بسم الله الرحمن الرحيم

لغة البرمجة جافا
Java Programming Language

الدرس الثالث عشر :
الأصناف (الصفوف) الداخلية في جافا

لماذا تعتبر الأصناف الداخلية جيدة ؟

صدق أولا تصدق، هناك فوائد كبيرة لاستخدام الأصناف الداخلية ولكن قبل الخوض في فوائدها سأقدم خلفية مختصرة عن الأصناف الداخلية.

تتواجد الأصناف الداخلية ضمن الأصناف الأخرى حيث يعتبر الصنف العادي عضو مباشر من الحزمة والتي تعرف كصنف عالي المستوى. أصبحت الأصناف الداخلية متاحة في إصدار جافا 1.1 حيث كان لها أربعة أنواع:

- أصناف الأعضاء الساكنة (الستاتيكية)
- أصناف الأعضاء
- الأصناف المحلية
- الأصناف المجهولة

دعونا نلقي نظرة سريعة على كل من هذه الأنواع:

باختصار فإن صنف العضو الساكن (الستاتيكي) هو عبارة عن عضو ساكن من صنف. يملك صنف العضو الساكن (الستاتيكي)، وبشكل مماثل للطرق الساكنة، إمكانية الولوج إلى جميع الطرق الساكنة التابعة للصنف الأب أو للصنف العالي المستوى.

يعرف صنف العضو بشكل مشابه لصنف العضو الساكن بأنه عضو من صنف ولكن باختلاف نوع السكون حيث أن الصنف العضو هو عبارة عن مثال(instance) محدد يملك إمكانية الولوج إلى كل الطرق والأعضاء حتى إلى المرجع this التابع للأب.

يصرح عن الأصناف المحلية ضمن كتلة من الشيفرة (code) وتعتبر مرئية ضمن تلك الكتلة مثل أي متحول آخر ضمن الطريقة .

أخيرا الصنف المجهول عبارة عن صنف محلي لا اسم له.

للإجابة بدقة عن السؤال السابق سوف يكون تركيزي على أصناف الأعضاء الداخلية والأصناف المجهولة حيث أن هذه الأنواع هي التي سوف تصادفها عادة و تستخدمها. يمكن تقسيم فوائد الأصناف الداخلية إلى ثلاث فئات:

- فائدة غرضية التوجه
- فائدة تنظيمية
- فائدة الاستدعاء

الفائدة غرضية التوجه

إن أهم ميزة للصنف الداخلي هو أنه يسمح لك بتحويل الأشياء إلى أغراض الشيء الذي يجعل شيفرتك غرضية التوجه بشكل أكبر مما هي الحال عليه عند عدم استخدام الأصناف الداخلية.

بإلقاء نظرة على الصنف العضو نجد أن المثال(instance) الخاص به هو عبارة عن عضو من مثال الأب حيث يملك إمكانية الولوج إلى كل عضو وطريقة في الأب. هذا لا يبدو كثيرا من النظرة الأولى لأننا قمنا سابقا بذكر إمكانية الولوج لطريقة موجودة في الصنف الأب. من ناحية أخرى فإن الصنف العضو يسمح لنا بأخذ المنطق (logic) من الأب وتحويله إلى غرض (object). على سبيل المثال قد يكون للصنف الشجرة طريقة رئيسية وعدة طرق مساعدة تقوم بعملية البحث أو التحرك ضمن الشجرة. من وجهة نظر غرضية التوجه فإن الشجرة هي شجرة وليست خوارزمية بحث، ولكن حتى تتم عملية البحث فأنت بحاجة إلى معرفة ببنية الأشجار.

نستنتج من ذلك أن الصنف الداخلي يسمح لنا باستخلاص المنطق (logic) ووضعه في الصنف الخاص به، نكون بذلك قد أزلنا الجملة الوظيفية من المكان الذي لا تنتمي إليه ووضعناها في الصنف الخاص بها.

نستطيع وباستخدام الصنف الداخلي أن نفصل بين خوارزمية البحث وبين الشجرة وفي حال أردنا تغيير خوارزمية البحث نستطيع ببساطة تبديلها بصنف جديد.

الفائدة التنظيمية

من وجهة نظر تنظيمية فإن الأصناف الداخلية تسمح لنا بتنظيم بنى الحزم عن طريق استخدام طريقة فراغات الاسم (namespaces). يمكن وضع الأصناف ضمن أصناف أخرى بدلا من جمع كل شيء في حزمة واحدة.
بدون استخدام الأصناف الداخلية فإننا مجبرون على استخدام البنية التسلسلية (الهرمية) التالية:

```
package1
   class 1
      class 2
      ...
      class n
...
package n
```

أما عند استخدام الأصناف الداخلية نستطيع عمل الآتي:

```
package 1
   class 1
   class 2
      class 1
      class 2
      ...
      class n
```

يمكن أن تقدم الأصناف الداخلية بنية تسلسلية والتي قد تناسب أصنافك بشكل أكبر.

فائدة الاستدعاء

تقدم أعضاء الأصناف الداخلية والأصناف المجهولة طريقة مناسبة لتعريف الاستدعاء. أكثر الأمثلة وضوحا على ذلك يتعلق بشيفرة واجهة المستخدم الرسومية (GUI)، إلى غير ذلك من المجالات التي يمكن أن تستخدم فيها تطبيقات الاستدعاء. معظم تطبيقات جافا الخاصة بواجهة المستخدم الرسومية (GUI) تملك عدة عناصر تفعل استدعاء الطريقة ()actionPerformed . لسوء الحظ فإن معظم المطورين كانت لهم نوافذهم الخاصة بهم لتنفيذ ال ActionListener حيث أنه يمكن أن يصبح لديك تصور عن العنصر الذي قام بالفعل (action) عن طريق كتابة التعليمة switch في الطريقة ()actionPerformed .
إليك المثال التالي عن التنفيذ الفردي(monolithic implementation) :

```
public class SomeGUI extends JFrame implements ActionListener
{
   protected JButton button1;
   protected JButton button2;
   ...
   protected JButton buttonN;


   public void actionPerformed(ActionEvent e)
   {
      if(e.getSource()==button1)
      {
         // do something
         }
      else if(e.getSource()==button2)
      {
```

يجب أن يدق جرسا منبها داخل دماغك في حال رؤيتك لكتلة تحوي تعليمة switch أو لكتلة ضخمة تحوي إحدى التعليمتين if / if else . إن مثل هذه البنى تعتبر سيئة لأنه بمجرد حدوث تغيير في جزء من الشيفرة فهذا يتطلب تغييرات مطابقة في تعليمة switch. أما أصناف الأعضاء الداخلية والأصناف المجهولة فإنها تسمح لنا بالابتعاد عن تغيير الطريقة actionPerformed() .
نستطيع تعريف الصنف الداخلي الذي ينفذ ال ActionListener بأنه هو كل عنصر نريد الاستماع إليه. يمكننا من ناحية أخرى تجنب استخدام تعليمات switch الكبيرة واستخدام عمليات تغليف الأفعال الإضافية بدلا عنها حيث أن استخدام الطريقة الأخيرة قد يطور الآداء. عندما نستخدم تعليمة switch والتي يوجد فيها ن مقارنة فإننا نستطيع أن نتوقع كمعدل وسطي ن / 2 مقارنة. تسمح لنا الأصناف الداخلية بتهيئة مراسلات من 1 إلى 1 بين مؤدي الفعل ومنتظره وهذا ما قد يحدث أثرا قويا على آداء تطبيقات واجهة المستخدم الرسومية (GUI) الكبيرة.
تبدو الطريقة المجهولة على الشكل التالي:

```
public class SomeGUI extends JFrame
{
   ... button member declarations ...

   protected void buildGUI()
   {
      button1 = new JButton();
      button2 = new JButton();
      ...

      button1.addActionListener(
         new java.awt.event.ActionListener()
         {
            public void
actionPerformed(java.awt.event.ActionEvent e)
            {
               // do something
            }
         }
      );
      .. repeat for each button
```

أما عند استخدام الأصناف الداخلية فإن نفس البرنامج يبدو على الشكل التالي:

```
public class SomeGUI extends JFrame
{
   ... button member declarations

   // inner class definitions
   class Button1Handler implements ActionListener
   {
      public void actionPerformed(ActionEvent e)
   {
         // do something
      }
   }
   ... define an inner member class for each button

   protected void buildGUI()
   {
```

```
// initialize the buttons
button1 = new JButton();
button2 = new JButton();
...

// register an inner class action listener instance
// for each button
button1.addActionListener(new Button1Handler());
.. repeat for each button
```

نستطيع نقل أي منطق (logic) يظهر في تنفيذ الطريقة actionPerformed() إلى صنف داخلي وذلك يعود إلى كون الأصناف الداخلية تملك إمكانية الولوج إلى كل شيء موجود في الصنف الأب.
بالنسبة لي فأنا افضل استخدام الأصناف الداخلية للاستدعاء، ولكن بالطبع فإن هذه مسألة تعود إلى رأيك الشخصي، ولكنني أشعر أن استخدام عدد كبير من الأصناف المجهولة يسبب الفوضى في الشيفرة ويصبح استخدام هذه الأصناف صعبا عندما تكون أكثر من سطر أو سطرين.

المساوئ

كأي شيء آخر يجب أن تدرس المساوئ مع الفوائد حيث يوجد للأصناف الداخلية مساوئها الخاصة به. من وجهة نظر تهتم بالصيانة فإن مطوروا الجافا ذوي الخبرة القليلة قد يجدون صعوبة في فهم الأصناف الداخلية. أيضا عندما تستخدم الأصناف الداخلية فإن مجمل الأصناف في شيفرتك يصبح كبيرا. علاوة على ذلك فإن معظم أدوات الجافا ظهرت بدعم قليل للأصناف الداخلية. على سبيل المثال لقد استخدمت VisualAge for Java المعد من قبل شركة IBM من أجل كتابة شيفرتي اليومية واستطعت ترجمة الأصناف الداخلية في هذا البرنامج ولكنني لم أجد فيه عارض (مستعرض) للأصناف الداخلية. عوضا عن ذلك كنت بحاجة لكتابة الصنف الداخلي ضمن تعريف الصنف مباشرة مما جعل استعراض الصنف الداخلي عملية صعبة. نظرا لضعف عملية إتمام الشيفرة في ال VisualAge فإنه من الصعب عليك الترميز أثناء الكتابة ضمن تعريف الصنف أو أثناء استخدامك للصنف الداخلي.

ساهم بنشر الكتاب ولك الأجر والثواب إن شاء الله
لا تنسوني من صالح دعائكم
تم بحمد الله