الجمهورية العربية السورية

وزارة التربية

المركز الوطني للمتميزين

**N**

**C**

D



2014 / 2015

تطبيق أندرويد خاص بالمركز الوطني للمتميزين



**Android Application special for NCD**

مشروع علمي لطلاب المركز الوطني للمتميزين في مادة المعلوماتية بعنوان

تطبيق أندرويد خاص بالمركز الوطني للمتميزين

**للعام الدراسي 2014– 2015**

**تقديم الطلاب:**

أحمد الصالح محمد زياد قباقيبي

محمد بجعه محسن الخيِّر

**بإشراف المدرّسة:** ميس درويش

**..:::الفهــــرس :::...**

|  |  |
| --- | --- |
| المــــوضــــوع | الصفحــة |
| [مخطط سير المشروع](#a1) ..................... [قسم نظري](#a2)  ..................... [قسم عملي](#يلاي) | **6**  **7** |
| [المقدمة](#a4) .............. | **8** |
| [أهمية المشروع](#أهمية) – [أهداف المشروع](#أهداف) .............. | **9** |
| [طبقات وبيئات نظام التشغيل Android](#a8)  .............. | **12** |
| [لمحة عن البرمجة كائنية التوجٌّه](#ثاني) .............. | **14** |
| [لمحة عن لغة البرمجة جافا](#ثالث) .............. | **21** |
| [انتشار نظام التشغيل Android وأهم ما يميزه](#a12) | **29** |
| [بعض أنواع الأجهزة التي تعمل باستخدام نظام التشغيل أندرويد](#ثث) | **31** |
| [إصدارات نظام التشغيل أندرويد](#ثال) | **32** |
| [نبذة عن قواعد البيانات](#فصل1) | **36** |
| [تعليمات نظام قواعد إدارة المعطيات SQL](#فصل2) | **39** |
| [الجداول والاستعلامات والعمليات عليها](#فصل3) | **45** |

|  |  |
| --- | --- |
| [القسم العملي](#القسمالعملي) | 58 |
| [النتائج](#النتائج) | 63 |
| [المقترحات](#المقترحات) | 63 |
| [الخاتمة](#الخاتمة) | 64 |
| [الإهداء](#إهداء) | 65 |
| [المصادر والمراجع](#المصادروالمراجع) | 66 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| فهرس الأشـــــــكال | | | | | | |
| (1) | يوضح طبقات نظام التشغيل Android | 13 | **(2)** | مثال الطُّلَّاب | 16 | |
| (3) | المصباح الكهربائي | 20 | **(4)** | الكلمات المحجوزة في اللغة(java) | 23 | |
| (5) | سير عمل بعض التعليمات بلغة الجافا | 26 | **(6)** | الرموز التي تظهر في مربع الحوار | 27 | |
| (7) | تزايد استخدام الأندرويد بالنسبة لباقي أنظمة تشغيل الهواتف المحمولة | 29 | **(8)** | تعداد مستخدمي الأندرويد | 30 | |
| (9) | الساعات الذكية | 31 | **(10)** | مدى انتشار كل نسخة من نسخ Android | 34 | |
| (11) | الجدول في قاعدة البيانات | 37 | **(12)** | جملة Select | 40 | |
| (13) | الاختيار الشرطي | 41 | **(14)** | عوامل المقارنة نستطيع استخدامها مع Where | 42 | |
| (15) | المعاملات في SQL | 42 | (16) | المعاملات المنطقية في SQL | 43 | |
| (17) | قواعد أسبقية المعاملات | 44 | (18) | عوامل المقارنة | 47 | |
| (19) | المعاملات التي يمكننا استخدامها لتقييم الاستعلامات متعددة السجلات | 48 | (20) | أهم أنواع البيانات التي يمكن استخدامها في إنشاء الجداول | 54 | |

|  |  |
| --- | --- |
|  |  |
|  |
|  |

**(مخطط سير المشروع)**

|  |  |
| --- | --- |
| الأبواب | الفصول |
| القســــــــــــم النظـــري | |
| الباب الأول  (تركيبة نظام التشغيل Android ولغة البرمجة جافا) | فصل أول:  طبقات وبيئات نظام التشغيل Android |
| فصل ثان:  لمحة عن البرمجة كائنية التوجه OOP |
| فصل ثالث:  لمحة عن لغة البرمجة JAVA |
| الباب الثاني  (معلومات عن نظام تشغيل Android) | فصل أول:  انتشار نظام التشغيل Android وأهم ما يميزه |
| فصل ثان:  بعض أنواع الأجهزة التي تعمل باستخدام نظام التشغيل Android |
| فصل ثالث:  إصدارات نظام التشغيل Android |
|  |  |

|  |  |  |
| --- | --- | --- |
| الباب الثالث  (قواعد البيانات SQL) | فصل أول:  نبذة عن قواعد البيانات | |
| فصل ثان:  تعليمات نظام إدارة قواعد المعطيات SQL | |
| فصل ثالث:  الجداول والإستعلامات والعمليات عليهما | |
|  | |
|  | |
| |  | | --- | | القســــــــــــم العمـلي | | | |
| الباب الأول:  التعليمات والواجهات في (Android Studio) | فصل أول: الواجهات و الأدوات | |
| فصل ثان: التعليمات في Android Studio | |
|  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |
|  |  | المقدمة  لقد كنا طلاباً في مرحلة التعليم الأساسي ثم قررنا الذهاب إلى المركز الوطني للمتميزين وكل أحد منا جاء ليحقق هدفاً معينا ليثبت نفسه ويبني وطنه الأم سورية، لكن من أين سمعنا بهذا المركز؟  سيجيب البعض بأنه أراد أن يجرب شيئاً جديداً فقدم والبعض الآخر من باب أن أهله أرادوا ذلك و.... إلخ.  ولكن هناك البعض من المبدعين والمتميزين خارجاً لم يقدموا إلى هذا المركز ليس لأنهم فشلوا باختبار القبول بل لأن الفرصة لم تتسنى لهم ليعرفوا بشيء يدعى المركز الوطني للمتميزين أو أنهم أخذوا عنه فكرة خاطئة.  لذلك قررنا من خلال مشروعنا هذا أن نعرف الطلاب والعالم الخارجي في وطننا بهذا المركز من خلال عدة نواحي كأهداف هذا المركز وإطلاع الآخرين على الخدمات التي يرعانا بها المركز وقد سعينا لأن نجعل هذا التطبيق منتشراً من خلال عدة وسائل وأهمها بألّا نجعله باللغة العربية فقط واخترنا نظام تشغيل Android ليعمل عليه التطبيق كونه الأكثر شيوعاً وانتشاراً في هذا الوقت ونأمل بأن يتعرف أكبر عدد ممكن من الناس ليس فقط في بلدنا بهذا المركز وما يقدمه لنا من خدمات تربوية وحياتية ليعرف العالم بهذا المشروع الذي يسعى في النهاية لأن نسمو بوطننا الغالي ونبنيه بأيدينا.... |  |

**إشكالية البحث:**

تأتي إشكالية البحث من أهمية نظام التشغيل Android وما حققه من انتشار واسع وخصوصاً في شريحة الشباب ولعدم وجود تطبيق تعريفي خاص بالمركز الوطني للمتميزين يكون رديفاً للأساليب التعريفية الأخرى.

**أهمية المشروع:**

إنشاء تطبيق تعريفي خاص بالمركز يقوم بالتعريف بأقسام المركز و طبيعة الدراسة فيه والهدف من إنشائه وتسهيل التواصل معه كون نظام Android أكثر أنظمة التشغيل انتشاراً ومقدرته الكبيرة على التعريف بالمركز بالطريقة المثلى بالإضافة إلى وسائل التعريف الأخرى.

***أعضاء المشروع***

**أهداف المشروع:**

* **من خلال الجانب النظريّ:**
* . إنشاء وسيلة جديدة للتعريف عن المركز الوطني للمتميزين
* . نقل صورة المركز إلى مجالات ونطاقات أوسع لتعزيز فرصة جلب العقول الأكثر كفاءة وقدرة على مواكبة الدراسة النَّموذجيَّة.
* **كما هدفنا من خلال الجانب العمليّ إلى:**
* . تحويل فكرة التطبيق إلى مثال واقعي يمكن تطويره.
* . اكتساب مهارة في التعامل مع بيئة التطوير Android ولغة البرمجة .Java

**القسم النظري**

**البــــــــــــــــــــــــــــــــــــــــــــاب الأول:**

لمحة عن لغة البرمجة جافا

تركيبة نظام التشغيل أندرويد

ولغة البرمجة جافا

طبقات وبيئات تطوير نظام التشغيل أندرويد

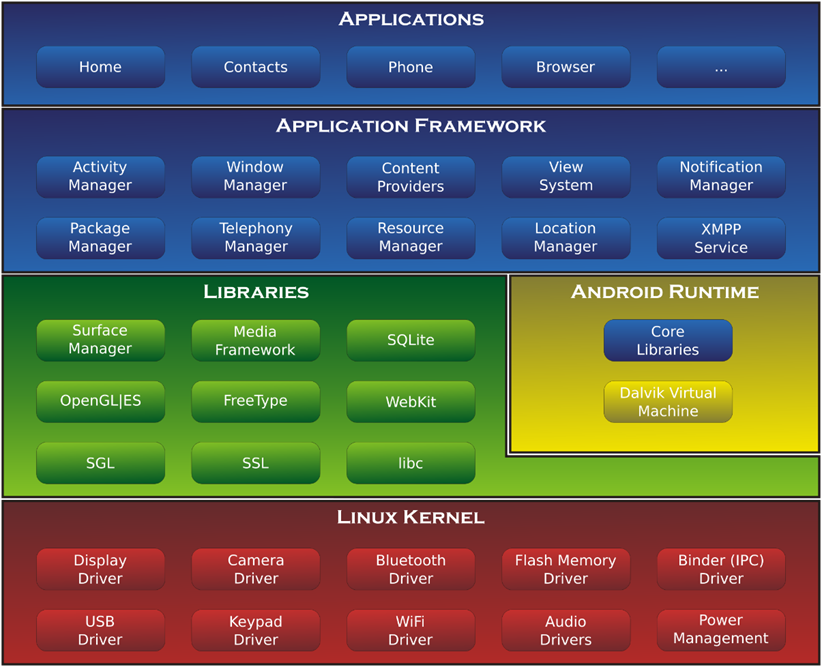
لمحة عن البرمجة كائنيَّة التوجُّه



فصل أول: طبقات وبيئات تطوير نظام التشغيل أندرويد

يتألف نظام الأندرويد برمجياً من عدة طبقات من حيث الوظائف وفي الصورة شكل يوضح بنية النظام من حيث الوظائف:

1. نواة اللينوكس: هي النواة المبني النظام عليها وتضمن الـ Drivers للمكونات المادية ليعمل النظام بشكل صحيح على الجهاز المنزل النظام عليه.
2. المكتبات: تتضمن الأكواد التي تقدم الخصائص الرئيسية للنظام مثلاً مكتبة الـ SQLite تقدم الدعم للتطبيقات حيث تقدم لها قاعدة البيانات التي يحتاجها التطبيق ليخزن عليها بياناته، أداة WEB Kit تقدم الإضافات والتوابع التي تمكن من الوصول للإنترنت.
3. Android Runtime: في نفس طبقة المكتبات وتقدم مجموعة من المكتبات المركزية التي تمكن المطورين من كتابة تطبيقات بلغة الجافا وتضمن أيضاً آلة المحاكاة البرمجية Dalvik (التي تتيح لكل تطبيق أندرويد بأن يعمل على حده ويؤدي وظائفه) هي التي تحوي المترجم compiler الذي يشغل تطبيقات الأندرويد حيث أنها مصممة خصيصاً للأندرويد على الأجهزة التي تعمل بالبطارية مع ذاكرة مستقلة ومعالج.
4. التطبيقات الهيكلية: تعطي القدرات المتعددة لنظام الأندرويد للتعامل مع التطبيقات ليستطيع المطورون الاستفادة منها باستخداماتها في تطبيقاتهم التي يقومون تطويرها
5. التطبيقات: وهي الطبقة السطحية نجد فيها التطبيقات الرئيسية المتعلقة بالجهاز كالهاتف والأسماء والمتصفح و.إلخ ، كما نجد أيضاً التطبيقات التي نحملها وننصبها من متجر الأندرويد وأي تطبيق تتم برمجته فمكانه هو هذه الطبقة.



الشكل رقم (1) يوضح طبقات نظام التشغيل Android

وتتم برمجة تطبيقات الأندرويد بلغة الجافا عبر بيئة التطوير Eclipse أو Android Studio

بالإضافة إلى حزمة الـ Android SDK التي تحوي أدوات لتطوير التطبيقات.

SDK (Software Development Kit)

الحزمة التطويرية البرمجية والتي تحوي المجموعات/الصفوف والدوال التي يمكن استخدامها في تطوير تطبيقات الأندرويد والحزم لضمان عمل التطبيق على نسخ النظام المختلفة.

Android Studio :هي منصة لكتابة [التطبيقات](http://ar.wikipedia.org/wiki/%D8%AA%D8%B7%D8%A8%D9%8A%D9%82) تسهل على المطورين كتابة [الشيفرة المصدرية](http://ar.wikipedia.org/wiki/%D8%A7%D9%84%D8%B4%D9%8A%D9%81%D8%B1%D8%A9_%D8%A7%D9%84%D9%85%D8%B5%D8%AF%D8%B1%D9%8A%D8%A9) لتطبيقات [أندرويد](http://ar.wikipedia.org/wiki/%D8%A3%D9%86%D8%AF%D8%B1%D9%88%D9%8A%D8%AF)، كما تسمح للمطور بمعاينة هيئة تطبيقه على مختلف قياسات [الشاشات](http://ar.wikipedia.org/wiki/%D8%A7%D9%84%D8%B4%D8%A7%D8%B4%D8%A7%D8%AA) بشكل فوري أثناء التطوير، وتسهّل تطوير التطبيقات متعددة [اللغات](http://ar.wikipedia.org/wiki/%D8%A7%D9%84%D9%84%D8%BA%D8%A7%D8%AA).

ميزات Android Studio :

1. يعمل على عدة منصات. ([Windows](http://ar.wikipedia.org/wiki/Windows), [Mac OS X](http://ar.wikipedia.org/wiki/Mac_OS_X), [Linux](http://ar.wikipedia.org/wiki/Linux))
2. سهولة في الاستخدام.
3. يعطي معاينة فورية لمختلف أنواع الأجهزة اللوحية أو المحمولة.
4. مخصص للبرمجة لمنصة أندرويد.

فصل ثاني: لمحة عن البرمجة كائنيَّة التوجُّه(OOP)

Object Oriented Programming)):

قبل البداية في شرح البرمجة كائنية التوجه سنحاول التطرُّق إلى نوعين كانا سائدين قبلها وهما:

* البرمجة الإجرائيَّة: وهي أوَّل تقنية كان من المفترض أن تظهر وقد ظهرت بالفعل، وهي تركِّز على إجراء البرنامج في خطوات واضحة ومحددة لا تحيد عنها وهي عبارة عن كتلة واحدة ، وعلى الرغم من تقديمها الكثير للمبرمجين إلا أنَّ الخوارزميَّات تصبح أكثرَ تعقيداً عندما يتعامل المبرمج مع مشاريع كبيرة الحجم. أيضاً لم يكن بإمكان الأشخاص العمل كفريق عمل لأنه لا يوجد تقسيم واضح في الخوارزميَّة، فهو عبارة عن كتلة واحدة، أيضاً عند القيام بصيانة البرنامج فإن الأمر يصبح أكثر تعقيداً وخاصةً عند تتبّع سير البرنامج لمعرفة مكان مما جعل المبرمجين يشبهون هذا النوع بأنه "معكرونة".
* البرمجة الهيكليّة :أتت البرمجة الهيكليّة لحل المشاكل التي تعاني منها البرمجة الإجرائيَّة إلا أنها لم تقدِّم الكثير؛ سوى تقسم الخوارزميَّة إلى العديد من الإجراءات بالإضافة لعدم تمكن المستخدم من إعادة استخدام هذه الإجراءات في بعض الحالات في برامج أخرى وفي بعض اللغات التي تملك ميزة المرجعيَّات والمؤشرات إذ لا يمكن للإجراء أن يعيد سوى قيمة واحدة، بالإضافة لهذا لا يمكن استخدام المتغيرات العامة بكثرة فهي تجعل عمليَّة تتبّع سير البرنامج عمليَّة مستحيلة، وبسبب أنَّ بعض الإجراءات تتطلَّب تعتمد على وجود هذه المتغيِّرات العامَّة في البرنامج فلن نستطيع استخدام هذا الإجراء في برامج أخرى لأنَّ هذا الإجراء ليس مستقلَّاً كما يخيَّل للبعض، من أجل كلِّ هذه العيوب والنواقص في البرمجة الهيكلية والإجرائيَّة ظهرت البرمجة كائنيَّة التوجُّه والتي لم ترتكز إلا على مفهوم البرمجة. [[1]](#footnote-1)
* البرمجة كائنيَّة التوجّه OOP: ترتكز البرمجة كائنيَّة التوجّه في وجودها ليس على الإجراءات والكائنات؛ فالأصناف (الصفوف classes) هي الوحدة الأساسيَّة في هذه البرمجة.
* تتألف الأصناف من متغيِّرات ودوال، وبمعنى برمجي كائني التوجّه بشكل بحت (دون التدخّل في لغات البرمجة)

فالصنف يتألف من شيئين اثنين هما الخواص والسّلوك ومن إحدى أهم اللغات كائنية التوجه هي لغة الجافا والتي هي عماد تطبيقنا.

تعريف الصِّنف: عبارة عن قالب يعرِّف مجموعة من الخواص والسُّلوك كما هي موجودة في العالم الحقيقي. ومثالاً على الأصناف فقد اقتبسنا من كتاب الإكسير في سي++ ما يلي: لو أردنا عمل برنامج لتسجيل الطلَّاب في الجامعة فمن الممكن أن نقسِّم البرنامج إلى عدة أصناف وهي صنف الطَّالب؛ صنف الكلِّيَّة أو القسم و صنف مسجِّل الطُّلَّاب(عمادة القبول والتسجيل)، وسنأخذ مثال الطَّالب أوَّلاً: يتألَّف صنف الطَّالب من متغيرات و دوال، من أمثلة الدوال لدى الطالب دالة اختيار الكلية المرغوب بها أما بالنسبة لصنف الكلية أو القسم فمن أهم المتغيرات لديه هي اسم الكلية واسم التخصص و الدرجة التي يقبل على أساسها الطالب أما بالنسبة للدوال فمن أهمها دالة القبول المبدئي (والتي تتأكد من توافق شروط القبول مع معايير الكلية وبالتالي تقبل الطالب)؛ وبإمكاننا هنا وضع متغير آخر ألا وهو مصفوفة الطلاب المقبولين قبولاً مبدئياً و متغير آخر هو مصفوفة الطلاب المقبولين قبولاً نهائياً ولن نتعرض هنا على دوال أخرى مثل دالة الفصل النهائي لأننا نكتب هنا برنامج لتسجيل الطلاب بالنسبة للصنف، بالنسبة للصنف الأخير وهو صنف مسجل الطلاب أو بالمعنى الأصح عمادة القبول والتسجيل؛ صنف عمادة القبول والتسجيل يتألف من هذه المتغيرات: مصفوفة الطلاب الراغبين بدخول الجامعة وتحوي أيضاً قائمةً بالتخصُّصات المرغوبة؛ ومصفوفة أيضاً تحوي أسماء الكليات وأقسامها ومن الدوال دالة تقوم بتسجيل الطلاب في قوائم القبول المبدئي (تتأكد من توافق الشروط العامّة للجامعة مع الطلاب) ودالة أخرى تقوم بإرسال اسم الطالب ودرجته إلى الكلية ودالَّة ثالثة تستقبل أسماء الطلاب المقبولين قبولاً نهائيّاً في الكلِّيات وبالتالي فبإمكاننا تمثيل هذه الأصناف في الأشكال في الصفحة التالية :



شكل رقم (2) يوضِّح مثال الطُّلَّاب

كما نلاحظ فقد أتممنا تصميم البرنامج في دقائق قليلة ولم نحتج فقط إلا للقليل من التركيز؛ أما لو حاولنا أن نكتب هكذا برنامج عن طريق البرمجة الهيكلية فسوف يكون معقّداً لدرجة غير معقولة، فقد أتت فكرة الكائنات لتبسيط المسائل المعقّدة وجعلها تبدو بسيطة ولتجعل صيانة البرنامج أكثر تقدماً وسرعة.

الفرق بين الكائن والصنف هو أنّ المستخدم عندما يُعلِم المترجم أن هناك صنفاً جديداً فلن يحجز له أيَّ ذاكرة فقط يكتفي بإعلامه أما عندما ينشئ كائناً جديداً فالمترجم يحجز له مكاناً في الذاكرة مثلاً:

فنحن نعلم المترجم أن هناك كائناً x من الصنف int قيمته هي "!"Hello World فيحجز له المترجم في الذاكرة. فالصنف هو Int والكائن هو x.

مبادئ البرمجة الكائنية:

1-التجريد: لتستفيد أقصى استفادة من ميزات الوراثة فعليك استعمال هذا المبدأ دائماً في جميع أصنافك، فهذا المبدأ يحفظ لك الوقت والجهد، ويزيد من الإنتاجية ومن ميزة إعادة الاستخدام.

لنفرض أنه طلب منك إنشاء برنامج تسجيل الطلاب في الجامعة هذا الشَّهر وفي الشهر القادم ستقوم بإنشاء برنامج إدارة الموظفين لشركة ما. بالطبع فإن أوَّل ما تفكِّر فيه هو أنَّك ستقوم بإنشاء البرنامجين كلٌّ على حدة ولكن الوراثة مع مبدأ التجريد هي التي تعطيك إمكانية الاستفادة من مزايا البرمجة الكائنيَّة، فبدلاً من أن تقوم بإنشاء صنفٍ للطالب وصنف للموظفين في تلك الشَّركة... لم لا تقوم بتجريد هذين الكائنين وتنظر إليهما ليس على أساس أنهما موظفين أو طلاب بل على أساس أنهم أشخاص؛ وبالتالي تقوم بإنشاء صنف اسمه شخص ثم تأتي بعد ذلك إلى الصِّنفين الطالب والموظَّف وتقوم بتوريثهما صنف الشَّخص. وليس ذلك فحسب بل تقوم بإنشاء صف اسمه الطَّالب الجامعي وتورثه صفات الأب (الصنف الطَّالب). أيضاً حينما تقوم بإنشاء صنفِ تسجيل الطلاب وصنف تسجيل الموظفين فلماذا لا تقوم بتجريدهما والنَّظر على أنَّه صنف تسجيل الأشخاص مثلاً... ولنفرض أنَّه بعد مدَّةٍ معيَّنة طلب منك إنشاء صنف لتسجيل الطُّلَّاب العسكريِّين فحينها لن تقوم بالإعادة من الصِّفر، وإنَّما كلُّ ما عليك هو الرُّجوع إلى مكتبة الأصناف التي أنشأتها مسبقاً وتقوم بتوريث صنف الطلاب صفة (الطالب العسكري) في صنفٍ جديد. مبدأ التجريد واسعٌ جدَّاً ومفيدٌ جدَّاً فهو مساعدٌ أوَّل في زيادة الإنتاجيَّة في وقتٍ أقصر.

2-الوراثة: مفهوم الميراث هو معيارٌ بالفعل في كل لغات OOP. إنَّه سبيلٌ لأخذِ صفِّ موجودٍ بالفعل والإضافة إلى وظيفته (كتغييره مثلاً). الميراث هو غالباً طريقٌ لإعادةِ تشغيل الخوارزمية عن طريق ترك "الصفٍّ الأساسي" نفسه، وتصليح الأشياء هنا وهناك لإنتاج ما تريد. على أيَّة حـال، الميراث ليس فقط السبيل لإنتاج صفوفٍ جديدة من صفوفٍ موجودة بالفعل. يمكنك أيضاً أن تضمِّن كائناً داخل تركيب صفِّك الجديد.

الوراثة هي أحد أهم مبادئ البرمجة الكائنية وهي تحظى بدعم كبير من لغات البرمجة الكائنية مثل (C++). تقوم الوراثة بتحقيق تمثيل العالم الواقعي أو على الأقل السير خطوات تجاه هذا الهدف.

* الفرق بين الوراثة في العالم الحقيقي وعالم البرمجة: لنفهم أولاً معنى الوراثة في العالم الحقيقي وسنفهمه على مستوى الأب وابنه فالأب يقوم بتوريث صفاته إلى الابن؛ أي أن الذي قام بفعل الوراثة هو الأب أو بمعنى برمجي الصنف الأساس، أما الشبل أو الصنف المشتق فهو الذي يأخذ من الأب أي أنه يستقبل، وهذا هو المعنى البديهي للوراثة. إلا أن الوضع يختلف بالنسبة للوراثة في عالم البرمجة. فالذي يقوم بالوراثة هنا ليس الأب ولكن الابن، لا تحاول فهم الوراثة على أن هناك صنف أساس يقوم بتوريث صفاته إلى أبنائه بل افهمه على مبدأ أن الصنف الذي أنشأته صنف مستقل عن جميع الأصناف الأخرى، وأنه هو الذي يحدد خياراته وشكله فبإمكانه أن يتوارث من أي صنف يريد بشرط أن يخدم الغرض من إنشاء هذا الصنف، وبإمكانه أيضاً أن يتوارث من عدة أصناف دفعة واحدة وليس صنفاً أو صنفين فقط؛ وهذا ما نطلق عليه التوارث المتعدد، وعموماً ينقسم التوارث إلى قسمين:
  1. التوارث العام: وفيه يملك الصنف الابن الصنف الأب بجميع أعضائه الخاصَّة والعامَّة.
  2. التوارث الخاص: وفيه يملك الصنف الابن طريقة الاستخدام للصنف الأب، فجميع أعضاء الأب تتحول إلى أعضاء خاصَّة لدى الصنف الابن.

3-تعددية الأشكال: لوحدك، قد تأخذ تسعةَ أشهرٍ لتكتشفَ وتفهمَ تعدُّدَ الأشكال، حجرَ الزَّاويةِ في OOP. خلال أمثلةٍ صغيرة بسيطةٍ سوف ترى كيفية إنشاءَ أسرةٍ من الأشكال بالميراث ومعالجة الكائنات في تلك الأسرة خلالَ الصفِّ الأساسيِّ المشترك. تعدُّديَّة الأشكال تمكنك من معالجة الكائنات في هذه الأسرة بشكلٍ عام، ما يعني أنَّ معظم خوارزميتك لا تعتمد على نوعٍ معيَّن من المعلومات. هذا يجعل برامجك قابلةً للامتداد، ذلك لبناء برامجٍ وصيانةِ خوارزميَّةٍ أسهلَ وأرخص.

الكبسلة أو التغليف : هي إخفاء المعلومات عن مستخدم الصنف فمن الممكن تشبيه الصنف أنه صندوقٌ أسود له معلومات لاستخدامه فإذا أخذنا مثال الصراف الآلي فأنت تقوم بإدخال بطاقتك الإتمانية ورقمها السري لتجري بعض العمليات والتي لا يهمك أن تعرفها وتخرج لك ما تريد من الصراف؛ بهذه الطريقة يمكن تشبيه الكبسلة أو التغليف؛ لا يهمك أن تعرف ماذا يحدث في الصراف وهذا أحد الأسباب وهنتك سبب آخر وهو أن البنك لا يريدك أن تعبث بالصراف فإذا كان بإمكانك تغيير برنامج الصراف وبالتالي تغيير برنامج البنك على ما تشتهيه نفسك فقد تحصل كارثة اقتصادية في البلاد، وهذا أيضاً على صعيد البرمجة الكائنية فمن جهة لا يهمك ما يحدث داخل الصنف ومن جهةٍ أخرى فإنه لا ينبغي لك أن تعبث بمحتويات الصنف الداخليَّة. هذه هي فائدة الكبسلة، وعلى صعيد الخوارزميَّات فهناك كلمتان (public) والتي تعني أن الأعضاء الذين تحتها هم أعضاء عامَّة فبالإمكان التَّغيير فيهم والكلمة الأخرى هي (private) والتي تعني أن الأعضاء الذين تحتها هم أعضاءٌ غير مرئيين خارج الطَّبقة أي أعضاء مكبسلين أو مغلَّفين.

الأعضاء ومحدِّدات الوصول:

أعضاء الصِّنف: هم جميع الدوال والمتغيِّرات التي تمَّ تعريفها ضمن هذا الصِّنف.

ولضمان أنك تقوم بتطبيقٍ فعلي للبرمجة الكائنيَّة ولمبدأ الكبسلة فعليك كبسلة جميع الأعضاء لا يوجد قاعدةٌ عامَّةٌ لذلك ولكن طبيعة البرمجة الكائنيَّة هي من تفرض عليك ذلك، فجميع أعضاء الصِّنف لن تطلب لها أن تكون ظاهرةً للعيان لأنها هي اللب الداخلي للصنف، أو يمكن أن نعرِّفها على أنها هي الحالة الداخليَّة للصِّنف، فالمتغيرات (المتحــوِّلات) إذا تغيَّرت فستتغيَّر طبيعة البرنامج الذي تقوم به، أو المهمَّة التي يقوم بها هذا الصِّنف بعكس الدَّوال الأعضاء فيمكننا فهم الدَّوال الأعضاء على أنَّها هي المحرِّك للمتغيّرات والمتغيِّرات يجب أن تبقى مخفيَّة عن الجميع ما عدا هذه الدوال والتي تعرف كيف تتصرَّف معها. فحينما ترغب في أن يكون أحد المتغيِّرات الأعضاء ثابتاً (لا يتغيَّر أبداً) مهما فعلت إلا وفق شروطٍ معيَّنة فستقوم بكبسلة هذا العضو المتغيِّر وكتابة دالة تعرف كيف تتصرف مع هذا المتغيِّر. ومحدِّدات الوصول ليست قاعدة برمجيَّة بل هي رؤيةُ أفضل لكتابة برامج أسهل للصيانة والتطوير.

تابعا البناء والهدم:

* تابع البناء: حينما نقوم بكتابة بيانات أيَّ صنف فإنه وبقليل من التفكير سنستنتج أنه لا يمكنك وضع أي قيمة ابتدائية لأي من بيانات هذا الصنف؛ والسبب في ذلك أنك لا تقوم بحجز ذاكرة لهذا الصنف فكيف تحجز ذاكرة في الأساس لعنصرٍ من عناصره. ومن أجل حل هذه المشكلة تم وضع دوال خاصَّة تسمَّى دوال البناء، ودالة البناء لها نفس اسم الصنف... وهذه الدالة لا تعيد أية قيم حتى القيمة (void) ولكن بإمكانها أن تأخذ أي عددٍ من المعاملات وبإمكانك زيادة تحميل مثل هذه الدَّوال.
* تابع الهدم: بعد أن تنتهي من الكائن الذي تعمل عليه فمن الضروري أن تقوم بهدمه أو حذفه حتى تتحرر الذاكرة؛ وهذا ما تقوم به دالة الهدم؛ بإمكانك أن تحذف الأعضاء الذين لا تريدهم مثل المؤشرات والمرجعيَّات وحذف الكائن بالكامل. وهذه الدالة لا تعيد أية قيم حتى القيمة (void) ولكن بإمكانها أن تأخذ أي عددٍ من المعاملات وبإمكانك زيادة تحميل مثل هذه الدَّوال.

الاحتواء (أو التركيب): يعرَّف الاحتواء على أن تركيب أحد الأصناف يعتمد على صنف آخر مثلاً إذا كان لدينا صف السيارة، لذلك فإن الصنف محرك يعتبر محتوى في الصنف السيارة، يمكن وصف العلاقة بين الصنفين بأنها علاقة (يمتلك) أي أنَّ الصنف السيارة يمتلك الصنف المحرِّك، يعتبر هذا الكلام ضرورياَّ للغاية حينما تصل لمواضيع الوراثة وكيف تفرق بين العلاقات بين الكائنات أهي علاقة توارث أم علاقة احتواء.

أول لغة كائنية ناجحة ظهرت في الوجود هي (Smalltalk) سنقوم في هذه الفقرة بأخذ مبادئ هذه اللغة ومجاراتها فيما يخدمنا لفهم أفضل لما تعنيه الكائنات في البرمجة.

تتألف(ٍSmalltalk) من خمسة مبادئ هي:

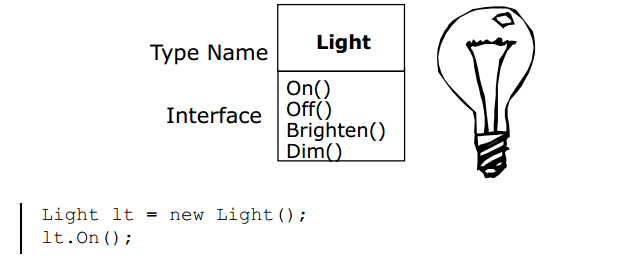
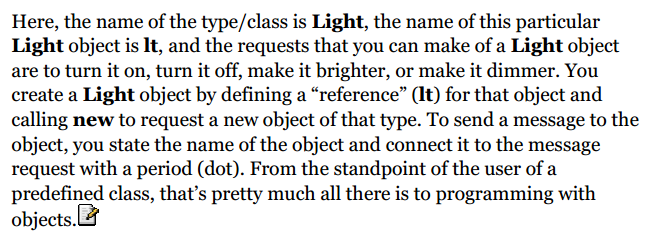
* كل شيء هو كائن.
* البرنامج هو عبارة عن كائنات تتفاعل مع بعضها بواسطة إرسال الرسائل.
* كل كائن يملك ذاكرة خاصة به مبنيَّة على الكائنات الأخرى.
* لكل كائن نوع من البيانات (صنف)
* جميع الكائنات من نفس النوع تتفاعل بواسطة نفس الرسائل.

من هذه المبادئ نستنتج أنَّها تتكلم عن الكائنات وليس الأصناف وبالتالي فإن عليك أن تتذكر أنَّ البرمجة الكائنيَّة قائمة على الكائنات وليس الأصناف.

لكل كائن واجهة: أولاً نبدأ بتعريف الواجهة: وهي البيانات العامة للكائن، ويجب عليك أنت صنع كائن جيد حيث أن هذا هو الاتجاه في البرمجة الكائنيَّة.

* أولاً: الواجهة هي الطريقة الوحيدة حتى يتفاعل هذا الكائن مع الكائنات الأخرى توابعاً كان أو متغيرات أو كائنات.
* ثانياً: البيانات العامة يجب ألا تكون هي اللب الرئيسي للكائن لأنها لو كانت كذلك لاستطاع المستخدم أن يعبث بمحتويات هذا الكائن، ليس عن قصد بل عن خطأ، لأنه لا يعرف ما هي الأشياء المهمة لهذا الكائن التي هي الواجهة، لذلك عندما يريد المستخدم تغيير إحدى البيانات (في الصِّنف) فسيكون التَّغيير أكثر أماناً باستخدام التَّوابع (get) و (set). من هنا يجب عليك الفصل بين الواجهة والمعالجة؛ فالمعالجة يجب أن تكون داخليَّة وليست خارجيَّة. فمثلاً إذا قمنا بكتابة صنف طالب فحينها يجب علينا إخفاء البيانات المهمة والتي نود معالجتها.
* إخفاء المعالجة في أغلب الأحيان يتطلب منك جعل البيانات الفعلية للأصناف مثل صنف طالب مخفيَّة (درجة الطالب، معدل الطالب، عمر الطالب) مخفية عن العالم الخارجي وبالتالي فأنت ستسمح فقط للتوابع الأعضاء بمعالجة هذه البيانات المهمة وبالتالي فأنت قمت بإخفاء المعالجة نهائياً.

ندعِّم الكلام السَّابق بمثالٍ مأخوذٍ من كتاب التفكير في سي شارب للمبرمج لاري أوبرين:

 ****

شكل رقم (3) يوضح المصباح الكهربائي

الاحتفاظ بكائناتك:

إنَّه برنامجٌ بسيطٌ إلى حدٍّ لا بأسَ به الذي لديه فقط كمِّيَّة ثابتة من الكائنات بأعمارٍ معروفة.

عموماً، برامجك سوف تُنشئُ دوماً كائناتٍ جديدة بتشكيلةٍ من التَّواقيت التي ستكون معلومة فقط بينما يكون البرنامج في وضع التشغيل. بالإضافة، لن تعلم الكمِّيَّةَ حتى وقت التَّشغيل أو حتى النوع الدَّقيق للكائنات التي تحتاجها. لحل مشكلة البرمجة العامَّة، سوف تحتاج لإنشاء أيِّ عددٍ من الكائنات، في أي وقت، وفي أيِّ مكان. هذا الفصل يستكشف في أعماق مكتبةَ Collection التي تدعمها ADO.NET لاحتجاز الكائنات بينما تقوم أنت بالتعامل معهم: المصفوفات البسيطة وحافظات أكثرَ تطوُّراً (تراكيب البيانات data structures). هذا الفصل يغطي أيضاً أساسيَّات ADO.NET.[[2]](#footnote-2)

فصل ثالث: لمحة عن لغة البرمجة جافا:

لغة الجافا:

هي لغة حديثة أنتجتها شركة سن (Sun Company) عام 1995م لتناسب التطبيقات الحديثة. وهي تناسب تطبيقات الإنترنت حيث أصبحت هي قلب برمجة الإنترنت بما توفره من إمكانيات.   
تتصف لغة الجافا بالصفات التالية :

1. لغة برمجية تدعم مبادئ الكائنية.
2. لها بيئة تشغيل خاصة بها JVM.
3. تقوم على لغة C و.C++
4. تعمل على معظم نظم التشغيل.

بيئة التشغيل (JVM):

الحروف JVM اختصار للعبارة (Java Virtual Machine) وهي فكرة قامت جافا بإنشائها لتجعل لغة جافا تعمل على جميع أو معظم أنظمة التشغيل. وتقوم الفكرة على إنشاء طبقة وسيطة (Software) كأنها برنامج تشغيل للبرامج Runtime لكل نظام تشغيل يتم إنزاله أولاً على الأجهزة بحيث تفهم هي برامج جافا وتفسرها لنظام التشغيل ثم الجهاز ولهذا كان من مزايا لغة جافا أنها تعمل على كثير من نظم التشغيل الموجودة بعد إعداد JVM الخاصة بمعظم أنظمة التشغيل ... فلا يهم إذا كان البرنامج مكتوب لنظام التشغيل WINDOWS) ) أو ([UNIX](http://www.c4arab.com/qamoos/mean.php?word=UNIX)  ) و المهم أن البرنامج يكتب ثم يحمل إلى الجهاز وعلى الجهاز يوجد JVM للنظام الموجود وبالتالي يعمل البرنامج.

* أهم مميزات الجافا:

أكثر لغات البرمجة إثارة حيث تمكننا من الآتي:

* تتميز لغة الجافا بمميزات خاصة مما يجعله
* إضافة الحركة والصوت إلى صفحات الويب.
* كتابة الألعاب والبرامج المساعدة.
* إنشاء برامج ذات واجهة مستخدم رسومية.
* تصميم برمجيات تستفيد من كل مميزات الانترنت.
* توفر لغة الجافا بيئة تفاعلية عبر الشبكة العنكبوتية وبالتالي تستعمل لكتابة برامج تعليمية للإنترنت عبر برمجيات المحاكاة الحاسوبية للتجارب العلمية وبرمجيات الفصول الافتراضية للتعليم الإلكتروني والتعليم عن بعد.[[3]](#footnote-3)
* نسخ لغة الجافا:

1. J2EE: هي اختصار ل Java 2 Enterprise Edition وهي تزودنا بالتطبيقات الكبيرة على مستوى الشركات الكبيرة.
2. J2SE: هي اختصار ل Java 2 Standard Editionيتم من خلالها دراسة اللغة وإنشاء التطبيقات أساسيةDesktop Application)).
3. J2ME: هي اختصار ل Java 2 Micro Edition فهي تخص ال (wireless devices) بشكل عام يعني على أجهزة الجوال وغيرها.

|  |  |  |
| --- | --- | --- |
| الكلمات المحجوزة في اللغة | | |
| Public | Finally | Abstract |
| Return | Float | Boolean |
| Short | For | Break |
| Static | If | Byte |
| Super | Implements | Case |
| Switch | Import | catch |
| Synchronized | Instanceof | char |
| This | Int | class |
| Throw | Interface | continue |
| Throws | Long | default |
| Transient | Native | Do |
| True | New | double |
| Try | Null | else |
| Void | Package | extends |
| Volatile | Private | false |
| While | Protected | final |

شكل رقم (4) يوضح الكلمات المحجوزة في اللغة

ما هو ال JDK؟

Java Developers Kit (JDK) عدة تطوير الجافا تعتبر هذه الأداة الأولى التي وضعها مطورو الجافا بشركة صن ميكروسيستمز (Sun Micro Systems) ورغم وجود أدوات برمجة أخرى من عدة شركات منافسة إلا أنه من الأفضل أن نأخذ الأمر ممن طوره وهذه الأداة قد صدرت في عدة إصدارات ويمكن تحميل هذه الأدوات من موقع صن ميكروسيستمز (Sun Micro Systems).

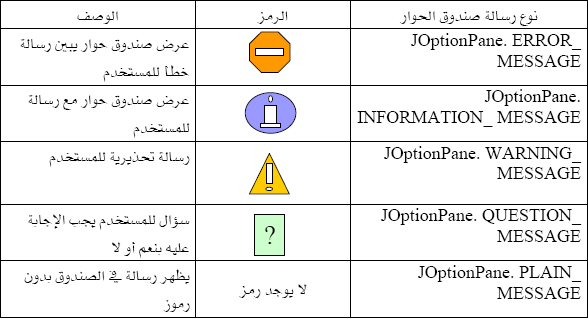
يمكن كتابة برامج الجافا في محرر نصوص عادي وتنفيذه في بيئة ال DOS.

ملاحظات خاصة باللغة:

* اللغة حساسة للأحرف الصغيرة الكبيرة.
* جمل التعليق في الجافا كما في لغة الـ C //لسطر واحد و /\* ------\*/ لأكثر من سطر.
* يمكن استخدام أسطر فارغة لا تؤثر في الترجمة.
* يجب حفظ الملف الخاص بالبرنامج بامتداد .java

|  |  |
| --- | --- |
| تعليمات برمجية | وصفها |
| System.out.print("---") | وهذه الجملة تقوم بإظهار الجمل النصية وكذلك أي معلومات أو بيانات في نافذة الأوامر (command window) حيث يتم تنفيذ برامج الجافا. ويظل المؤشر في نفس السطر. |
| System.out.println("---") | نفس التعليمة السابقة ولكن تتم الطباعة وينتقل المؤشر سطر جديد. |
| Integer.parseInt  ( firstNumber ); | تقوم هذه الدالة بتحويل القيمة النصية الممرره للدالة -firstNumber -إلى قيمة صحيحة. وتسند الدالة إلى متغير من نوع Int. |
| Double.parseDouble (firstNumber) | تقوم هذه الدالة بتحويل القيمة النصية الممرره للدالة -firstNumber -إلى متغير من نوع Double. وتسند الدالة إلى متغير من نوعDouble. |
| import javax.swing.JOptionPane; | هذه العبارة يتم إضافتها في بداية البرنامج مثل عبارة #include<stdio.h> في لغة ال C.  تعمل هذه العبارة على تضمين الكائن المسمى JOptionPane من الحزمة المسماة javax.swing. وهذه الحزمة تحتوي على كثير من الكائنات الخاصة بالرسومات والتعامل مع المستخدم والتي تسهل إدخال وإخراج البيانات من خلال مربعات الحوار. |
| Import javax.swing.\*; | هذه العبارة يتم إضافتها في بداية البرنامج مثل عبارة #include<stdio.h> في لغة ال C.  تعمل هذه العبارة على تضمين كافة الكائنات الموجودة في الحزمة javax.swing. |
| JOptionPane.showInputDialog( "---" ); | يتم استدعاء الدالة showInputDialog () الموجودة في الكلاس JOptionPane والتي تعمل على استقبال مدخلات من المستخدم على هيئة String. ولها وسيط واحد والذي يمثل عبارة توضيحية تظهر في مربع الحوار. و يتم اسناد القيمة المدخلة من قبل المستخدم الى متغير من نوع String. |
| System.exit(0); | تستخدم هذه الدالة لإنهاء التطبيق. |
| JOptionPane.showMessageDialog  (null,"" ,"" , ); | يتم استدعاء الدالة showMessageDialog () من الكلاس المسمى JOptionPaneوهذه الدالة تقوم بإظهار مربع حوار لعرض المخرجات ولها وسيطين أو أربعة:  الأول: دائماً سيكون الكلمة null، وهو يحدد المكان الذي يظهر فيه صندوق الحوار، وفي هذه الحالة فإن صندوق الحوار سوف يظهر في منتصف الشاشة.  الثاني: فهو النص المراد إظهاره.  الثالث: يمثل النص الذي سوف يظهر في سطر العنوان لمربع الحوار.  الرابع: يمثل الرمز الذي يبين نوع مربع الحوار.  ويوجد مجموعة من الرموز التي يمكن إظهارها في صندوق الحوار لتساعد المستخدم في معرفة نوع صندوق الحوار والرسالة التي تظهر فيه و هذه الرموز في الجدول التالي: |

شكل رقم (5) يوضح سير عمل بعض التعليمات بلغة الجافا



شكل رقم (6) يوضح الرموز التي تظهر في مربع الحوار

**البــــــــــــــــــــــــــــــــــــــــــــاب الثــــــــــــــــــــــــــــــــانــــــــــــــي:**

معلومات حول نظام التشغيل Android

إصدارات نظام التشغيل Android

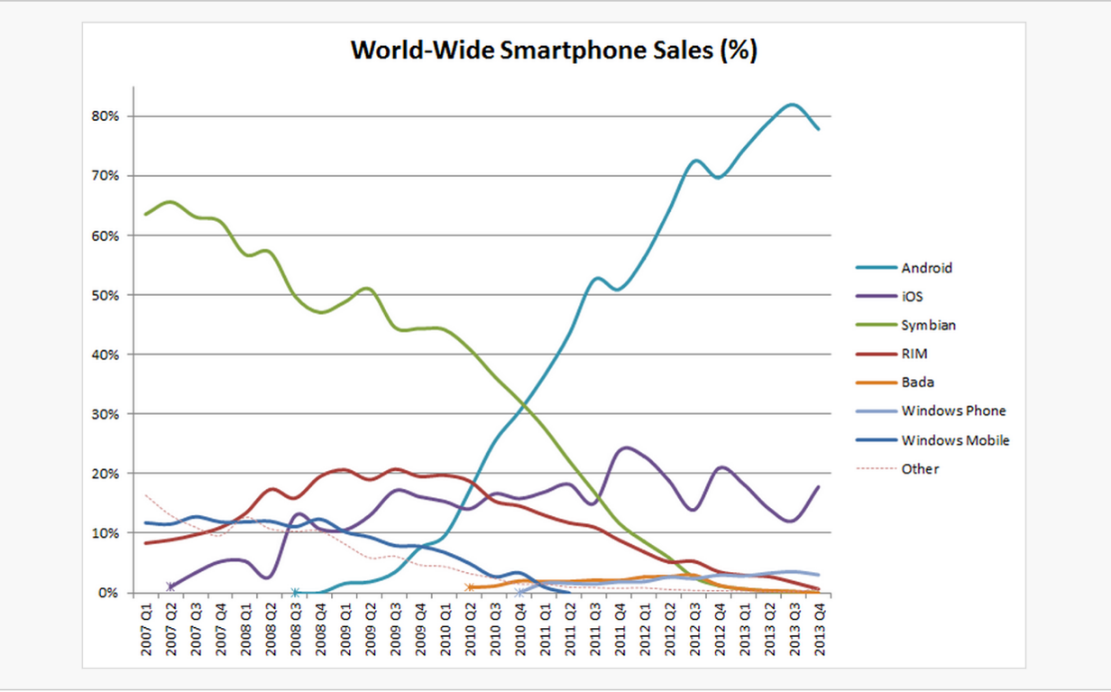
انتشار نظام التشغيل Android وأهم ما يميزه

بعض أنواع الأجهزة التي تعمل باستخدام نظام التشغيل Android

فصل أول: انتشار نظام التشغيل Android وأهم ما يميزه

في أغسطس[[4]](#footnote-4) 2005، قامت شركة جوجل بشراء النظام من الشركة المالكة وهي توزع النظام حالياً(Android OS) وأدوات التطوير (developer tools) بالمجان وضم المؤسسين السابقين وعلى رأسهم المهندس أندرو روبن الذي استمر بالعمل على تطوير النظام، ويعتبر هو المسؤول المباشر عن وصول نظام أندرويد إلى النجاح الذي نشهده حالياً.

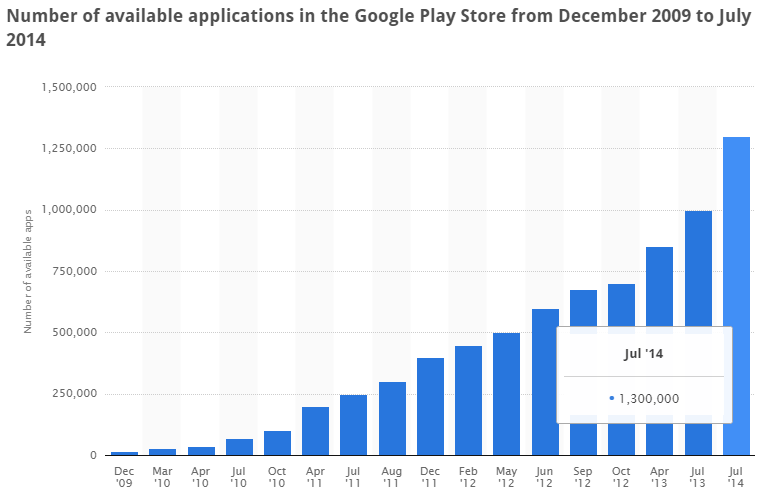
وفي الصورة مخطط يوضح تزايد استخدام الأندرويد بالنسبة لباقي أنظمة تشغيل الهواتف المحمولة:



شكل رقم (7) يوضح تزايد استخدام الأندرويد بالنسبة لباقي أنظمة تشغيل الهواتف المحمولة

يتم تطوير الأندرويد من قبل التحالف المفتوح للهواتف النقالة الذي تديره شركة جوجل.

ويحتوي متجر جوجل بلاي على ملايين التطبيقات والألعاب المجانية للأندرويد وهو المتجر الرسمي للأندرويد.

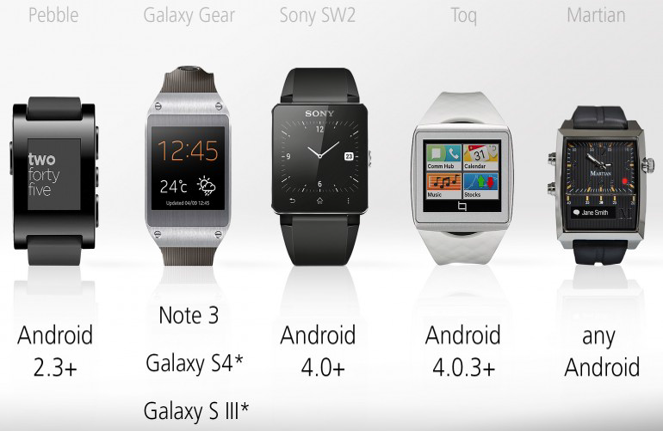
وصل عدد المستخدمين إلى 1 مليار بتاريخ 4/تموز/2001 وعدد التطبيقات مليون وربع تطبيق في متجر الأندرويد من ألعاب وتطبيقات خدمية.

شكل رقم (8) يوضح تعداد مستخدمي الأندرويد

* بعض خصائص التي يدعمها النظام:

1. التخزين: تستخدم التطبيقات قواعد بيانات SQLite)) لتخزين بياناتها.
2. التواصل والشبكات: يدعم خدمات (GSM),(IDEN),(CDMA),(UMTS),(Bluetooth),(Wifi),(WiMax),(LTE)
3. خدمات الرسائل: SMS) , (MMS) ).
4. تصفح الإنترنت: بواسطة الأداة مفتوحة المصدر WEB Kit.
5. التكامل مع ال ـHardware: منح خواص التحكم والتعامل مع الحساسات والكاميرا والـ GPS وحساس البوصلة وحساس الخادم الوكيل الموجودة في الجهاز.
6. دعم الوسائط: يدعم صيغ الصور والفيديو والموسيقا: ,(3GP), (MPEG) (MP4),(MP3),(WAV),(JPEG),(PNG),(BMP),(GIF),(AAC), .
7. تعدد اللمس.
8. تعدد المهام.
9. دعم وسائط الفلاش.
10. مشاركة الشبكة: يدعم مشاركة الهاتف للإنترنت عن طريق تكوين نقطة وصول لاسلكية من الجهاز.

فصل ثان: بعض أنواع الأجهزة التي تعمل باستخدام نظام التشغيل Android

أولاً: الساعات الذكية (Smart Watches)

ميزاتها:

* الأوامر الصوتية.
* مكالمات الهاتف.
* تشغيل بعض التطبيقات والنصوص.
* تحديد المواقع.
* شاحن لاسلكي.

شكل رقم(9) يوضح الساعات الذكية (ٍSmart Watches)

* عداد خطوات المشي.
* عداد للنبض القلبي.

ثانياً: شاشة التلفاز التفاعلية الذكية (Smart TV platform)

ميزاتها:

* عرض النصائح المعتمدة على المشاهدة كالتذكير بمواعيد الطعام.
* تشغيل بعض تطبيقات الوسائط.
* ألعاب.
* خدمات البحث الصوتي للإجابة عن التساؤلات.

ثالثاً: نظارات جوجل الذكية (**Google Glass**) من ميزاتها:

* تستجيب للأوامر الصوتية.
* تسجل الفيديو.
* تعرض وتتصفح المعلومات من خلالها على الويب مثلاً.

رابعاً: نظام التشغيل الداخلي للسيارات:

Android Hardware: in‐Car Entertainment and Navigation System))

بعض الميزات:

* نظام معلومات.
* هاتف.
* قادر على تشغيل الأصوات.

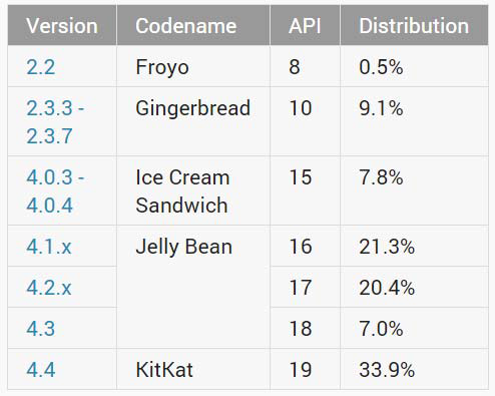
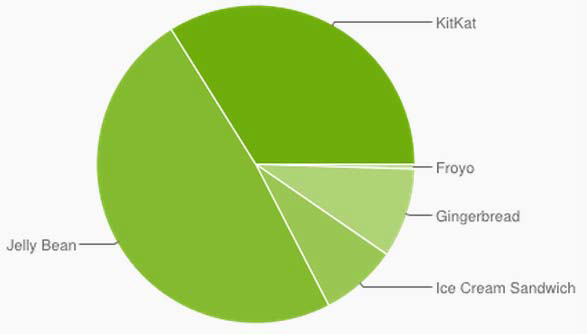
فصل ثالث: إصدارات نظام التشغيل Android

أهم ما يميز الأندرويد عن غيره أنه مفتوح المصدر كما ذكرنا سابقاً وأن أغلب التطبيقات مجانية وأيضاً لا ننسى أن أسماء إصدارات النظام مرتبة أبجدياً وكل إصدار هو اسم لنوع من أنواع الحلويات[[5]](#footnote-5)

إصدارات النظام وتاريخه:

1. 1.5 cupcake
2. 1.6 donut
3. 2.0 eclair
4. 2.2 froyo
5. خبز الزنجبيل : (Gingerbread) وهو التحديث رقم 2.3 للأندرويد. وفيه تم تحسين واجهة المستخدم، لوحة المفاتيح لينة ونسخ / لصق ميزات، وتحسين أداء الألعاب، وأضاف دعم SIP) المكالمات عبر بروتوكول الإنترنت).
6. قرص العسل: (Honeycomb) وهو التحديث رقم 3.0 للأندرويد، الذي يدعم الشاشات الكبيرة، ويقدم العديد من الميزات الجديدة في واجهة المستخدم، ودعم المعالجات متعددة النواة وتسريع أجهزة الرسومات. وأول جهاز تم طرح هذا الاصدار فيه هو، جهاز موتورولا زوم اللوحي (Xoom). وقد تفرع من نفس التحديث
7. قرص العسل 3.1: صدر في مايو 2011، وتم فيه تقديم الدعم لأجهزة الإدخال إضافية، ووضع المضيف USB لنقل المعلومات مباشرة من الكاميرات والأجهزة الأخرى، وأفلام وتطبيقات وكتب غوغل.
8. قرص العسل 3.2: صدر في يوليو 2011، وتم فيه التحسين لمجموعة أوسع من أحجام الشاشات الجديدة “تكبير إلى ملء” وضع التوافق الشاشة، تحميل ملفات الوسائط مباشرة من بطاقة SD، ودعم الشاشة API الموسعة.
9. ايس كريم ساندويتش (Ice Cream Sandwich): وهو التحديث رقم 4.0 للأندرويد، وقد أُعلن عنه يوم 19 أكتوبر 2011، وفيه تم إضافة ميزات قرص العسل إلى الهواتف الذكية، وإضافة الميزات الجديدة بما في ذلك التعرف على الوجه، ورصد استخدام شبكة البيانات والتحكم، الاتصالات الموحدة والشبكات الاجتماعية، والتحسينات التصوير الفوتوغرافي، وتبادل المعلومات باستخدام NFC. ساندويتش الآيس كريم 4.0.4.
10. هلام الفول (Jelly Bean): أندرويد 4.1 (جيلي بين) هو الإصدار الأخير للأندرويد الذي يتوفر على الهواتف ومن ابرز مميزاته سريعة وناعمة واكثر ما يميزه هو البحث الصوتي وخدمة Google Now التي تجلب لك المعلومة المناسبة في الوقت المناسب بالإضافة لتواجد كل مميزات الاصدار السابق ساندويتش الآيس كريم أو يمكن القول انه فرع منه. وفي 29 أكتوبر تشرين الثاني أعلنت جوجل عن تحديث جديد أندرويد 4.2 بذات الاسم جيلي بين.
11. أندرويد كيت كات (KitKat): وهو الإصدار رقم 4.4 الذي جاء ليضيف مميزات برمجية لتجربة استخدام أفضل مثل الالة ART والعديد من التغيرات في التصميم وقد وتم الكشف عنه بتاريخ 30 أكتوبر 2013
12. 5.0 lollipop

نسب وإحصائيات استخدام نسخ النظام موضح بالصورة:



شكل رقم(10) يوضح مدى انتشار كل نسخة من نسخ Android

**البـــــــــــــــــــــــــــــــــــــــــاب الثالــــــــــــــــــــــــــــــــــــــث:**

قواعد البيانات SQL

فصل أول: **نبذة عن قواعد البيانات:**

**المقصود بقاعدة البيانات**

هي مجموعة مترابطة من البيانات يمكن استخدامها لاستخراج المعلومات التي تريدها أو هي عبارة عن تجمع لكمية كبيرة جداً من البيانات والمعلومات وعرضها بطريقة تسهل من الاستفادة منها فعلى سبيل المثال فإن دليل الهاتف هو أبسط قاعدة بيانات يمكن أن تتعامل معها فيمكنك تسجيل اسماء أصدقائك وأرقام التليفونات الخاصة بهم حتى يمكنك فيما بعد أن تبحث عن اسم أحد الأصدقاء بإدخال رقم الإشراك أو العنوان ……وهكذا.

**نظم إدارة قواعد البيانات**

نظم إدارة قواعد البيانات ( DBMS) هي مجموعة من البرامج الجاهزة التي تقوم بتنفيذ جميع الوظائف والمهام المطلوبة من قاعدة البيانات فعلى سبيل المثال بعد إدخال بيانات العاملين إلى قاعدة بيانات شئون العاملين فإنك قد تحتاج إلى ترتيب اسماء العاملين أبجدياً فإن مثل هذا العمل يطلق عليه إدارة قاعدة البيانات، ويتكون نظام إدارة قواعد البيانات من مجموعة من البرامج والملفات التي تتشابك مع بعضها لحل مشكلة أو تحويل نظام يدوي إلى نظام يعمل بالحاسب مثل تحويل حسابات المخازن من أنظمة ودفاتر يومية إلى نظام وملفات تستخدم بواسطة الحاسب الآلي.

**أنواع قواعد البيانات**

قواعد بيانات هرمية (Hierarchy Database)

قواعد بيانات شبكية (Network Database)

قواعد بيانات علائقية (Relational Database)

يقتصر استخدام كلاً من قواعد البيانات الهرمية والشبكية على الحاسبات الكبيرة، أما قواعد البيانات العلائقية Relational Databaseفإنها أكثر استخدما وشهرة مع الحاسب وإليها ينتمي Oracle For Windows.

**لغة الاستفسارات الهيكلية(SQL)**

تمكنك لغة) SQL اختصاراً للكلمات Structured Query Language) من إدارة قواعد البيانات بشكل كامل وإجراء جميع العمليات القياسية كإنشاء الجداول وتعبئتها بالبيانات، أو إجراء الاستعلامات عليها وكذلك الربط بين الجداول المختلفة، ولكن كيف يمكننا عمل ذلك بها؟

كما وضحنا سابقا فإن قاعدة البيانات عبارة عن مكان أو مستودع كبير لتخزين البيانات المختلفة، تريحك قاعدة البيانات من عناء تخزين بياناتك في ملفات منفصلة وكما أنك تحتاج إلى إجراء عمليات البحث والتصنيف لهذه البيانات عن طريق خوارزميات البحث المعقدة فتسهل لك ذلك بأنها تعطيك واجهة سهلة للتعامل مع البيانات، بحيث لا تحتاج إلى كل هذا وتكون البيانات في قاعدة البيانات مخزنة في عدة جداول Tables وكما نعلم فالجدول يتكون من صفوف Rows وأعمدة Columns هكذا:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

شكل رقم (11) يوضح الجدول في قاعدة البيانات

الجدول السابق يتكون من ثلاثة صفوف وثلاثة أعمدة، وفي قواعد البيانات فإننا نسمي الصفوف بالسجلات Records ونسمي الأعمدة بالحقول Fields، فما هي الحقول وما هي السجلات؟

يقوم الحقل الواحد في الجدول بتخزين معلومة معينة، فمثلا عندما يكون لدينا قاعدة بيانات بأرقام الهواتف فإننا سنحتاج إلى حقلين (أو عامودين) واحد للاسم والثاني لرقم الهاتف، أما السجلات (أو الصفوف) فيحتوي كل منها على مجموعة من الحقول، فترى بأن السجل الواحد في مثالنا يحتوي على معلومتين مختلفتين هما الاسم ورقم الهاتف، لذلك يمكننا القول بأن قاعدة البيانات تتكون من الجداول والجداول تتكون من السجلات والسجلات تتكون من الحقول، وكل حقل يحتوي على معلومة واحدة، أي أن الحقل هو أصغر وحدة قاعدة البيانات.

**عبــاراتSQL**

تنقسم عبارات SQL إلى ثلاث فئات رئيسية وهي كالتالي:

**لغة تعريف البيانات DDL: Data Definition Language**

هي مجموعة من أوامرSQL تستطيع من خلالها إنشاء وتعريف الكائنات في قاعدة البيانات، وتقوم هذه الأوامر بإنشاء أو إسقاط أو تغيير كائن قاعدة البيانات، وهي اختصاراً للكلمات (Data Definition Language)، ومن أهم أوامرها Create , Alter, Drop, Rename, Truncate.

**لغة معالجة البيانات DML Data Manipulation Language**

هي مجموعة من العبارات التي تستطيع من خلالها معالجة البيانات الموجودة في قاعدة البيانات من إدراج البيانات أو تحديثها أو تحديد أو حذف البيانات، وهي اختصارا للكلمات (Data Manipulation Language)، ومن أهم أوامرها Insert, Update, Delete.

**لغة التحكم والصلاحيات: DCL Data Control Language**

هي مجموعة من أوامرSQL تستطيع من خلالها منح أو سحب امتيازات استخدام كائن قاعدة البيانات، وهي اختصارا للكلمات، وهي اختصارا للكلمات (Data Control Language)، ومن أهم أوامرهاGrant، Revoke.

وإيجازاً للقول فإن الشكل التالي يوضح الفئات الرئيسية لعبارات SQL

**استخدام SQL\*Plus**

يعتبر برنامج SQL\*PLUS أحد منتجات شركة Oracle، وهو واجهة التخاطب مع قاعدة البيانات، ويستخدم في إنشاء قواعد البيانات ومنح السماحيات، وإنشاء الجداول وتعديلها واسترجاع البيانات من الجداول.

**استخدام SQL\*Plus**

يعد استخدام SQL\*Plusمن أسهل الطرق لتعلم لغة SQL، للدخول على SQL\*Plus يتم ذلك بالطريقة التالية:

* من قائمة start أشر إلى Programs، ثم Oracle -OraHome9i، ومنهApplication Development، ثم SQL\*Plus ستظهر لك الشاشة التالية تسأل عن اسم المستخدم وكلمة المرور
  + اسم المستخدم (user name): اسم معرف حساب مستخدم oracle الذي تم تحديده للمستخدم.
  + كلمة المرور (password): كلمة المرور الخاصة بحساب المستخدم.
  + سلسلة المصنف(Host): سلسلة الأحرف التي تحدد مكان تواجد قاعدة البيانات، وتسمى بسلسلة الاتصال (connect string) وتستخدم عندما تكون متصلاً بشبكة وoracle الذي تريد الوصول إليه قد تم إعداده على جهاز أخر غير الحاسب الذي تقوم بالتعامل منه مع oracle.

**استخدام (SQL LITE):**

* + إن جميع تطبيقات الأندرويد تحتاج إلى مجموعة من المعلومات و البيانات التي يجب تسجيلها و حفظها في مكان آمن و محمي من الاختراق كنظم إدارة قواعد البيانات من جهة التعليمات فإن نسخة SQL LITE لا تختلف من حيث كتابة أو وظيفة أي تعليمة لكن هي قابلة للعمل مع مختلف إصدارات أندرويد و التعامل معها ، ما يميز نظم إدارة قواعد المعطيات عن هو الحاجة لتخزين كميات متزايدة من المعلومات يومياً و شهرياً و سنوياً سواء لناحية المستخدم أو الشركة المصنعة إذ أن المزيد من التكنولوجيا و التطور و الراحة يقابل كمية معلومات أكبر و بذل جهد عظيم لتحقيق ذلك.

فصل ثان: تعليمات نظام إدارة قواعد المعطيات SQL

* ما هو الاستعلام؟

إن الاستعلام ما هو إلا سؤالا تسأله عن بياناتك. فأنت قد تسأل، مثلاً:

أ-أي الطلبات جاءت من العملاء المقيمين في باريس؟

ب-من الذي كان أفضل موظف مبيعات خلال الربع الأخير من السنة؟

ج-ما هي الحسابات المستحقة؟

يقوم Access بجمع البيانات التي تجيب على سؤالك من جدول أو أكثر. إن هذه البيانات هي المجموعة الحيوية (إذا كنت تستطيع التحرير فيها)، أو لقطه (إذا كنت لا تستطيع التحرير فيها).

وفي كل مره تشغل فيها الاستعلام تحصل على أحدث المعلومات في المجموعة الحيوية. ويقوم Access إما بعرض المجموعة الحيوية أو اللقطة لمشاهدتها، أو تنفيذ إجراء ما فيها مثل حذف بعض البيانات أو تحديثها.

* + طرق عرض الاستعلام:

يمكنك العمل بالاستعلام في طريقتي العرض:

1. طريقة عرض التصميم.
2. طريقة عرض صفحة البيانات.

1 ــ طريقة عرض التصميم:

لإنشاء أو تحديث هيكل الاستعلام، قم بالعمل في طريقة العرض تصميم. هنا تسأل أسئلة عن بياناتك لتحديد لأي بيانات تريدها، وتحديد كيفية ترتيبها.

2 ــ طريقة عرض صفحة البيانات:

لإضافة، تحرير، تحليل البيانات نفسها -- البيانات المحتواة بداخل المجموعة الحيوية أو فئة السجل التي تجيب عن الأسئلة التي استوضحتها.

جملة Select

الأمر Select يمكنك من استعادة البيانات من داخل قاعدة البيانات، وتأخذ جملة Select الشكل التالي:

**SELECT [DISTINCT] {\*, *column* [*alias*],...}**

**FROM *table;***

شكل رقم (12) يوضح جملة Select

تمثل عبارة Select طريقة مختصرة لإخبار SQL بأننا نريد إظهار جميع أعمدة الجدول.

1. Select: تحدد ماهي الأعمدة المراد إظهارها.
2. From: تحدد المكان الذي نريد استعادة البيانات منه (جدول أو أكثر).
3. الفاصلة المنقوطة: تخبر SQL بأن العبارة كاملة وجاهزة للتنفيذ.
4. الأقواس [] تشير على أن ما بداخل هذه الأقواس اختيارياً.

اختيار أعمدة محددة

يمكننا اختيار أعمدة محددة، وهو أمر في غاية السهولة حيث تقوم بكتابة اسماء الأعمدة بعد عبارة Select، مع وضع فاصلة بين اسماء الأعمدة، ثم تحدد الجدول بعد From.

جملة الشرط Where Clause

كان اختيارنا سابقاً للأعمدة يجري على كافة الصفوف التي يحتويها الجدول، أردنا إظهار صفوفاً تحتوي على قيماً معينة، كإظهار الموظفين الذين تتجاوز مرتباتهم 3000ريال أو الذين يقطنون مدينة ما، لفعل مثل ذلك سنحتاج إلى وضع كلمة Where ضمن عبارة Select، يأخذ الاختيار الشرطي الشكل التالي:

SELECT [DISTINCT] {\*, *column* [*alias*], ...}

FROM *table*

[WHERE *condition(s)*];

شكل رقم (13) يوضح الاختيار الشرطي

تلميح

* تأتي جملة Where Clause بعد جملة From Clause.
* يلي Whereشرط أو عدة شروط.
* توجه عبارةWhere) ) الأمر إلى أوراكل بالبحث في قاعدة البيانات واستعادة الصفوف التي تتوافق مع شرط البحث.

الأحرف النصية Character Strings، والتاريخ في جملة Where لابد من وضعها بين علامتى تنصيص (' '.Single quotation استخدام عوامل المقارنة(Comparison Operators)

تمنحنا SQL ستة عوامل للمقارنة نستطيع استخدامها مع Where، كما هو مبين بالجدول التالي:

|  |  |
| --- | --- |
| المعنى Meaning | المعامل Operator |
| يســـاوي | = |
| أكبر من أو أقل من | >or < |
| أكبر من أو يساوي | >= |
| أقل من أو يساوي | <= |
| لا يساوي | <> Or! = |

شكل رقم (14) يوضح عوامل المقارنة التي نستطيع استخدامها مع Where

* تتضمن SQL أربعة معاملات أخرى بالإضافة للمعاملات التي الإشارة إليها في النقطة السابقة يمكن استخدامها مع جملة Where، كما هو مبين بالجدول التالي:

|  |  |
| --- | --- |
| المعني Meaning | المعامل Operator |
| بين قيمتين | BETWEEN…..AND…… |
| تتيح لك اختيار أي قيمة من List | IN(list) |
| يشابه أو يماثل | LIKE |
| فارغاً | IS NULL |

شكل رقم (15) يوضح المعاملات في SQL

* وضع قيم المجموعة بين قوسين.
* فصل قيم المجموعة عن بعضها بالفاصلة.
* يمكن وضع القيم بأي ترتيب.
* عندما تكون قيم المجموعة حرفية أو تاريخ لابد من وضعها بين علامتي تنصيص(' ') .
* يمكن استخدام NOT IN لعكس الاختيار.
* استخدام المعامل Like:
* يمكنك استخدام Like مع جملة Where لاستعادة واسترجاع السجلات التي قيم عمود فيها تشابه سلسلة أحرف نبحث عنها، ليس هذا فحسب بل يمكننا من استرجاع السجلات التي تكون قيم عمود فيها مطابقة جزئياً لمجموعة أحرف نبحث عنها.
* تستخدم LIKE مع الأعمدة التي يكون نوع بياناتها حرفي.
* اختصار لأي تتابع من سلسلة أحرف غير معينة مؤلفة من أي قيمة وبأي طول.
* اختصار لأي حرف وحيد.

استخدام معامل IS NULL:

قبل استخدام معامل (IS NULL) مع (Where)لابد من الإجابة على ماهي القيمة NULL؟ تستخدم القيمة NULL للإشارة إلى عمود لا يحتوي على بيانات، حيث لا تعني القيمة NULL صفر أو مسافة Blank Space.

استخدام عوامل المقارنة الأخرى (Logical Operators)

تتضمن SQL ثلاثة معاملات منطقية بالإضافة للمعاملات التي الإشارة إليها في سابقاً يمكن استخدامها مع جملة Where، كما هو مبين بالجدول التالي:

|  |  |
| --- | --- |
| المعني Meaning | المعامل Operator |
| و | AND |
| أو | OR |
| النفي | NOT |

شكل رقم (16) يوضح المعاملات المنطقية في SQL

* استخدام معامل AND

يستخدم ANDمع Where لاستعادة السجلات التي تحقق كافة الشروط المحددة في جملة Where، حيث يعني AND أنه لاستعادة السجلات لابد من تحقق كافة الشروط.

* استخدام معامل OR

نستخدم معامل Or مع Where لاستعادة السجلات التي تحقق شرطا واحداً من عدة شروط، حيث يعني المعامل Or أن استعادة البيانات تتم إذا تحقق أي شرط من الشروط

* استخدام معامل NOT

استخدام المعامل NOT يمكننا من استثناء سجلات معينة من اختيارنا، حيث يمكننا من استرجاع السجلات المعاكسة لاختيارنا.

|  |  |
| --- | --- |
| المعامل Operator | الترتيب Order Evaluated |
| جميع عوامل المقارنة | 1 |
| NOT | 2 |
| AND | 3 |
| OR | 4 |

شكل رقم (17) يوضح قواعد أسبقية المعاملات

* ترتيب النتائج باستخدام Order by

ترتب السجلات باستخدام Order by وفقا لقيم العمود الذي نحدده ترتيباً تصاعدياً أو تنازليا، ويمكن الترتيب بناء على قيم عمود أو عدة أعمدة، وعند كتابة جملة Order by يجب إتباع الآتي:

* Order by تتبع جملة from{table name} مباشرة في حالة عدم وجود Where {Condition(s)}
* Order by تتبع جمل Where {Condition(s)} مباشرة.
* يوجد لدينا خياران عند استخدام Order by
  1. Order by {column name} ASC: يسمح هذا الخيار بالترتيب التصاعدي
  2. Order by {column name} desc: يسمح هذا الخيار بالترتيب التنازلي.
* يمكن الترتيب على أساس عدة أعمدة، ويجب الفصل بين اسماء الأعمدة بفاصلة (,) .
* يوجد لدينا طريقتان لكتابة جملة Order by وهما
  1. عن طريق اسم العمود
  2. عن طريق رقم العمود النسبي (موقعه في جملة SELECT)
* القيمة Null وتأثيراتها

تستخدم القيمة null للإشارة إلى عمود لا يحتوي على بيانات بمعني وجود قيم فارغة داخل الحقول ومن أشهر عيوب القيمة null صعوبة إجراء أي عمليات حسابية على سجلات تحتوي بعضها على قيم فارغة.

* استخدام القيمة Null
* تستخدم القيمة null للإشارة إلى عمود لا يحتوي على بيانات، ولا تعني القيمة null الصفر أو المسافة الفارغة blank space
* عند استخدام القيمة null في التعبيرات الحسابية يكون الناتج null.

فصل ثالث: الجداول والاستعلامات والعمليات عليهما:

**الربط بين جدولين أو أكثر**

كثيراً ما نحتاج إلى استرجاع بيانات من جدولين أو أكثر في استعلام واحد Query، لإجراء مثل ذلك يتطلب منا إنشاء علاقة بين الجدولين أو الجداول، حيث تكمن القوة الحقيقية لقاعدة بيانات علائقية في قدرتها على استرجاع سجلات من جداول مختلفة في استعلام واحد، وتوجد لدينا العديد من الأسباب التي تدعونا إلى إنشاء علاقة بين الجداول منها:

* دمج الأعمدة من جدولين أو أكثر.
* اختيار أعمدة موجودة في جدول واحد بناء على شرط يطبق على عمود أخر (self-join).

**استرجاع بيانات من جدولين أو أكثر (Displaying Data from Multiple Tables)**

عند استرجاع بيانات من جدولين أو أكثر هناك عدة اعتبارات يجب مراعاتها

1. وضع اسماء الأعمدة في جملة Select.
2. وضع اسماء الجداول في جملة From، على أن يتم الفصل بينها بالفاصلة.
3. إضافة شرط الربط في جملة Where.
4. الصيغة العامة لاسترجاع بيانات من جدولين

SELECT table1.column, table2.column

FROM table1, table2

WHERE table1.column1 = table2.column2;

**استخدام وإنشاء استعلام فرعي(Sub query)**

تمكننا SQLمن جعل الاستعلامات متداخلة مع بعضها البعض بشكل نموذجي، بحيث ينتج عن الاستعلام الداخلي (inner query) قيماً تفحص في قسم الشرط للاستعلام الخارجي(outer query) لمعرفة فيما إذا كان الشرط سيتحقق عندها أم لا [[6]](#footnote-6)

**كيف تعمل الاستعلامات الفرعية**

نفترض أنك تريد كتابة استعلاما لاسترجاع من يأخذ مرتباً أكثر من محمد؟ نجد أننا نحتاج إلى استعلامين

الأول: ما هو مرتب محمد؟

الثاني: البحث عمن يأخذ مرتباً أعلى من محمد؟

ولحل مثل تلك المشكلة لابد من دمج الاستعلامين معا بمعنى استخدام استعلام داخل استعلام أخر، حيث يعيد الاستعلام الداخلي أو الفرعي يعيد قيمة يستخدمها الاستعلام الرئيسي أو الخارجي لمعرفة فيما إذا كان سيتحقق الشرط عندها أم لا.

* ينفذ الاستعلام الفرعي أو الداخلي أولاً ثم ينفذ الاستعلام الرئيسي أو الخارجي.
* يولد أو يعيد الاستعلام الداخلي نتيجة يستخدمها الاستعلام الخارجي لمعرفة ما إذا سيتحقق الشرط عنها أم لا.

الصيغة العامة لعمل واستخدام الاستعلامات الفرعية Sub queries.

**أنواع الاستعلامات الفرعية Types Of Sub queries**

* يوجد لدينا نوعين من الاستعلامات الفرعية وهي كالتالي:
* 5-2-1 الاستعلامات الفرعية التي تعيد سجل واحد
* single-row sub queries))
* 5-2-2 الاستعلامات الفرعية التي تعيد أكثر من سجل واحد
* Multiple-row sub queries))
* 5-2-3 الاستعلامات الفرعية التي تعيد أكثر من عمود واحد
* Multiple-column sub queries)
* الاستعلامات الفرعية التي تعيد سجل واحد
* كما سبق وتعلمنا أن single-row sub queries)) تعيد سجل واحد فقط، ويستخدم ذلك النوع من الاستعلامات مع عوامل المقارنة الموضحة بالجدول التالي:

|  |  |
| --- | --- |
| المعني Meaning | المعامل Operator |
| يســـاوي | = |
| أكبر من | > |
| أكبر من أو يساوي | >= |
| أقل من | < |
| أقل من أو يساوي | <= |
| لا يساوي | <>Or! = |

شكل رقم (18) يوضح عوامل المقارنة

**الاستعلامات الفرعية التي تعيد أكثر من سجل واحد**

تعيد الاستعلامات الفرعية متعددة السجلات أكثر من سجل، وتتطلب هذه الأنواع من الاستعلامات الفرعية معاملاً يمكن استخدامه لتقييم عدة قيم، فعلى سبيل المثال لا الحصر يمكننا استخدام المعامل IN مع الاستعلامات الفرعية متعددة الأسطر لأنه يقيم مصفوفة من القيم، والجدول التالي يوضح المعاملات التي يمكننا استخدامها لتقييم الاستعلامات متعددة السجلات.

شكل رقم (19) يوضح المعاملات التي يمكن استخدامها لتقييم الاستعلامات متعددة السجلات

|  |  |
| --- | --- |
| الوصف | المعامل Operator |
| يكون المحدد مساوياً لأي من القيم التي يعيدها الاستعلام الفرعي لهذا الشرط حتى يصبح محققا(True) | IN |
| يقارن المحدد بكل قيمة يعيدها الاستعلام الفرعي، ويكون الشرط محققا (True)إذا كان أي عنصر من المجموعة يحقق الشرط | ANY |
| يقارن المحدد بكل قيمة يعيدها الاستعلام الفرعي، ويكون الشرط محققاَ | ALL |

* < any: تعني أقل من minimum
* > any: تعني أقل من maximum
* = any: تعادل تكافئ معامل IN

**إضافة سجل واحد إلى الجدول**

كما وضحنا سابقا أن عبارة insert لها صيغتين الأولي والتي نحن بصددها تمكننا من إضافة سجل واحد إلى الجدول في نفس الوقت، والصيغة العامة لها تأخذ الشكل التالي:

INSERT INTO *table* [(*column* [*, column...*])]

VALUES *(value* [*, value...*])*;*

مما سبق نجد أن الصيغة العامة لعبارة insert تتألف مما يلي:

* Insert, into, values هي كلمات محجوزة في sql.
* Table ويشير إلى اسم جدول وحيد في قاعدة البيانات ، وبالتحديد الجدول الذي نريد إدراج السجلات به.
* Column: وتشير إلى أعمدة الجدول.
* لائحة من القيم محاطة بين قوسين تقوم sql بإدراجها في الأعمدة المقابلة لها على الترتيب.
* يجب أن تتطابق الأعمدة والقيم المقابلة لها في نوع البيانات data types.
* لائحة القيم لابد أن تأخذ نفس ترتيب الأعمدة في الجدول المراد إضافة السجلات إليه.
* لابد من وجود علاقة واحد إلى واحد بين الأعمدة ولائحة القيم، بمعني أن توجد قيمة واحدة من أجل كل عمود وبنفس الترتيب.
* قد نجهل قيمة أحد الحقول، وذلك بكتابة كلمة NULL، بشرط ألا يكون العمود المقابل للقيمة معرفاً بالخاصية NOT NULL.
* قد تسبب INSERT مشكلة في التجانس والتكامل المرجعي (وسيتم تناول ذلك عند إنشاء العلاقات بين الجداول لاحقا في هذا الفصل).
* لابد من وضع الأحرف النصية والتاريخ بين علامتي تنصيص مفردة' '.
* للتأكد من تنفيذ العبارة السابقة وإضافة السجل إلى جدول emp قم بتنفيذ العبارة التالية:

SQL>Select \* From emp;

**إضافة عدة سجلات إلى الجدول:**

نلاحظ أنه ليس من المنطقي أن تقوم بإدخال 20,000 سجلاً مثلاً في أحد الجداول باستخدام الصيغة الأولي من عبارة insert، ولإضافة عدة سجلات في نفس الوقت تمكننا الصيغة الثانية من عبارة insert بإضافة عدة سجلات إلى جدولنا مختارة من جدول آخر، والصيغة العامة لها هي:

INSERT INTO *table(*[(*column* [*, column...*]*)*

Select ([(*column* [*, column...*])])

[where conditions]);

مما سبق نجد أن الصيغة العامة لعبارة insert تتألف مما يلي:

1. Insert, into هي كلمات محجوزة في Sql.
2. Table ويشير إلى اسم جدول وحيد في قاعدة البيانات ، وبالتحديد الجدول الذي نريد إدراج السجلات به.
3. Column: وتشير إلى أعمدة الجدول.

عبارة select: هي عبارة select عادية تستخدم لانتخاب الأعمدة التي ستنسخ محتوياتها إلى الجدول المراد إضافة السجلات إليه.

**حذف بيانات موجودة في قاعدة البيانات (Deleting Record)**

نستخدم عبارة DELETE لحذف سجلات من جدول ما، وتأخذ الصيغة العامة لعبارة DELETE الشكل التالي:

مما سبق نجد أن الصيغة العامة لعبارة Delete تتألف مما يلي:

SQL>DELETE [FROM] *table*

[WHERE *condition*];

1. Delete: هي أحد الكلمات المحجوزة في Sql.
2. From: هي أحد الكلمات المحجوزة في Sqlويليها اسم الجدول المراد حذف السجلات منه

WHERE: هي أحد الكلمات المحجوزة فيSql ويليها شرط حذف السجلات ، وهي عبارة اختيارية للحد من عدد السجلات المحذوفة ، فإذا تم حذف عبارة where من جملة delete فإن هذا يعني حذف كافة السجلات.[[7]](#footnote-7)

**تعديل البيانات الموجودة في قاعدة البيانات**

Using the update command))

يمكننا استخدام update لتعديل القيم الموجودة في جدول ما، والصيغة العامة لعبارة update تأخذ الشكل التالي:

SQL> UPDATE *table*

SET *column* = *value* [, *column* = *value*]

[WHERE *condition*];

مما سبق نجد أن الصيغة العامة لعبارة (update)تتألف مما يلي:

1. Update: من الكلمات المحجوزة في, وهو أمر بتحديث البيانات فيSql.
2. Table: ويشير إلى اسم الجدول المراد التعديل في بياناته.
3. Set: وتحتوي على سلسلة من عمليات التحديث ، حيث نقوم بوضع اسم العمود الذي نقوم بتعديل بياناته كمعامل أول ، ويحتوي المعامل الثاني على القيمة الجديدة ، ويفصل بين المعاملين إشارة المساواة.
4. Where: وهي عبارة اختيارية يتم استخدامها لتحديد السجلات التي يراد التعديل بها ، فإذا تم حذف عبارة Where فإن التعديل سيتم على كل سجلات الجدول.

**تحديث كافة السجلات (Update ALL Record)**

يمكننا تحديث كافة السجلات من جدول ما بحذف عبارة where من جملة Update فإن هذا يعني تحديث كافة السجلات، والصيغة العامة لها هي:

SQL> UPDATE emp

2 SET sal = 6000

3 WHERE empno = 555;

1 row updated.

SQL> UPDATE *table*

SET column = value [, column = value];

**تحديث سجلات محددة** (Update Specific Record)

**كيفية إنشاء جداول وإضافة محددات من نوع (primary key، foreign key)**

مما سبق نجد أنه يوجد في لغة SQL ثلاث عبارات تسمي بعبارات تعريف البيانات وهي تسمح لنا بتعريف ومحي وتعديل الجداول في قاعدة البيانات، وهذه العبارات هي:

1. CREATE: تستخدم لتعريف جدول جديد في قاعدة البيانات.
2. DROP: تستخدم لمحي جدول موجود من قاعدة البيانات.
3. ALTER: تستخدم لتغيير بنية الجدول الموجود.

قواعد البيانات العلائقية قادرة على تنفيذ كافة عبارات SQL في أي وقت بشرط أن يكون له الحق في تنفيذ تلك الأوامر والعبارات، فلكي تقوم بإنشاء ومحي وتعديل الجداول لابد من أن تتأكد من مدير قاعدة البيانات أنه يمنحك حق تعريف ومحي وتعديل الجداول.

**إنشاء وتعديل وإسقاط الجداول**

**إنشاء الجداول (create table)**

تمكننا عبارة create من إنشاء الجداول في SQL، وتأخذ عبارة Create الشكل التالي[[8]](#footnote-8):

SQL> CREATE TABLE [*schema*.]*Table*

(*column* *datatype* [DEFAULT *expr*]);

مما سبق نجد أن الصيغة العامة لعبارة Create تتألف مما يلي:

1. create: من الكلمات المحجوزة فيSql،وتخبر DBMS بإنشاء كائن في قاعدة البيانات.
2. Table: من الكلمات المحجوزة فيSql، وتخبر DBMSبنوع الكائن المراد إنشائه في قاعدة البيانات.
3. Table name: ويستخدم لتحديد اسم الجدول ، ويجب أن يبدأ اسم الجدول بحرف نصي ويمكن أن يحتوي أحرف وأرقام والشرطة ،ويجب ألا يتجاوز طوله 30 حرفا ،و لا يسمح بالفراغات ،ولكن يسمح بالشرطة السفلية (underscore).
4. Column names: وتستخدم لتحديد اسماء الأعمدة المكونة للجدول المراد إنشائه، وتنطبق نفس القواعد الخاصة بالتسمية على اسم الجدول على اسماء الأعمدة ، ويجب أن تحاط مجموعة تعريف الأعمدة بين قوسين.
5. Column data type: يحتوي تعريف الأعمدة على نوع البيانات ،وطولها ودقتها.
6. Constraints: وهي إعدادات اختيارية يمكن استخدامها لتمثل قيود للمحافظة على تكامل قاعدة البيانات.

وهناك عدة اعتبارات يجب مراعاتها عند إنشاء الجداول وهي كالتالي:

* 1. يجب وضع أقواس تضم اسماء الأعمدة ونوع بياناتها.
  2. يتم وضع فاصلة (, ) بين تعريف أي عمود وأخر.
  3. يجب أن يكون اسم العمود فريدا داخل الجدول.
  4. لا يمكن استخدام الكلمات المحجوزة كأسماء أعمدة في الجدول.
  5. يتم استخدام الفاصلة المنقوطة لإخبار SQL أن الجملة انتهت وجاهزة للتنفيذ.
  6. الجدول التالي يوضح لنا أهم أنواع البيانات التي يمكن أن نستخدمها عند إنشاء الجداول في SQL

شكل رقم (20) يوضح أهم أنواع البيانات التي يمكن أن نستخدمها عند إنشاء الجداول في SQL

|  |  |
| --- | --- |
| نوع البيانات | الوصف |
| CHAR(n) | يعرف هذا النوع من البيانات عمود من نوع البيانات الحرفية بطول (n) حرف، وجميع الحقول متساوية في الطول، n=<255 |
| VARCHAR(n) | يعرف هذا النوع من البيانات عمود من نوع البيانات الحرفية بطول (n) حرف، وتختلف الحقول في الطول وتتجاوز 255 |
| NUMBER(n) | يعرف عمود من نوع بيانات من نوع رقم، مع حد أقص لعدد الأرقام (n)، ويمكن أن تأخذ (n) كحد أقصى 105 |
| NUMBER(n,d) | يعرف عمود من نوع البيانات number، ويحتوي عدد أرقام قدره (n)، والتي تتضمن القيمة (d) التي توضح عدد الأرقام العشرية بعد الفاصلة. |
| INTEGER | يشبه نوع البيانات NUMBER، ولكنه لا يأخذ غير القيم الرقمية الصحيحة، ولا يقبل أي أرقام عشرية بعد الفاصلة. |
| LONG | يعرف عمود من نوع البيانات الحرفية، وأقصي طول له 65.535، ولا يمكن تعريف سوي عمود واحد من هذا النوع في الجدول، ولا يمكن استخدام هذا العمود ضمن عبارات Whereأو في الاستعلامات الفرعية أو الدوال أو التعابير أو ضمن الفهارس. |
| RAW(n) | يعرف هذا النوع عمود يحتوي سجل بيانات بطول قدره n |
| SMALLINT | يعرف عمود يخزن أرقاماً موجبة وسالبة صغيرة تقع في النطاق +32, 767)، (-32, 767 |
| DATE | يعرف عمود نوع بياناته تاريخ. |
|  |  |

**تعديل الجدول (altering the table)**

يستخدم alter table في تعديل بنية جدول موجود وتغيير محددات الجدول بعد إنشائه وفيما يلي الخيارات التي يمكن استخدامها مع الأمر alter:

1. Add: يستخدم هذا الخيار لإضافة أعمدة جديدة أو شروط إلى الجدول ويأخذ الشكل التالي:

SQL> ALTER TABLE *table*

ADD (*column datatype* [DEFAULT *expr*]

[, *column datatype*]...);

1. Modify: ويستخدم لتغيير محددات موجودة ويأخذ الشكل التالي:

SQL>ALTER TABLE *table*

MODIFY(*column datatype* [DEFAULT *expr*]

[, *column datatype*]...);

1. Disable: يستخدم لإلغاء تفعيل شرط ما في الجدول.
2. Enable: يستخدم لإعادة تفعيل شرط كنا قد قمنا بإلغاء تفعيلة.
3. Drop: لإزالة شرط ما بشكل دائم من الجدول.
4. Set unused: يستخدم لتحديد عمود أو أكثر كأعمدة غير مستخدمة ، بحيث يمكن حذف تلك الأعمدة من الجدول عندما تصبح مصادر النظام ملائمة لذلك.
5. Drop column: يستخدم لإزالة عمود ما من الجدول

**إضافة محددات إلى الجدول من نوع (Primary key, foreign key)**

**المحافظة على تكامل قاعدة البيانات**

يوجد في أوراكل بعض الأدوات لحماية قاعدة البيانات، تسمي هذه الأدوات بالشروط Constraints، وتقوم هذه الشروط بعدة وظائف هامة منها:

1. التأكد من المفتاح الأساسي أو الرئيسي unique.
2. فرض التكامل المرجعي بمعني التأكد من أن السجلات الأبناء الموجودة في جداول مرتبطة، تمتلك سجل أب.
3. التأكد من أن الأعمدة المرتبطة بالشروط تحتوي دائما قيمة بداخلها NOT NULL.
4. التأكد من وضع القيمة الافتراضية DEFAULT VALUE في عمود ما.
5. التأكد من عموداً ما يحتوي على قيمة ما، وأن هذه القيمة موجودة ضمن نطاق محدد من القيم The Check Constraint

**الشرط CHECK (فحص وجود قيمة)**

يستخدم هذا الشرط للتأكد من أن قمة ما لعمود ما تقع بين قيم مجموعة محددة، عندما تستخدم الشرط CHECK سيقوم أوراكل بمقارنة أية قيمة يتم إدخالها مع مجموعة القيم المحددة في المجموعة.

**الشرط not null (ليس فارغاً)**

نستخدم الشرط not null للتأكد من أن العمود دوماً سيحتوي قيمة بداخله، وخاصة ما يتم وضع هذا الشرط على عمود أو أعمدة المفتاح الأساسي للجدول، لكي تقوم بوضع الشرط not null على عمود ما، قم بوضع العبارة not null بعد نوع بيانات العمود.

**الشرط unique (وحيد)**

يستخدم الشرط unique للتأكيد من أن القيمة الموجودة بالعمود المشروط هي قيمة وحيدة في جميع سجلات الجدول، بمعني عدم السماح بتكرار القيم، فيقوم هذا الشرط بعمله عبر إنشاء فهرس وحيد (unique index) على هذا العمود، يعتبر الشرط unique أداة جيدة للمحافظة على هذه الخاصية في قاعدة البيانات، فعلى سبيل المثل يمكن اعتبار عمود "number\_social\_security""الضمان الاجتماعي " الموجود في جدول employee لا يحتوي سوي قيمة وحيدة، مع العلم بأن العمود emp\_id " " هو المفتاح الأساسي في الجدول بمعني أنه لابد أن يحتوي قيما فريدة ووحيدة و لا يسمح بالتكرار.

**الشرط primary key(المفتاح الأساسي)**

يستخدم الشرط primary key للمحافظة على تكامل عمود أو أعمدة المفتاح الأولي، حيث يجعل هذا الشرط القيم الموجودة في المفتاح في عمود مشروط به تتمتع بالشرطين

Not null, unique معاً، فعند تعريف هذا الشرط يتم إنشاء فهرس وحيد unique index ضمني وإنشاء شرط not null ضمني أيضاً على العمود أو الأعمدة المشروطة بشرط المفتاح الأساسي، ويمكن اعتبار تعريف هذا الشرط كجزء من تعريف العمود أو جزء من تعريف الجدول ، حيث إذا تم تعريف هذا الشرط على عدة أعمدة (حالة مفتاح أساسي مركب) فإن تعريف هذا الشرط سيكون جزءاً من تعريف الجدول.

**الشرط foreign key (المفتاح الخارجي)**

يستخدم الشرط foreign key كأداة للتأكيد من الجداول المرتبطة لا تحتوي على سجلات غير منتمية إلى سجل أب، ويحتوي الشرط foreign key على خيار يسمح بحذف كل السجلات الأبناء المرتبطة مع سجل الأب عند حذف سجل الأب وهذا الخيار هو

ON DELETE CASCADE،يمكننا إنشاء مفتاح الربط الخارجي كجزء من تعريف العمود عند إنشاء الجدول ، حيث يتم استخدام كلمة REFERENCES ويليها اسم الجدول الذي يحتوي السجل الأب وذلك يكون بعد تعريف العمود ولسنا بحاجة لذكر اسم العمود من الجدول الأب ، حيث إذا لم تقوم بكتابة اسم العمود الموافق من الجدول الأب يفترض أوراكل أنه هو المفتاح الأساسي.

**تعديل تعريف شرط ما (Modifying Constraints)**

يمكننا إضافة شرط للجدول وذلك بعد إنشائه، كما يمكننا إزالة تلك الشروط أو إلغاء تفعيلها أو تفعيلها وذلك باستخدام alter table ويمتلك هذا الأمر عدة خيارات يمكن استخدامها مع الشروط وهي موضحة كالتالي:

1. ADD (إضافة): يستخدم لإضافة شرط جديد إلى الجدول.
2. MODIFY (تعديل): يستخدم لإضافة شرط إلى عمود موجود.
3. DROP (حذف) : يستخدم لحذف الشرط من الجدول.
4. DISABLE (إلغاء تفعيل): يسمح بإدخال البيانات دون النظر في فيما إذا كانت تحقق الشرط أم لا ، ولكن مع بقاء الشرط كما هو في قاموس البيانات.
5. ENABLE (تفعيل): يقوم بالتحقق من أن جميع البيانات التي يتم إدخالها والموجودة مسبقاً تحقق الشرط.[[9]](#footnote-9)

**القسم العملي**

**الباب الأول:**

التعليمات والواجهات في Android Studio

فصل أول: الواجهات والأدوات

المكونات الأساسية في لغة ال(XML): أولاً هناك نوعان من الخيارات نستطيع التعامل فيها مع ال(activity):

1. Design : هي النافذة التي تري المستخدم شاشة ال(activity) لكي يصمم شكلها و يضعها في المكان الذي يشعره هو مناسباً فضلاً عن المكوِّنات الأخرى المختلفة .
2. Text: هي النافذة التي تعتبر ترجمة كتابيَّة للكائنات الموجودة على شاشة ال Design.

نستطيع في XML أن نضع الكثير من الأدوات ومنها سنذكر التي استخدمناها في عمل التَّطبيق:

* الأزرار: وهي كما ذكرنا يتحكَّم بها المستخدم ويتحكَّم بآليَّة عملها وبوظائفها.
* أزرار الصُّورة: هي أزرار بعملها ولكن في ظاهرها هي صورة.
* النصوص: هي ما يظهر على التَّطبيق من كتابات وتوجيهات تساعد المستخدم وتخدمه في احتياجاته المختلفة حسب نوعيَّة التَّطبيق.
* Scrollview: هي الأداة التي تخول المستخدم ليقلِّب النَّص على الشَّاشة عندما يكون طويلاً أو أكبر من شاشة الموبايل.
* Menu.XML: هي القائمة التي تظهر للمستخدم عند ضغطه على زر القائمة.

نتحدَّث الآن عن بنية ساعدتنا في إظهار المعلومات الأساسيَّة عن التطبيق Manifests :

والمقصود هنا بالمعلومات الأساسية هي اسم التَّطبيق وترتيب الإصدار للتطبيق وأسماء شاشات العرض المستخدمة Activities والوسائط الموجودة بداخله Intents.

يوجد أيضاً ملف ضروري في برمجة أندرويد وهو المصادر res وهو اختصار لكلمة resources وهو أيضاً بلغة XML فيه:

Drawable: يحوي الصُّور المستخدمة في التطبيق حيث يتم إدخالها يدوياً من قبل المستخدم وتُدخل بدقَّات مختلفة حسب الطَّلب.

Layout: تحوي صفحات بلغة XML كما ذكرنا وهي ذاتها ال(activities).

Menu: وفيها القائمة المنسدلة التي تحوي خصائص التَّطبيق عند الضَّغط على الزر اليمين، أو من الزر في الموبايل.

Values: وتحوي القيم المخزَّنة فيها للتطبيق من نوع السَّلاسل المحرفيَّة وهي تقوم بتسهيل العمل على المبرمجين

حيث يستطيع المبرمج ترجمة تطبيقه الخاص إلى عدد اللغات الي يريد عن طريقها وتقوم بتخفيف الأسطر عن البرنامج وجعل هذه النُّصوص ملفَّاً مرفقاً والأهم من ذلك عامَّاً يستطيع الكلُّ استخدامه.

وهناك الطبقة الثَّانية التي وفَّرتها برمجة الأندرويد للمستخدمين وهي JAVA عن طريق الأصناف التي يتم ربطها بملفات الXML . تتيح هذه الأصناف التحكُّم بالبنية الدَّاخليَّة للتطبيق وليس الخارجيَّة (فهذه وظيفة Layout) حيث تربط الكائنات الموجودة في التطبيق وتقوم بإنشاء الأحداث عليها.

فصل ثان: التعليمات في Android Studio

تعليمات مهمَّة استخدمناها:

مجموعة التَّعليمات التَّالية ستقوم بربط زر صورة بحدث فتح موقع فيس بوك على المستعرض:

Image Button b1 = (Image Button) findViewById (R.id.F\_IB);

عرَّفنا صورةً تشكِّلُ زرَّاً b1 وقبل أن نبدأ يجب أن نعطي لمحة عن مبدأ القصر:

صنف الأزرار الموجود في التعليمة السَّابقة على يسار إشارة الإسناد يرث من صنف View لكي يرجع قيمة View والهدف منها موضَّح على يمين إشارة الإسناد فهو يبحث عن id ل (View) وليس (Button). هذه التَّعليمة تربط بين الزِّر الموجود في ملف XML <التَّصميم> عن طريق البحث عن معرِّفه id الذي يوجد في ملفات Text في XML

b1.setOnClickListener (new View.OnClickListener ()

هنا نرى المستمع للنقر ولتسهيل مفهومه حاول أن تتخيَّل أنَّك تضغط على زر الجرس فكلَّما ضغطت على الزِّر أتتك استجابة من الجرس فيرن كذلك هذا التَّابع فهو ينتظر من المستخدم النَّقر على الزر المعرَّف بال id السَّابق لكي يقوم بالاستجابة وهنا نرى تكملة التعليمة الممثَّلة بالاستجابة:

{

@Override

public void on Click(View v)

Intent i = new Intent(android.content.Intent.ACTION\_VIEW,Uri.parse("<http://www.facebook.com/)>")

});

* عند بداية أي تابع نجد البادئة Override حيث تعطي المستخدم إيعازاً بإنشاء أمر برمجي.
* صرَّحنا عن تابع on Click الذي يحتاج إلى قيمة من نوع View
* صرَّحنا بغاية التَّابع في إنشاء حدث. ACTION\_VIEW ويتم إرجاع القيمة من Url إلى Uri وهو المنفذ إلى المستعرض لكي يقوم بعرض الملفات.

النتــــــــائج:

1. لغة برمجة JAVA هي من اللغات كائنية التوجه بالتالي هي بيئة تفاعلية في نظام أندرويد عبر برمجيات المحاكاة الحاسوبية للتجارب العملية وبرمجيات الدراسة الافتراضية، والتعلم عن بعد.
2. لغة JAVA الكائنية التوجه سيكون لها مستقبل باهر بالنسبة للغات الأخرى غير كائنية التوجه لأنها تعمل عقل المبرمج بقضايا منطقية من الحياة الواقعية وتترجم أفكار المبرمجين (الواقعية) إلى بيئة وهمية مترابطة متماسكة.
3. SQL في النهاية هي لتنظيم التطبيق من أجل ضمان النتائج الأمثل التي ترضي المبرمج.
4. ظهور كائنية التوجه جعلت من البرنامج سواء البسيط أو المعقد بيئة وهمية تحتاج إلى المنطق لحل المسائل التي كانت تعتبر مستحيلة الحل بالنسبة للبرمجة الإجرائية والهيكلية.
5. إن مستقبل الربط بين نظم إدارة قواعد البياناتSQL) ) و لغات البرمجة و خاصة كائنية التوجه سيشكل أحد أبرز معالم القرن الواحد والعشرين و خاصة التطبيقات التي تتضمن كم معلومات هائلا حيث ستشكل نظم إدارة قواعد البيانات الهيكل التنظيمي و البنيوي للتطبيقات الحديثة العالية التقنية.

المقترحات:

1. استكمال العمل على التطبيق لأنه بادرة إيجابية لتوسيع دائرة التعريف بالمركز من خلال الاعتماد على قدرة انتشار نظام أندرويد الكبيرة وبساطته إضافة إلى عكس الصورة الحقيقية لقدرات طلاب المركز الوطني للمتميزين على مواكبة التكنولوجيا وتسخيرها لخدمتهم.
2. متابعة تطوير التطبيق عن طريق إضافة الموديل الذي سيتم تحميله قريباً على شبكة الإنترنت بحيث يمكن للطالب تصفح الدرس عن طريقه وأيضاً أن توضع الدروس على شكل Backup أو صور ضمن ملف وضغطه وتنظيمهم في قواعد بيانات مع إمكانية تحميلهم إضافة لمجموعة معلومات تدعم خدمة الاستعلام والبحث في المركز.
3. دعم تعليم برنامج Android studio أو JAVA لمساعدة الأجيال القادمة في مواكبة العصر والتطورات دون الحاجة للرجوع إلى الخلف فالعصر القادم هو عصر التقانة والتكنولوجيا العالية وأيضاً لتأمين منطلق علمي يمكن الجيل القادم من الاستمرار في العمل بناءً على قاعدة وأسس متينة وعلمية.
4. الاستمرار الدائم في تطوير التطبيق لضمان زيادة مدى فعاليته التعريفية وضمان تزامنه مع التكنولوجيا.

**الخاتمة:**

إن الهدف النهائي من تطبيقنا هذا هو توسيع دائرة التعريف بالمركز الوطني للمتميزين من خلال توسيع مجالات تواصله بالانتقال إلى تطبيق أندرويد ما يتيح معرفته لمعظم الجيل الحديث والشاب خصوصاً بالإضافة إلى إثبات جوهر وحقيقة إنشاء المركز الوطني للمتميزين والذي يهدف لتنمية المواهب ودعمها لتصبح قادرة على البحث والإبداع والعودة بأفضل نتائج ممكنة.

أيضاً نسعى من خلال هذا التطبيق إلى إدخال المركز الوطني للمتميزين في مجال التقنية والتكنولوجيا الحديثة وإدخالها ضمن الإطار العام للمركز لما تشكله من عصب للحياة الحالية والمستقبلية، وكذلك إدخاله عالم المنافسة في التطبيقات الحديثة من خلال أي بيئة تطوير (الأندرويد حالياً)، بحيث يشكل التطبيق (الحالي والمستقبلي) أحد عوامل التعريف بالمركز ويشكل لاحقاً نقطة ارتكاز تكنولوجية تعود بالنفع العلمي والمعرفي والاجتماعي على المركز الوطني للمتميزين.

**إهداء**

إلى من قدموا لنا الأمل والدعم المعنوي ... إلى من شجعونا ووقفوا بجانبنا في أصعب اللحظات ... إلى من أخذوا بيدنا، وأناروا درب العلم أمامنا ... إلى آبائنا وأمهاتنا.

إلى من كان عوناً وسنداً لنا ... إلى من أمدنا بالمعلومات القيمة التي ساهمت في إنجاز هذا المشروع ... إلى مشرفة مشروعنا المدرسَّة ميس درويش.

إلى راعي التميز والعلم والإبداع ... سيادة الرئيس الدكتور الفريق المفدّى بشار حافظ الأسد.

إلى صاحبة الفضل الأكبر في تنشئة جيل متعلم ... إلى من وضعت حجر الأساس لبناء "سورية العلم".... إلى سيدة سورية الأولى السيدة أسماء الأسد.

**وأخيــــــــــــــراً:**

إلى التي عانت وقدمت الكثير لتخرج أجيالاً سلاحها العلم في سبيل بناء صرح حضارة راقية متقدمة ....... إلى " أمنا الغالية سورية ".

**المصادر والمراجع:**

**الموسوعات:**

1. Microsoft® Encarta® 2009.

**المصادر العربية:**

1. الإكسير في C++، تأليف سلطان محمد الثبيتي، الطبعة الأولى، 1426ه.

**المصادر الأجنبية:**

1. Thinking In C#, Prentice Hall, Informatics, Larry O'Brien and Bruce Eckel, New Jersey 07458,Revision 1.0.

**المواقع الإلكترونية:**

1. <http://venturebeat.com/2010/10/27/google-exec-android-was-best-deal-ever/>

تاريخ الدخول 28 / 4 / 2015 الساعة: 4:35.

1. الجامعة الافتراضية السورية: [www.svuonline.org](http://www.svuonline.org)

تاريخ الدخول: 5 / 4 / 2015 الساعة: 2:20.

1. Oracle.com outage تاريخ الدخول: 8 / 4 / 2015 الساعة 12:30.
2. حزمة الأندرويد والأندرويد ستديو

<https://developer.android.com/sdk/installing/>

<http://javatechig.com/android/installing-android-studio>

تاريخ الدخول: 29/ 4 / 2015 الساعة: 4:35.

1. مقال عن الأندرويد

<http://officialandroid.blogspot.com/>

تاريخ الدخول: 8/ 4 / 2015 الساعة: 1:30.

1. طبقات الأندرويد

<https://code.google.com/p/saaf/>

تاريخ الدخول: 25/4/2015 الساعة: 3:30.

1. معهد ماساشوستس: https:// web.mit.edu

تاريخ الدخول 26 / 4 / 2015 الساعة: 4:33.

1. https://Android SQLite database and content provid Tutorial

تاريخ الدخول: 6 / 5 / 2015 الساعة: 4:30.

1. https:// Android Developers/

تاريخ الدخول: 18 / 4 / 2015 الساعة 12:00.

1. <https://Android.database.sqlite>

تاريخ الدخول: 20 / 4 / 2015 الساعة 12:45.

1. <http://venturebeat.com/>

تاريخ الدخول:4/4/2015 الساعة: 4:30 مساءً.

1. <http://www.android.com/history> /

تاريخ الدخول: 11/5/2015 4:38 مساءً.

1. موقع شركة جافا<https://SUNJAVA.com>

تاريخ الدخول 5/3/2015 الساعة: 4:30 مساءً.

1. الإكسير في C++، تأليف سلطان محمد الثبيتي، الطبعة الأولى، 1426هجري. [↑](#footnote-ref-1)
2. Thinking In C#, Prentice Hall, Informatics, Larry O'Brien and Bruce Eckel, New Jersey 07458,Revision 1.0. [↑](#footnote-ref-2)
3. موقع شركة جافا https://SUNJAVA.com/ تاريخ الدخول 5/3/2015 الساعة: 4:30 مساءً [↑](#footnote-ref-3)
4. <http://venturebeat.com/2010/10/27/google-exec-android-was-best-deal-ever/>

   تاريخ الدخول:4/4/2015 الساعة: 4:30 مساءً [↑](#footnote-ref-4)
5. <http://www.android.com/history/>

   11/5/2015 4:38 PMتاريخ الدخول: [↑](#footnote-ref-5)
6. الجامعة الافتراضية السورية: www.svuonline.org تاريخ الدخول: 5 / 4 / 2015 الساعة: 2:20

   Oracle.com outage تاريخ الدخول : 8 / 4 / 2015 الساعة : 12:30

   . [↑](#footnote-ref-6)
7. Android Developersتاريخ الدخول: 18 / 4 / 2015 الساعة 12:00.

   android.database.sqlite تاريخ الدخول : 20 / 4 / 2015 الساعة 12:45 [↑](#footnote-ref-7)
8. معهد ماساشوستس: web.mit.edu تاريخ الدخول 26 / 4 / 2015 الساعة: 4:33 [↑](#footnote-ref-8)
9. W3School : [www.w3School.com](http://www.w3School.com) تاريخ الدخول 28 / 4 / 2015 الساعة :2:35

   Android SQLite database and content provid Tutorial تاريخ الدخول : 6 / 5 / 2015 الساعة : 4:30 [↑](#footnote-ref-9)