هام جدا شرح مبسط عن الكلاس و الأوبجيكت في الجافا. شرح راح يساعدك كثير

سؤال, مقالة, مناقشة, عام, أسئلة-المجموعة-الأول

WassimDaily **13ي**

تبينه: قد يأتي في التقييم النهائي لمسار برمجه تطبيقات الأندرويد. سؤال عن الكلاس او الأوبجيكت. لا تخلوا اسأله المعاهد و الجامعات عنها

مفهوم Class مقابل مفهوم Object

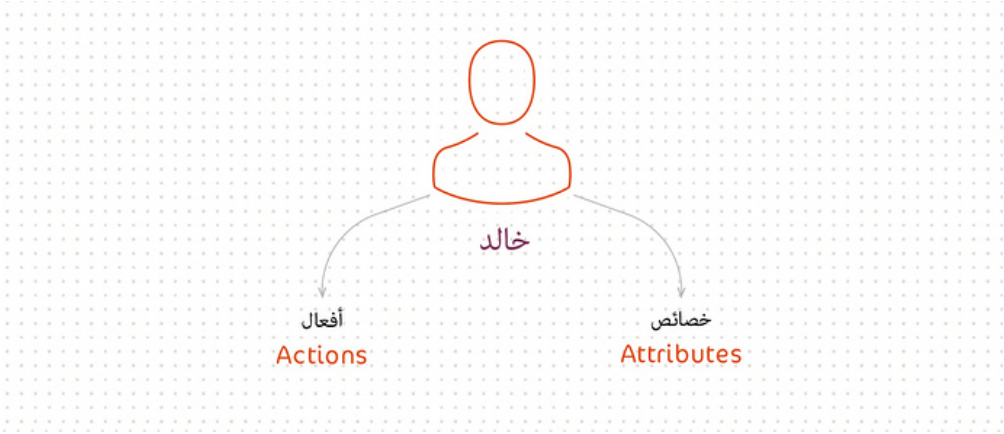
للبدء بشكل قوي و ثابت في عالم برمجة الكائنات أو Object Oriented Programming فنحن نحتاج إلى فهم عدد من القواعد البسيطة التي ستساعدنا بإذن الله تعالى على فهم الكثير من الأمور المتعلقة بهذا المفهوم بشكل جيد و قوي، بحيث يسهل أيضاً كل ما يخص هذا المفهوم من مواضيع متقدمة بنيت على تلك القواعد

**

1. القاعدة الأولى :

** كل شيء عبارة عن كائن، Everything is an object، وهذا يعني أن كل ما تنظر إليه من حولك هو عبارة عن كائن Object، أي السيارة، الطيارة، الكمبيوتر، وحتى أنت كشخص تعتبر كائن. وأما القاعدة الثانية فتقول : كل كائن يتكون من خصائص Attributes و أفعال Actions.

لتوضيح الأمر، دعنا نفرض أن لدينا شخص اسمه خالد كما هو موضح في الشكل 1-1 وهو عبارة عن كائن Object، ووفقاً للقواعد السابقة، سيحتوي على خصائص Attributes، وأفعال Actions.



الشكل 1-1 يوضح شكل الكائن ويوضح الخصائص المرتبطة به Attributes والأفعال Actions التي يقوم بها

لو قمنا باستبدال كائن الشخص (خالد) هنا بأي كائن آخر، فسنجد أن الخصائص والأفعال ستكون موجودة معه وستتغير بناءً على نوع الكائن نفسه، **وهنا نقول أن الكائن مهما كان نوعه، سيتكون من خصائص و أفعال**

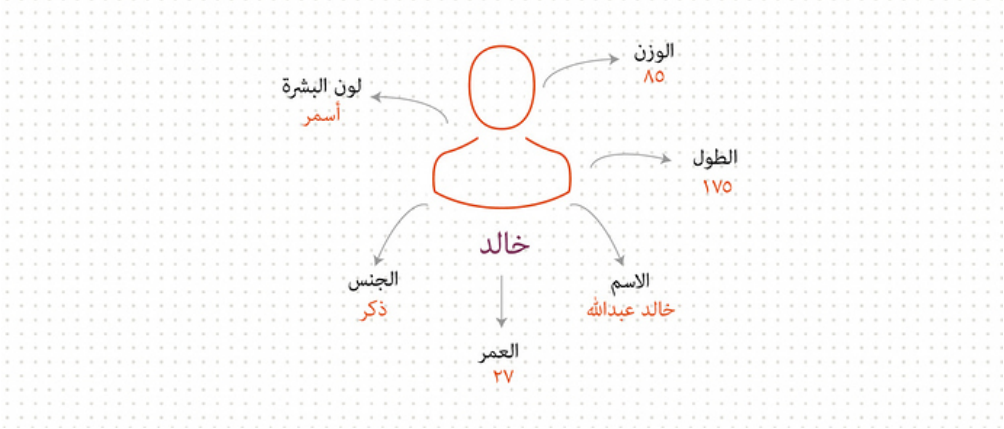
خصائص الكائن Object Attributes

تعلمنا مما سبق، أن كل شيء عبارة عن كائن، وأن كل كائن يتكون من خصائص Attributes وأفعال Actions، وبناءً عليه، سنقوم بالتفصيل في أول مكون من مكونات الكائن، وهو الخصائص Attributes. ويمكننا القول بأنها عبارة عن مواصفات الكائن، بكلام آخر، الخصائص Attributes هي الأشياء التي تأتي على صيغة name = value، ماذا يعني هذا الكلام؟.

الآن لو أردت شراء سيارة، فإنك ستسأل عن سعرها، لونها، وغيرها من المواصفات التي تهمك أثناء الشراء، وكما قلنا أن السيارة عبارة عن كائن، فإن سعرها، ولونها تمثل خصائص Attributes لهذا الكائن (السيارة)، ولكي نوضح المقصود باستخدام صيغة name = value، أو اسم الخاصية = قيمتها، لاحظ كيف نضع المواصفات فيما يلي

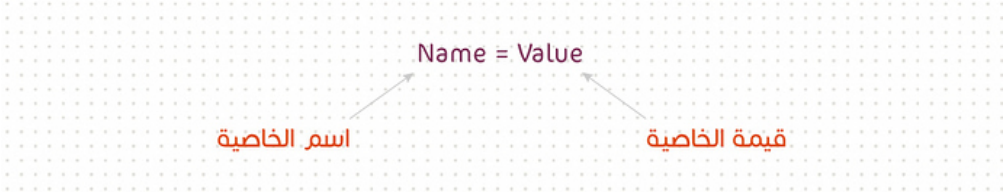
```
car.color = red
car.price = 66500
```

نستنتج مما سبق أن **color** تمثل اسم الخاصية **name**، وأن **red** تمثل قيمة الخاصية **value**، ونفس الأمر ينطبق على الخاصية **price**، ولتأكيد الفهم أكثر، دعنا نقول أننا أردنا كتابة الخصائص **Attributes** للشخص الذي أسميناه خالد، عندها ستكون خصائصه **كما هو موضح في الشكل 1-2.**



الشكل 1-2 قائمة بخصائص Attributes الكائن الذي يمثل شخص اسمه خالد

تعتمد معرفة الخصائص لكائن ما على مدى معرفة الشخص الذي يحاول وصف ذلك الكائن من ناحية المعلومات التي يمتلكها حول ذلك الكائن، ومن ثم كتابتها باستخدام صيغة **name = value** ***كما هو موضح في الشكل 1-3.***



الشكل 1-3 القاعدة العامة لخصائص الكائن Object Attributes

أفعال الكائن Object Actions

سنتحدث هنا عن ثاني مكون من مكونات الكائن، وهو الأفعال **Actions**، وقد تجدها بإسم **Behavior** و **Methods** وغيرها، **ويمكننا القول بأنها عبارة عن الأشياء التي يقوم بها الكائن، أو بكلام آخر، هي الأشياء التي يفعلها الكائن.**

لو عدنا للكائن السابق، السؤال هنا، ما هي الأشياء التي يستطيع أن يفعلها خالد؟، المشي، والجري، والتوقف، وغيرها، وهذه كلها أفعـــال **Actions**، **والشكل 1-4 يوضح قائمة ببعض الأفعال.**

الشكل 1-4 قائمة ببعض الأفعال Actions التي يمكن لخالد القيام بها

يظهر لنا هنا جلياً أن مفهوم الأفعال هو مفهوم بسيط للغاية، كوننا نقوم بكل تلك الأشياء بشكل يومي تقريباً، و قد تم ذكرها على سبيل المثال لا الحصر كون الشخص يستطيع القيام بالكثير من الأفعال، وقس على ذلك لبقية الكائنات من الأنواع الأخرى.

ماهو Class و Object و العلاقه بينهما.

نستنتج من **الشكل 1-5** أن **الكائن Object** عبارة عن نسخة من **Class**، وهذه هي العلاقة، أي أن **Class** قد ننشيء منه عدة نسخ **Instances**، وجميع هذه النسخ هي عبارة عن كائنات **Objects**، ونستطيع أن نستنتج أيضاً أن الكائن لا يمكن الحصول عليه من غير

Class، أي يبنى **Class** أولاً و من ثم تصدر منه النسخ المختلفة من الكائنات.

يمكننا القول أن **Class هو عبارة عن طريقة لوصف الكائنات التي ستنشأ منه**، أو أن **Class** يمثل القالب أو الشكل العام للكائنات التي ستستنسخ منه، وقد يكون مثال صنع الحلويات جيداً في هذا السياق، فقد نقوم بتصميم قالب على شكل نجمة، ومن ثم نصب المادة فيه لتنتج لنا حلوى على شكل نجمة، هنا نقول أن القالب الذي نصب فيه هو **Class**، وأن الحلويات التي تنتج على شكل نجمة هي الكائنات، بحيث تمثل كل حلوى على شكل نجمة كائناً مستقلاً.

بعد هذا التوضيح نقول، يقوم المبرمج غالباً بكتابة **Class ليصف من خلال لغة البرمجة التي يعمل عليها شكل الكائنات التي ستصدر منه، وبعد ذلك يصبح Class مجرد نوع جديد في تلك اللغة Data Type**، ليستخدم مثله مثل بقية الأنواع الأخرى المعرفة مسبقاً في اللغة أو التي أتت ضمن المكتبات البرمجية التي أتت مع تلك اللغة، بكلامٍ آخر، نحن نبني **Class** لنعرف نوع بيانات جديد يصف لنا شكل البيانات المراد تخزينها في ذلك النوع.

التمثيل البرمجي لكل من Class و Object

تدعم لغة Java مفهوم Object Oriented Programming، وهذا يعني أنها لغة تتعامل بمفهوم Classes و Objects، ولكي نعرف الكمبيوتر بشكل كائن ما و تفاصيله، لابد لنا من إنشاء Class يشرح تلك التفاصيل، والكائنات التي ستنتج منه تتكون من خصائص Attributes وأفعال Actions، وعندما نأتي إلى الخصائص Attributes فإنه سيتم تمثيلها بما يسمى المتغيرات Variables برمجياً، وعندما نأتي إلى الأفعال Actions فإنه سيتم تمثيلها كذلك برمجياً بما يسمى الدوال Functions أو Methods.

لتوضيح الأمر، دعنا نفرض أن لدينا Person Class، والذي سيمثل شخص ما، فإنه لابد لنا من تعريف مجموعة متغيرات Variables تمثل الخصائص Attributes مثل name و age وغيرها، وأيضاً لابد لنا من عمل مجموعة من الدوال Methods تمثل الأفعال Actions مثل eat و sleep وغيرها، وهنا نستطيع القول بأن الكائن Object من المفهوم النظري ما هو إلا هو عبارة عن مجموعة من الخصائص Attributes والأفعال Actions، وأن الكائن Object من المفهوم البرمجي هو عبارة عن مجموعة من المتغيرات Variables والدوال Methods.

الشيفرة 1-1 توضح Person Class والذي ستنتج منه الكائنات الأخرى التي تمثل أشخاصاً بعينهم مثل «خالد» سابقاً

```
1. class Person {
2.
3.   public String name;
4.   public int age;
5.   public boolean gender;
6.
7.   public void eat() {...}
8.   public void sleep() {...}
9.   public void run() {...}
10. }
```

الشيفرة 1-1 شيفرة برمجية توضح Person Class مكتوب بلغة Java

لاحظ كيف تم فتح الأقواس من السطر 1 إلى السطر 10، ومن ثم وصف شكل الكائنات التي ستصدر من Person، أي كأننا نقول أن كل شخص سيحتوي على ثلاثة خصائص هي name و age و gender، وثلاثة أفعال يقوم بها هي eat و sleep و run، بعد ذلك فإن إنشاء أي كائن من النوع Person سيحتوي على تلك الأشياء، وهذا هو كل ما في الأمر بصورته البسيطة.

> تنبيه : تم وضع الخصائص Attributes في تعريف Person على أنها public لأغراض الشرح، ورغم أن هذا الأمر لا مشكلة فيه، إلا أنه يتعارض مع ما يسمى بإخفاء البيانات.

بعد أن قمنا بإنشاء Person أصبح بإمكاننا الآن إنشاء عدد لا نهائي من الكائنات التي تكون من النوع Person، والشيفرة 1-2 توضح إنشاء كائن يمثل «خالد» الذي تحدثنا عنه سابقاً.

الشيفرة 1-2 شيفرة برمجية توضح إنشاء كائن أو متغير من Person Class

```
1. Person khaled = new Person();
2. khaled.name = "Khaled";
3. khaled.age = 25;
4. khaled.gender = true;
```

لاحظ المتغير (الكائن) khaled في السطر 1، وكيف استخدمنا كلمة new لإنشاء كائن أو نسخة جديدة من Person، ومن ثم قمنا بتعبئة البيانات الخاصة بتلك النسخة التي تمثل "خالد"، بحيث وضعنا الاسم والعمر والجنس، وهكذا لبقية الكائنات الأخرى التي ستنشأ من Person.

جميع الكائنات أو المتغيرات التي سنقوم بإنشائها ستتبع التفاصيل الموجود في تعريف Person، وهذا يعني أنه في حال قمنا بتعديل بعض الأمور في Person Class فإن جميع الكائنات Objects التي تم تعريفها في البرنامج ستتأثر بذلك التعديل، وهذا الأمر يعتبر ميزة وعيب في نفس الوقت، فهو ميزة في حال أدخلت تحسينات على الشيفرة البرمجية، فسيكون الأمر سريع ومرن، وقد يكون عيب في حال تم تعديله بالخطأ فإنه قد يعطي نتائج غير متوقعة على نتائج البرنامج الذي يستخدمه.

> *تنبيه : لغات البرمجة تختلف في كيفية إنشاء الكائنات، فمثلاً لغة Swift لا تستخدم كلمة new عند الإنشاء، وقس على ذلك بقية اللغات في اختلافها، ويبقى المفهوم واحد، وهذا هو المهم.*

MASAY **13ي**

التقييم أسئلة اختيارات

WassimDaily **13ي**

اي بضبط الاسئله الي بنهايه الدفعه