# ♥ Java ♥

المحاضرة الثامنة عشر الاثنين 4/12/2017

## محددات الوصول modifiers

لدينا أربع محددات وصول:

1. العام public
2. الخاص private
3. المحمي protected
4. الافتراضي package

تأتي محددات الوصول في ثلاث مواضع:

① قبل الصف

وأمام كل مكون من مكونات الصف: ② طرق و ③ معطيات أعضاء

* كل مكون من مكونات الصف يحتاج إلى محدد وصول خاص به لوحده.

الافتراضي package: إذا لم يوضع قبل الصف أو مكونات الصف أي محدد وصول بشكل صريح فإن الصف أو مكوناته يتبعون للمحدد الافتراضي وهو يعني أنه يمكن الوصول إليهم فقط من الصفوف التي ضمن نفس الحزمة أي نفس المجلد.

العام public: ★ إذا وضع قبل الصف فإنه يمكن الوصول إلى هذا الصف أينما كان موجوداً على الحاسب ليس شرطاً أن يكون بنفس الحزمة ولكن لا يمكن الوصول إلى مكوناته من أي مكان لأن المحدد على الصف فقط أي يمكن فقط التصريح عن كائنات منه من أي مكان أما مكوناته فتتبع محددات الوصول الخاصة بها.

```
public class Toto{
    int x; double y;}
```

يمكننا التصريح عن أغراض من هذا الصف من أي مكان ولكن لا يمكن الوصول إلى مكوناته من أي مكان لأننا لم نضع أي محدد لكل من x و y وبالتالي لا يتم الوصول إليهما إلا عن طريق الصفوف الموجودة ضمن نفس الحزمة لأن محدد الوصول لكل منهما هو الافتراضي.

★ إذا وضع قبل أي مكون من مكونات الصف (طريقة أو معطى صف) فإنه يمكن الوصول إليه من أي مكان وذلك لأنه أصبح عاماً وليس شرطاً أن يكونوا في نفس الحزمة (يستطيع قراءة المكون وتعديله)

Scanned by CamScanner

الخاص private: ★ إذا وضع قبل الصف فإنه لن يكون ممكناً الوصول له وبالتالي لا يمكن انشاء اغراض منه وبالتالي ليس هناك داعٍ من كتابته

★ إذا اتى قبل مكونات الصف فإنه لا يمكن الوصول إلى المكون المسبوق بـ private إلا ضمن الصف نفسه أي لا يمكن الوصول إليه من خارج الصف سواء من نفس الحزمة أم لا

المكون الخاص هو خاص بالصف فقط ويمكن الوصول إليه من ضمن الصف نفسه فقط

مثال:

```
public class Toto{
public int x;
private int y;
}
```

استطيع كتابة: Toto t=new Toto(); من أي صف آخر حتى ولم يكن في نفس الحزمة

واستطيع القيام بالعملية: T.x=5; من أي صف

ولكن الكتابة T.y=2; خاطئة لأن y هو معطى خاص.

المحمي protected: ★ لا يأتي قبل الصف لنفس الأسباب عندما يأتي الحدد الخاص قبل الصف.

★ اذا اتى قبل مكونات الصف فإن المكون المسبوق بـ protected يكون عاماً بالنسبة للصف وللصفوف التي ترتبط مع هذا الصف بعلاقة وراثة وخاصاً بالنسبة للصفوف الأخرى أي أنه يمكن الوصول إلى المكونات المحمية من داخل الصف نفسه ومن داخل الصفوف الأبناء لهذا الصف ولكن لا يمكن الوصول إليها عن طريق صف ليس ابن.

مثال:

```
class Toto{
public int x;
private char c;
float y;
void print(){
System.out.print(x+c+y);}
}
```

Scanned by CamScanner

| في نفس الحزمة | خارج الحزمة | التعليمات |
|---|---|---|
| ممكن لأن محدد الوصول هو الافتراضي | ممنوع لأن الصف ليس عام هو افتراضي | Totot=newTotol(); |

بفرض أن الصف عام أي: public class Toto{

| داخل الحزمة | خارج الحزمة | التعليمات |
|---|---|---|
| صحيحة لأن x عام | عملية صحيحة لأن x هو متغير عام | t.x=5; |
| عملية صحيحة لأن y هو افتراضي أي ضمن الحزمة | عملية خاطئة حتى لو كان الصف عام ولكن y هو افتراضي | t.y=2; |
| عملية خاطئة لأن c هو خاص أي ضمن الصف فقط | عملية خاطئة لأن c هو خاص أي ضمن الصف فقط | t.c='H'; |
| عملية صحيحة لأن الطريقة هي افتراضية أي ضمن الحزمة | عملية خاطئة لأن الطريقة هي افتراضية أي في نفس الحزمة | t.print(); |

## الباني الافتراضي له محدد وصول نفس محدد وصول صفه

ملاحظة: المعطى البياني الخاص هو أي فقط داخل صفه ومنه لا استطيع رؤيته أو تعديله أو القيام بأي عملية عليه خارج صفه ولكن عندما أريد أن أتيح امكانية رؤيته خارج صفه فسوف اتيحها بالطريقة التي أريد دون تغيير خصوصيته

* أي عضو بياني خاص استطيع أن اتيح امكانية رؤيته بالشروط التي أريد عن طريق كتابة دالة عامة ترجع قيمته اسمها get ويمكننا تعديل قيمته عن طريق كتابة دالة عامة اسمها set
* كل معطى خاص يزود مع دوال عامة داخل الصف
set تمكننا من اعطاء قيمة للمعطى البياني الخاص خارج الصف
get تمكننا من الحصول على قيمة المعطى البياني الخاص من خارج الصف

Scanned by CamScanner

مثال:

```
class Girle{
  int tall;
private int age;
 Girle(){tall=157; age=41;}
 Girle(int tall, int age){
   this.tall=tall;
   this.age=age;}
   void print1(){
     System.out.print(185+"cm "+21+"years");
                  }
    private void print(){
     System.out.print(tall+","+age);}
     public void set_age(int a){
        age=a;}
     public void get_age(){
          return age;}
                }
class Mymain{
public static void main(String args[]){
 Girle g=new Girle();
   g.print();  ← خطأ كون الدالة خاصة
   g.print1();  → 185cm 21years
System.out.print(g.age) → خطأ كون age خاص
System.out.print(g.get_age()); → 41
    g.age=15; → خطأ
     g.set_age(15);
    }
    }
```

في نفس الحزمة:

Scanned by CamScanner

ملاحظات:

* عندما قمت بالعملية System.out.print(g.getage()) فإنه سيطبع القيمة الموجودة في الـ object عند بنائه لأن الدالة تقصيرها هي ارجاع القيمة نفسها ولكن لو كانت الدالة بالشكل

```
public int get_age(){
    return 22;
}
```

فعند استدعاء الدالة يكون ناتج الطباعة هو 22

ولو قمنا بكتابة الدالة بالشكل:

```
public int get_age(){
    return age_20;
}
```

فعند استدعاء الدالة يكون ناتج الطباعة هو 21

الخلاصة: في كل صف استطيع ان اضع مكونات عامة او خاصة او محمية.

* المكونات العامة كل من هو خارج الصف يستطيع الوصول لها والتعامل معها من ناحية قراءتها أو تعديلها

* المكونات المحمية كي فقط لمن يرتبط بعلاقة وراثة مع هذا الصف الوصول لها والتعامل معها قراءة أو تعديل.

* المكونات الخاصة لا يحق لأي كان الوصول لها أو قراءتها أو التعديل سوى الصف الموجودة به.

إذا اردنا نحن اتاحة عملية تغيير أو قراءة من الخارج فإن ذلك يتم من خلال دوال عامة ضمن الشروط التي أريد!

كن لنفسك كل شيء

انتهت المحاضرة

Scanned by CamScanner