Abstract y Virtual

10

Programación II y Laboratorio de Computación II

Edición 2018

Abstracción

- Definición Formal:
  - Efecto de separar conceptualmente algo de algo.

- El modificador abstract se utiliza para indicar que una clase está incompleta y que sólo se va a utilizar como una clase base.

Clases abstractas

- No se puede crear una instancia de una clase abstracta directamente, y es un error en tiempo de compilación utilizar el operador new en una clase abstracta.
- Aunque es posible tener variables y valores cuyos tipos en tiempo de compilación sean abstractos, tales variables y valores serán null o contendrán referencias a instancias de clases no abstractas derivadas de los tipos abstractos.
- Se permite que una clase abstracta contenga miembros abstractos, aunque no es necesario.
- No se puede sellar una clase abstracta.

Clases abstractas

- Cuando una clase no abstracta se deriva de una clase abstracta, la clase no abstracta debe incluir implementaciones reales de todos los miembros abstractos heredados; por lo tanto, reemplaza a estos miembros abstractos.
- Las clases abstractas se sitúan en la cima de la jerarquía de clases.
- Establecen la estructura y significado del código.
- Facilitan la creación de marcos de trabajo, esto es posible ya que las clases abstractas poseen una información y un comportamiento común a todas las clases derivadas de un marco de trabajo.

Abstracción

```
abstract class A
{
}
class B : A
{
    public void G() { }
}

// ...

A a = new A(); // ERROR!
B b = new B(); // Correcta!
```

Abstracción

```
abstract class A
{
    public abstract void F();
}
abstract class B : A
{
    public void G() { }
}
class C : B
{
    public override void F()
    {
        // Implementación real de F
    }
}
```

Miembros abstractos

- Cuando una declaración de método de instancia incluye un modificador abstract, se dice que el método es un método abstracto.
- Las declaraciones de métodos abstractos sólo se permiten en clases abstractas.
- Aunque, por definición formal, un método abstracto es también implícitamente un método virtual, no puede tener el modificador virtual.

Miembros abstractos

- Dada la definición formal, podríamos decir:
  - Una declaración de método abstracto introduce un nuevo método virtual, pero no proporciona una implementación del método.
  - Por esto es necesario que las clases derivadas no abstractas proporcionen su propia implementación mediante el reemplazo del método.
  - Debido a que un método abstracto no proporciona una implementación real, el cuerpo-del-método de un método abstracto consiste simplemente en un punto y coma.
  - Son métodos y propiedades que se declaran sin implementación.

Miembros abstractos

```
// System.Drawing

public abstract class Forma
{
    public abstract void Pintar(Graphics g, Rectangle r);

    public abstract int Area { get; }
}
```

```
public abstract class Shape
{
    public abstract void Paint(Graphics g, Rectangle r);
}
public class Ellipse : Shape
{
    public override void Paint(Graphics g, Rectangle r)
    {
        g.DrawEllipse(r);
    }
}
public class Box : Shape
{
    public override void Paint(Graphics g, Rectangle r)
    {
        g.DrawRect(r);
    }
}
```

Miembros Virtuales

- El modificador *virtual* sirve para métodos, propiedades, indexadores o evento declarado y permitir que este sea anulado en una clase derivada.
- Cuando una declaración de método de instancia incluye un modificador *virtual*, se dice que el este es un método virtual.
- Si no existe un modificador virtual, se dice que el método es un método no virtual.

Miembros Virtuales

- La implementación de un método no virtual es invariable, o sea es la misma tanto si se invoca un método en una instancia de la clase en la que se declaró o en una instancia de una clase derivada.
- En cambio, en un método virtual se puede sustituir por clases derivadas. El proceso de sustitución de la implementación de un método virtual heredado es conocido como reemplazamiento del método.

Miembros Virtuales

- En la invocación de un método virtual, el tipo en tiempo de ejecución de la instancia para la que tiene lugar la invocación determina la implementación del método real a invocar.
- Cuando se invoca un método no virtual, el factor determinante es el tipo en tiempo de compilación de la instancia.

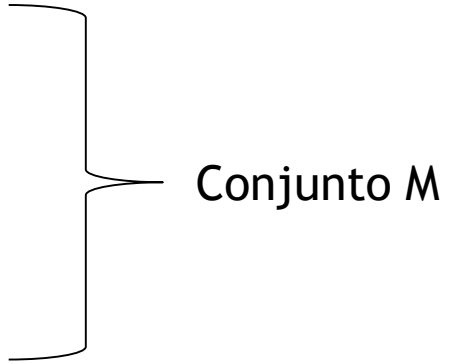
Miembros Virtuales

- Podemos concluir que cuando se invoca un método denominado N con una lista de argumentos A en una instancia con un tipo C en tiempo de compilación y un tipo “r” en tiempo de ejecución (donde “r” es C o una clase derivada de C):
- La invocación se procesa de la forma siguiente:
  - En primer lugar, la resolución de sobrecarga se aplica a C, N y A para seleccionar un método específico M del conjunto de métodos heredados y declarados por C.
  - A continuación, si M es un método no virtual, se invoca M.
  - Si no, M es un método virtual, y se invoca la implementación más derivada de M con respecto a R.

Miembros Virtuales

```
public class C
{
    public virtual void N(int a1, int a2)
    { }
    public virtual void N(int a1, int a2, int a3)
    { }
    public virtual void Z(int a1, int a2)
    { }
}

public class C_Derivada : C
{
    public override void N(int a1, int a2)
    { }
}
```



Redefinición de Métodos

- El modificador override es necesario para ampliar o modificar la implementación abstracta o virtual de un método, propiedad, indixador o evento heredado.
- Un método override proporciona una nueva implementación de un miembro que se hereda de una clase base. El método invalidado por una declaración override se conoce como método base invalidado.
- El método base reemplazado debe tener la misma firma que el método override.

Redefinición de Métodos

- No se puede reemplazar un método estático o no virtual. El método base reemplazado debe ser virtual, abstract u override.
- Una declaración override no puede cambiar la accesibilidad del método virtual. El método override y el método virtual deben tener el mismo modificador de nivel de acceso.
- No se pueden usar los modificadores new, static o virtual para modificar un método override.
- Una declaración de propiedad de invalidación debe especificar exactamente el mismo modificador de acceso, tipo y nombre que la propiedad heredada, y la propiedad invalidada debe ser virtual, abstract u override.

Redefinición de Métodos

```
public class C
{
   public override string ToString()
   {
         return "Mi propio ToString()";
   }
}
```

Virtuales vs No Virtuales

```
class A {
    public void F() { Console.WriteLine("A.F"); }
    public virtual void G() { Console.WriteLine("A.G"); }
}
class B : A {
    new public void F() { Console.WriteLine("B.F"); }
    public override void G() { Console.WriteLine("B.G"); }
}
class Test {
    static void Main()
    {
        B b = new B();
        A a = b;
        a.F();
        b.F();
        a.G();
        b.G();
    }
}
```