تحسين الكود - استخدام Magic Number, Refactoring - Magic Numbers

للكاتب: Mohammad Elsheimy

قمت بكتابة هذا الدرس سابقا في موقعي مع الدوت نت.

نظرة خاطفة

درسنا في هذا اليوم يتكلم عن أحد الطرق لتحسين الكود أو ما نسميها بعملية Refactoring. يتكلم عن طريقة استخدام Magic Numbers لجعل الكود أسهل وأبسط في القراءة والكتابة.

مقدمة

تحسين الكود أو ما نسميه علميا بـ Refactoring هي أفكار أو طرق يتم تطبيقها على الكود لجعله أبسط وأسهل في القراءة والكتابة ولجعل عملية التنقيح Debuggingأو مراجعة Revising الكود أسهل وأبسط.

هناك طرق عديدة جدا للـ Refactoring ومنها استخدام الـ Magic Numbers.

أرقام Magic Numbers هي عبارة عن أرقام لها دلالات معينة. فعلى سبيل المثال إذا كنا نريد تحويل عدد من المترات إلى سنتيميترات فسوف نقوم بالضرب في 100. وبالعكس، إذا كنا نريد تحويل السنتيميترات إلى مترات فسوف نقوم بالقسمة على 100. ولذلك يعتبر 100 هو رقم سحري Magic Number. فبدلا من كتابتك لهذا الرقم مرات عديدة في الكود يمكنك جعل هذا الكود في متغير Variableأو حتى في متغير ثابت Constant وتقوم بدلا من كتابة الرقم 100 تقوم بكتابة اسم المتغير أو هذا الثابت.

لنأخذ مثال آخر أكثر تعقيدا فربما الرقم 100 هو رقم بسيط وسهل. إذا كنا نريد عمل برنامج يقوم بالتحويل بين المترات والأقدام فسوف يكون عندنا رقم سحري Magic Number وهو (3.28083989501312335958005249343830هذه أعلى دقة لهذا الرقم). فإنك إذا كنت تريد التحويل إلى الأقدام فسوم تقوم بالضرب في هذا الرقم وإذا كنت تريد العكس بالتحويل على المترات فسوف تقوم بالقسمة على هذا الرقم. فبدلا من أن تقوم بكتابة هذا الرقم بهذه الدقة في كل مكان في الكود وإذا كنت تريد تغييره فسوف يكون هذا شبه مستحيل أن تبحث في كل الكود عن هذا الرقم وتقوم بتغييره. ولكن الحل الأفضل هو أن تقوم بتسجيل هذا الرقم في متغير وتقوم بكتابة فقط اسم المتغير في كل دالة بدلا من الرقم وهكذا تستطيع التعامل مع الكود الأسهل وأبسط بالإضافة إلى أنه سيكون أسهل في القراءة.

التطبيق

الكود التالي يوضح دالتين واحدة للتحويل من المتر إلى القدم والأخرى للتحويل بالعكس من القدم إلى المتر. لاحظ كيفية قراءة وكتابة الكود.

```
// C# Code

   public static double ConvertMetersToFeet(double meters)
   {
       return meters * 3.28083989501312335958005249343830;
   }

   public static double ConvertFeetToMeters(double feet)
   {
       return feet / 3.28083989501312335958005249343830;
   }

   ------------------------------
```

```
   ' VB.NET Code

   Public Function ConvertMetersToFeet(meters As Double) As Double
       Return meters * 3.2808398950131233595800524934383
   End Function

   Public Function ConvertFeetToMeters(feet As Double) As Double
       Return feet / 3.2808398950131233595800524934383
   End Function
```

يصبح كودنا بالشكل التالي Magic Number من نوع Refactoring بعد أن نقوم بتحسين الكود أو بمعنى أصح عمل:

```
// C# Code

   public const double MeterFeet =
3.2808398950131233595800524934383;

   public static double ConvertMetersToFeet(double meters)
   {
       return meters * MeterFeet;
   }

   public static double ConvertFeetToMeters(double feet)
   {
       return feet / MeterFeet;
   }

   ------------------------------

   ' VB.NET Code

   Public Const MeterFeet As Double =
3.2808398950131233595800524934383

   Public Function ConvertMetersToFeet(meters As Double) As Double
       return meters * MeterFeet
   End Function

   Public Function ConvertFeetToMeters(feet As Double) As Double
       return feet / MeterFeet
   End Function
```

خاتمة

تعلمنا في هذا الدرس كيفية تحسين الكود Refactoring عن طريق استخدام Magic Numbers لجعل الكود أسهل وأيسر وأسهل في عملية التنقيح والمراجعة. سوف نقوم بإذن الله تعالى بشرح طرق أكثر لتحسين الكود في الدروس القادمة.