز افتراضي، أو على جهاز يعمل بنظام أندرويد. 	أنفّذ



<ul style="list-style-type: none"> • الخطة المعدّة سابقاً. • الطلب الخاص بدائرة الإحصاء. • حاسوب. • برنامج Android Studio. • جهاز افتراضي أو جهاز يعمل بنظام الأندرويد. 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات. • حوار ومناقشة. 	<ul style="list-style-type: none"> • أتحقّق من ملاءمة اختيار الواجهة المقترحة للتطبيق. • أتأكّد من ملاءمة تصميم واجهة التطبيق. • أتحقّق من تنفيذ التطبيق وظهوره على الجهاز بشكل مناسب. 	<p>أتحقّق</p>
<ul style="list-style-type: none"> • حاسوب • شاشة عرض • جهاز افتراضي أو جهاز يعمل بنظام أندرويد. 	<ul style="list-style-type: none"> • حوار ومناقشة. • تعلم تعاوني/مجموعات. 	<p>أوثق البيانات التي تتعلق بـ:</p> <ul style="list-style-type: none"> • العناصر (views) المستخدمة للإدخال والايخراج. • أقدم عرضاً تقديمياً. • إعداد ملف بالحالة (تصميم واجهة تطبيق أندرويد بسيطة لإحصاء عدد السكان). 	<p>أوثق وأقدم</p>
<ul style="list-style-type: none"> • نماذج للتقويم معدّة سابقاً • طلب دائرة الإحصاء. 	<ul style="list-style-type: none"> • حوار ومناقشة. 	<ul style="list-style-type: none"> • رضا دائرة الإحصاء عن تصميم التطبيق. • ملاءمة التصميم للمعايير والمواصفات. 	<p>أقوم</p>



أبحث عن ميزات صور png.



تصميم تطبيق أندرويد (1)

نشاط نظري (1):

أبحث عن دور xml في برمجة تطبيقات الأندرويد.

xml في الأندرويد:

لغة ترميز، لها مهمّات عدّة في الأندرويد، أهمّها تصميم واجهات التطبيق، وهي اللغة المستخدمة في ملف `AndroidManifest.xml`، وفي ملفات مجلد `.res`.

طريقة الكتابة في ملفات xml:

1. الوسم

كلمة مفتاحيّة تصنّف معلومات معيّنة، مثل (صورة، نص،....)، يكتب هذا الوسم بين `< >` ، كما يأتي:

وسم ابتدائي `< ScrollView >`

وسم إغلاقي `< /ScrollView >`

2. العنصر:

الكلمات بين الوسم البدائيّ والنهائيّ، وتُسمّى محتوى العنصر.

البيانات المكتوبة ضمن وسم البداية تُسمّى خصائص العنصر.

```
< string name="app_name" > My App </string >
```

تكون (name) هي الخاصيّة، وقيمتها (app_name) بينما (My App) هي المحتوى في هذا المثال.

تعريف ملف xml:

هذا السطر أجده في بداية كلّ ملف xml لتعريف أنّ هذا هو ملف xml.

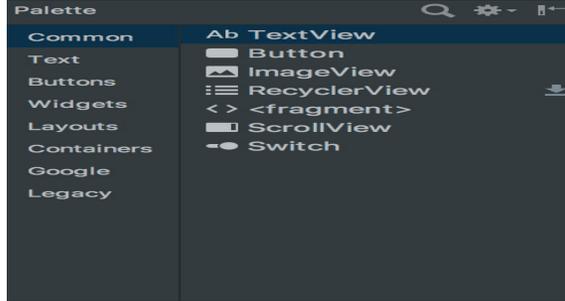
```
<?xml version="1.0" encoding="utf-8"?>
```



عناصر الواجهات وخصائصها (views, viewGroup):

1. العناصر views:

عناصر لها خصائص ووظائف مختلفة تختلف وفقاً لنوع العنصر، كالاتي:

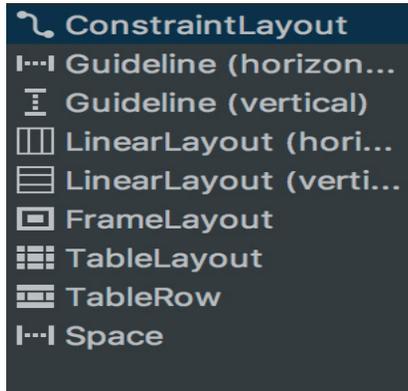


أهم العناصر التي أستخدمها:

- button: عبارة عن زر (كبسة) عند الضغط عليه يُنفذ أمراً ما.
- TextView: تُستخدم لعرض نصّ معيّن.
- Plain Text: تُستخدم لإدخال نصّ أو قيمة، وتكون بأشكالٍ عدّة، منها ما يُستخدم لإدخال رقم سريّ، أو إيميل، أو، ويقروها البرنامج على شكل نصّ.
- imageView: تُستخدم لعرض صورة.

2. ViewGroup: حاوية تحتوي على عدة عناصر views تُسمّى layout، أمثلة كما في

الصورة:



أناقش:

لكلّ view خصائص تختلف عن غيره، لكن هناك خصائص مشتركة.



نشاط نظري + عملي (2):

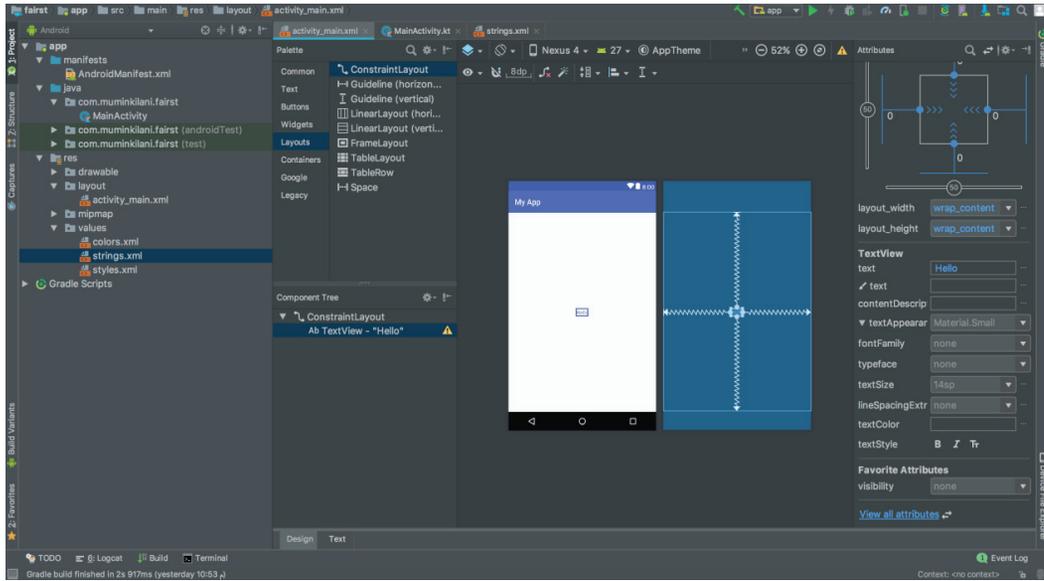


أضيف Button وأحاول أن أكتشف قائمة الخصائص.

نشاط عملي (3):



افتح مشروع جديد وأسميه باسمي كما تعلمنا سابقاً وانتقل لملف layout، تلقائياً تكون TextView مضافة في وسط layout أختارها، ومن قائمة البرنامج اليمنى تكون الخصائص موجودة، كما في الصورة الآتية:



الخصائص الرئيسية للعناصر (views):

- الارتفاع /layout_Height / العرض layout_width:
 - يكون مساوياً ل layout الرئيسية match_parent.
 - يكون ملائماً لما في داخل view تلقائياً ويُسمى wrap_content.
 - نستطيع أن نحدده بقياس رقمي وحدته (بيكسل) px أو dp.
- النص Text: بعض ال views تحتوي على نص يمكن تغييره من الخاصية (text)، وله عدة خصائص، مثل:
 - لون الخطّ textColor.
 - حجم الخطّ textSize ويقاس بوحدة sp.
 - شكل الخطّ textStyle.



3. الخلفيّة background .
4. Padding: تُحدّد بُعدَ ما في داخل العنصر عن حدود العنصر الأربعة (أعلى، أسفل، يمين، يسار).
5. Layout_Margin: تُحدّد بُعد العنصر عن محيطه.
6. ID: أهمّ خاصيّة، ويجب أن تكون قيمته فريدة، بواسطته يرتبط العنصر بالكود البرمجيّ.

نشاط عملي (4):



أطبق الخصائص السابقة على العنصر textView الموجود في نشاط (3).

تصميم الواجهات layout:

أكثر الواجهات استخداماً:

1. ConstraintLayout:

هذه الواجهة تسمح بوضع العناصر في أيّ مكان، لكن يجب الانتباه عند استخدام هذه الواجهة، يجب ربط العنصر من جميع الجهات.

2. linearLayout:

لها نوعان، نوع أفقيّ يرتّب العناصر بشكل أفقيّ Horizontal، ونوع عموديّ يرتّب العناصر بشكل عموديّ vertical.



نشاط عملي (5):



أعمل واجهة تفيد في عمل تطبيق لحساب الوزن المثالي.

أحضّر المعطيات:

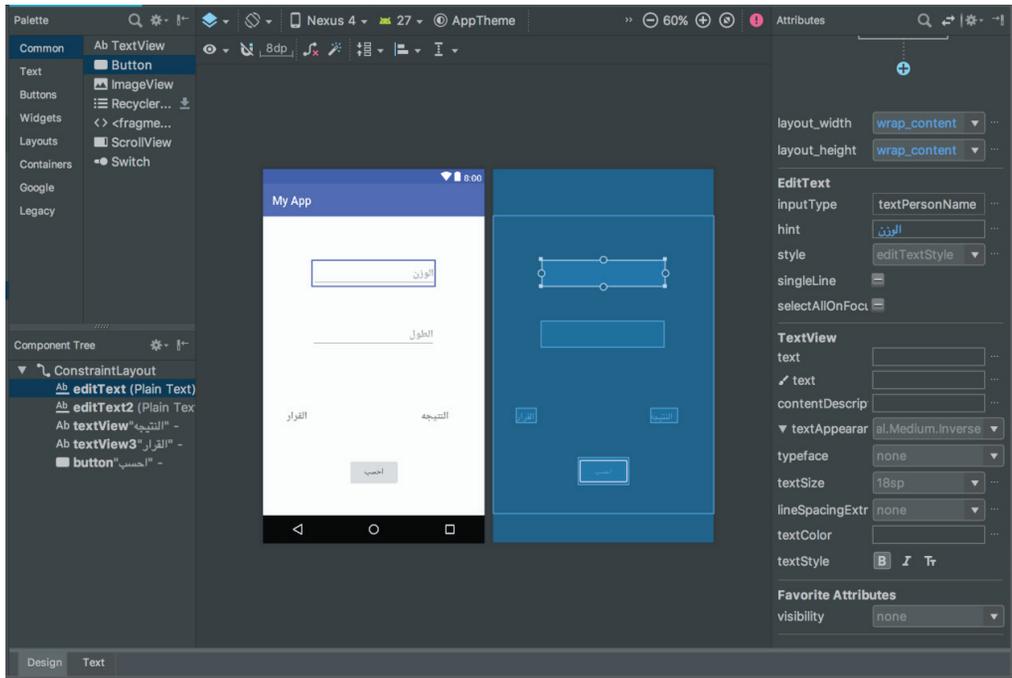
قانون الوزن المثالي (الوزن المثالي = الوزن الحالي ÷ (الطول)²)، مؤشر النحافة أقلّ من ١٨,٥، مؤشر الوزن المثالي يتراوح بين ١٨,٥-٢٤,٩، مؤشر الزيادة بالوزن يتراوح بين ٢٥-٢٩,٥، مؤشر السمنة أكثر من ٣٠.

أفتح الأندرويد ستوديو، ثمّ أنشئ مشروعاً جديداً وأسمّيه **Weight App**، بعد أن يفتح البرنامج بشكل كامل، استناداً إلى المعطيات أختار العناصر التي أريد استخدامها، وهي كما يأتي:

- عنصران من نوع plain text: لإدخال الوزن الحالي، والطول.
- عنصران من نوع text view: لإظهار النتيجة القانون، وإظهار هل الوزن مثالي أم لا.
- عنصر من نوع button: عند الضغط عليه يُنفذ العمليات البرمجية الخاصة لحساب الوزن المثالي.

◀ الآن أبدأ بالعمل :

أضيف العناصر إلى الواجهة الرئيسيّة من خلال السحب والاسقاط في المكان المناسب، أتأكد من أنّ لكلّ عنصر ID خاصّ به، كما هو موضّح في الصورة الآتية:

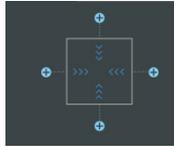


ملحوظة:

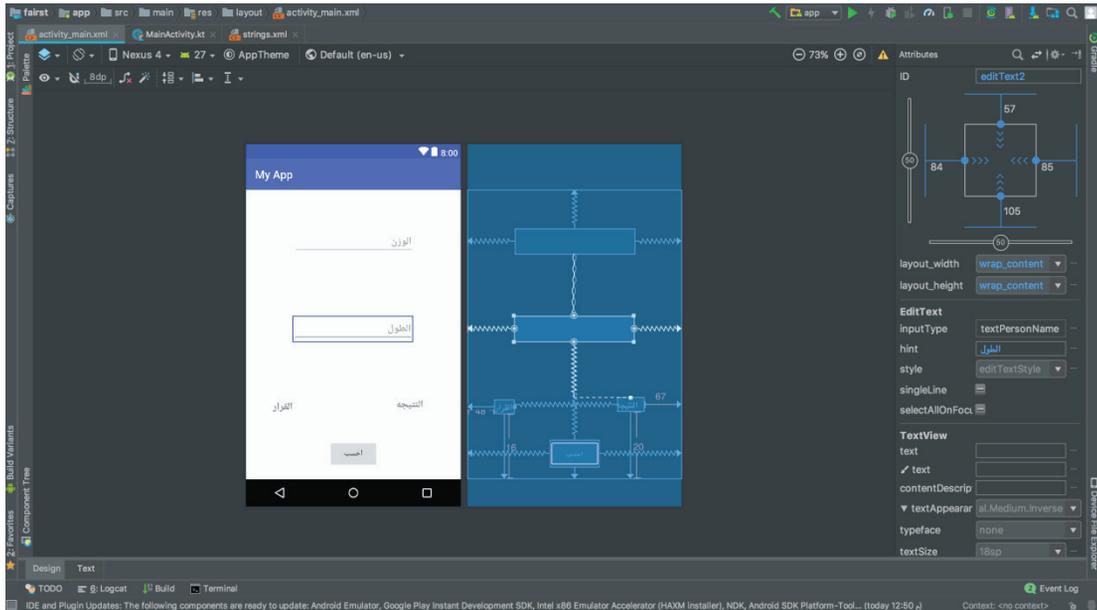
عند تسمية plain text أنتقل إلى خانة hint وأكتب وظيفة العنصر.

أناقش: أكتشف مع زملائي فائدة hint.

بما أنّ الواجهة المستخدمة هي ConstraintLayout، إذن يجب ربط العناصر، عند الضغط على العنصر تظهر دوائر في منتصف الأضلاع، يتم توصيل هذه النقاط بالشكل المناسب، أو من قائمة الخصائص يوجد الشكل الآتي:



أضغط على إشارة (+) يقوم بربط العنصر تلقائياً، وبعد ربط جميع العناصر ينتج معي كالاتي:



ملحوظة:

عند ربط عنصر بآخر من احدى الجهات، لا يجب اعادة ربط العنصر الثاني بالأول من نفس الجهة.

الخطوة الأخيرة أشغّل الجهاز الافتراضي، ثمّ أعملُ اختباراً للبرنامج من خلال الضغط على زر التشغيل **run** الموجود في شريط الأدوات في الأعلى، ويظهر التطبيق كالتالي:



الأسئلة

- 1- أوضح الفرق بين `LinearLayout` الأفقية والعمودية بالرسم؟
- 2- ما هي الخصائص الرئيسية للعناصر؟
- 3- ما هي طريقة الكتابة في ملفات `xml`؟

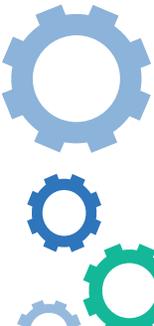


الموقف التعليمي التعلّمي (4 - 3):

تصميم تطبيق أندرويد (٢)

وصف الموقف التعليمي: ذهب محسن إلى مشغل تطبيقات الهاتف المحمول في مدرسة حيفا المهنية، لمساعدته في تصميم واجهة تطبيق أندرويد بسيطة لإيجاد الفروق بين صورتين.

العمل الكامل			
خطوات العمل	وصف الموقف الصّفي	المنهجية (استراتيجية التعلم)	الموارد
أجمع البيانات وأحلّها	<p>أجمع البيانات من محسن عن:</p> <ul style="list-style-type: none"> الصور التي يريد إضافتها. إصدار نظام الأندرويد الذي يستخدمه. <p>أجمع البيانات عن:</p> <ul style="list-style-type: none"> العناصر (views) المستخدمة للإدخال والإخراج. (viewGroup) الواجهة الرئيسية التي تحوي العناصر. 	<ul style="list-style-type: none"> حوار ومناقشة. التعلم التعاوني/ مجموعات البحث العلمي. 	<ul style="list-style-type: none"> وثائق (طلب محسن). التكنولوجيا (روابط إلكترونية متنوعة من مصادر موثوقة، حاسوب، إنترنت، هاتف يعمل بنظام الأندرويد).
أخطّط وأقرّر	<ul style="list-style-type: none"> أصنّف البيانات (الصور المستخدمة، نظام الأندرويد المستخدم). أحدّد المعطيات (الصور المستخدمة). أحدّد نوع (viewGroup) الواجهة الرئيسية المستخدمة. أحدّد العناصر (views) المستخدمة لإعداد الواجهة. إعداد جدول زمني للتنفيذ. 	<ul style="list-style-type: none"> تعلم تعاوني/ مجموعات حوار ومناقشة. 	<ul style="list-style-type: none"> الشبكة العنكبوتية.
أنفّذ	<ul style="list-style-type: none"> اختيار عدد من الواجهات المقترحة قابلة للاستخدام في تطبيق الأندرويد. تفق مع محسن على اختيار واجهة التطبيق المناسبة. أصمّم واجهة التطبيق. أعمل اختباراً للمشروع على جهاز افتراضي، أو على جهاز يعمل بنظام أندرويد. 	<ul style="list-style-type: none"> تعلم تعاوني/ مجموعات حوار ومناقشة. عصف ذهني/ استمطار الأفكار. 	<ul style="list-style-type: none"> حاسوب. برنامج Android Studio. جهاز افتراضي، أو جهاز يعمل بنظام أندرويد.



<ul style="list-style-type: none"> • الخطة المعدّة سابقاً • الطلب الخاص بمحسن . • حاسوب . • برنامج Android Studio . • جهاز افتراضي أو جهاز يعمل بنظام الأندرويد . 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات • حوار ومناقشة . 	<ul style="list-style-type: none"> • أتحقّق من ملاءمة اختيار الواجهة المقترحة للتطبيق . • أتأكّد من ملاءمة تصميم واجهة التطبيق . • أتحقّق من تنفيذ التطبيق وظهوره على الجهاز بشكل مناسب . 	<p>أتحقّق</p>
<ul style="list-style-type: none"> • حاسوب • شاشة عرض • جهاز افتراضي أو جهاز يعمل بنظام أندرويد . 	<ul style="list-style-type: none"> • حوار ومناقشة . • تعلم تعاوني/مجموعات عمل . 	<p>أوثق البيانات التي تتعلّق بـ:</p> <ul style="list-style-type: none"> • العناصر (views) المستخدمة للإدخال والإخراج . • أقدم عرضاً تقديمياً . • إعداد ملف بالحالة (تصميم واجهة تطبيق أندرويد بسيطة لإيجاد الفروق بين صورتين) . 	<p>أوثق وأقدم</p>
<ul style="list-style-type: none"> • نماذج للتقويم معدّة سابقاً • طلب محسن . 	<ul style="list-style-type: none"> • حوار ومناقشة . 	<ul style="list-style-type: none"> • رضا محسن عن تصميم التطبيق . • ملاءمة التصميم للمعايير والمواصفات . 	<p>أقوم</p>



◀ **أناقش:** أفضل استخدام `linearLayout` الأفقيّة لتطبيق الموقف التعليمي السابق.





تصميم تطبيقات الأندرويد (٢)



أعمل واجهة لتطبيق يعمل على التبديل بين صورتين، وتغيير اسم الصورة في كل مرة.

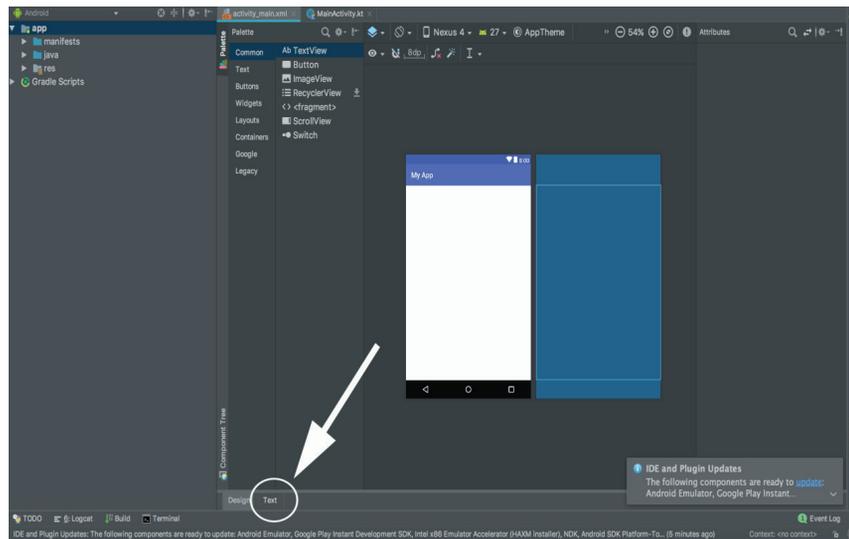
أحضّر المعطيات: الصور المراد عرضها، واسم كل صورة.

أفتح الأندرويد ستوديو، ثم أنشئ مشروعاً جديداً وأسميه My images، بعد أن يفتح البرنامج بشكل كامل، استناداً إلى المعطيات أختار العناصر التي أريد استخدامها، وهي كما يأتي :

- عنصر من نوع image view: لعرض الصور.
- عنصر من نوع text view: لإظهار اسم الصورة.
- عنصر من نوع button: عند الضغط عليه ينفذ العمليات البرمجية الخاصة لتغيير الصور.

◀ الآن أبدأ بالعمل :

بما أنّ العناصر ترتيبها عمودياً أستخدم linearLayout الأفقية، لتغيير نوع الواجهة أدخل على كود xml وأتبع الخطوات الآتية:



عند الدخول على كود xml أستبدل المظلل بالازرق بـ `LinearLayout`، كما يلي :

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

</android.support.constraint.ConstraintLayout>
```

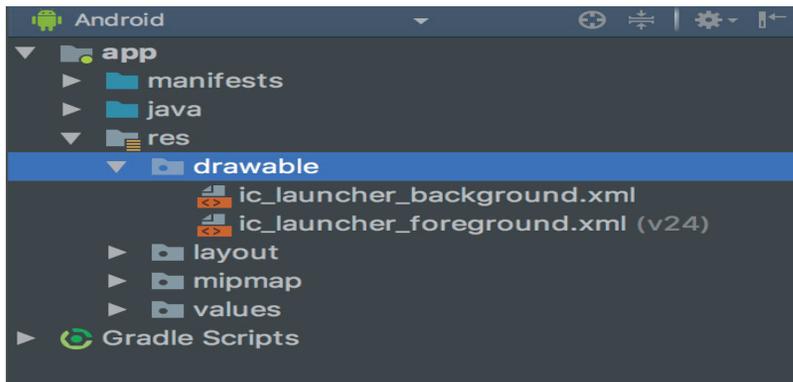
بعد التأكد من تغيير الاسم أكون قد استبدلت الواجهة، ويتبقى إضافة نوع الواجهة أفقية أم عمودية، كما في الصورة الآتية في النص المظلل :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

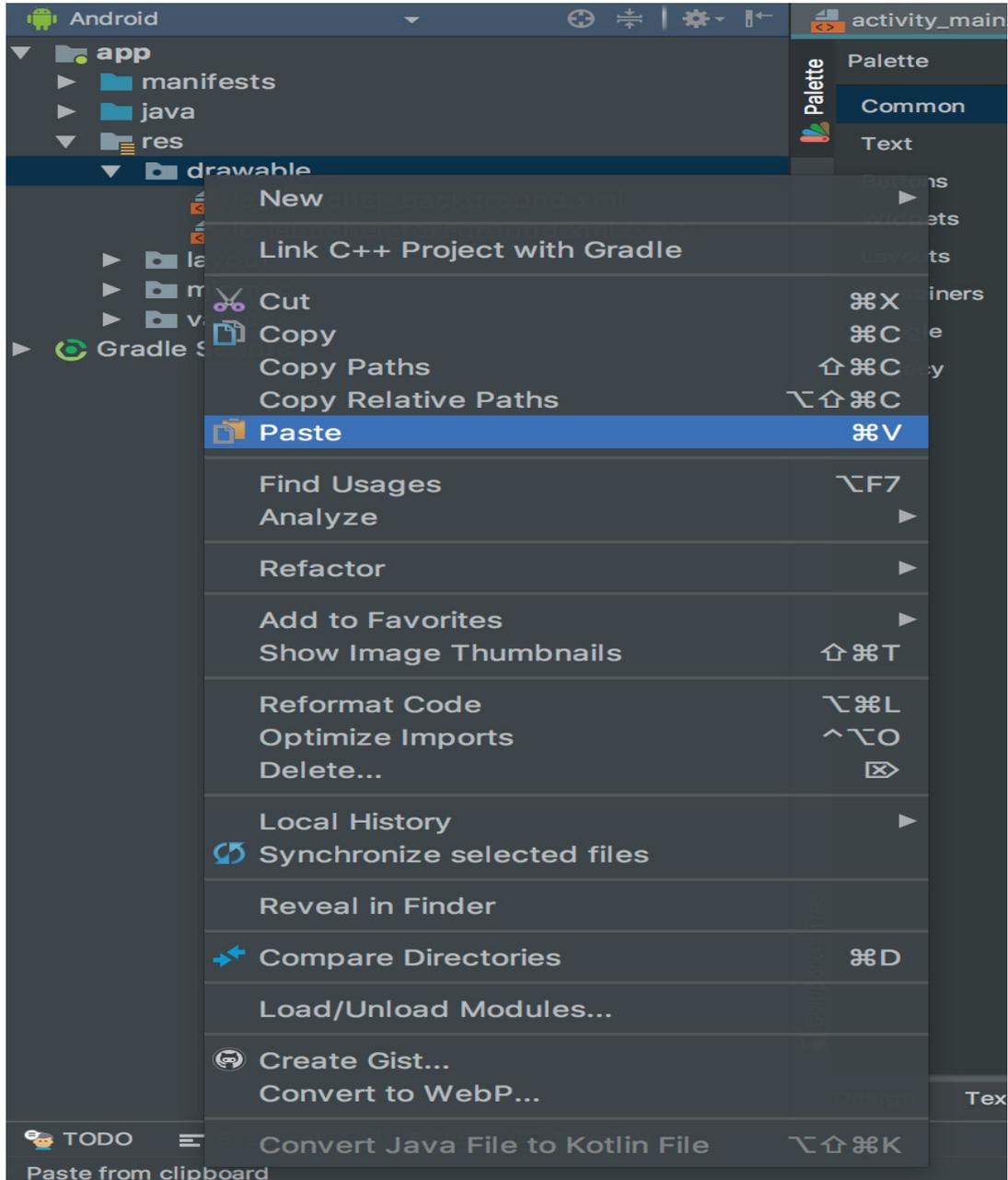
</LinearLayout>
```

أعود لواجهة التصميم :

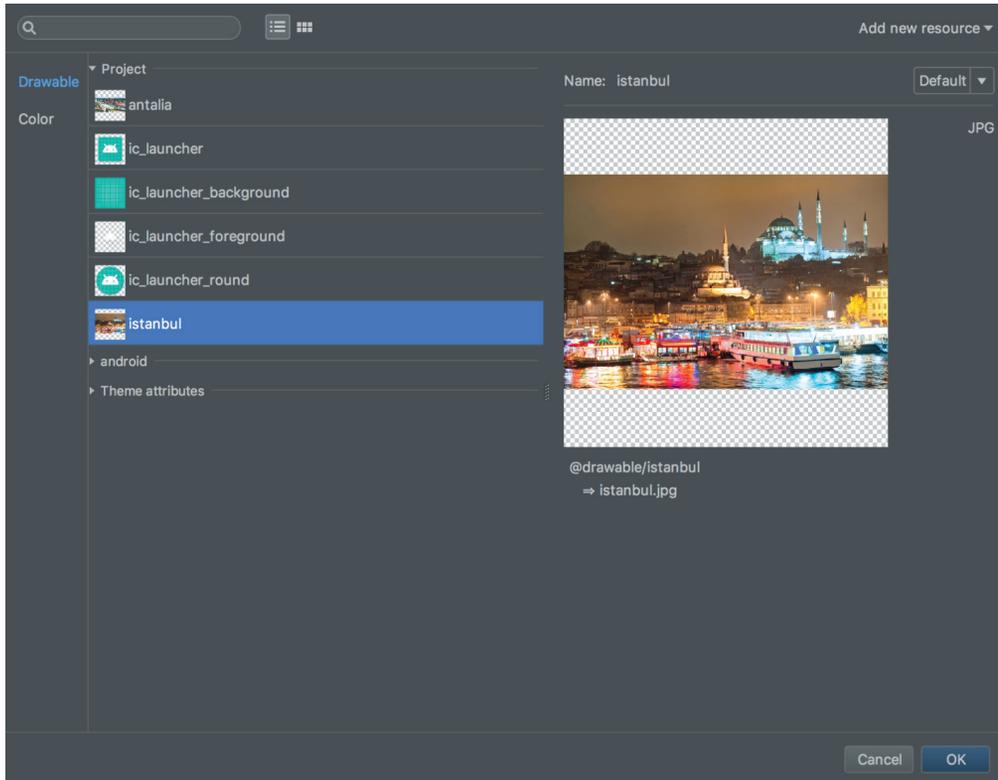
قبل أن أبدأ يجب وضع الصور التي أريدها في مجلد `drawable` من خلال نسخها من مكان تواجدها (سطح المكتب، ملف الصور، المستندات، ...)، وألصقها في الملف، مع مراعاة الخصائص التي ذكرتها سابقاً، كما يأتي :



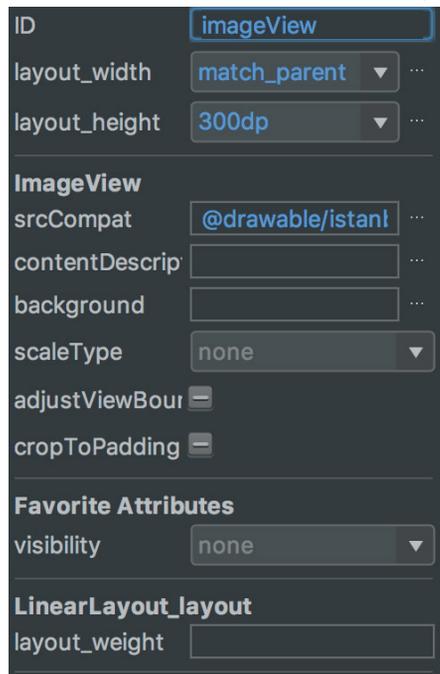
بالنقر على زر الفأرة اليمين تظهر نافذه كما يأتي، أختار أمر اللصق (Paste):



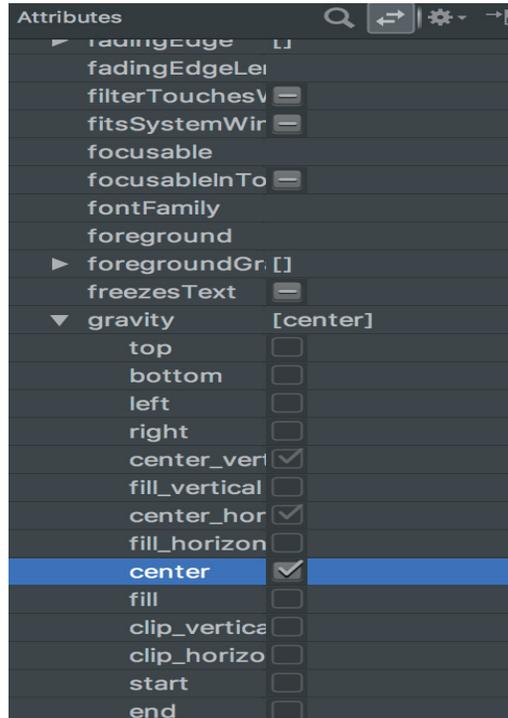
أضيف imageView من خلال السحب والإفلات، ستظهر نافذه لأختار الصورة الافتراضية التي أريد وضعها، أختار من القائمة الجانبيّة في النافذة التي تظهر مجلد Drawable كما في الشكل الآتي:



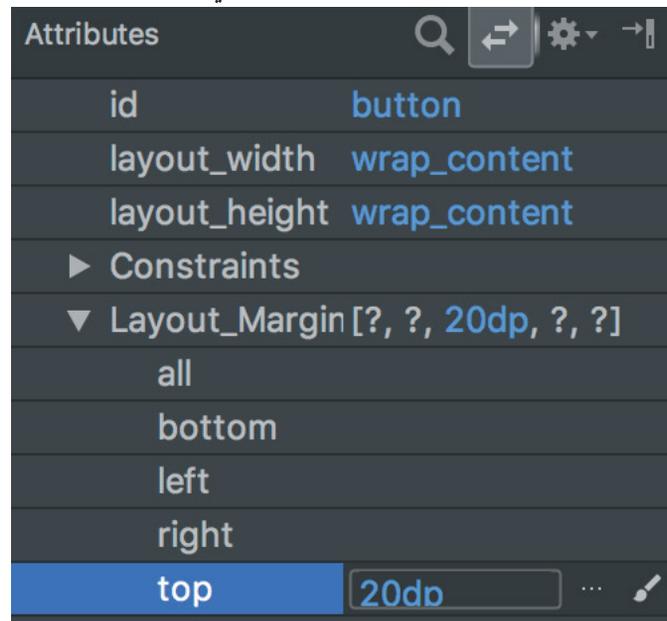
أغير بعض خصائص الصورة لكي تلائم جهاز الأندرويد، وأضيف id للعنصر، كما يلي:



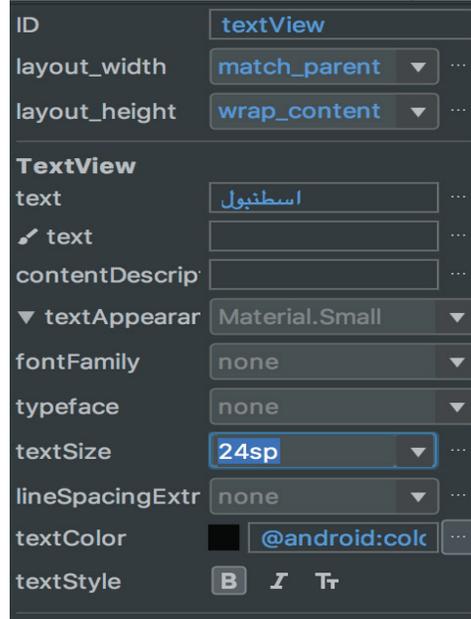
وبذلك تكون الصورة جاهزة، أنتقل للعناصر المتبقية، وأضيف `textView` و `button`، أضيف الخصائص اللازمة. عند إضافة `textView`، `button` ألاحظ أنّ النص غير منسق، أذهب إلى الخيارات، ومن خيار `gravity` أختار `center` لوضع النص في منتصف العنصر، كما يلي:



أيضاً من خيارات `Layout_Margin` من أعلى أبعد `20dp`، كما يلي:



الصورة التالية توضح بعض الخصائص التي اعطيته ل TextView :



أشغل المشروع والاحظ النتائج:



الأسئلة

- 1- كيف يتم اضافة صور في مجلد drawable؟
- 2- كيف يتم توسيط النص الموجود داخل العنصر؟



الموقف التعليمي التعلّمي (4 - 4):

تأهيل تطبيق أندرويد (١)

وصف الموقف التعليمي: زار معلّم رياضة مشغلاً لتطبيقات الهاتف المحمول في مدرسة يافا المهنيّة، لمساعدته في إنشاء تطبيق أندرويد بسيط لحساب الوزن المثاليّ.

العمل الكامل			
الموارد	المنهجية (استراتيجية التعلم)	وصف الموقف الصّفيّ	خطوات العمل
<ul style="list-style-type: none">• وثائق (طلب معلّم الرياضة، نشرات، كتب ومصادر متنوعة).• التكنولوجيا (روابط إلكترونية متنوعة من مصادر موثوقة، حاسوب، إنترنت، هاتف يعمل بنظام الأندرويد).	<ul style="list-style-type: none">• حوار ومناقشة.• التعلم التعاوني/ مجموعات• البحث العلميّ.	<p>أجمع البيانات من معلّم الرياضة عن:</p> <ul style="list-style-type: none">• قانون حساب الوزن المثاليّ.• المعطيات لحساب الوزن.• إصدار نظام الأندرويد المثبت على جهازه ونوعه. <p>أجمع البيانات عن:</p> <ul style="list-style-type: none">• العناصر (views) المستخدمة للإدخال والإخراج.- نوع المتغيّرات المستخدمة.- الجمل الشرطيّة المستخدمه.	<p>أجمع البيانات وأحلّها</p>
<ul style="list-style-type: none">• وثائق (نشرات، كتب ومصادر متنوعة)• الشبكة العنكبوتيّة.	<ul style="list-style-type: none">• تعلم تعاوني/مجموعات.• حوار ومناقشة .	<ul style="list-style-type: none">• أصنّف البيانات (الطول، الوزن، نظام الأندرويد المستخدم).• أحدّد المعطيات (الطول، الوزن).• أحدّد صيغة قانون حساب الوزن المثاليّ.• أحدّد المتغيّرات.• أحدّد الجمل الشرطيّة المناسبة.• أحدّد واجهات التصميم المستخدمة.• إعداد جدول زمنيّ للتنفيذ.	<p>أخطّط وأقرّر</p>



<ul style="list-style-type: none"> • حاسوب . • برنامج Android Studio . • جهاز افتراضي أو جهاز يعمل بنظام أندرويد . 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات حوار ومناقشة . • عصف ذهني/استمطار الأفكار . 	<ul style="list-style-type: none"> • أتفق مع المعلم على اختيار واجهة التطبيق المناسبة . • أصمّم واجهة التطبيق . • أكتب الكود البرمجي لحساب الوزن المثالي، وربطه مع عناصر الواجهة . • أعمل اختباراً للمشروع على جهاز افتراضي، أو على جهاز يعمل بنظام أندرويد . 	<p>أنفذ</p>
<ul style="list-style-type: none"> • الخطة المعدة سابقاً • الطلب الخاص بمعلم الرياضة . • حاسوب . • برنامج Android Studio . • جهاز افتراضي أو جهاز يعمل بنظام أندرويد 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات حوار ومناقشة . 	<ul style="list-style-type: none"> • أتحمّق من اختيار واجهة التطبيق الملائمة . • أتأكّد من ملاءمة تصميم واجهة التطبيق . • أتحمّق من تنفيذ التطبيق وظهوره على الجهاز بشكل صحيح . • أختبر التطبيق بعد إدخال المعطيات والضغط على زر التنفيذ، لحساب الوزن المثالي . • أتأكّد من عدم ظهور أيّ خطأ برمجيّ . • أتحمّق من ظهور نتائج صحيحة . 	<p>أتحمّق</p>
<ul style="list-style-type: none"> • حاسوب • شاشة عرض • جهاز افتراضي او جهاز يعمل بنظام أندرويد . 	<ul style="list-style-type: none"> • حوار ومناقشة . • تعلم تعاوني/ مجموعات عمل . 	<ul style="list-style-type: none"> • أوثّق البيانات التي تتعلّق بـ: • العناصر (views) المستخدمة للإدخال والإخراج . - نوع المتغيّرات المستخدمة . - الجمل الشرطية المستخدمة . • أقدم عرضاً تقديمياً . • أقدم التطبيق المطلوب للعروسين . • فتح ملف بالحالة (إنشاء تطبيق أندرويد بسيط لحساب الوزن المثالي) . 	<p>أوثّق وأقدم</p>
<ul style="list-style-type: none"> • نماذج للتقويم . • معدّة سابقاً • طلب المعلم . 	<ul style="list-style-type: none"> • حوار ومناقشة . 	<ul style="list-style-type: none"> • مدى فعالية التطبيق في حساب الوزن المثاليّ . • رضا المعلم عن التطبيق . • ملاءمة التطبيق للمعايير والمواصفات . 	<p>أقوم</p>

أناقش: أستطيع أن أعدّل على الكود البرمجيّ من خلال استخدام التطبيق.

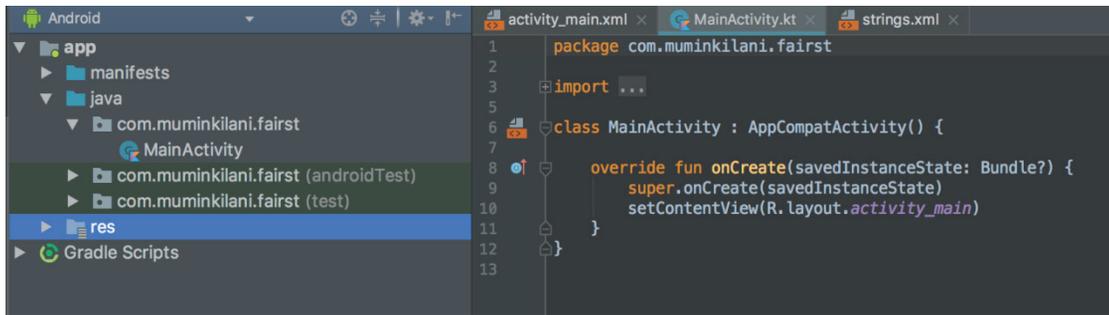


تأهيل تطبيق أندرويد (١)

الكود البرمجي:

هو الأوامر والتعليمات التي يكتبها مطورو البرامج بإحدى لغات البرمجة، وظيفتها في الأندرويد التحكم في واجهات التطبيق، وبجميع العمليات المطلوب إجراؤها، ولا يستطيع المُستخدم إجراء التعديلات عليها من خلال استخدامه للتطبيق .

لكي أتمكن من الوصول إلى الكود البرمجيّ أذهبُ إلى مجلد جافا ثم إلى ملف الكود الرئيسي، كما في الشكل الآتي:



```

1 package com.muminkilani.fairst
2
3 import ...
4
5
6 class MainActivity : AppCompatActivity() {
7
8     override fun onCreate(savedInstanceState: Bundle?) {
9         super.onCreate(savedInstanceState)
10        setContentView(R.layout.activity_main)
11    }
12
13 }

```

يوضح الشكل أعلاه الكود البرمجيّ للمشروع، يتكوّن الكود من أمور عدّة:

- اسم الحزمة للتطبيق ككل، وهي أوّل سطر في الكود.
- قائمة import، تستخدم لاستيراد المكتبات الضرورية لكتابة الأكواد.
- السطر السادس اسم الكلاس ويرث كلاس AppCompatActivity() وهو معدّ مسبقاً من قبل الشركة.
- عند وراثة كلاس AppCompatActivity() يجب استدعاء الدالّة onCreate (..)، هذه الدالّة تعني أنّها أوّل دالّة يتمّ استدعاؤها عند بدء التطبيق على الهاتف.
- هذه جملة ربط ملف واجهة التصميم بالكود البرمجيّ. setContentView(R.layout.activity_main)





- أكمل النشاط (5) الموجود ضمن الموقف التعليمي الثاني، وأكتب الأكواد اللازمة لعمل التطبيق بشكل صحيح.
- أبحث عن العمليات اللازمة، ومعرفة وقت تنفيذ هذه العمليات، أجد أن هذه العمليات جميعها تُنفذ عند الضغط على عنصر الحساب؛ لذلك تكون العمليات داخل هذا العنصر، كما هو موضح في الشكل الاتي:

```

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    Calculate.setOnClickListener { it: View!

        val l= Length.text.toString().toDouble()
        val w=weight.text.toString().toDouble()

        val c = w/(l*l)

        result.text= c.toString()

        when {
            c<18.5 -> resolution.text="نحيف"
            c in 18.5..24.9 -> resolution.text="وزن مثالي"
            c in 25.0..29.5 -> resolution.text="زيادة في الوزن"
            c >= 30 -> resolution.text="سمين"
            else -> resolution.text="تأكد من المدخلات"
        }
    }
}
    }
}

```

result : id الخاص بعنصر النتيجة.
 resolution : id الخاص بعنصر القرار.

Calculate : id الخاص بعنصر أحسب.
 Length : id الخاص بعنصر الطول.
 weight : id الخاص بعنصر الوزن.



```
Calculate.setOnClickListener{
```

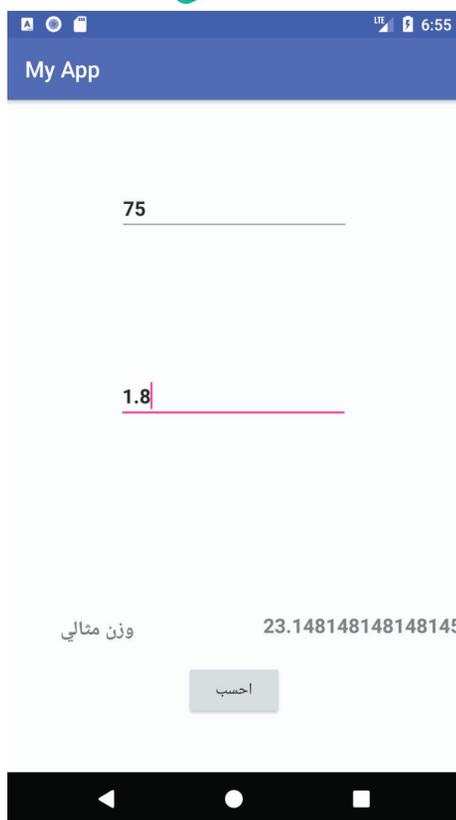
- كتابه الأوامر التي يتم تنفيذها عند الضغط على هذا العنصر

```
}
```

- `w.text.toString()`

تعني هذه العبارة إمّا جلب بيانات، أو إدخال بيانات وفق استخدامها في الكود، وعند إدخال البيانات أو إخراجها يجب أن تكون سلسله نصيّة بغضّ النظر عن نوع البيانات؛ لهذا السبب أستخدم عبارة `.toString()`.

أعمل تشغيل للمشروع، ثم اختبر التطبيق وافحص النتائج:



الأسئلة

1- ما فائدة `toString()`؟

2- ما هي الجملة البرمجية الخاصة بربط الكود البرمجي بملف واجهة التصميم؟

الموقف التعليمي التعلّمي (4 - 5):

تأهيل تطبيق أندرويد (٢)

وصف الموقف التعليمي: توجّه عروسان إلى مشغلٍ لتطبيقات الهاتف المحمول في مدرسة يافا المهنية، لمساعدتهما في إنشاء تطبيق أندرويد بسيط لعرض صور حفل الزفاف الخاصّ بهم.

العمل الكامل			
الموارد	المنهجية (استراتيجية التعلم)	وصف الموقف الصّفي	خطوات العمل
<ul style="list-style-type: none"> • وثائق (طلب العروسين، نشرات، كتب ومصادر متنوعة). • التكنولوجيا (روابط إلكترونية متنوعة من مصادر موثوقة، حاسوب، إنترنت، هاتف يعمل بنظام الأندرويد). 	<ul style="list-style-type: none"> • حوار ومناقشة. • التعلم التعاوني/ مجموعات • البحث العلمي. 	<p>أجمع البيانات من العروسين عن:</p> <ul style="list-style-type: none"> • صور حفل الزفاف. • إصدار نظام الأندرويد المثبت على جهازه ونوعه. <p>أجمع البيانات عن:</p> <ul style="list-style-type: none"> • العناصر (views) المستخدمة للإدخال والإخراج. • نوع المتغيّرات المستخدمة. • الجمل الشرطية المستخدمة. 	أجمع البيانات وأحلّها
<ul style="list-style-type: none"> • وثائق (نشرات، كتب ومصادر متنوعة) • الشبكة العنكبوتية. 	<ul style="list-style-type: none"> • تعلّم تعاوني/مجموعات • حوار ومناقشة. 	<ul style="list-style-type: none"> • أصنّف البيانات (صور حفل الزفاف ، نظام الأندرويد المستخدم). • أحدّد المعطيات (صور حفل الزفاف). • أحدّد المتغيرات. • أحدّد الجمل الشرطية المناسبة. • أحدّد واجهات التصميم المستخدمة. • إعداد جدول زمنيّ للتنفيذ. 	أخطّط وأقرّر
<ul style="list-style-type: none"> • حاسوب. • برنامج Android Studio. • جهاز افتراضي أو جهاز يعمل بنظام أندرويد. 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات • حوار ومناقشة. • عصف ذهنيّ/استمطار الأفكار. 	<ul style="list-style-type: none"> • أتفق مع العروسين على اختيار واجهة التطبيق المناسبة. • أصمّم واجهة التطبيق. • أكتب الكود البرمجيّ الخاص بمعرض الصور، وربطه مع عناصر الواجهة. • أعمل اختباراً للمشروع على جهاز افتراضي، أو على جهاز يعمل بنظام أندرويد. 	أنفّذ



<ul style="list-style-type: none"> • الخطة المعدّة سابقاً. • الطلب الخاص بالعروسين. • حاسوب. • برنامج Android Studio. • جهاز افتراضي أو جهاز يعمل بنظام أندرويد. 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات • حوار ومناقشة. 	<ul style="list-style-type: none"> • أتحدّق من اختيار واجهة التطبيق الملائمة. • أتأكّد من ملاءمة تصميم واجهة التطبيق. • أتحدّق من تنفيذ التطبيق وظهوره على الجهاز بشكل صحيح. • أختبر التطبيق عند الضغط على أزرار التنقل بين الصور. • أتأكّد من عدم ظهور أيّ خطأ برمجيّ. • أتحدّق من ظهور نتائج صحيحة. 	<p>أضبط</p>
<ul style="list-style-type: none"> • حاسوب • شاشة عرض • جهاز افتراضي أو جهاز يعمل بنظام أندرويد. 	<ul style="list-style-type: none"> • حوار ومناقشة. • تعلم تعاوني/مجموعات عمل. 	<p>أوثّق البيانات التي تتعلّق بـ:</p> <ul style="list-style-type: none"> • العناصر (views) المُستخدمة للإدخال والإخراج. • نوع المتغيّرات المستخدمة. • الجمل الشرطيّة المستخدمة. • أقدم عرضاً تقديمياً. • أقدم التطبيق المطلوب للمعلّم. • فتح ملف بالحالة (إنشاء تطبيق أندرويد بسيط لعرض صور حفل الزفاف). 	<p>أوثّق وأقدّم</p>
<ul style="list-style-type: none"> • نماذج للتقويم معدّة سابقاً • طلب العروسين. 	<ul style="list-style-type: none"> • حوار ومناقشة. 	<ul style="list-style-type: none"> • مدى فعاليّة التطبيق في عرض الصور. • رضا العروسين عن التطبيق. • ملاءمة التطبيق للمعايير والمواصفات. 	<p>أقوم</p>



◀ **أناقش:** من الأفضل أن يكون id العنصر يعبّر عن عمل العنصر.



تأهيل تطبيق أندرويد (٢)

- أكمل النشاط الموجود ضمن الموقف التعليمي الثالث، وأكتب الأكواد اللازمة لعمل التطبيق بشكل صحيح.
- أبحث عن العمليات اللازمة، ومعرفة وقت تنفيذ هذه العمليات، أجد أن هذه العمليات جميعها تُنفَّذ عند الضغط على عنصر تبديل الصورة؛ لذلك تكون العمليات داخل هذا العنصر.

```
val image = arrayOf(R.drawable.istanbul,R.drawable.antalia)
```

يفيد السطر السابق عمل مصفوفة من الصور المستخدمة، حتى يتم التبديل بينهما، لاستدعاء الصور من الملف الخاص فيها اكتب R التي تشير إلى ملف resources ثم نقطه حتى أتمكن من الوصول لما في داخل resources، وهنا أريد الوصول إلى ملف drawable ثم إلى الصورة.

```
val text = arrayOf("انطاليا","اسطنبول")
```

يفيد السطر السابق عمل مصفوفة بأسماء الصور بترتيب الصور نفسها في مصفوفة الصور.

```
class MainActivity : AppCompatActivity() {
    val image = arrayOf(R.drawable.istanbul,R.drawable.antalia)

    val text = arrayOf("اسطنبول","انطاليا")

    var cout = 0

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }

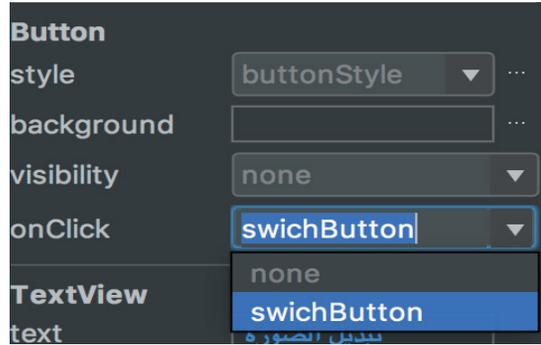
    fun swichButton() {
        if (cout==0){
            imageView.setImageResource(image[1])
            textView.text=text[1]
            cout=1
        }else{
            imageView.setImageResource(image[0])
            textView.text=text[0]
            cout=0
        }
    }
}
```

في الشكل السابق :

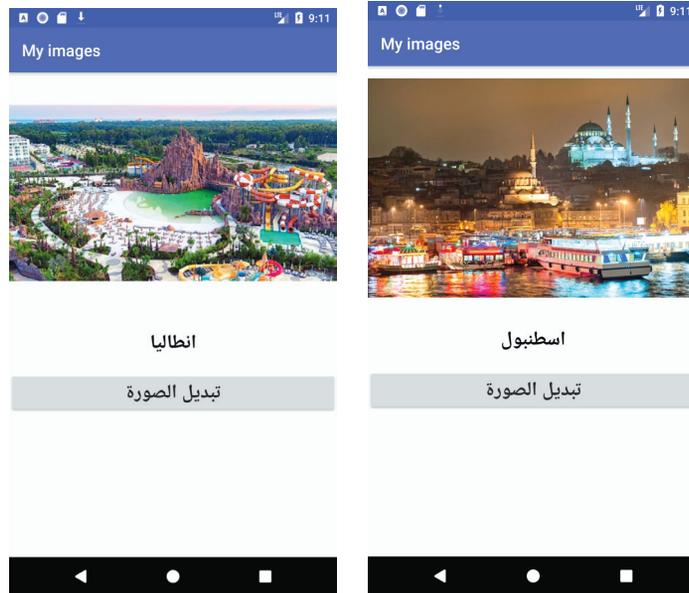
1. عرفت coutn من نوع Int يعمل كعداد حتى يتم التبديل بين الصور.
2. استخدمت طريقة الدالة (function) لعملية النقر على الزر، عند كتابة الدالة يجب أن توضع خارج onCreate.



3. لتعيين صورته من داخل الكود استخدمت `setImageResource`.
4. لربط الاقتران بالزر (button) أنتقل إلى تصميم الواجهة وأظلل الزر (button) ومن قائمة الخصائص عند كلمة `onClick` أختار الدالة الذي أنشأتها كما يأتي:



أعمل تشغيل للمشروع، ثم اختبر التطبيق وافحص النتائج:



الأسئلة

- 1- ما هو الكود المسؤول عن تغيير الصورة في `imageView` من الكود البرمجي؟
- 2- كيف يتم الوصول الى صورة موجودة في مجلد `drawable` من الكود البرمجي؟



الموقف التعليمي التعلّمي (4 - 6):

تأهيل تطبيق أندرويد (3)

وصف الموقف التعليمي: ذهبت معلّمة رياض الأطفال إلى مشغل لتطبيقات الهاتف المحمول في مدرسة يافا المهنية، لمساعدتها في إنشاء تطبيق أندرويد بسيط لتعليم الأطفال على الألوان.

العمل الكامل			
الموارد	المنهجية (استراتيجية التعلم)	وصف الموقف الصفي	خطوات العمل
<ul style="list-style-type: none"> • وثائق (طلب المعلمة ، نشرات، كتب ومصادر متنوعة). • التكنولوجيا(روابط إلكترونية متنوعة من مصادر موثوقة، حاسوب، إنترنت، هاتف يعمل بنظام الأندرويد) 	<ul style="list-style-type: none"> • حوار ومناقشة. • التعلّم التعاوني/ مجموعات. • البحث العلمي. 	<p>أجمع البيانات من المعلمة عن:</p> <ul style="list-style-type: none"> • مجموعة الألوان التي تريد تعليمها للاطفال. • إصدار نظام الأندرويد المثبت على جهازه ونوعه. <p>أجمع البيانات عن :</p> <ul style="list-style-type: none"> • العناصر (views) المستخدمة للإدخال والإخراج. • نوع المتغيّرات المستخدمة. 	<p>أجمع البيانات وأحلّها</p>
<ul style="list-style-type: none"> • وثائق (نشرات، كتب ومصادر متنوعة) • الشبكة العنكبوتية. 	<ul style="list-style-type: none"> • تعلّم تعاوني/مجموعات حوار ومناقشة. 	<ul style="list-style-type: none"> • أصنّف البيانات(مجموعة الألوان، نظام الأندرويد المستخدم). • أحدّد المعطيات(مجموعة الألوان). • أحدّد المتغيّرات. • أحدّد واجهات التصميم المستخدمة. • إعداد جدول زمنيّ للتنفيذ. 	<p>أخطّط وأقرّر</p>
<ul style="list-style-type: none"> • حاسوب. • برنامج Android Studio. • جهاز افتراضي أو جهاز يعمل بنظام أندرويد. 	<ul style="list-style-type: none"> • تعلّم تعاوني/مجموعات حوار ومناقشة. • عصف ذهنيّ/استمطار الأفكار. 	<ul style="list-style-type: none"> • أتفق مع المعلمة على اختيار واجهة التطبيق المناسبة. • أصمّم واجهة التطبيق. • أكتب الكود البرمجيّ الخاص بعرض الألوان وربطه مع عناصر الواجهة. • أعمل اختباراً للمشروع على جهاز افتراضي، أو على جهاز يعمل بنظام أندرويد. 	<p>أنفّذ</p>



<ul style="list-style-type: none"> • الخطة المعدة سابقاً. • الطلب الخاص بالمعلمة. • حاسوب. • برنامج Android Studio. • جهاز افتراضي أو جهاز يعمل بنظام أندرويد 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات • حوار ومناقشة. 	<ul style="list-style-type: none"> • أتحدّق من اختيار واجهة التطبيق الملائمة. • أتأكّد من ملاءمة تصميم واجهة التطبيق. • أتحدّق من تنفيذ التطبيق وظهوره على الجهاز بشكل صحيح. • أختبر التطبيق عند الضغط على أزرار الألوان يعطي نتائج صحيحة. • أتأكّد من عدم ظهور أي خطأ برمجيّ. • أتحدّق من ظهور نتائج صحيحة. 	<p>أتحدّق</p>
<ul style="list-style-type: none"> • حاسوب. • شاشة عرض • جهاز افتراضي أو جهاز يعمل بنظام أندرويد. 	<ul style="list-style-type: none"> • حوار ومناقشة. • تعلم تعاوني/مجموعات عمل. 	<p>أوثق البيانات التي تتعلّق بـ:</p> <ul style="list-style-type: none"> • العناصر (views) المستخدمة للإدخال والإخراج. • نوع المتغيّرات المستخدمة. • أقدم عرضاً تقديمياً. • أقدم التطبيق المطلوب للمعلمة. • فتح ملف بالحالة (إنشاء تطبيق أندرويد بسيط لتعليم الأطفال على الألوان الأساسيّة). 	<p>أوثق وأقدم</p>
<ul style="list-style-type: none"> • نماذج للتقويم معدة سابقاً. • طلب المعلمة. 	<ul style="list-style-type: none"> • حوار ومناقشة. 	<ul style="list-style-type: none"> • مدى فعالية التطبيق في عرض الألوان. • رضا المعلمة عن التطبيق. • ملاءمة التطبيق للمعايير والمواصفات. 	<p>أقوم</p>

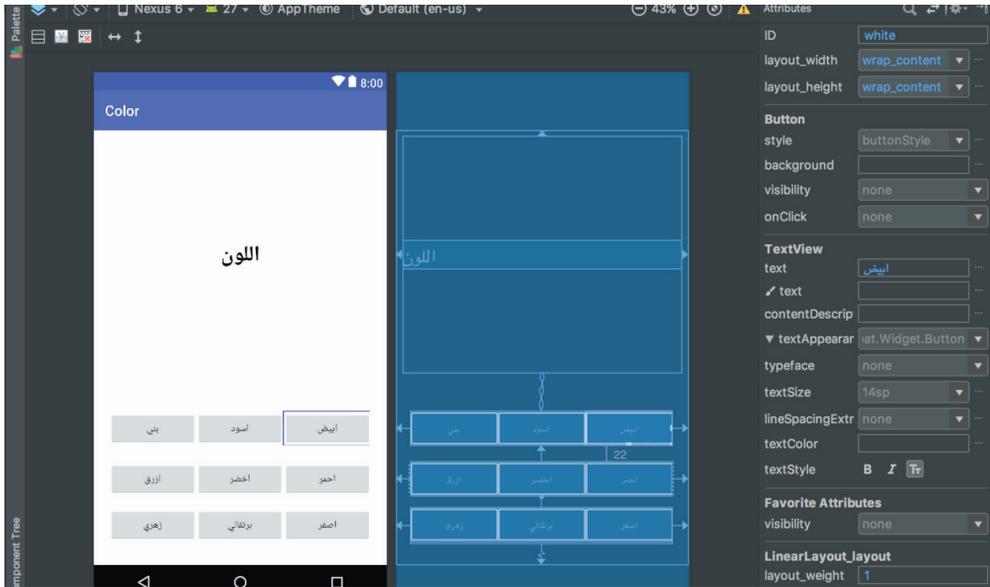


◀ **أناقش: أستطيع وضع viewgroup داخل viewgroup.**



تأهيل تطبيق أندرويد (٣)

- أعمل تطبيقاً لعرض الألوان من خلال الضغط على اسم اللون. أحضر المعطيات: الألوان الأساسية وعددها وأسماءها. ثم أفتح برنامج الأندرويد ستوديو وأنشئ مشروعاً جديداً وأسميه Color، بعد أن يفتح البرنامج بشكل كامل، استناداً إلى المعطيات أختار العناصر التي أريد استخدامها، وهي كما يأتي:
- أربعة عناصر من نوع `LinearLayout`: لتغيير الألوان، وإضافة الأزرار في داخلها.
 - عنصر من نوع `text view`: لإظهار اسم اللون.
 - تسعة عناصر من نوع `button`: عند الضغط عليه يُنفذ العمليات البرمجية الخاصة بتغيير اللون، وفق اسم الزر.
- أبدأ بالعمل:** أصمم الواجهة المناسبة لعمل التطبيق، أضيف أولاً `LinearLayout`، وإضافة ٣ عناصر من نوع `button` في كل `horizontal LinearLayout`، ومن خصائص `LinearLayout` أجعل الارتفاع ملائماً لما في داخله من عناصر كما تعلمت سابقاً، وأضيف `LinearLayout` لكي تعرض الألوان في داخلها، وأضيف أيضاً داخلها `textview` لعرض اسم اللون، كما يلي:



أعطي كل عنصر id يدل على عمله، وأعطي `LinearLayout` التي أريد عرض الألوان عليها id حتى أستطيع استدعاؤها في الكود البرمجي.



أنتقل إلى الكود البرمجي:

أبحث عن العمليات اللازمة، ومعرفة وقت تنفيذ هذه العمليات، أجد أنّ هذه العمليات جميعها تُنفذ عند الضغط على العناصر التي تحمل أسماء الألوان؛ لذلك تكون العمليات داخل هذه العناصر عند الضغط عليها كالآتي:

```
white.setOnClickListener{
```

```
color.setBackgroundColor(Color.WHITE) // تغيير اللون
```

```
colorName.text= white.text.toString()// textView في button لعرض النص الموجود على
```

```
}
```

هذه الأوامر في زر اللون الأبيض، ألاحظ استخدام id الخاص بـ `LinearLayout` لإعطائها لوناً للخلفيّة، ومن المتوقع أنّ تتغيّر صيغته الكود مستقبلاً، وتصبح كما يأتي:

```
color.backgroundColor = Color.WHITE
```

`Color` هو صنف (class) جاهز يحتوي على الألوان.

أكمل الأوامر لباقي أزرار الألوان بالطريقة نفسها، وينتج الكود الآتي:

```
black.setOnClickListener { it: View!
    color.setBackgroundColor(Color.BLACK)

    colorName.text = black.text.toString()
}
brown.setOnClickListener { it: View!
    color.setBackgroundColor(Color.parseColor( colorString: "#5c4033"))

    colorName.text = brown.text.toString()
}
red.setOnClickListener { it: View!
    color.setBackgroundColor(Color.RED)

    colorName.text = red.text.toString()
}
green.setOnClickListener { it: View!
    color.setBackgroundColor(Color.GREEN)

    colorName.text = green.text.toString()
}
blue.setOnClickListener { it: View!
    color.setBackgroundColor(Color.BLUE)

    colorName.text = blue.text.toString()
}
yellow.setOnClickListener { it: View!
    color.setBackgroundColor(Color.YELLOW)

    colorName.text = yellow.text.toString()
}
orange.setOnClickListener { it: View!
    color.setBackgroundColor(Color.parseColor( colorString: "#ff6600"))

    colorName.text = orange.text.toString()
}
pink.setOnClickListener { it: View!
    color.setBackgroundColor(Color.parseColor( colorString: "#ff6ec7"))

    colorName.text = pink.text.toString()
}
```



ألاحظ أنّ مكتبة الألوان لا تحتوي على جميع الألوان، ولإضافة لونٍ خارجيٍّ أجلب رقم اللون، أو ما يُسمى كود لون html، وأستخدم الجملة الآتية:

```
color.setBackgroundColor(Color.parseColor("#5c4033"))
```

وهكذا أكون قد أنهيت الكود البرمجيّ، وبقيّ اختبار التطبيق، أعمل على تشغيل المشروع لإظهاره على الجهاز الافتراضيّ، وأتحقّق من النتائج.



ملحوظة:

أستطيع جلب كود اللون من الموقع الآتي : <https://html-color-codes.info>.

الأسئلة



- 1- ما الكود البرمجي المسؤول عن تغيير الخلفية؟
- 2- ما الأمر البرمجي المسؤول عن اضافة نص على `textView`؟





أسئلة الوحدة

السؤال الأول:

اختر رمز الإجابة الصحيحة فيما يأتي :

- 1- ما اسم البرنامج المستخدم في برمجة تطبيقات الأندرويد بلغة كوتلن؟
١. visual studio . ٢. Android studio . ٣. Net beans . ٤. photo shop .
- 2- ما المجلد الذي يحتوي على الملفات الخاص في تصميم الواجهات؟
١. res . ٢. java . ٣. manifests . ٤. libs .
- 3- أي من الآتي يُعدُّ وحدة قياس الخطّ المعتمدة في الأندرويد ستوديو؟
١. px . ٢. dp . ٣. cm . ٤. sp .
- 4- ما الخاصية التي يتم ربط العنصر (view) مع الكود البرمجي بواسطتها؟
١. id . ٢. text . ٣. hint . ٤. manifest .
- 5- ما المجلد الذي يحتوي الصور الخاصة بالمشروع؟
١- drawable - ٢- values - ٣- Java - ٤- manifest -

السؤال الثاني:

وضّح الفرق بين view و viewgroup .

السؤال الثالث:

عرّف المصطلحات الآتية: xml ، sdk ، الكود البرمجيّ .

السؤال الرابع:

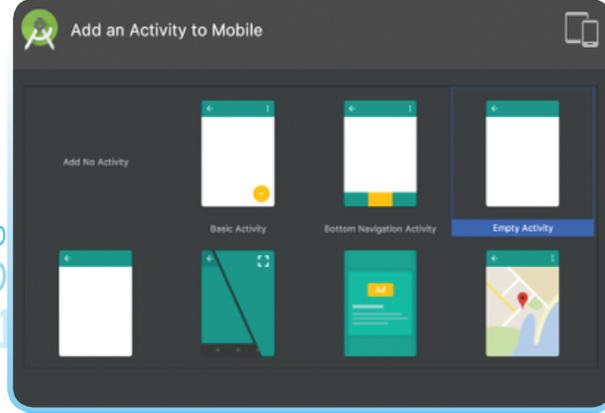
وضّح طريقة إعداد محاكي الأندرويد .

مشروع الوحدة



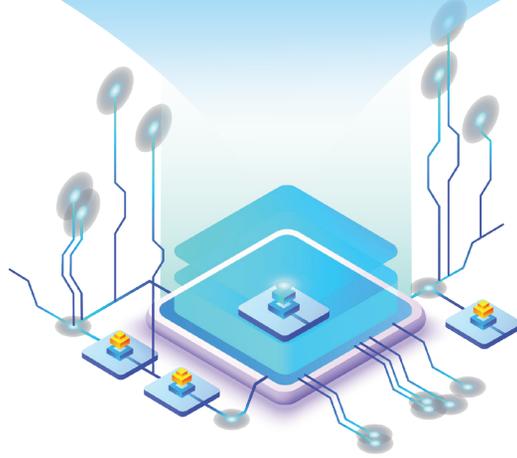
عمل تطبيق بسيط لحساب العمر .

برمجة الواجهات Activity



10011101
011101
01011011100010

10010110
1001011011100
1001011



أناقش:

تُعدُّ Activity الواجهة
التي تظهر للمستخدم



يُتَوَقَّعُ من الطلبة بعد دراسة هذه الوحدة والتفاعل مع انشطتها ان يكونوا قادرين على برمجة الواجهات **Activities** من خلال تحقيق الاتي:

- ✓ التعرف على دورة حياة **Activity**.
- ✓ معرفة الايماءات **Gestures**.
- ✓ انشاء القوائم **Menus** والتعامل معها.
- ✓ انشاء الواجهات (**Activities**) والتنقل بينها باستخدام **Intent**.
- ✓ نقل بيانات بين الواجهات **Activity - Intent**.

الكفايات المهنية المُتوقَّع من الطلبة امتلاكها بعد الانتهاء من دراسة هذه الوحدة، والتفاعل مع أنشطتها:

- الاستعداد التام لتقبُّل ملحوظات الزبون وانتقاداته.
- القدرة على الإقناع.

أولاً: الكفايات الحرفية.

- امتلاك مهارة التعامل مع activity والإيماءات.
- التعامل مع القوائم وإنشائها.
- التعامل مع الأخطاء.
- إنشاء أكثر من واجهة activity والتنقل بينها.
- نقل بيانات بين الواجهات activities.
- التخاطب بين التطبيقات واستخدام الكاميرا.
- تقييم النتائج.

ثالثاً: الكفايات المنهجية.

- توظيف التعلُّم التعاوني.
- امتلاك مهارة البحث العلمي، والقدرة على توظيف أساليبه.
- القدرة على التخطيط.
- القدرة على استمطار الأفكار.
- استخدام المصادر والمراجع المختلفة.
- توثيق نتائج العمل وعرضها.
- تقييم النتائج وعرضها.
- توظيف التكنولوجيا والبرامج الإحصائية وتمثيل البيانات.
- جمع البيانات وتحليلها وتبويبها.

ثانياً: الكفايات الاجتماعية والشخصية.

- العمل ضمن فريق.
- تقبُّل التغذية الراجعة.
- احترام رأي الآخرين.
- القدرة على إدارة الحوار وتنظيم النقاش.
- مصداقية التعامل مع الزبون.
- حفظ خصوصية الزبون.
- الاستعداد للاستجابة لطلبات الزبون.
- التحلي بأخلاقيات المهنة أثناء العمل.
- التمتع بالفكر الريادي أثناء العمل.
- استخلاص النتائج ودقة الملاحظة.
- الاتصال والتواصل الفعال.
- الاستعداد للاستفسار والاستفادة من ذوي الخبرة.
- الاستفادة من مقترحات الآخرين.
- امتلاك مهارة التأمل الذاتي.

قواعد الأمن والسلامة:

- الالتزام بالجلسة الصحية أمام الحاسوب.
- إراحة العينين كل ربع ساعة.
- الجلوس في غرفة فيها إضاءة مناسبة وتهوية جيّدة.
- الابتعاد عن الحاسوب كل ساعة، والقيام ببعض التمارين الرياضية.



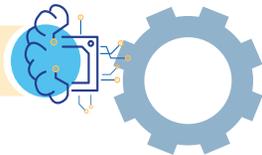
التعرّف إلى دورة حياة الواجهة activity.

وصف الموقف التعليمي: زار سعد مشغلاً لتطبيقات الهاتف المحمول في مدرسة سعد صايل المهنيّة؛ لمساعدته في إنشاء تطبيق أندرويد للتعرف على دورة حياة الواجهة activity.

العمل الكامل			
خطوات العمل	وصف الموقف الصّفي	المنهجية (استراتيجية التّعلم)	الموارد
أجمع البيانات وأحلّها	<ul style="list-style-type: none"> • أجمع البيانات من سعد عن: - إصدار نظام الأندرويد المثبت على جهازه ونوعه. - ما يريد معرفته عن activity. • أجمع البيانات عن: - العناصر (views) المستخدمة للإدخال والإخراج. - الجمل البرمجية الخاصة بدورة حياة activity. 	<ul style="list-style-type: none"> • حوار ومناقشة. • التعلم التعاوني/ مجموعات. • البحث العلمي. 	<ul style="list-style-type: none"> • وثائق (طلب سعد، نشرات، كتب ومصادر متنوعة). • التكنولوجيا (روابط إلكترونية متنوعة من مصادر موثوقة، حاسوب، إنترنت).
أخطط وأقّر	<ul style="list-style-type: none"> • أصنّف البيانات (نظام الأندرويد المستخدم onCreate ، onStart ، onResume ، onPause ، onStop ، onRestart ، onDestroy). • أحدّد ما يأتي: - المعطيات: (onCreate ، onStart ، onResume ، onPause ، onStop ، onRestart ، onDestroy). - واجهة التطبيق المستخدمة. - الجمل البرمجية المستخدمة في دورة حياة activity. - إعداد خطة للتنفيذ. 	<ul style="list-style-type: none"> • تعلّم تعاوني/ مجموعات. • حوار ومناقشة. 	<ul style="list-style-type: none"> • وثائق (نشرات، كتب ومصادر متنوعة). • التكنولوجيا (روابط إلكترونية متنوعة من مصادر موثوقة، عن كيفية تطوير تطبيق أندرويد).

<ul style="list-style-type: none"> • حاسوب. • برنامج Android Studio. • جهاز افتراضي، أو جهاز يعمل بنظام أندرويد. 	<ul style="list-style-type: none"> • تعلمٌ تعاونيٌّ/مجموعات. • حوار ومناقشة. • عصف ذهنيٌّ/استمطار الأفكار. 	<ul style="list-style-type: none"> • اتَّفَق مع سعد على الواجهة المستخدمة. • أصمَّم واجهة التطبيق. • أكتب الكود البرمجي. • أعمل اختباراً للمشروع على جهاز افتراضي، أو على جهاز يعمل بنظام أندرويد. 	<p>أُنقِذ</p>
<ul style="list-style-type: none"> • الخطة المُعدَّة سابقاً. • الطلب الخاص بسعد. • حاسوب. • برنامج Android Studio. • جهاز افتراضي، أو جهاز يعمل بنظام أندرويد. 	<ul style="list-style-type: none"> • تعلمٌ تعاونيٌّ/مجموعات. • حوار ومناقشة. 	<ul style="list-style-type: none"> • أتحقَّق من اختيار واجهة التطبيق الملائمة. • أتأكَّد من ملاءمة تصميم واجهة التطبيق. • أتحقَّق من تنفيذ التطبيق وظهوره على الجهاز بشكل صحيح. • أختبر التطبيق. • أتأكَّد من عدم ظهور أيِّ خطأ برمجيِّ. 	<p>أتحقَّق</p>
<ul style="list-style-type: none"> • حاسوب. • شاشة عرض • جهاز افتراضي، أو جهاز يعمل بنظام أندرويد. 	<ul style="list-style-type: none"> • حوار ومناقشة. • تعلمٌ تعاونيٌّ/مجموعات. 	<p>أوثِّق البيانات التي تتعلَّق بـ:</p> <ul style="list-style-type: none"> • العناصر (views) المستخدمة للإدخال والايخارج. • الجمل البرمجيَّة الخاصة بدورة حياة activity. • إعداد عرض تقديميِّ. • فتح ملف بالحالة (إنشاء تطبيق أندرويد للتعرفُف إلى activity). 	<p>أوثِّق وأقِّم</p>
<ul style="list-style-type: none"> • طلب سعد، وثائق نشرات، كتب ومصادر متنوعة، وروابط إلكترونية متنوعة من مصادر موثوقة، عن كيفية تطوير تطبيق أندرويد. • نشرات عن المعايير والمواصفات الخاصة بتطوير تطبيق أندرويد. 	<ul style="list-style-type: none"> • حوار ومناقشة. 	<ul style="list-style-type: none"> • مدى فعاليَّة التطبيق في إظهار دورة حياة activity. • رضا سعد عن التطبيق. • ملاءمة التطبيق للمعايير والمواصفات. 	<p>أقِّم</p>

أناقش: أهميَّة وجود activity في التطبيق.





أنعلّم: دورة حياة الواجهة activity

activity

هي الواجهة التي تظهر للمستخدم، وتتكوّن من كود برمجيّ، وملف واجهة التصميم (layout)، ويجب أن يحتوي التطبيق على activity واحدة على الأقلّ.

دورة حياة activity:

عند تشغيل أيّ تطبيق على الجهاز وظهور activity، فإنّ هذه الواجهة تمرّ بمراحل عدّة، وهي:

1. onCreate(): يتمّ استدعاؤها عند إنشاء Activity أول مرّة.

2. onStart(): يتمّ استدعاؤها قبل ظهور Activity على الشاشة.

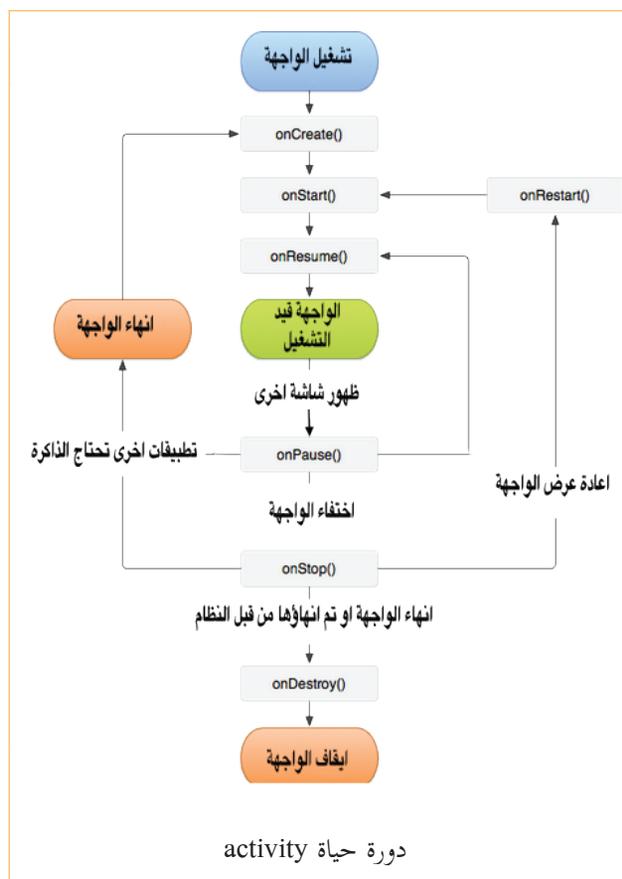
3. onResume(): يتمّ استدعاؤها عند ظهور الشاشة أمام المستخدم لأول مرّة، أو عند استعادة الواجهة بعد استدعاء (onPause).

4. onPause(): يتمّ استدعاؤها عندما تكون activity في حالة pause وتكون activity التي قبلها في حالة onResume، ويتمّ استدعاء هذه الدالة غالباً عندما أضغط على زر Back في جهاز الهاتف.

5. onStop(): يتمّ استدعاؤها عندما تكون Activity مخفية؛ أي غير ظاهرة للمستخدم، وتتمّ عندما أضغط على زر الهوم.

6. onRestart(): يتمّ استدعاؤها عندما تكون Activity في حالة stop، ثمّ تعود للظهور مرّة أخرى.

7. onDestroy(): يتمّ استدعاؤها عند تدمير Activity بواسطة النظام، وهنا تتمّ إزالة activity من الذاكرة، وتتمّ العمليّة عند إغلاق التطبيق بشكل نهائيّ.



عبارة عن تعليمات تظهر لبضع ثوانٍ ثم تختفي، عادة تظهر في أسفل الواجهة، وطريقة كتابتها في الكود كالتالي:

```
Toast.makeText(this,"النص المراد عرضه",Toast.LENGTH_LONG).show()
```

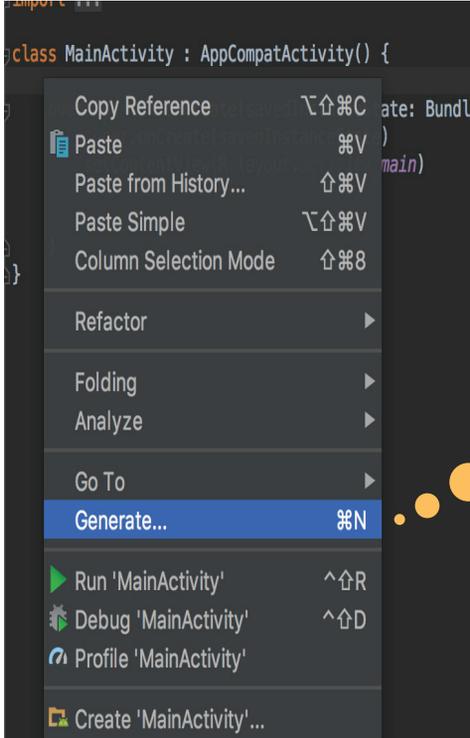
مكوّنات toast:

- كلمة this تشير إلى الصنف الذي نكتب بداخله.
- النص المراد عرضه ويعرض بين علامتي التنصيص " " .
- المدة الزمنية: هناك مدّتان نستخدمهما لعرض toast: مدّة طويلة تقريباً تصل إلى ٧ ثوانٍ، وتُكتب هكذا Toast.LENGTH_LONG، ومدّة قصيرة مدّتها تصل تقريباً نصف المدّة الطويلة، وتُكتب Toast.LENGTH_SHORT.
- .show(): تُستخدم ليتمّ عرض toast، وبدونها لا يتمّ عرضه.

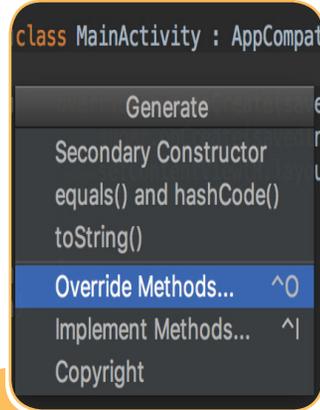
نشاط عملي: أنشئ تطبيقاً يقوم يوضح دورة حياة الواجهة activity، وإظهار الحالات في toast.

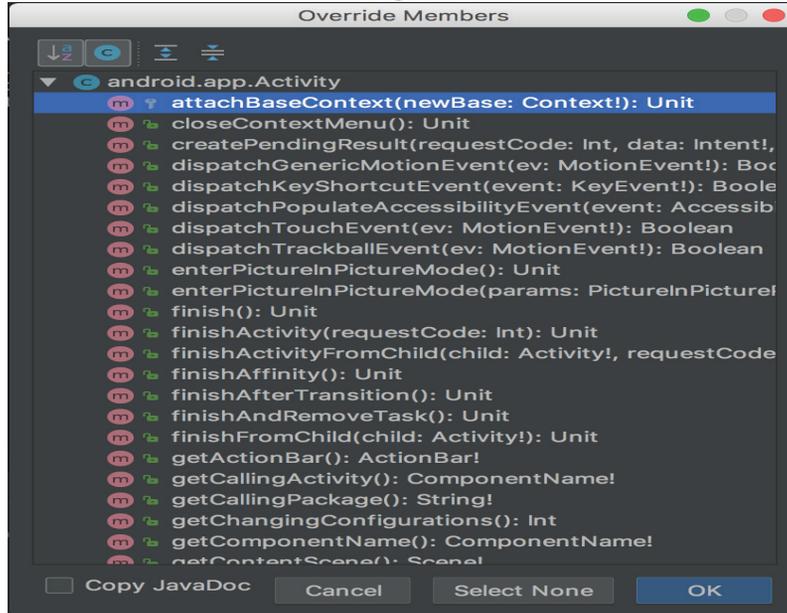
أفتح برنامج android studio ثم أنشئ مشروعاً جديداً، وعندما يفتح البرنامج بشكل كامل، أنتقل إلى الكود البرمجيّ:
أضغط برز الفأرة اليمين في منطقته خالية، وأكمل كما يأتي:

-1-



-2-





عند ظهور الشاشة رقم ٣ أبحث عن مراحل الواجهة activity عن طريق كتابة الاسم، وأختار ما أريد استخدامه، وأضيف باقي المراحل بالطريقة نفسها، ويصبح الكود كالآتي:

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
}

override fun onStart() {
    super.onStart()
}

override fun onStop() {
    super.onStop()
}

override fun onResume() {
    super.onResume()
}

override fun onRestart() {
    super.onRestart()
}

override fun onPause() {
    super.onPause()
}

override fun onDestroy() {
    super.onDestroy()
}
```

أحاول إدراج جميع المراحل مرة واحدة.



بعد إضافة الحالات أضيف toast، وأضع النصّ باسم الحالة كما يأتي:

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
    Toast.makeText( context: this, text: "onCreate", Toast.LENGTH_LONG).show()
}

override fun onStart() {
    Toast.makeText( context: this, text: "onStart", Toast.LENGTH_LONG).show()
    super.onStart()
}

override fun onStop() {
    Toast.makeText( context: this, text: "onStop", Toast.LENGTH_LONG).show()
    super.onStop()
}

override fun onResume() {
    Toast.makeText( context: this, text: "onResume", Toast.LENGTH_LONG).show()
    super.onResume()
}

override fun onRestart() {
    Toast.makeText( context: this, text: "onRestart", Toast.LENGTH_LONG).show()
    super.onRestart()
}

override fun onPause() {
    Toast.makeText( context: this, text: "onPause", Toast.LENGTH_LONG).show()
    super.onPause()
}

override fun onDestroy() {
    Toast.makeText( context: this, text: "onDestroy", Toast.LENGTH_LONG).show()
}
```

أشغل المشروع، وأختبر التطبيق، وأفحص النتائج:



1. أذكر مراحل دورة حياة activity، مع شرح موجز.
2. ما طريقة كتابة toast؟



التعرّف إلى الإيماءات Gestures.

وصف الموقف التعليمي: زار جاسر مشغلاً لتطبيقات الهاتف المحمول في مدرسة غزّة المهنيّة؛ لمساعدته في إنشاء تطبيق أندرويد للتعرف إلى الإيماءات Gestures.

العمل الكامل			
الموارد	المنهجية (استراتيجية التعلّم)	وصف الموقف الصّفيّ	خطوات العمل
<ul style="list-style-type: none"> وثائق (طلب جاسر، نشرات، كتب ومصادر متنوعة). التكنولوجيا (روابط إلكترونية متنوعة من مصادر موثوقة، حاسوب، إنترنت). 	<ul style="list-style-type: none"> حوار ومناقشة. التعلم التعاوني/ مجموعات. البحث العلميّ. 	<ul style="list-style-type: none"> أجمع البيانات من محمود عن: - إصدار نظام الأندرويد المثبت على جهازه ونوعه. - ما يريد معرفته عن الإيماءات. أجمع البيانات عن: - العناصر (views) المستخدمة للإدخال والإخراج. - الجمل البرمجية الخاصّة بالإيماءات. 	أجمع البيانات وأحلّها
<ul style="list-style-type: none"> وثائق (نشرات، كتب ومصادر متنوعة). التكنولوجيا (روابط إلكترونية متنوعة من مصادر موثوقة). 	<ul style="list-style-type: none"> تعلم تعاوني/ مجموعات. حوار ومناقشة. 	<ul style="list-style-type: none"> أصنّف البيانات (نظام الأندرويد المستخدم، onDown، onTouchEvent، onFling ..). أحدّد: - المعطيات: (..onTouchEvent، OnDown). - واجهة التطبيق المستخدمة. - الجمل البرمجية المستخدمة في الإيماءات. إعداد خطة للتنفيذ. 	أخطط وأقرّر
<ul style="list-style-type: none"> حاسوب. برنامج Android Studio. جهاز افتراضيّ، أو جهاز يعمل بنظام أندرويد. 	<ul style="list-style-type: none"> تعلم تعاوني/ مجموعات. حوار ومناقشة. عصف ذهنيّ / استمطار الأفكار. 	<ul style="list-style-type: none"> أتفق مع جاسر على الواجهة المستخدمة. أصمّم واجهة التطبيق. أكتب الكود البرمجيّ. أعمل اختباراً للمشروع على جهاز افتراضيّ، أو على جهاز يعمل بنظام أندرويد. 	أنفذ

<ul style="list-style-type: none"> • الخطة المعدة سابقاً. • الطلب الخاص بجاسر. • حاسوب. • برنامج Android Studio. • جهاز افتراضي، أو جهاز يعمل بنظام أندرويد. 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات. • حوار ومناقشة. 	<ul style="list-style-type: none"> • أتحمق من اختيار واجهة التطبيق الملائمة. • أتأكد من ملاءمة تصميم واجهة التطبيق. • أتحمق من تنفيذ التطبيق وظهوره على الجهاز بشكل صحيح. • أختبر التطبيق. • أتأكد من عدم ظهور أي خطأ برمجي. 	<p>أتحمق</p>
<ul style="list-style-type: none"> • حاسوب. • شاشة عرض. • جهاز افتراضي، أو جهاز يعمل بنظام أندرويد. 	<ul style="list-style-type: none"> • حوار ومناقشة. • تعلم تعاوني/مجموعات. 	<ul style="list-style-type: none"> • أوثق البيانات التي تتعلق ب: - العناصر (views) المستخدمة للإدخال والإخراج. • الجمل البرمجية الخاصة بالإيماءات. • إعداد عرض تقديمي. • فتح ملف بالحالة (إنشاء تطبيق أندرويد للتعرف إلى activity). 	<p>أوثق وأقدم</p>
<ul style="list-style-type: none"> • طلب جاسر، وثائق نشرات، كتب ومصادر متنوعة، وروابط إلكترونية متنوعة من مصادر موثوقة، عن كيفية تطوير تطبيق أندرويد للتعرف إلى الإيماءات. • نشرات عن المعايير والمواصفات الخاصة بتطوير تطبيق أندرويد. 	<ul style="list-style-type: none"> • حوار ومناقشة. 	<ul style="list-style-type: none"> • مدى فعالية التطبيق في إظهار دورة حياة activity. • رضا جاسر عن التطبيق. • ملاءمة التطبيق للمعايير والمواصفات. 	<p>أقوم</p>

نشاط للاطلاع (1): أبحث مع زملائي عن عمل كل إيماء من الإيماءات.



أتعلم: الإيماءات Gestures



الإيماءات Gestures:

التفاعلات التي يستطيع المستخدم عملها على شاشة الهاتف الذكي، وهذه الإيماءات كما يأتي:

```
override fun onTouchEvent(event: MotionEvent?): Boolean {...}
override fun onShowPress(p0: MotionEvent?) {...}
override fun onSingleTapUp(p0: MotionEvent?): Boolean {...}
override fun onDown(p0: MotionEvent?): Boolean {...}
override fun onFling(p0: MotionEvent?, p1: MotionEvent?, p2: Float, p3: Float) {...}
override fun onScroll(p0: MotionEvent?, p1: MotionEvent?, p2: Float, p3: Float) {...}
override fun onLongPress(p0: MotionEvent?) {...}
override fun onDoubleTap(p0: MotionEvent?): Boolean {...}
override fun onDoubleTapEvent(p0: MotionEvent?): Boolean {...}
override fun onSingleTapConfirmed(p0: MotionEvent?): Boolean {...}
```

الدالة `onTouchEvent` هي الدالة المسؤولة عن إظهار نتائج الإيماءات ما عدا `onDoubleTap`.

نشاط عملي (2): أنشئ تطبيقاً يوضح الإيماءات ويظهرها في `textView`.

أنشئ مشروعاً جديداً، عندما يفتح البرنامج بشكل كامل، أذهب إلى ملف `layout`، ويكون موجود تلقائياً `textView`، أعطها `id` لكي أستخدمها في الكود البرمجي، ثم أنتقل إلى الكود. في الكود أجعل الصنف يرث `Interfaces` اثنين، وهما: `OnGestureListener`, `OnDoubleTapListener` كما يأتي:

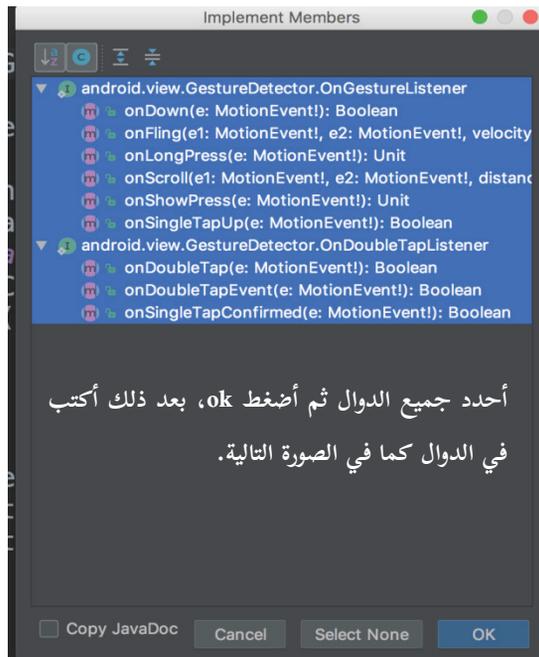
```
class MainActivity : Activity(), GestureDetector.OnGestureListener, GestureDetector.OnDoubleTapListener
```

ملحوظة: يتم وراثة أكثر من صنف من خلال وضع فاصلة (,) بعد الصنف الموروث أولاً.

إظهار البرنامج علامة خطأ، تعني الالتزام باستدعاء دوال موجودة في `Interfaces`.



أضغظ على إشارة الضوء الحمراء وأختار Implement members :



```
override fun onShowPress(p0: MotionEvent?) {
    text.text="on Press"
}

override fun onSingleTapUp(p0: MotionEvent?): Boolean {
    text.text="on single tap"
    return true
}

override fun onDown(p0: MotionEvent?): Boolean {
    text.text="on Down"
    return true
}

override fun onFling(p0: MotionEvent?, p1: MotionEvent?, p2: Float, p3: Float): Boolean {
    text.text="on fling"
    return true
}

override fun onScroll(p0: MotionEvent?, p1: MotionEvent?, p2: Float, p3: Float): Boolean {
    text.text="on scroll"
    return true
}

override fun onLongPress(p0: MotionEvent?) {
    text.text="on long press"
}

override fun onDoubleTap(p0: MotionEvent?): Boolean {
    text.text="on double tap"
    return true
}

override fun onDoubleTapEvent(p0: MotionEvent?): Boolean {
    text.text="on double tap event"
    return true
}

override fun onSingleTapConfirmed(p0: MotionEvent?): Boolean {
    text.text="on SingleTapConfirmed"
    return true
}
```



أستدعي الدالة `onTouchEvent()`، بالضغط من لوحة المفاتيح على `ctrl+o` تظهر قائمة الدوال، وأبحث عن الدالة المطلوبة، أو أقوم باستدعائها كما تعلمت سابقاً. أناقش ذلك مع زملائي.

```
private lateinit var g : GestureDetectorCompat

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
    this.g = GestureDetectorCompat(context: this, listener: this)

    g.setOnDoubleTapListener(this)
}

override fun onTouchEvent(event: MotionEvent?): Boolean {
    this.g.onTouchEvent(event)
    return super.onTouchEvent(event)
}
```

في الكود السابق عرفت متغيراً من نوع `GestureDetectorCompat`، وأعطيته خاصية `private`؛ أي أن استخدامه فقط في هذا الصنف، أيضاً قمت بإضافة `lateinit`، هذه الخاصية تُستخدم لتوضيح أن المتغير سيُعاد تعريفه في وقت لاحق. في السطر البرمجي الآتي أعرف المتغير الذي انشأته سابقاً `GestureDetectorCompat`، وهذا المتغير يستقبل قيمتين: الأولى هي الصنف الذي أريد أن أطبق الإيماءات فيه، وهو الصنف الحالي لذلك أكتب `this`، أما القيمة الثانية وهي قيمة المستمع (`listener`)؛ أي أين سيتم التنصت على وجود الإيماءات، وأكتب أيضاً `this`، وتعني أن التنصت يتم في الصنف الحالي.

```
this.g = GestureDetectorCompat(this,this)
```

أستطيع الاستغناء عن `this` في `this.g`، ولكن الأفضل وجودها.



هذا السطر البرمجي خاص بالتنصت على النقر المزدوج، ويعني أن المتغير `g` سيأخذ قيمة `onDoubleTap` من هذا الصنف، ويُظهر نتيجته:

```
g.setOnDoubleTapListener(this)
```

أضغط على `ctrl+o`، وأبحث عن `onTouchEvent`، وأستخدمها كما يأتي:

```
override fun onTouchEvent(event: MotionEvent?): Boolean {
    this.g.onTouchEvent(event)
    return super.onTouchEvent(event)
}
```



السطر البرمجيّ الآتي يأخذ قيمة الإيماء الذي يحدث على الشاشة:

```
this.g.onTouchEvent(event)
```

أشغل المشروع، وأختبر الإيماءات Gesture.



1. أعرف الإيماءات Gesture.
2. أذكر ثلاث أنواع من الإيماءات Gesture.



القوائم Menus.

وصف الموقف التعليمي: زار حسّان مشغل تطبيقات الهاتف المحمول في مدرسة عسقلان المهنيّة؛ لمساعدته في إنشاء تطبيق أندرويد يحتوي على بياناته الشخصية، ويستطيع التحكّم بلون الخلفيّة.

العمل الكامل			
الموارد	المنهجية (استراتيجية التعلّم)	وصف الموقف الصّفيّ	خطوات العمل
<ul style="list-style-type: none"> وثائق (طلب حسّان، نشرات، كتب ومصادر متنوعة). التكنولوجيا (روابط إلكترونية متنوعة من مصادر موثوقة، إنترنت، حاسوب). 	<ul style="list-style-type: none"> حوار ومناقشة. التعلّم التعاوني/ مجموعات. البحث العلميّ. 	<ul style="list-style-type: none"> أجمع البيانات من حسّان عن: <ul style="list-style-type: none"> - إصدار نظام الأندرويد المثبت على جهازه ونوعه. - المعلومات الشخصية. - الألوان التي يريد إضافتها للخلفيّة. أجمع البيانات عن: <ul style="list-style-type: none"> - العناصر (views) المستخدمة للإدخال والإخراج. - القائمة المستخدمة لتغيير اللون. - الجمل الشرطيّة المستخدمة. 	أجمع البيانات وأحلّها
<ul style="list-style-type: none"> وثائق (نشرات، كتب ومصادر متنوعة). التكنولوجيا (روابط إلكترونية متنوعة من مصادر موثوقة). 	<ul style="list-style-type: none"> تعلّم تعاوني/ مجموعات. حوار ومناقشة. 	<ul style="list-style-type: none"> أصنّف البيانات: (نظام الأندرويد المستخدم، الاسم، العمر، تاريخ الميلاد، درجة التعليم، المهارات، المواهب، الخبرات). أحدّد: <ul style="list-style-type: none"> - المعطيات: (الاسم، العمر، تاريخ الميلاد، درجة التعليم، المهارات، المواهب، الخبرات). - العناصر المستخدمة. - الجمل البرمجيّة الشرطيّة. - القائمة المستخدمة. - إعداد خطة للتنفيذ. 	أخطّط وأقرّر

<ul style="list-style-type: none"> • حاسوب. • برنامج Android Studio. • جهاز افتراضيّ، أو جهاز يعمل بنظام أندرويد. 	<ul style="list-style-type: none"> • تعلّم تعاونيّ/ مجموعات. • حوار ومناقشة. • عصف ذهنيّ/استمطار الأفكار. 	<ul style="list-style-type: none"> • أتفق مع حسان على الواجهة المستخدمة والألوان. • أصمّم واجهة التطبيق. • أضيف قائمة علويّة. • أكتب الكود البرمجيّ. • أعمل اختباراً للمشروع على جهاز افتراضيّ، أو على جهاز يعمل بنظام أندرويد. 	<p>أنفذ</p>
<ul style="list-style-type: none"> • الخطة المعدة سابقاً. • الطلب الخاصّ بحسان. • حاسوب. • برنامج Android Studio. • جهاز افتراضيّ، أو جهاز يعمل بنظام أندرويد. 	<ul style="list-style-type: none"> • تعلّم تعاونيّ/ مجموعات. • حوار ومناقشة. 	<ul style="list-style-type: none"> • أتحقّق من اختيار واجهة التطبيق الملائمة. • أتأكد من ملائمة تصميم واجهة التطبيق. • أتحقّق من تنفيذ التطبيق وظهوره على الجهاز بشكل صحيح. • أختبر التطبيق. • أتأكد من عدم ظهور أي خطأ برمجيّ. 	<p>أتحقّق</p>
<ul style="list-style-type: none"> • حاسوب. • شاشة عرض. • جهاز افتراضيّ، أو جهاز يعمل بنظام أندرويد. 	<ul style="list-style-type: none"> • حوار ومناقشة. • تعلّم تعاونيّ/ مجموعات عمل. 	<ul style="list-style-type: none"> • أوثّق البيانات التي تتعلّق ب:- - العناصر (views) المستخدمة للإدخال والإخراج. - الجمل البرمجيّة المستخدمة. - القائمة المستخدمة. • إعداد عرض تقديميّ. • فتح ملف بالحالة (إنشاء تطبيق أندرويد للتعرف إلى activity). 	<p>أوثّق وأقدم</p>
<ul style="list-style-type: none"> • طلب حسان، وثائق نشرات، كتب ومصادر متنوعة، وروابط إلكترونية متنوعة من مصادر موثوقة، خاصة بعرض البيانات والتحكّم بالخلفية. • نشرات عن المعايير والمواصفات الخاصة بتطوير تطبيق أندرويد. 	<ul style="list-style-type: none"> • حوار ومناقشة. 	<ul style="list-style-type: none"> • مدى فعاليّة التطبيق في عرض البيانات والتحكّم بالخلفيّة. • رضا حسان عن التطبيق. • ملائمة التطبيق للمعايير والمواصفات. 	<p>أقوم</p>

أناقش: تُستخدم قوائم menus فقط لإعدادات التطبيق.



أتعلم: القوائم menus

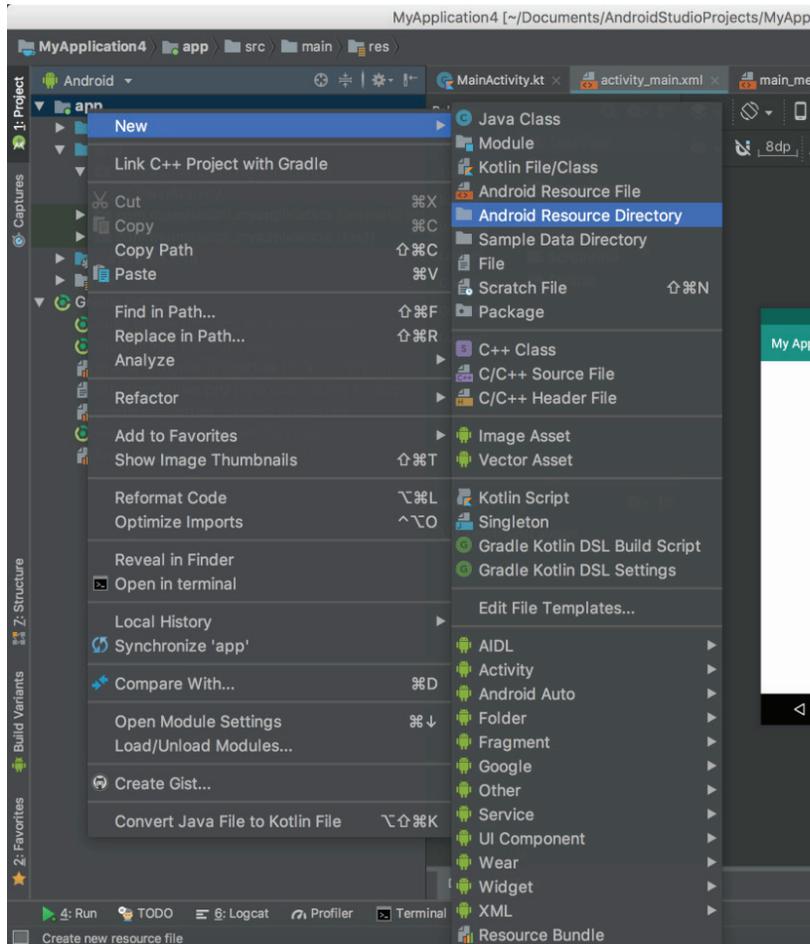


القوائم menus:

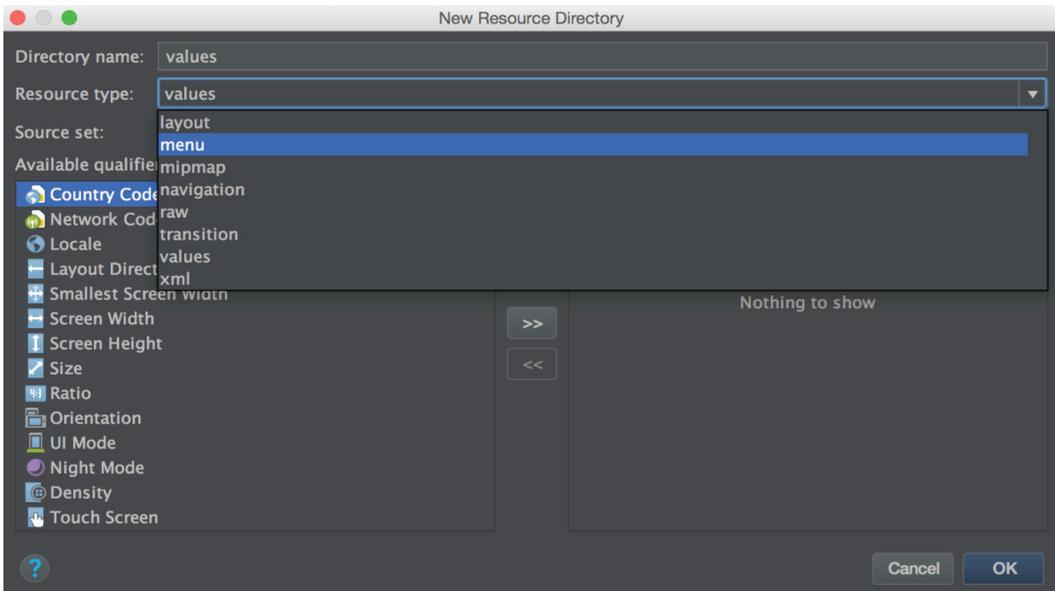
القائمة العلوية: هي القائمة التي تكون في أعلى التطبيق وعادة تكون قائمة إعدادات، ويُرمز لها في التطبيق بثلاث نقاط عمودية تدلّ عليها.

نشاط عملي (1): أنشئ تطبيقاً يحتوي على قائمة علوية بداخلها أسماء الفصول الأربعة، وعندما أضغط على أي خيار يظهر اسم الفصل بـ toast.

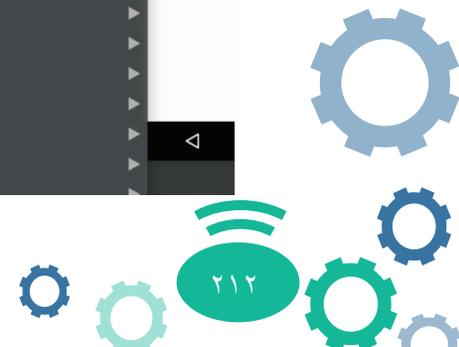
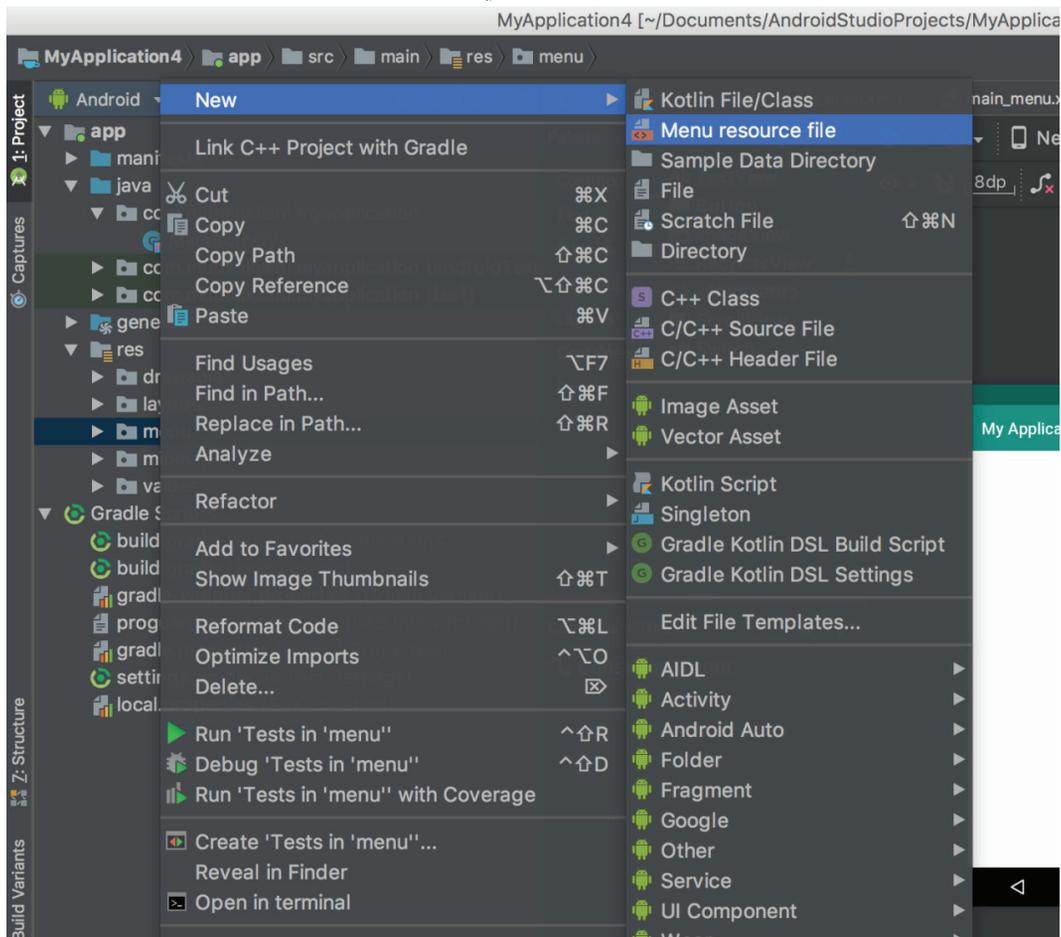
أفتح الأندرويد ستوديو وأنشئ مشروعاً جديداً وأسميه، وعندما يفتح البرنامج بشكل كامل أمسح جميع العناصر التي بداخل ملف layout الرئيسي.
الآن بزر الفأرة اليمين أضغط على ملف app، وأختار (Android Resource Directory)، كما في الصورة الآتية:

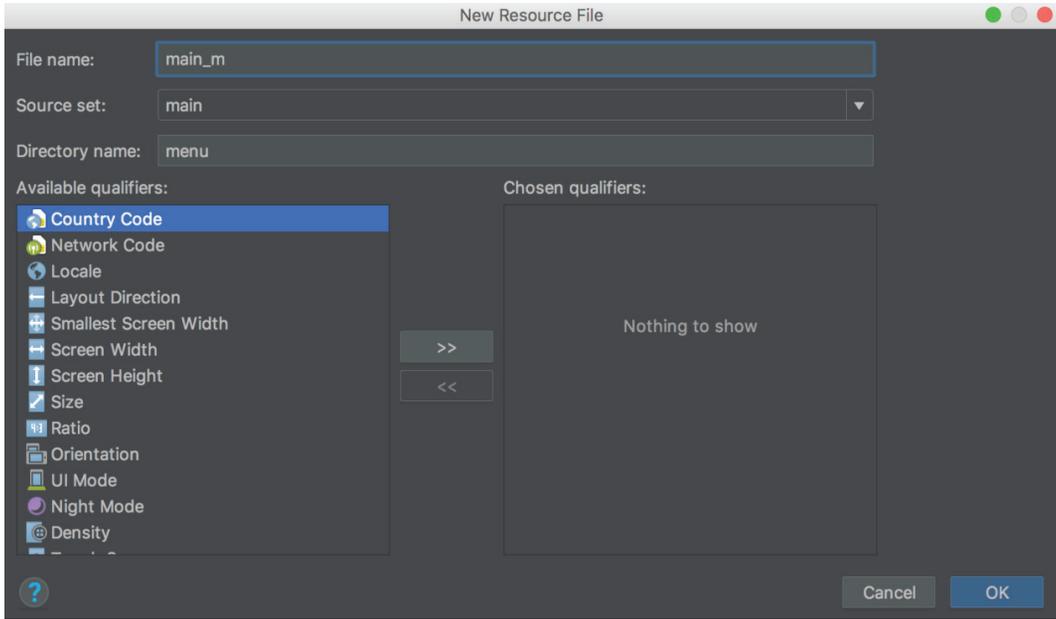


ثم أختار (menu) من (Resource type)، وأضغط على زر (OK)، كما يأتي:

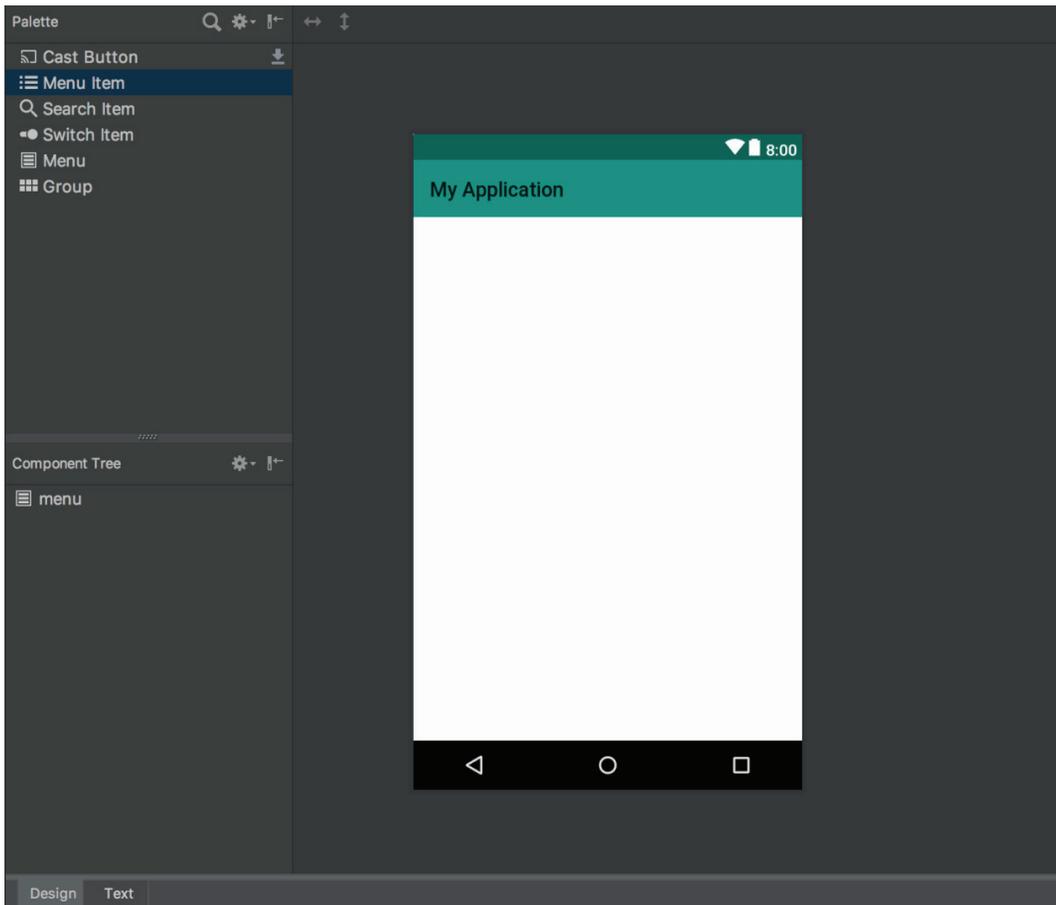


بعد إنشاء المجلد (menu) الخاص بالقائمة، أضغط على مجلد menu بزر الفأرة اليمين لأنشئ ملف xml خاصاً بالقائمة، اختار (New -> Menu resource file)، كما يأتي:

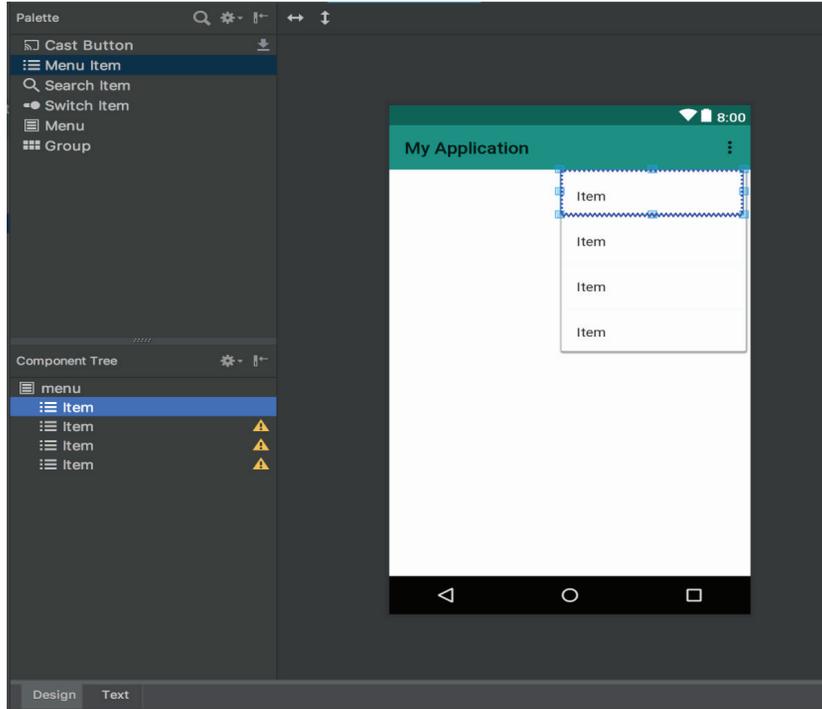




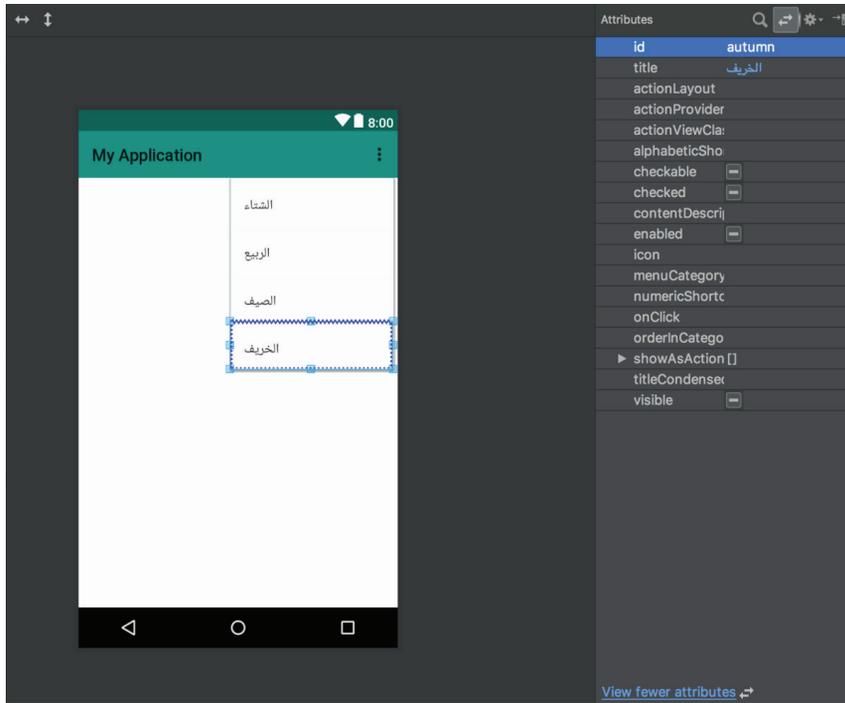
أكتب اسم لملف xml الخاص بالقائمة، في خانة file name، ثم تظهر لي نافذة كالآتي:



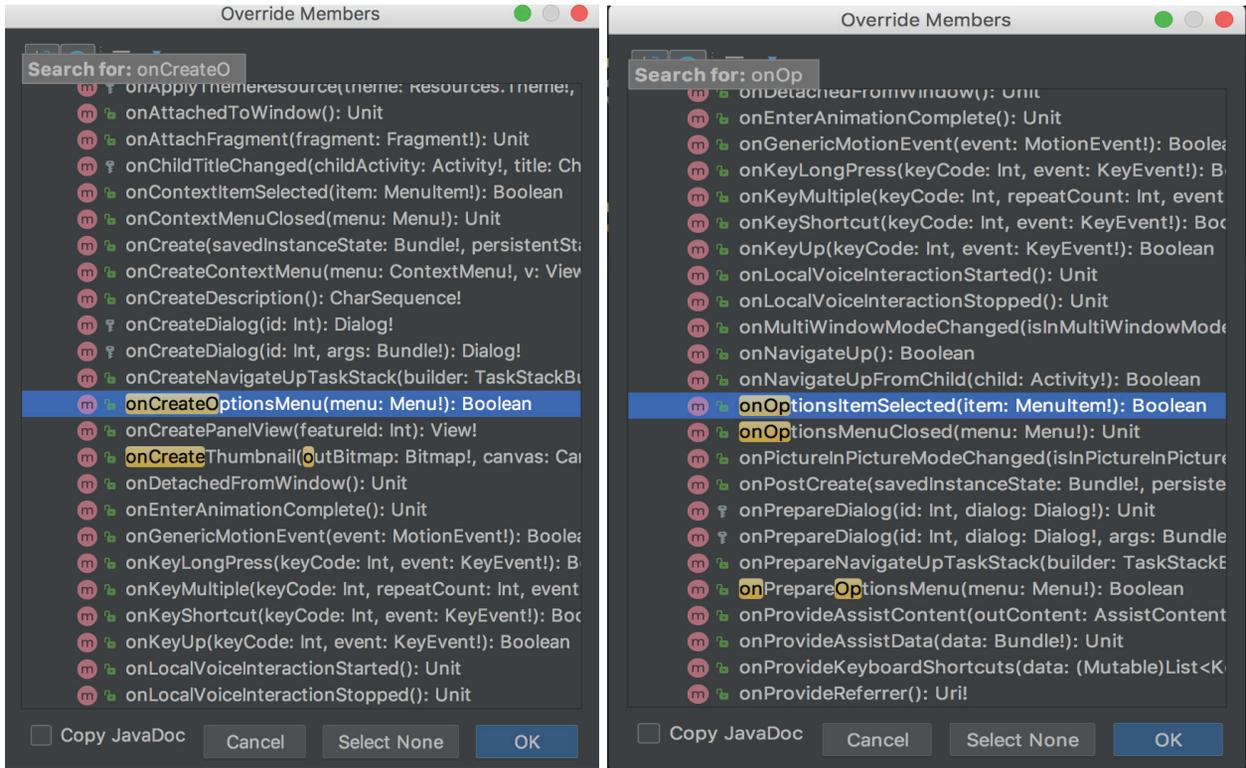
الآن أبدأ بإضافة العناصر للقائمة من خلال السحب menu item، والإفلات في الأسفل عند كلمة menu، حتى تصبح كما يأتي:



أضغط على كل عنصر في القائمة، وأغيّر الاسم (title) والمعرف (ID)، وفي المثال هنا أُغَيّر الاسم باسم الفصل، وأعطيه id مناسباً.



انتقل إلى الكود البرمجي، وأعمل استدعاء للدالتين onCreateOptionsMenu، onOptionsItemSelected كما تعلمت سابقاً.



```

override fun onCreateOptionsMenu(menu: MenuItem?): Boolean {
    return super.onCreateOptionsMenu(menu)
}

override fun onOptionsItemSelected(item: MenuItem?): Boolean {
    return super.onOptionsItemSelected(item)
}

```

- onCreateOptionsMenu: أكتب بداخل هذه الدالة كوداً لربط ملف xml الخاص بالقائمة مع الكود البرمجي.
- onOptionsItemSelected: أكتب بداخل هذه الدالة الأوامر اللازمة عند الضغط على عنصر في القائمة.

```

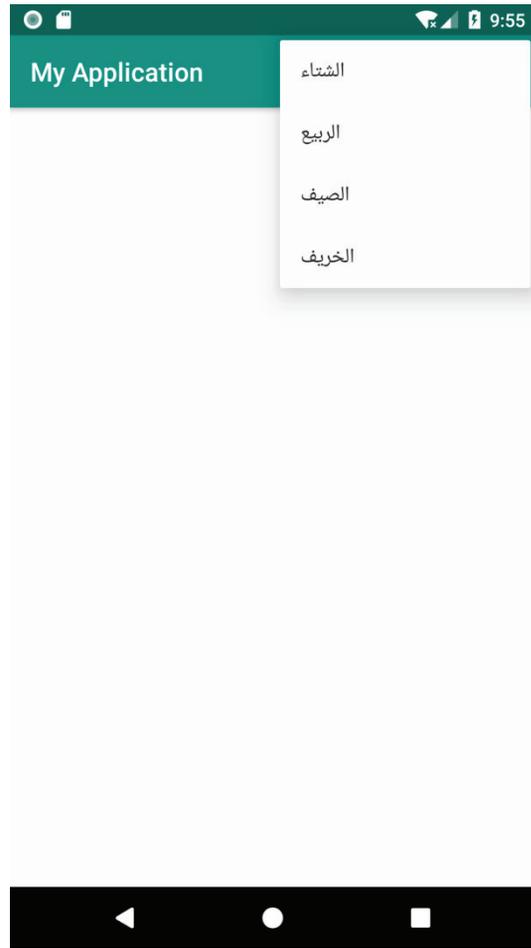
override fun onCreateOptionsMenu(menu: MenuItem?): Boolean {
    menuInflater.inflate(R.menu.main_menu, menu)
    return super.onCreateOptionsMenu(menu)
}

override fun onOptionsItemSelected(item: MenuItem?): Boolean {
    when (item!!.itemId) {
        R.id.winter -> {
            Toast.makeText(context, item.title, Toast.LENGTH_SHORT).show()
        }
        R.id.spring -> {
            Toast.makeText(context, item.title, Toast.LENGTH_SHORT).show()
        }
        R.id.summer -> {
            Toast.makeText(context, item.title, Toast.LENGTH_SHORT).show()
        }
        R.id.autumn -> {
            Toast.makeText(context, item.title, Toast.LENGTH_SHORT).show()
        }
    }
    return super.onOptionsItemSelected(item)
}

```



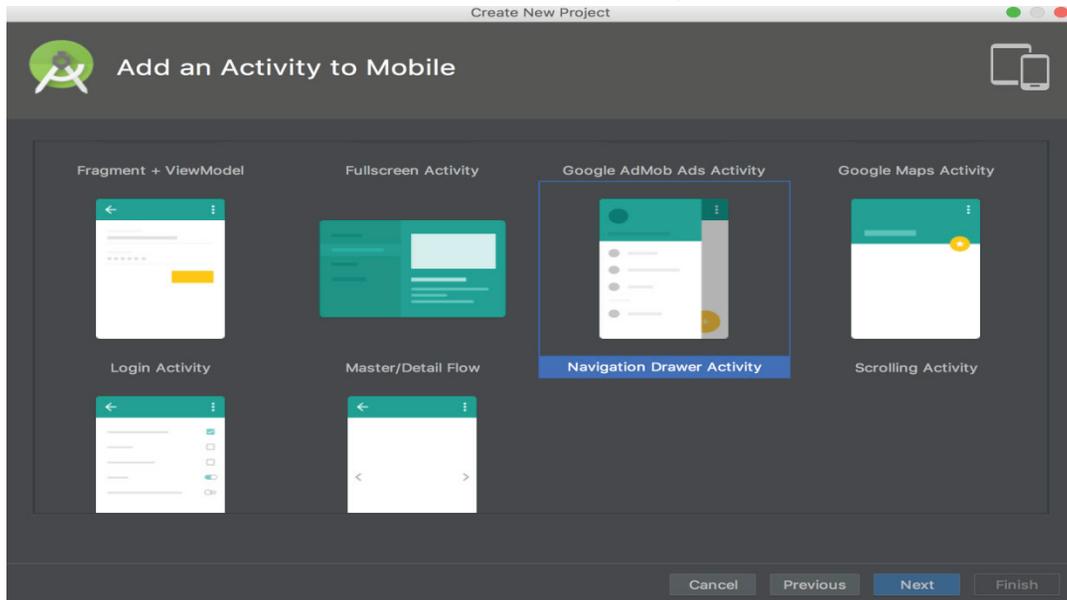
- السطر الآتي يقوم بربط ملف xml بالكود (R.menu.main_menu, menu) menuInflater.inflate
- menuInflater: هي دالة جاهزة استخدمتها لعملية الربط، وتستقبل قيمتين: الأولى هي مكان ملف xml وأحصل عليه بكتابة R.menu.main_menu، والقيمة الثانية اسم القائمة وفق تعريفها في الدالة الرئيسية (menu: Menu?).
- استخدمت الجملة الشرطية when لمعرفة أي خيار (item) تم الضغط عليه عن طريق مقارنة id الخاص بكل خيار (item).
 - أكتب item.getItemId للوصول لجميع id الخاصة بالعنصر (item) المخزنة، وأضعها في الجملة الشرطية when، ولأستخدم الـ id الخاص بكل عنصر (item) في الجملة الشرطية أكتب R.id.winter.
 - أكتب item.title لعرض النص الموجود في العنصر (item).
 - أشغل المشروع، وأتأكد من عمل التطبيق بالشكل الصحيح.



هي القائمة التي تظهر من جانب الواجهة، إمّا بالسحب، أو عن طريق الضغط على أيقونة القائمة، وهي عبارة عن ثلاثة خطوط أفقيّة.

نشاط للاطلاع (2): أبحث مع زملائي عن أسماء تطبيقات تستخدم القائمة الجانبيّة.

لعمل تطبيق يحتوي على قائمة جانبيّة أفتح الأندرويد ستوديو، وأنشئ مشروعاً جديداً، عند إنشاء المشروع أختار (Navigation Drawer Activity)، كما في الصورة الآتية:

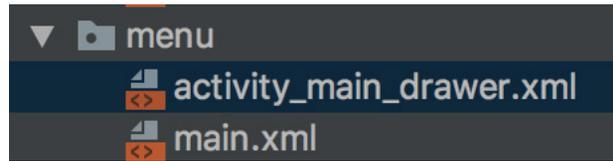


عندما يفتح البرنامج بشكل كامل، يظهر ملف xml الرئيسي يحتوي على `textView`، أضع لها id لأتعامل معها برمجياً في وقت لاحق.

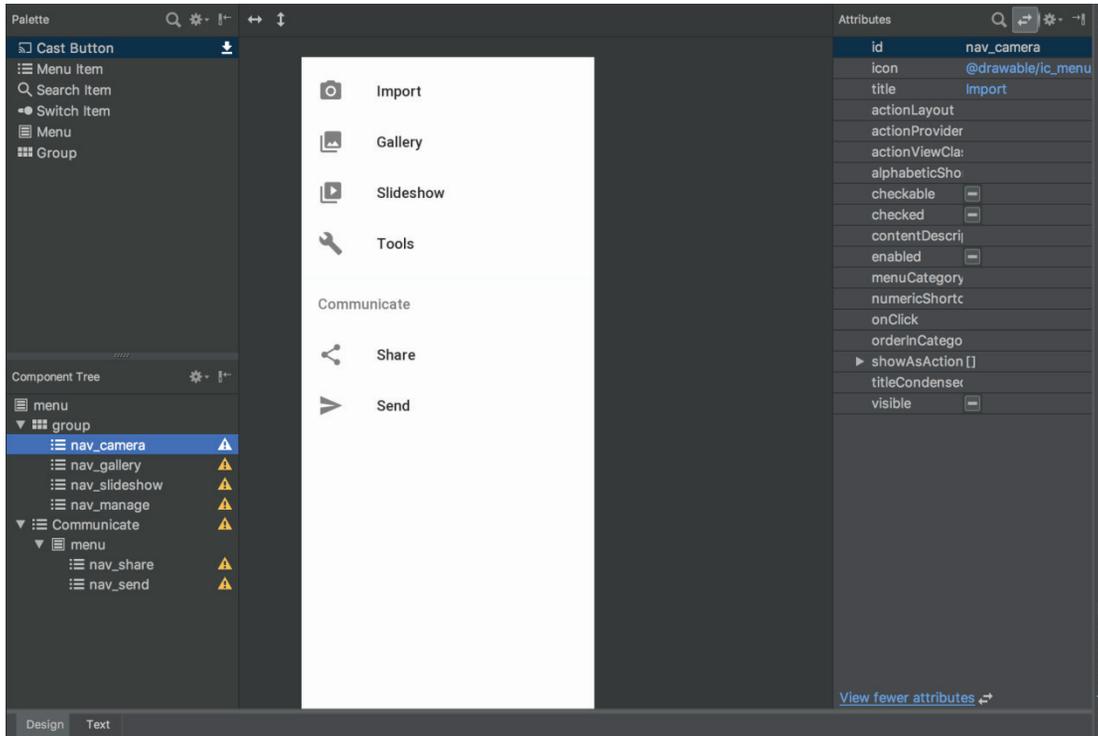
مكوّنات مشروع القائمة الجانبيّة:

- ملف كود واحد.
- أربعة ملفّات xml داخل مجلّد layout.
- ملفّان في مجلّد menu.



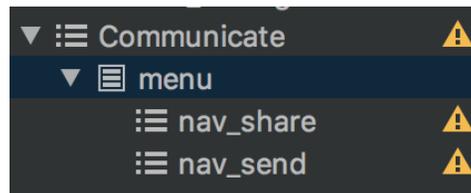


1. الملف activity_main_drawable.xml يحتوي على عناصر القائمة الجانبية:



القائمة اليسرى هي العناصر التي أريد إضافتها للقائمة، سوف أستخدم منها group و menu item، أما الأولى فتستخدم لتجميع عدة عناصر مع بعضها لأغراض الترتيب والتنظيم، كما يمكن الاستغناء عنها، وتتم إضافتها ثم إضافة item بداخلها عن طريق السحب والإفلات.

استخدمت item وبداخلها menu جديدة:



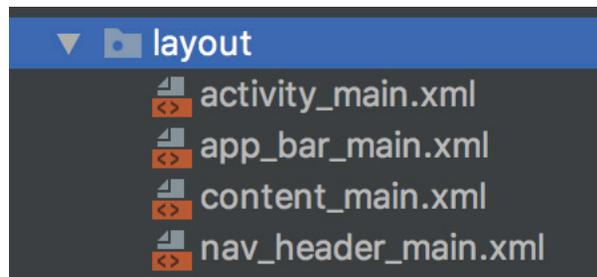
كلمة communicate هي اسم item وأستطيع تغييرها، وفائدة وجود menu داخل item لتوضيح أن هذا القسم جديد داخل القائمة الرئيسية، ويكون عنوان القسم اسم item مع عدم إمكانية الضغط على اسم القسم في التطبيق. القائمة اليمنى تحتوي على خصائص العناصر، وأهم العناصر التي سوف أستخدمها هي: id، الأيقونة (icon)، الاسم (title)، chekable.



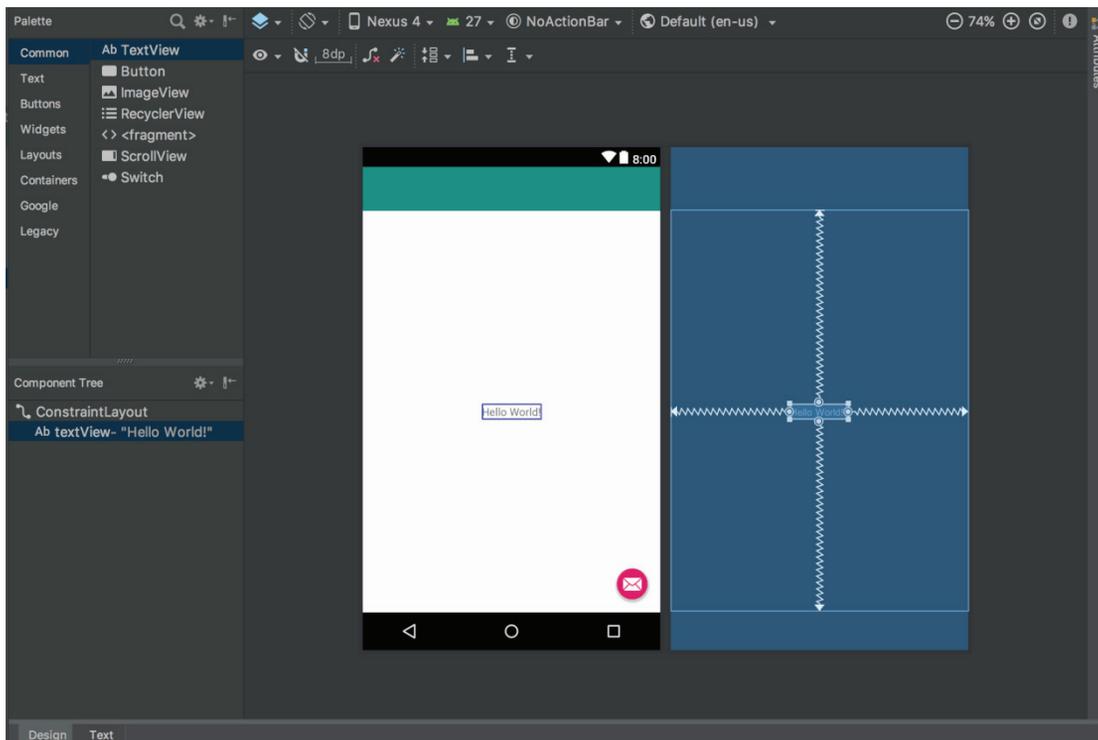
- Id: هو اسم فريد استخدمه لربط العنصر في الكود البرمجيّ.
- Icon: هي أيقونة العنصر التي تعبّر عن اسم العنصر وتُؤخَذُ من ملف drawable، ومن الضروريّ أن تكون أبعادها قليلة، مثلاً: 50 طول * 50 عرض، وتظهر باللونين الأبيض والأسود.
- Title: هو اسم ل item الذي سوف يظهر للمستخدم.
- Chekable: تطلّل العنصر في القائمة الذي تمّ الضغط عليه، وتمّ اختياره.

2. ملف main.xml: هذا ملف للقائمة العلوية التي تمّ شرحها سابقاً.

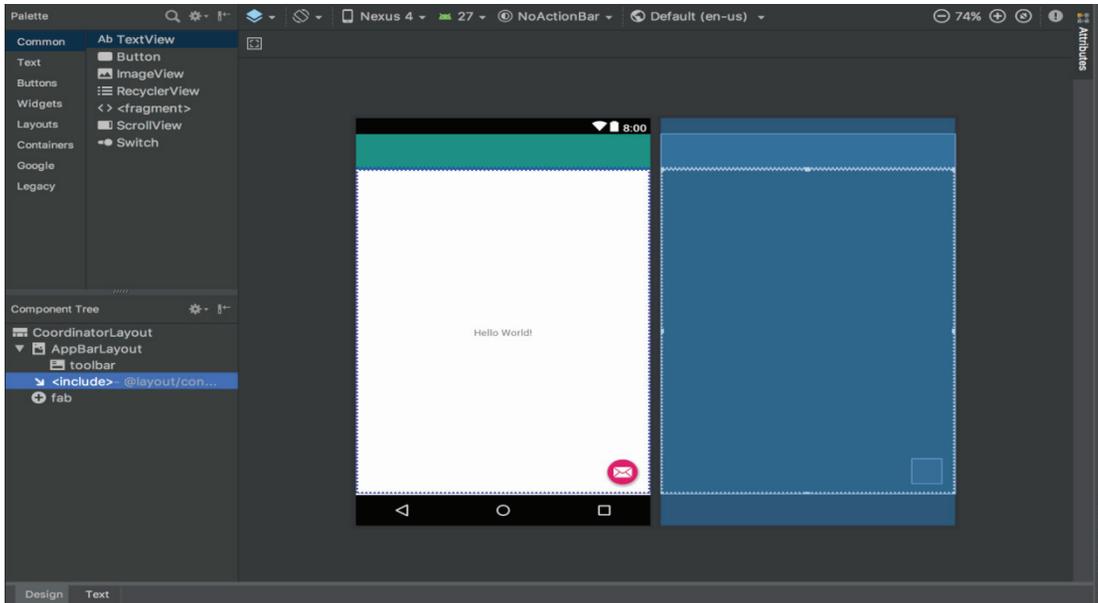
مجلّد layout: ألاحظ وجود أربعة ملفات بداخله كما يأتي:



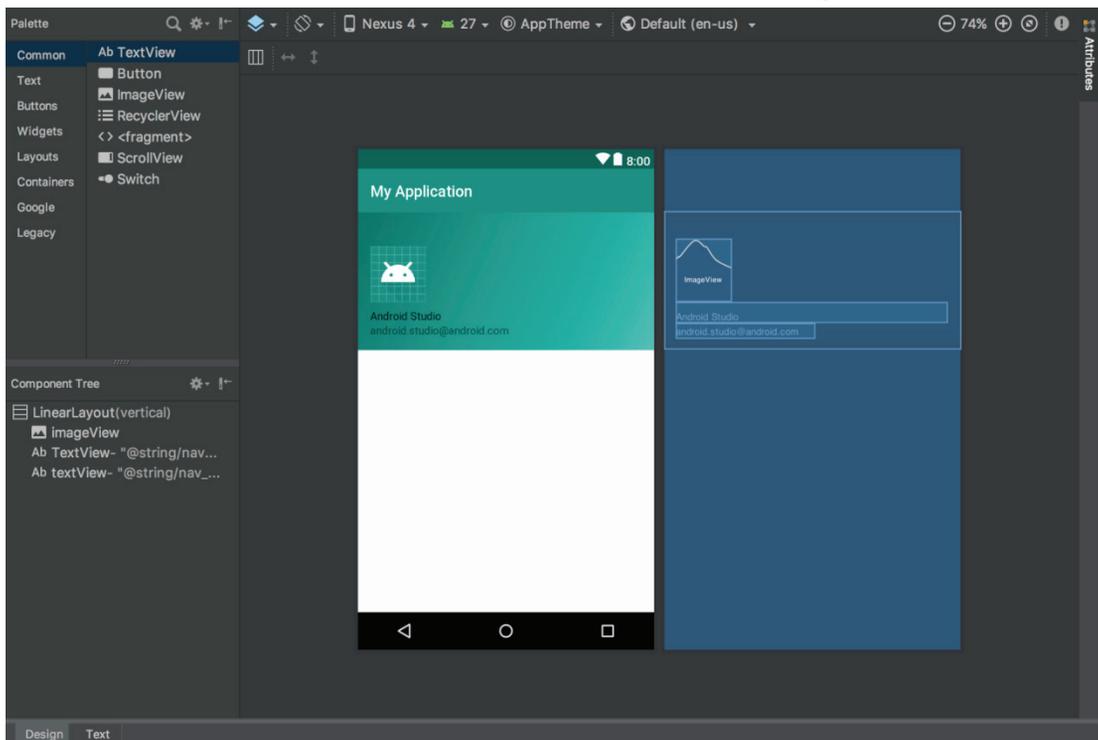
- ملف content_main: للواجهة التي تحتوي على العناصر والأمر التي أريد عرضها، هنا أترك textView موجودة وأعطيها id؛ لكي أعرض فيها ما تمّ اختياره من القائمة.



- ملف `app_bar_main`: يتحكّم بشرط الأداة الموجود فوق الواجهة، ويحتوي على اسم التطبيق والقائمة العلويّة، وأعمل في هذا الملف `include` لملف `content_main`؛ أي أنّ هذا الملف المسؤول عن إظهار ملف `content_main`



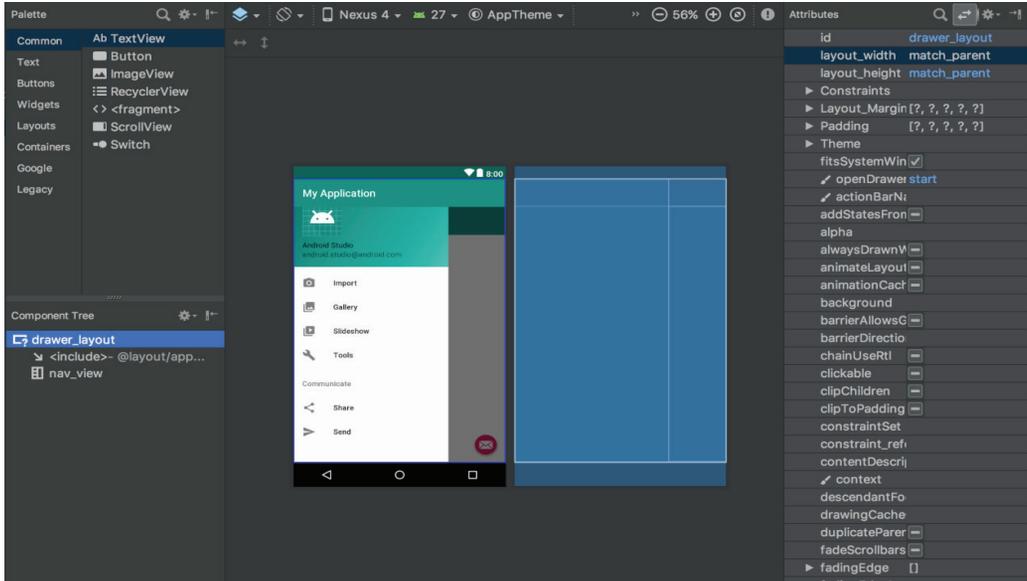
- ملف `nav_header_main`: يتحكّم بالشكل الذي يكون أعلى القائمة الجانبيّة، ويمكن تغيير الخلفيّة والصورة والنصوص، وإضافة الطابع الخاصّ الذي أريده.



- ملف `activity_main`: الملف الرئيسي الذي سوف أجمع فيه جميع الملفات سابقة الذكر، أولاً أضيف `nav_header_main`، وفي داخلها أضيف ملف `nav_header_main`، والملف الذي يحتوي على القائمة الجانبية، كما في الصورة الآتية:

headerLayout	@layout/nav_header_main
layout_gravity	[start]
menu	@menu/activity_main_drawer

اضيف include يحتوي على ملف `app_bar_main` وهكذا أكون قد انتهيت ملفات الواجهة.



ملف `main_activity` الكود البرمجي:

ما أريد معرفته والتعامل معه في الكود البرمجي الجزء الآتي:

```

override fun onNavigationItemSelected(item: MenuItem): Boolean {
    // Handle navigation view item clicks here.
    when (item.itemId) {
        R.id.nav_camera -> {
            textView.text=item.title
        }
        R.id.nav_gallery -> {
            textView.text=item.title
        }
        R.id.nav_slideshow -> {
            textView.text=item.title
        }
        R.id.nav_manage -> {
            textView.text=item.title
        }
        R.id.nav_share -> {
            textView.text=item.title
        }
        R.id.nav_send -> {
            textView.text=item.title
        }
    }

    drawer_layout.closeDrawer(GravityCompat.START)
    return true
}

```



الكود خاصّ بالقائمة الجانبيّة لإعطاء أوامر الضغط، وأريد إظهار النصّ الموجود في عناصر القائمة على TextView عند الضغط على عنصر محدّد، فيصبح الكود كالآتي:

```
override fun onNavigationItemSelected(item: MenuItem): Boolean {
    // Handle navigation view item clicks here.
    when (item.itemId) {
        R.id.nav_camera -> {
            // Handle the camera action
        }
        R.id.nav_gallery -> {

        }
        R.id.nav_slideshow -> {

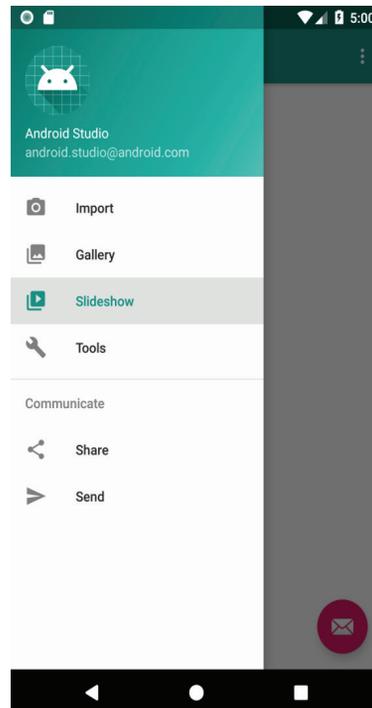
        }
        R.id.nav_manage -> {

        }
        R.id.nav_share -> {

        }
        R.id.nav_send -> {

        }
    }
    drawer_layout.closeDrawer(GravityCompat.START)
    return true
}
```

أشغل المشروع، وأختبر التطبيق، وألاحظ النتائج:



نشاط عملي (3): أنشئ تطبيق يحتوي على قائمة جانبيّة، بداخلها أسماء زملائي، وعند الضغط على اسم أي شخص تظهر معلومات عن هذا الشخص في الشاشة الرئيسيّة.



1. أذكر نوعين من القوائم؟
2. ما مكوّنات القائمة الجانبيّة؟



التنقل ونقل بيانات بين الواجهات.

وصف الموقف التعليمي: استضافت معلّمة رياضيات خبيراً مشغول تطبيقات الهاتف المحمول في مدرسة عسقلان المهنيّة؛ لمساعدتها في إنشاء تطبيق يقوم بحساب مساحة الأشكال الهندسيّة ومحيطها (يكون إدخال البيانات في واجهة والنتائج في واجهة أخرى).

العمل الكامل			
خطوات العمل	وصف الموقف الصّفيّ	المنهجية (استراتيجيةّ التّعلم)	الموارد
أجمع البيانات وأحلّها	<ul style="list-style-type: none"> • أجمع البيانات من المعلّمة عن: <ul style="list-style-type: none"> - الأشكال الهندسيّة. - قوانين المساحة والمحيط. - إصدار نظام الأندرويد. • أجمع البيانات عن: <ul style="list-style-type: none"> - العناصر (views) المستخدمة للإدخال والإخراج. - عدد الواجهات. - الجمل البرمجيّة المستخدمة في ربط ونقل البيانات بين الواجهات. 	<ul style="list-style-type: none"> • حوار ومناقشة. • التعلّم التعاوني/ مجموعات. • البحث العلميّ. 	<ul style="list-style-type: none"> • وثائق (طلب المعلّمة، نشرات، كتب ومصادر متنوعة). • التكنولوجيا (روابط إلكترونية متنوعة من مصادر موثوقة، حاسوب، إنترنت، هاتف يعمل بنظام الأندرويد).
أخطّط وأقرّ	<ul style="list-style-type: none"> • أصنّف البيانات: (نظام الأندرويد المستخدم، المربّع، الدائرة، المستطيل، ...، قانون المساحة، قانون المحيط). • أ حدّد: <ul style="list-style-type: none"> - المعطيات (المربّع، الدائرة، المستطيل، ...، قانون المساحة، قانون المحيط). - العناصر (views) المستخدمة. - عدد الواجهات. - الجمل البرمجيّة. • إعداد خطة للتنفيذ. 	<ul style="list-style-type: none"> • تعلّم تعاوني/ مجموعات. • حوار ومناقشة. 	<ul style="list-style-type: none"> • وثائق (نشرات، كتب ومصادر متنوعة) (روابط إلكترونية متنوعة من مصادر موثوقة).

<ul style="list-style-type: none"> • حاسوب . • برنامج Android Studio . • جهاز افتراضيّ، أو جهاز يعمل بنظام أندرويد . 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات حوار ومناقشة . • عصف ذهني / استمطار الأفكار 	<ul style="list-style-type: none"> • اختيار عدد من الواجهات المقترحة قابلة للاستخدام في تطبيق الأندرويد . • أتفق مع المعلمة على اختيار واجهات التطبيق المناسبة . • أصمّم واجهات التطبيق وأربط بينها، وأنشئه . • أعمل اختباراً للمشروع على جهاز افتراضيّ، أو على جهاز يعمل بنظام أندرويد . 	<p>أفقد³</p>
<ul style="list-style-type: none"> • النخطة المعدّة سابقاً . • الطلب الخاص بالمعلمة . • حاسوب . • برنامج Android Studio . • جهاز افتراضيّ، أو جهاز يعمل بنظام الأندرويد . 	<ul style="list-style-type: none"> • تعلم تعاوني/مجموعات حوار ومناقشة . 	<ul style="list-style-type: none"> • أتحقّق من ملاءمة اختيار الواجهات المقترحة للتطبيق . • أتأكد من ملاءمة التطبيق . • أتحقّق من تنفيذ التطبيق وظهوره على الجهاز بشكل مناسب . 	<p>أتحقّق³</p>
<ul style="list-style-type: none"> • حاسوب . • شاشة عرض . • جهاز افتراضيّ، أو جهاز يعمل بنظام أندرويد . 	<ul style="list-style-type: none"> • حوار ومناقشة . • تعلم تعاوني/ مجموعات عمل . 	<ul style="list-style-type: none"> • أوّلق البيانات التي تتعلّق ب: <ul style="list-style-type: none"> - العناصر (views) المستخدمة للإدخال والإخراج . - عدد الواجهات . - الجمل البرمجية المستخدمة في ربط ونقل البيانات بين الواجهات . • إعداد عرض تقديمي . • إعداد ملف بالحالة (إنشاء تطبيق يقوم بحساب مساحة الاشكال الهندسيّة ومحيطها) . 	<p>أوّلق وأقدم³</p>
<ul style="list-style-type: none"> • طلب المعلمة، وثائق نشرات، كتب ومصادر، وروابط إلكترونية متنوعة من مصادر موثوقة، ونشرات عن المعايير والمواصفات الخاصة بتطوير تطبيق أندرويد . 	<ul style="list-style-type: none"> • حوار ومناقشة . 	<ul style="list-style-type: none"> • رضا المعلمة عن التطبيق . • ملاءمة التطبيق للمعايير والمواصفات . 	<p>أقوم³</p>



أناقش: يمكن نقل أي نوع من البيانات بين الواجهات دون الأخذ بالاعتبار نوعها.

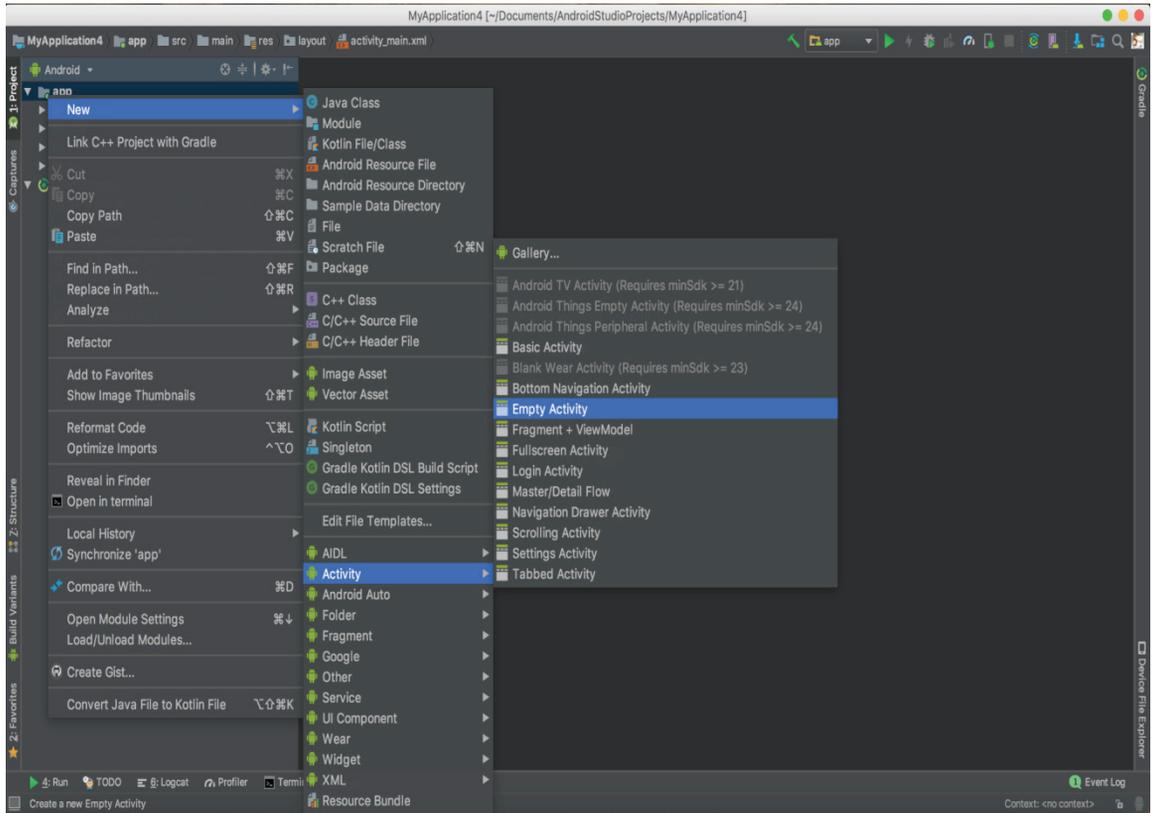


أنعلّم: التنقل بين الواجهات



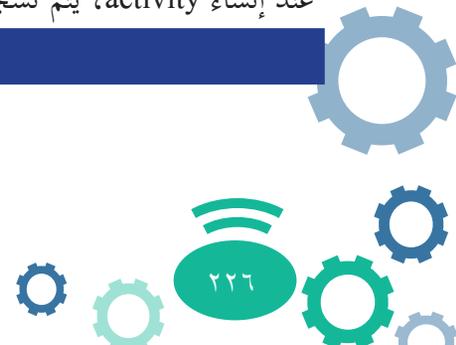
إنشاء activity جديدة في المشروع نفسه:

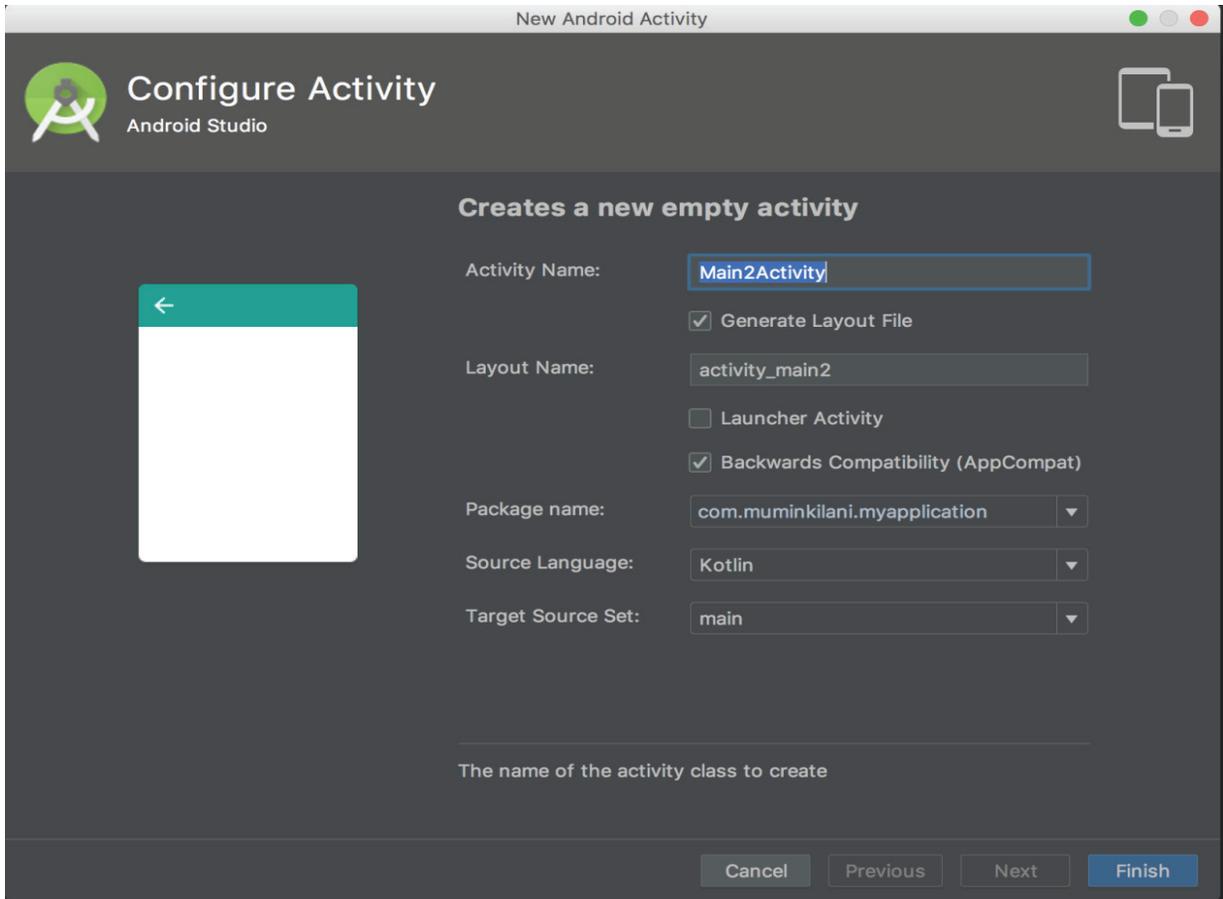
لإنشاء activity جديدة، أضغط بزر الفأرة اليمين على كلمة app، ثمّ new ثمّ activity، وأختار empty activity.



عند إنشاء activity، يتم تسجيلها بملف manifest تلقائياً.

```
<activity android:name=".Main2Activity" > </activity >
```



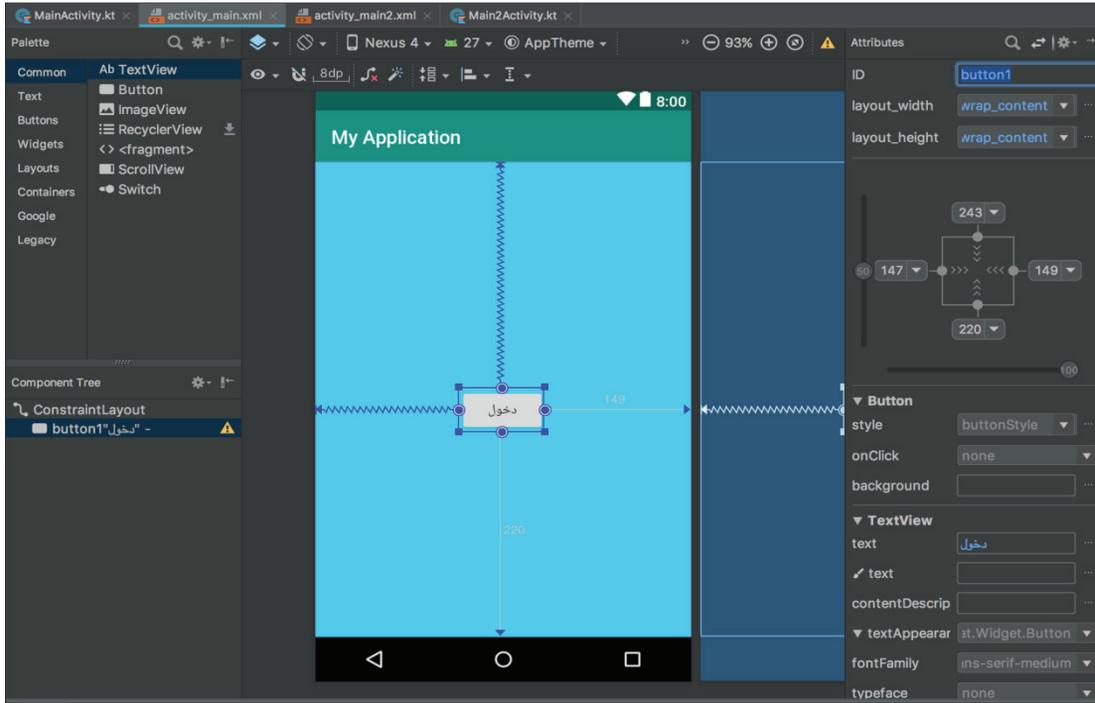


في الصورة أعلاه أستطيع تغيير اسم activity بالاسم الذي أريده.



نشاط عملي (1): أنشئ تطبيقاً يحتوي على واجهتين، لكلٍ منهما لون مختلف، وتحتوي كلٌ واحدة على button، للتنقل بين الواجهتين.

أفتح الأندرويد ستوديو، وأنتظر حتى يعمل بشكل كامل، ثم أنشئ activity جديدة، وأضع في كل واجهة button، ثم أعطي كل button id مختلفاً، وأغيّر لون كل واجهة بلون مختلف، كما يأتي:



انتقل إلى الكود البرمجي الخاص بـ activity الأولى، وأكتب كما يأتي:

```

1 package com.muminkilani.myapplication
2
3 import ...
4
5
6
7
8
9
10
11
12 class MainActivity : AppCompatActivity() {
13
14
15     override fun onCreate(savedInstanceState: Bundle?) {
16         super.onCreate(savedInstanceState)
17         setContentView(R.layout.activity_main)
18         button1.setOnClickListener { it: View!
19
20             val intent = Intent( packageContext: this, Main2Activity::class.java)
21             startActivity(intent)
22         }
23     }
24 }
25

```

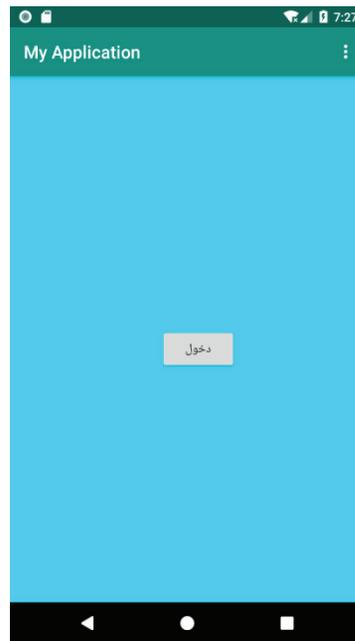
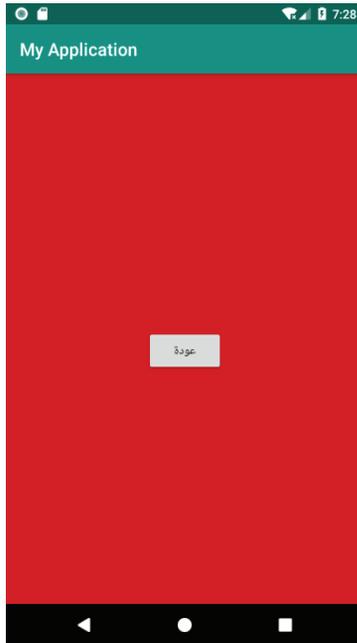
في داخل `button1.setOnClickListener` أنشئ كائناً من الصنف `Intent`، من وظائف `intent` التنقل من واجهة إلى أخرى، وعند تعريفه بعد إشارة المساواة ألاحظ أنه يستقبل قيمتين: الأولى هي الواجهة (`activity`) الحالية وأشير إليها بـ `this`، أما الثانية فهي الواجهة (`activity`) التي أريد أن انتقل إليها، أكتب اسمها، وبعدها أؤكد أنها من نوع `class` من خلال كتابة (`::class.java`).

أكتب جملة `startActivity`، وهي الدالة الخاصّة لبدء عمل `intent` تستقبل قيمة من نوع `intent` وهي الكائن الذي عرفته سابقاً.

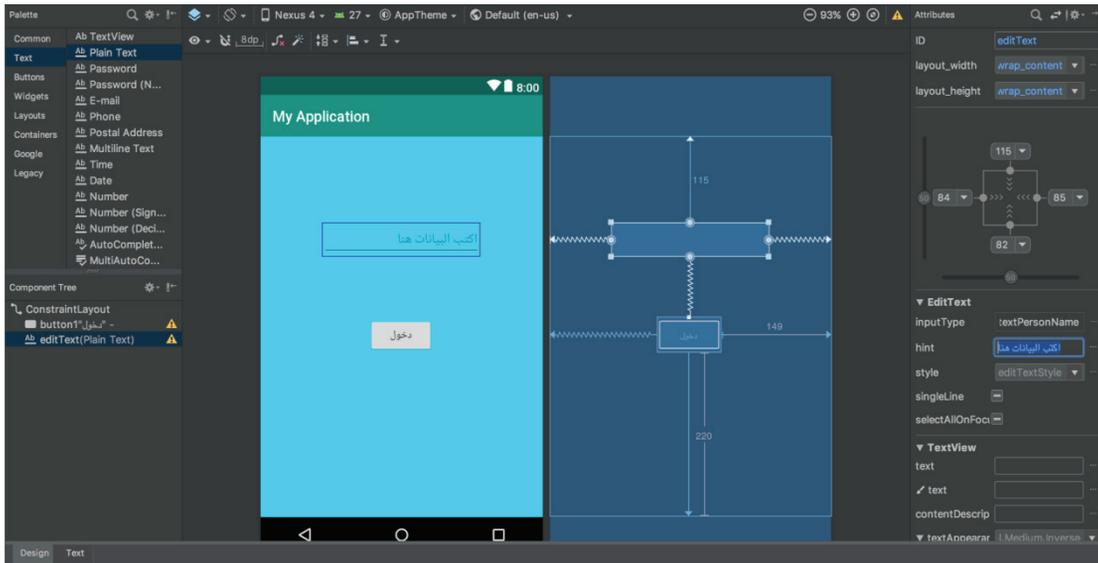
أنتقل إلى ملف كود الواجهة (activity) الثانية، وأكتب كما كتبت في ملف كود الواجهة الأولى لكن بالعكس، كما يأتي:

```
1 package com.muminkilani.myapplication
2
3 import ...
4
5
6
7
8 class Main2Activity : AppCompatActivity() {
9
10 override fun onCreate(savedInstanceState: Bundle?) {
11     super.onCreate(savedInstanceState)
12     setContentView(R.layout.activity_main2)
13     button2.setOnClickListener { it: View!
14         val intent = Intent( packageContext: this, MainActivity::class.java)
15         startActivity(intent)
16     }
17 }
18 }
19 }
20 }
```

المشروع جاهز للتنفيذ، أعمل تشغيلاً للمشروع، وألاحظ النتائج.



إذا أردت في نشاط (1) نقل بيانات بين الواجهات، أضيف planText (editText) في الواجهة الأولى لإدخال بيانات معيّنه، واضيف textView في الواجهة الثانية لإظهار البيانات المرسله من الواجهة الأولى، كما يأتي:



بعد إضافة العناصر أنتقل إلى الكود البرمجي للواجهة الأولى، أضيف السطر الآتي:

```
val intent = Intent( packageContext: this, Main2Activity::class.java)
intent.putExtra( name: "massage", editText.text.toString())
startActivity(intent)
```

```
intent.putExtra("massage",editText.text.toString())
```

intent هذا الكائن الذي عرفته سابقاً من كلاس Intent، من خواصّ intent هي putExtra الدالة التي تقوم بنقل البيانات، للوصول إليها أضع نقطة ثمّ putExtra، تستقبل putExtra قيمتين: الأولى هي اسم البيانات (name)، وهو عبارة عن مفتاح للقيمة الذي سيتم إرساله، أمّا القيمة الثانية فهي البيانات التي أريد نقلها، هنا أريد جلب البيانات المدخلة من editText؛ لذلك استخدم التحويلة toString؛ لأنّ البيانات المنقولة من نوع string.



أمّا في الواجهة الثانية فيكون الكود كالآتي:

```
9
10 override fun onCreate(savedInstanceState: Bundle?) {
11     super.onCreate(savedInstanceState)
12     setContentView(R.layout.activity_main2)
13
14     val bundle= intent.extras
15     textView.text = bundle["message"].toString()
16
17     button2.setOnClickListener { it: View!
18         val intent = Intent( packageContext: this, MainActivity::class.java)
19         startActivity(intent)
20     }
21 }
22
23 }
```

هنا عرّفت متغيّراً من نوع bundle، يقوم باستقبال وتخزين جميع البيانات الممرّرة إلى الواجهة الحالية على شكل المفتاح (الاسم)/البيانات (name/data)، أكتب بعد إشارة المساواة intent.extras؛ ليتم تخزين البيانات في bundle، ولجلب البيانات المراد استخدامها أكتب المفتاح الذي وضعته عند إرسال البيانات، كما في السطر الآتي:

```
textView.text = bundle["message"].toString()
```

أجلب البيانات المرتبطة بالمفتاح ["message"] من bundle، بما أنّ البيانات نوعها string لذلك استخدم (.toString) لتحويلها إلى string، وأخيراً تظهر البيانات في textView.

نشاط عملي (2): أنشئ تطبيقاً ينقل البيانات من الواجهة الأولى إلى الثانية، وإظهارها في textView و

toast، ثمّ نقل بيانات أخرى من الواجهة الثانية إلى الأولى وإظهارها في textView و toast.



1. ما الدالّة المسؤولة عن نقل البيانات بين activity وأخرى؟
2. كيف يتمّ الوصول إلى البيانات؟
3. ما الجملة الخاصّة لبدء عمل Intent؟
4. ما طريقة إنشاء activity جديدة؟



التخاطب بين التطبيقات.

وصف الموقف التعليمي: استعان ليث بمشغل تطبيقات الهاتف المحمول في مدرسة جنين المهنية؛ لمساعدته في إنشاء تطبيقين، يقوم الأول بالتقاط صورة وعرضها على التطبيق، ثم نقلها مع معلومات عنها إلى التطبيق الثاني.

العمل الكامل			
خطوات العمل	وصف الموقف الصّفي	المنهجية (استراتيجية التعلّم)	الموارد
أجمع البيانات وأحلّها	<ul style="list-style-type: none"> • أجمع البيانات من ليث عن: <ul style="list-style-type: none"> - إصدار نظام الأندرويد. - أبعاد الصورة. - المعلومات عن الصورة. • أجمع البيانات عن: <ul style="list-style-type: none"> - العناصر (views) المستخدمة للإدخال والإخراج. - عدد الواجهات. - عدد التطبيقات. - الجمل البرمجية المستخدمة. 	<ul style="list-style-type: none"> • حوار ومناقشة. • التعلّم التعاوني/ مجموعات. • البحث العلمي. 	<ul style="list-style-type: none"> • وثائق (طلب ليث، نشرات، كتب ومصادر متنوعة). • التكنولوجيا: (روابط إلكترونية متنوعة من مصادر موثوقة، هاتف، حاسوب، إنترنت، يعمل بنظام الأندرويد).
أخطط وأقرّر	<ul style="list-style-type: none"> • أصنّف البيانات (نظام الأندرويد المستخدم). • أحدّد: <ul style="list-style-type: none"> - المعطيات: (أبعاد الصورة، معلومات عن الصورة). - العناصر (views) المستخدمة. - عدد الواجهات. - عدد التطبيقات. - الجمل البرمجية. • إعداد خطة للتنفيذ. 	<ul style="list-style-type: none"> • تعلّم تعاوني/ مجموعات. • حوار ومناقشة. 	<ul style="list-style-type: none"> • وثائق (نشرات، كتب ومصادر متنوعة). • التكنولوجيا (روابط إلكترونية متنوعة من مصادر موثوقة).

<ul style="list-style-type: none"> • حاسوب . • برنامج Android Studio . • جهاز افتراضيّ، أو جهاز يعمل بنظام أندرويد . 	<ul style="list-style-type: none"> • تعلّم تعاونيّ/مجموعات . • حوار ومناقشة . • عصف ذهنيّ / استمطار الأفكار . 	<ul style="list-style-type: none"> • اختيار عدد من الواجهات المقترحة قابلة للاستخدام في تطبيق الأندرويد . • أتفق مع ليث على اختيار واجهات التطبيق المناسبة . • أصمّم وأبرمج واجهة التطبيقين الأول والثاني . • أعمل اختباراً للمشروع على جهاز افتراضيّ، أو على جهاز يعمل بنظام أندرويد . 	<p>أنفذ</p>
<ul style="list-style-type: none"> • الخطة المعدة سابقاً . • الطلب الخاصّ بليث . • حاسوب . • برنامج Android Studio . • جهاز افتراضيّ، أو جهاز يعمل بنظام الأندرويد . 	<ul style="list-style-type: none"> • تعلّم تعاونيّ/مجموعات . • حوار ومناقشة . 	<ul style="list-style-type: none"> • أتحقّق من ملاءمة اختيار الواجهة المقترحة للتطبيقين . • أتأكّد من ملاءمة التطبيقين . • أتحقّق من تنفيذ التطبيقين وظهورهما على الجهاز بشكل مناسب . 	<p>أتحقّق</p>
<ul style="list-style-type: none"> • حاسوب . • شاشة عرض . • جهاز افتراضيّ، أو جهاز يعمل بنظام أندرويد . 	<ul style="list-style-type: none"> • حوار ومناقشة . • تعلم تعاونيّ / مجموعات عمل . 	<ul style="list-style-type: none"> • أوّثق البيانات التي تتعلّق بـ: <ul style="list-style-type: none"> - العناصر (views) المستخدمة للإدخال والإخراج . - عدد الواجهات . - عدد التطبيقات - الجمل البرمجية المستخدمة . • إعداد عرض تقديميّ . • إعداد ملف بالحالة (إنشاء يقوم بالتقاط صورة وعرضها على التطبيق) . 	<p>أوثق وأقدم</p>
<ul style="list-style-type: none"> • طلب ليث، وثائق نشرات، كتب ومصادر، وروابط إلكترونية متنوعة من مصادر موثوقة عن كيفية تطوير تطبيق أندرويد، تختص بالتقاط الصور وعرضها ثم نقلها مع معلومات عنها إلى التطبيق الثاني . • نشرات عن المعايير والمواصفات الخاصة بتطوير تطبيق أندرويد خاص بالتنقل بين التطبيقات . 	<ul style="list-style-type: none"> • حوار ومناقشة . 	<ul style="list-style-type: none"> • رضا ليث عن التطبيقين . • ملاءمة التطبيقين للمعايير والمواصفات . 	<p>أقوم</p>

أناقش: أستطيع الوصول إلى أي تطبيق موجود على جهاز الأندرويد .



أُتعلَّم: التَّنقل بين التَّطبيقات



التَّنقل بين تَطبيقات منشآت على الجهاز نفسه:

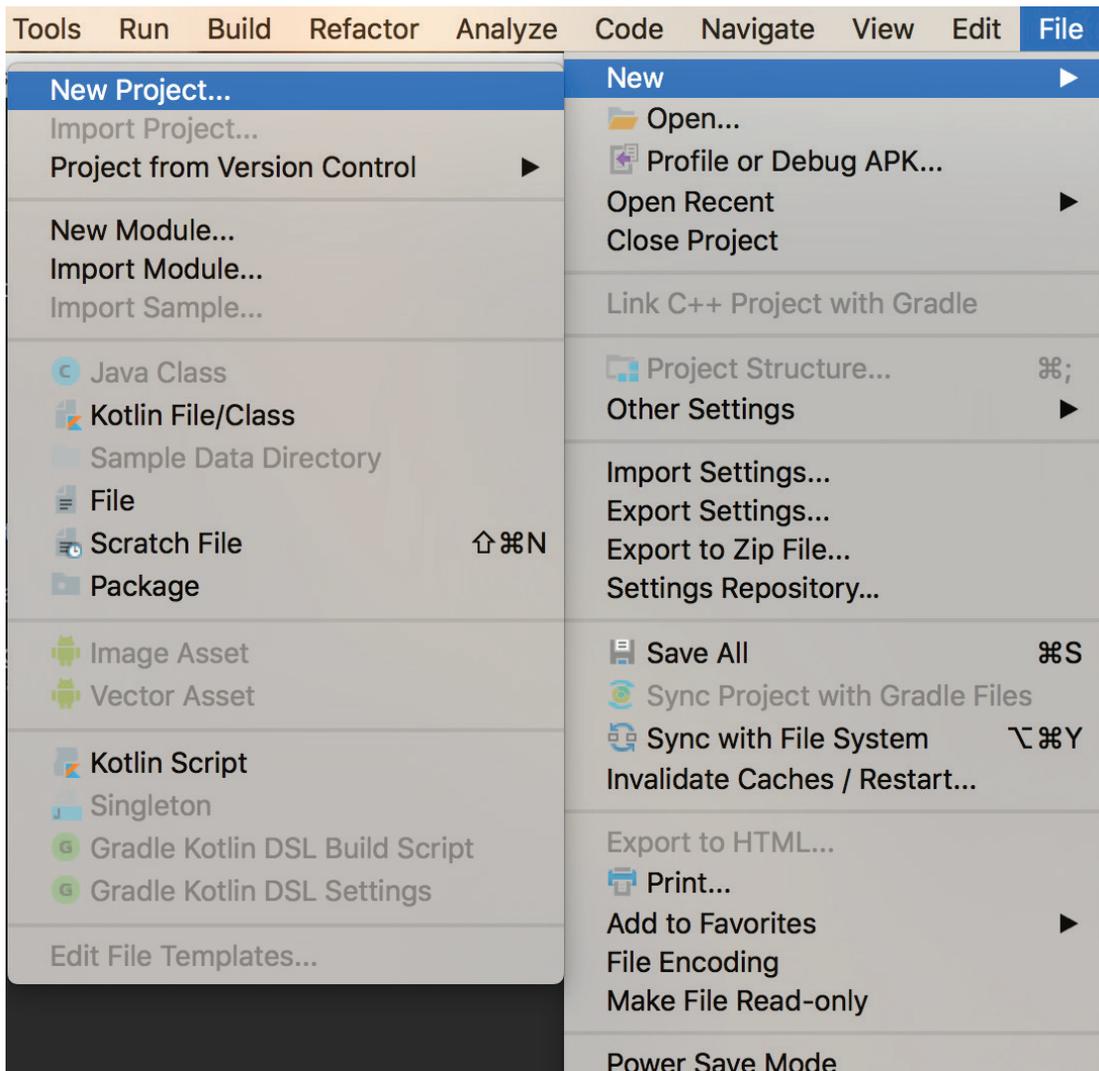


نشاط عملي (1): أنشئ تطبيقين يرسل كل منهما بيانات للآخر ويعرضها.



أنشئ المشروع الأول وأسمِّه App1، ثمَّ أنتظر حتى يفتح البرنامج بشكل كامل، ثمَّ أنشئ المشروع الثاني وأسمِّه App2 من خلال:

File->New->New Project



بعد أن يفتح البرنامج بشكل كامل، يصبح لديّ مشروعان قيّد التشغيل في الأندرويد ستوديو.



أصمّم واجهة التطبيقين الأول والثاني، كما في الصورة الآتية:



استخدمتُ `textView` لإظهار البيانات المُستقبلة من التطبيق الآخر، و `planText` لإدخال البيانات التي أريد إرسالها، أمّا `button` فيُستخدم لإرسال البيانات والانتقال إلى التطبيق الآخر.

أفتحُ الكود البرمجيّ الخاصّ بواجهة التطبيق الأول، ثمّ أكتب كما يأتي:

```
6
7 class MainActivity : AppCompatActivity() {
8
9     override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11         setContentView(R.layout.activity_main)
12
13         val b=intent.extras
14         if (b!=null){
15             resText.text = b["app2"].toString()
16         }
17
18         sendButton.setOnClickListener{ it: View!
19             val message = sendText.text.toString()
20             val launchIntent = packageManager.getLaunchIntentForPackage("com.muminkilani.app2")
21             launchIntent.putExtra( name: "app1",message)
22             startActivity(launchIntent)
23         }
24     }
25 }
26 }
```

في الكود السابق، أكتبُ الأوامر في داخل `sendButton`، أعرفُ متغيّراً لتخزين البيانات المُدخلة في `planText` بعد جلبها من خلال هذا السطر:

```
val message = sendText.text.toString()
```

ثمّ أنشئُ كائناً من صنف `Intent`، وأسمّيه `launchIntent`.

packageManager هي دالة رئيسية تتعامل مع إدارة حزمة التطبيق، واستخدامها هنا لكي أستطيع الوصول إلى دالة getLaunchIntentForPackage التي تحتوي على اسم حزمة التطبيق الذي أريد الانتقال إليه، ووظيفة هذه الدالة هي الانتقال من تطبيق إلى آخر.

أناقش: فائدة السطرين البرمجيّين الآتيين:



```
launchIntent.putExtra("app1",message)
startActivity(launchIntent)
```

أكتب الكود البرمجيّ الذي يستقبل البيانات من التطبيق الآخر ويظهرها في textView كما يأتي:

```
val b=intent.extras
if (b!=null){
    resText.text = b["app2"].toString()
}
```

أناقش: فائدة الجملة الشرطيّة في الكود السابق.



أنشئ كائناً من الصنف bundle، وأسميه b يقوم بخزن جميع البيانات التي تصل إلى activity الحالية كما تعلمت سابقاً، ثم أكتب جملة شرطيّة تقوم بفحص وجود بيانات داخل المتغيّر b أم لا، فإن وجد بيانات يظهرها على textView كما يأتي:

```
resText.text = b["app2"].toString()
```

أناقش: عمل السطر البرمجيّ السابق.



بعد الانتهاء أشغّل المشروع الأول، ثمّ بعد ذلك أنتقل إلى المشروع الثاني وأكتب الكود البرمجيّ نفسه، مع إجراء بعض التغييرات البسيطة كما يأتي:

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

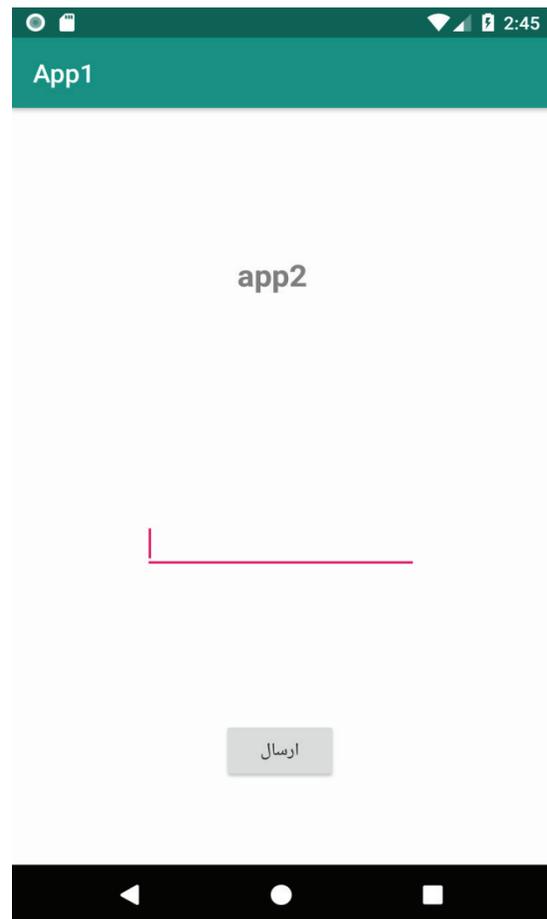
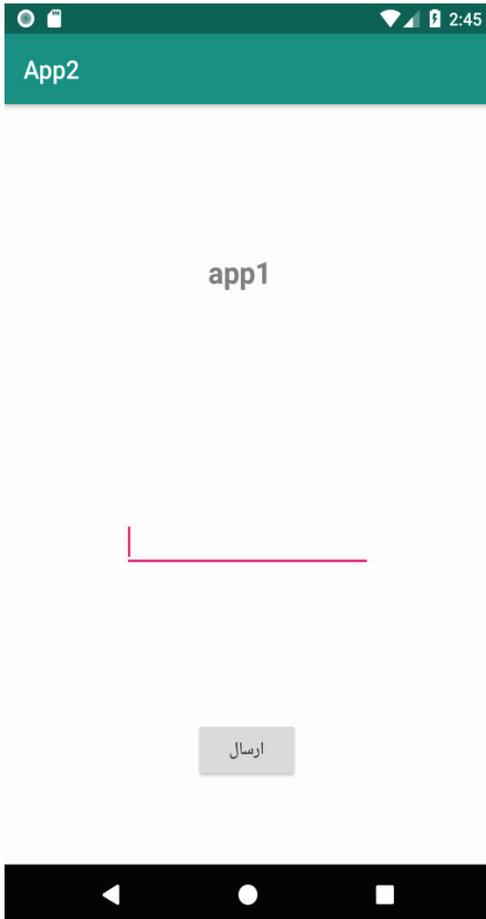
    val b=intent.extras
    if (b!=null){
        resText.text = b["app1"].toString()
    }

    sendButton.setOnClickListener{ it: View!
        val message = sendText.text.toString()
        val launchIntent = packageManager.getLaunchIntentForPackage("com.muminkilani.app1")
        launchIntent.putExtra( name: "app2",message)
        startActivity(launchIntent)
    }
}
```

نشاط نظري (2): أناقش مع زملائي عمل الكود البرمجيّ السابق، وأكتشف الاختلاف عن الكود البرمجيّ لمشروع التطبيق الأول.



أخيراً أشغّل المشروع الثاني على جهاز الأندرويد نفسه، ثمّ أختبر عمل التطبيقين.





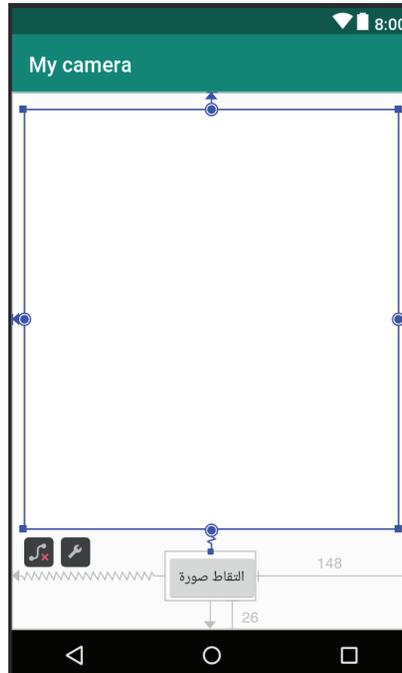
التنقل بين تطبيق تم أنشاؤه واستخدام الكاميرا:



نشاط عملي (3): أنشئ تطبيقاً يلتقط صورة ويظهرها داخل التطبيق.



أنشئ مشروعاً جديداً وأسميه Mycamera، وبعد أن يفتح البرنامج بشكل كامل أبدأ بتصميم الواجهة بإضافة imageView لعرض الصورة الملتقطة وأعطيتها id، و button لفتح الكاميرا والتقاط الصورة وأعطيه id:



أفتح الكود البرمجي، وأبدأ بكتابة الأكواد كما يأتي:

```
private val REQUEST_IMAGE = 1

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

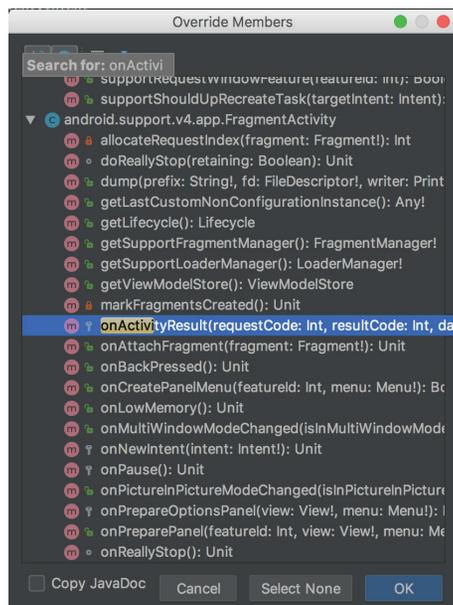
    button.setOnClickListener { it: View!
        val intent = Intent(MediaStore.ACTION_IMAGE_CAPTURE)
        startActivityForResult(intent, REQUEST_IMAGE)
    }

    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
        if (requestCode == REQUEST_IMAGE && resultCode == Activity.RESULT_OK) {
            val extras = data!!.extras
            val photo = extras["data"] as Bitmap
            imageView.setImageBitmap(photo)
        }
    }
}
```

أبدأ أولاً بالكتابة داخل button، أعرف intent وأعطيتها أمراً لفتح الكاميرا وهو: `MediaStore.ACTION_IMAGE_CAPTURE`. لبدء عمل intent الخاصة أكتب:

`startActivityForResult(intent, REQUEST_IMAGE)`

وتأخذ قيمتين: الأولى هي intent التي عرفتها، والثانية هي رقم activity التي أريد الانتقال إليها، وهنا عرفت قيمة ثابتة وأسميتها `REQUEST_IMAGE`، وأعطيتها قيمة 1، وأستطيع أن أعطيها قيمة أخرى بشرط أن تكون موجبة، وقد عرفت القيمة في البداية لكي أستخدمها في أكثر من مكان في الكود. لكي يتم استقبال الصورة أعمل استدعاء للدالة اسمها `onActivityResult` الخاصة بفحص التغييرات التي تحصل على الواجهة (activity) الحالية لاستقبال بيانات من وجهات (activities) أخرى، كما يأتي:



أحذف الكود الموجود داخل الدالة، وأكتب الكود البرمجي الآتي:

```
if (requestCode == REQUEST_IMAGE && resultCode == Activity.RESULT_OK) {  
    val extras = data!!.extras  
    val photo = extras["data"] as Bitmap  
    imageView.setImageBitmap(photo)  
}
```

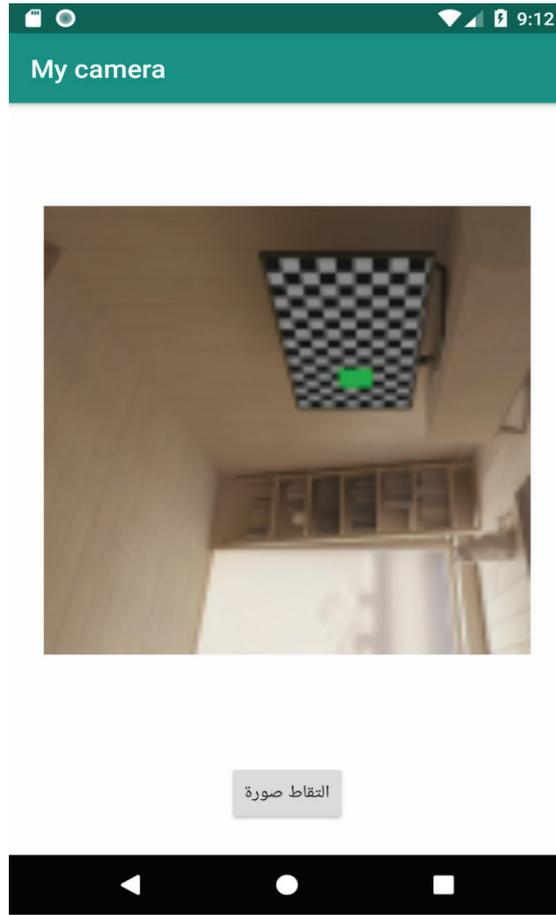
- أكتب جملة شرطية بداخلها `requestCode == REQUEST_IMAGE` تفحص القيمة المطلوبه هل هي مساوية لرقم `activity` التي أريد جلب البيانات منها أم لا.
 - أيضاً `resultCode == Activity.RESULT_OK` تفحص وجود نتائج في الواجهة؛ أي هل يوجد قيم تم استقبالها في الواجهة (هنا القيم هي صور)، وهذا ما تعنيه هذه الجملة `Activity.RESULT_OK`؛ أي أنه يوجد نتائج.
 - داخل الجملة الشرطية أنشئ كائناً من الصنف `Intent` أسميه `extras` لتخزين البيانات التي تم استقبالها. كلمة `data` هي متغير من نوع `Intent` معرف في الدالة الرئيسية `.onActivityResult`.
 - ثم أعرف متغيراً أسميه `photo`، وأخزن القيم الموجودة في `extras` التي تحمل مفتاحاً اسمه `"data"`.
 - `as Bitmap` وتعني أن البيانات هي من نوع `bitmap`.
 - أخيراً تم تخزين الصورة التي تم التقاطها من الكاميرا في `imageView` التي أضفتها في الواجهة باستخدام:
`imageView.setImageBitmap(photo)`
- بعد الانتهاء من كتابة الكود، وبما أن الكاميرا مكون مادي وخارجة عن إدارة التطبيق، يلزم إضافة طلب السماح للوصول، واستخدام الكاميرا في ملف `manifest`، وبدونه لا يعمل التطبيق ويعطي خطأ. كما يأتي:

```
<uses-feature android:name="android.hardware.camera" android:required="true"/>
```

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    package="com.muminkilani.mycamera">  
      
    <uses-feature android:name="android.hardware.camera" android:required="true"/>  
      
    <application  
        android:allowBackup="true"  
        android:icon="@mipmap/ic_launcher"  
        android:label="My camera"  
        android:roundIcon="@mipmap/ic_launcher_round"  
        android:supportRtl="true"  
        android:theme="@style/AppTheme"  
        tools:ignore="GoogleAppIndexingWarning">  
        <activity android:name=".MainActivity">  
            <intent-filter>  
                <action android:name="android.intent.action.MAIN" />  
                <category android:name="android.intent.category.LAUNCHER" />  
            </intent-filter>  
        </activity>  
    </application>  
</manifest>
```



أشغل المشروع، ثم اختبر التطبيق وأفحص النتائج.



1. ما وظيفة الدالة الآتية: `getLaunchIntentForPackage`؟
2. ما هو طلب السماح للوصول واستخدام الكاميرا الذي يُضاف في ملف `manifest`؟



أسئلة الوحدة

السؤال الأول: أضع دائرة حول رمز الإجابة الصحيحة فيما يأتي:

1. أي من الآتية لا يُعدّ من دورة حياة الواجهة activity؟

أ. onCreate . ب. onTouchEvent . ج. onPause . د. onStart .

2. ماذا تفيد كلمة lateinit؟

أ. إعادة تعريف المتغيّر في وقت لاحق .

ب. استخدام المتغيّر في أكثر من موضع .
د. لا يمكن استخدام المتغيّر إلا مرّة واحدة فقط .

لا يوجد لها فائدة .

3. ما وظيفة show . في كتابة toast؟

أ. تُكمل تركيب toast .

ب. إخفاء toast .
د. تُستخدم توقيتاً زمنياً ل toast .

ج. عرض toast .

4. ما الدالة التي تُستخدم لربط الكود البرمجيّ بملف xml الخاصّ بالقوائم؟

أ. onCreateMenu .

ب. onSelectItem .
د. onCreateOptionsMenu .

ج. onOptionsItemSelected .

5. ما الأمر الذي أكتبه لكي يبدأ عمل intent؟

أ. show(intent) .

ب. start(intent) .
د. shoeActivity(intent) .

ج. startActivity(intent) .

6. ماذا أستخدم لنقل بيانات من واجهة إلى أخرى؟

أ. intent.extras .

ب. intent.putExtra .
د. intent.putString .

ج. intent.getExsta .

7. ما وظيفة bundle في الواجهة activity التي تستقبل البيانات؟

أ. تخزين البيانات المرسلّة من واجهة أخرى .

ب. يخزّن فقط البيانات التي تحمل مفتاحاً محدّداً .
د. يتعامل مع البيانات من نوع string فقط .

ج. يُستخدم وسيلةً لنقل البيانات .



السؤال الثاني: ما وظائف intent؟

السؤال الثالث: أوضّح باختصار دورة حياة الواجهة activity.

السؤال الرابع: ما المقصود بالإيماءات؟ أذكر ثلاثة من أنواعها.

السؤال الخامس: ما مكونات toast؟

نشاط الوحدة: أنشي تطبيقاً من عدّة واجهات يعرض سيرتي الذاتية، بحيث تحتوي الواجهة الأولى على اسم مستخدم وكلمة مرور، ولا يمكن فتح التطبيق إلاّ بواسطتهما.



■ لجنة المناهج الوزارية:

د. بصري صيدم
د. وسام نخلة
د. بصري صالح
د. سمية النخالة
أ. ثروت زيد

■ المشاركون في ورشات تطبيقات الهواتف الذكية للصف الحادي عشر:

محمد الجوراني
سماح بعلوشة
جمال جودة
محمد قباجة
عاصم شراب

تمّ بحمد الله وتوفيقه

