Atributos y constructores

Programación en Java

Ángel García y Beltrán
Unidad docente de Informática Industrial
ETSII – UPM
11 de Octubre de 2010

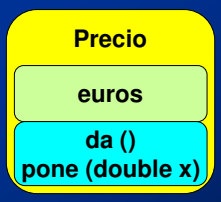
Índice

- Objetos y atributos
  - Variables de instancia (no `static`)
  - Variables de clase (`static`)
  - Constantes o variables finales (`final`)
- Constructores

11/10/2010

Ejemplo de declaración de una clase

```
// Contenido de Precio.java
public class Precio {
    // Atributo o variable miembro
    private double euros;
    // Metodos publicos
    public double da() {
        return euros;
    }
    public void pone(double x) {
        euros=x;
    }
}
```



11/10/2010

Ejemplo de uso de la clase `Precio`

```
// Contenido de PruebaPrecio.java
public class PruebaPrecio {
    public static void main (String [] args) {
        Precio p;           // Crea la referencia
        p = new Precio();   // Crea la instancia
        p.pone(56.8);       // Llamada al metodo pone
        System.out.println("Valor = " + p.da());
        Precio q = new Precio();
        q.pone(75.6);
        System.out.println("Valor = " + q.da());
    }
}
```

11/10/2010

Variables *de instancia*

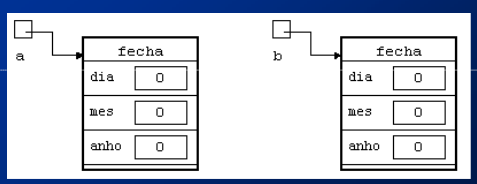
- Una ***variable de instancia*** es un atributo o variable miembro de una clase con las siguientes características:
  - Cada ***instancia*** de una clase tiene sus ***variables de instancia*** propias
  - En su declaración NO se utiliza la palabra `static`
  - Pueden declararse como `public` ó como `private` (acceso restringido dentro de la propia clase)
  - Pueden ser de un tipo primitivo o pertenecer a otra clase
- Ejemplo de declaración de variables de instancia...

11/10/2010

Ejemplo de variables *de instancia*

```
// Contenido parcial de Fecha.java
public class Fecha {
    // Declaracion de atributos o variables miembro
    private int dia;
    private int mes;
    private int anho;
    // Declaracion de metodos . . .
}
```

```
// Contenido parcial de PruebaFecha.java
// . . . Creacion de dos instancias de la clase Fecha
Fecha a = new Fecha();
Fecha b = new Fecha();
```



11/10/2010

Ejercicio propuesto (1)

- Completar la declaración de la clase `Fecha` con los siguientes métodos:

```
public void asigna(int dd,int mm,int aa)
public boolean esPrimeroDeMes()
public boolean esAnterior(Fecha otra)
public boolean esBisiesto()
public String daFecha()
```

- Construir un programa `PruebaFecha` que pruebe la clase `Fecha` con los métodos anteriores

11/10/2010

Una solución al ejercicio propuesto

```
public void asigna(int ndia, int nmes, int nanho) {
    dia = ndia;
    mes = nmes;
    anho = nanho;                // No es necesaria una sentencia return
}
public boolean esPrimeroDeMes() {
    return (dia == 1);
}
public boolean esAnterior(fecha otra) {
    return (anho < otra.anho) ||
           ((anho == otra.anho) && (mes < otra.mes)) ||
           ((anho == otra.anho) && (mes == otra.mes) && (dia < otra.dia));
}
public boolean esBisiesto() {
    return (anho % 400 == 0) ||
           ((anho % 4 == 0) && (anho % 100 != 0));
}
public String daFecha() {
    return (dia + "/" + mes + "/" + anho);
}
```

11/10/2010

Variables *de clase*

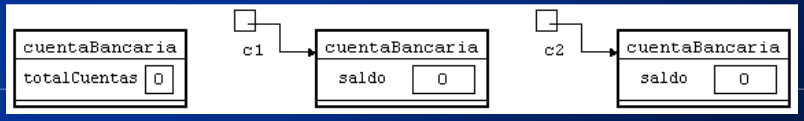
- Una *variable de clase* es un atributo o variable miembro de una clase con las siguientes características:
  - Cada *clase* tiene sus *variables de clase* propias y compartidas por todas las instancias de esa clase
  - En su declaración se utiliza la palabra `static`
  - Pueden declararse como `public` ó como `private` (acceso restringido dentro de la propia clase)
  - Pueden ser de un tipo primitivo o pertenecer a otra clase
- Ejemplo de declaración de variable de clase...

11/10/2010

Ejemplo de variable *de clase* (`static`)

```
public class CuentaBancaria {
    private double saldo;              // Variable de instancia
    public static int totalCuentas=0; // Variable de clase
    // Declaracion de metodos ...
    public void modificacion(double ingreso) {
         saldo += ingreso;
    }
    public static void inctotalCuentas() {
         totalCuentas++;
    }
}
```

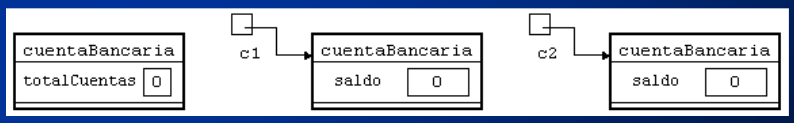
```
// Contenido parcial de PruebaCuentaBancaria.java
// Creacion de dos instancias de la clase CuentaBancaria
CuentaBancaria c1 = new CuentaBancaria();
CuentaBancaria c2 = new CuentaBancaria();
```



11/10/2010

Ejemplo de uso de CuentaBancaria

```
// Contenido de PruebaCuentaBancaria.java
public class PruebaCuentaBancaria {
 public static void main (String [] args) {
  CuentaBancaria c1 = new CuentaBancaria();
  c1.totalCuentas++;
  System.out.println("Total cuentas: " + c1.totalCuentas);
  CuentaBancaria c2 = new CuentaBancaria();
  c2.totalCuentas++;
  System.out.println("Total cuentas: " + c2.totalCuentas);
  // Acceso a traves de la clase:
  CuentaBancaria.totalCuentas++;
  System.out.println("Total cuentas: " + CuentaBancaria.totalCuentas);
        // Resto de sentencias . . .
 }
```



11/10/2010

Ejercicio propuesto (2)

- Completar la declaración de la clase `CuentaBancaria` con los siguientes métodos:

```
public double saldo()
public boolean enNumerosRojos()
public void movimiento(double valor)
public void transferencia(CuentaBancaria b)
public static void incrementoTotalCuentas()
```

- Construir un programa `PruebaCuentaBancaria` que pruebe la clase `CuentaBancaria` con los métodos anteriores

11/10/2010

Una solución al ejercicio propuesto

```
public double saldo() {
    return saldo;
}
public boolean enNumerosRojos() {
    return saldo<0;
}
public void movimiento(double valor) {
    saldo += valor;
}
public void transferencia(CuentaBancaria origen) {
    saldo += origen.saldo;
    origen.saldo=0;
}
public static void incrementoTotalCuentas () {
    totalCuentas++;
}
```

11/10/2010

Variables *finales* o constantes

- Una *variable final* es un atributo o variable miembro de una clase con las siguientes características:
  - Se declara con la palabra reservada `final`
  - No puede ser modificada una vez declarada e inicializada
  - La inicialización puede hacerse después de la declaración
  - Puede declararse como `static` (suele hacer por ahorro de memoria)
  - Puede declararse como `public` o como `private` (acceso sólo dentro de la propia clase)
  - Pueden ser de un tipo primitivo o pertenecer a otra clase
- Ejemplo de declaración de variable *final*...

11/10/2010

Ejemplo de variable *final*

```
// Contenido de Circulo.java
public class Circulo {
   // Atributos o variables miembro
   private double radio;
   // Constante de clase:
   private static final double PI = 3.141592;
   // Declaracion de metodos ...

   }
```

11/10/2010

Ejercicio propuesto (3)

- Completar la declaración de la clase `Circulo` con los siguientes métodos:

```
public void asignaRadio(double valor)
public double daRadio()
public double longitud()
public double area()
public boolean mayorQue(Circulo otro)
```

- Construir un programa `PruebaCirculo` que pruebe la clase `Circulo` con los métodos anteriores

11/10/2010

Una solución al ejercicio propuesto

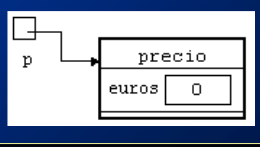
```
public void asignaRadio(double valor) {
    radio=valor;
}
public double daRadio() {
    return radio;
}
public double longitud() {
    return 2*PI*radio;
}
public double area() {
    return PI*radio*radio;
}
public boolean mayorQue(Circulo otroCirculo) {
    return (radio>otroCirculo.daRadio());
}
```

11/10/2010

Constructores

- Deben ser llamados al crear una instancia
- Inicializan los atributos de la instancia
- Todas las clases tienen un constructor *por defecto*:
  - Con el mismo identificador que la clase
  - No tiene parámetros
  - El constructor por defecto puede redeclararse
- Pueden declararse otros constructores (sobrecarga)

```
public class PruebaPrecio {
    public static void main (String [] args) {
        Precio p;              // Crea referencia
        p = new Precio();      // Llamada al constructor
    //  Resto del codigo ...
    }
}
```



11/10/2010

Ej. de declaración de constructores

- El constructor `Fecha()` puede redeclararse...
- También pueden declararse otros constructores...

```
public class Fecha {
    private int dia;
    private int mes;
    private int anho;
    public Fecha() {        // Redefinicion del constructor
        dia = 1;
        mes = 1;
        anho = 2000;
    }
    public Fecha(int ndia, int nmes, int nanho) {
        dia = ndia;
        mes = nmes;
        anho = nanho;
    }
// resto de declaraciones
```

11/10/2010

Ejemplo de uso de constructores

- En la creación de una instancia para la inicialización de los valores de sus atributos

```
/**
 *  Ejemplo de uso de la clase Fecha
 *  A. Garcia-Beltran - octubre, 2010
 */
public class PruebaFecha {
    public static void main (String [] args) {
        Fecha origen = new Fecha();
        Fecha actual = new Fecha(11,10,2010);
```

11/10/2010

Ejercicio propuesto (4)

- Modificar la clase `Fecha` con constructores.
  - Editar y compilar
- Modificar la clase `PruebaFecha` para que verifique el uso de los constructores de la clase `Fecha`.
  - Editar, compilar y ejecutar

11/10/2010

Más sobre constructores

```
public class CuentaBancaria {
  private double saldo;
  public static int totalCuentas=0;
  public CuentaBancaria( ) {
     this(0.0);      // Llamada al constructor de un parametro
  }
  public CuentaBancaria(double ingreso) {
     saldo = ingreso;
     incrementoTotalCuentas();
  }
  public static void incrementoTotalCuentas () {
     totalCuentas++;
  }
  public void transferencia(CuentaBancaria origen) {
     saldo += origen.saldo;
     origen.saldo=0;
  }
  public double saldo() {
     return saldo;
  }
}
```

11/10/2010

Ejemplo (2) de uso de constructores

```
public class testCuentaBancaria {
 public static void main (String [] args) {
  System.out.println("Total cuentas: " + CuentaBancaria.totalCuentas);
  CuentaBancaria c1 = new CuentaBancaria();
  System.out.println("Nueva cuenta con: " + c1.saldo() + " euros");
  System.out.println("Total cuentas: " + CuentaBancaria.totalCuentas);
  CuentaBancaria c2 = new CuentaBancaria(20.0);
  System.out.println("Nueva cuenta con: " + c2.saldo() + " euros");
  System.out.println("Total cuentas: " + CuentaBancaria.totalCuentas);

  System.out.println("Transferencia de cuenta 2 a cuenta 1");
  // La siguiente asignacion es peligrosa:
  c1.transferencia(c2);
  System.out.println("Cuenta 1 con: " + c1.saldo() + " euros");
  System.out.println("Cuenta 2 con: " + c2.saldo() + " euros");
 }
}
```

11/10/2010