Fundamentos de la Programación Orientada a Objetos

Definición de Clases

Programación Orientada a Objetos
Facultad de Informática

Juan Pavón Mestras
Dep. Ingeniería del Software e Inteligencia Artificial
Universidad Complutense Madrid

Basado en el curso Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling

Conceptos para la definición de clases

- Atributos
- Constructores
- Métodos
- Parámetros
- Sentencias de asignación
- Control de flujo: sentencias condicionales

Por dónde empezar: qué hacen los objetos

- Lo más importante: la ***visión externa*** de los objetos:
  - **Qué se puede hacer con los objetos**

- Ejemplo: Máquina expendedora
  - En principio una máquina muy sencilla
  - Proporciona billetes a un precio fijo
    - ¿Cómo se podría hacer para tener billetes de distintos precios?
  - ¿Cómo funciona?
    - El usuario mete dinero
    - El usuario pulsa un botón para solicitar el billete
  - La máquina tendrá que saber cuánto dinero se ha introducido

- Ver el proyecto *naive-ticket-machine*
  - *En el campus virtual está un fichero zip con los ejemplos del capítulo*

Estructura básica de la clase

- La visión interna

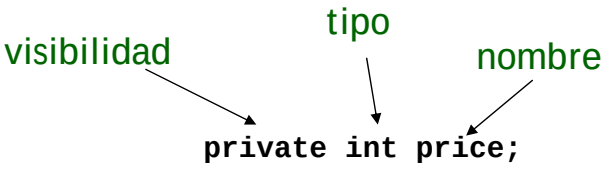
```
public class ClassName
{
    Fields
    Constructors
    Methods
}
```

Campos / Atributos

- Almacenan valores de un objeto
  - Definen el estado del objeto
- También se conocen como *variables de instancia*
- En BlueJ se puede ver con *inspect*

```
public class TicketMachine
{
    private int price;
    private int balance;
    private int total;

    Further details omitted.
}
```

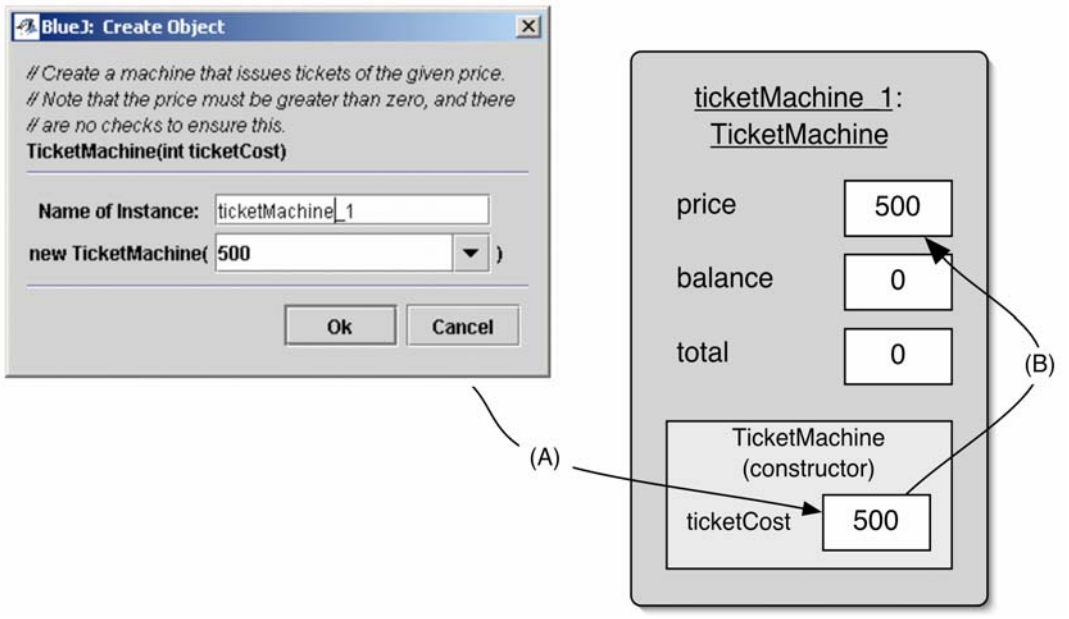


Constructores

- Tienen el mismo nombre que la clase
- Sirven para inicializar un objeto
  - Asignan valores iniciales a las variables
  - Pueden tener parámetros para estos valores

```
public TicketMachine(int ticketCost)
{
    price = ticketCost;
    balance = 0;
    total = 0;
}
```

Paso de datos a través de parámetros



Sentencias de asignación

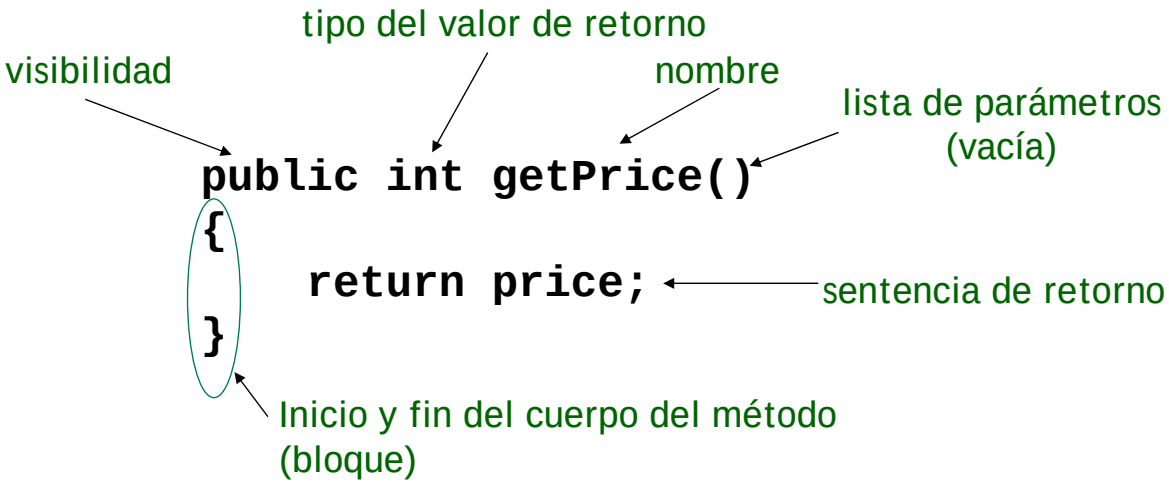
- Permiten almacenar valores en variables
  - Variable = expresión ;
    - Ejemplo: `price = ticketCost;`
  - Cada variable almacena un solo valor, luego el anterior se pierde

Métodos

- Hay muchos tipos
  - Acceso
  - Modificación
  - Acciones
  - Constructores
  - Destructores
- Tienen
  - Cabecera (*signatura*): nombre, parámetros, valor de retorno
  - Cuerpo (implementación): conjunto de sentencias
    - En C++ se pueden declarar en archivos diferentes
    - En Java se definen juntos

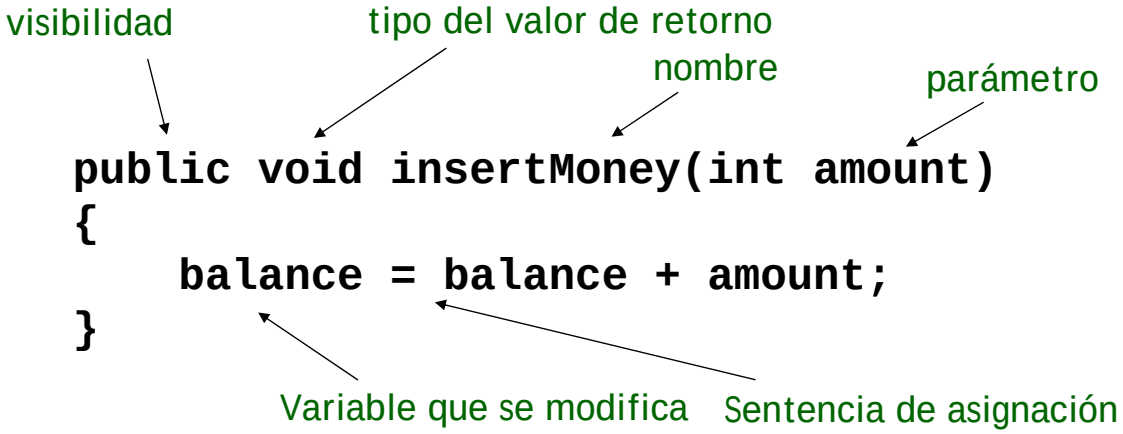
Métodos de acceso

- Proporcionan información sobre un objeto



Métodos de modificación

- Tienen una estructura similar a los métodos de acceso
- Pero sirven para cambiar el estado del objeto
  - Cambiando el valor de una o más variables
  - Utilizan normalmente sentencias de asignación
  - Suelen recibir parámetros de entrada



Métodos de impresión

- Para imprimir en la salida estándar
  - System.out.print (String); // no acaba la línea
  - System.out.println (String); // línea completa

```
public void printTicket()
{
    // Simulate the printing of a ticket.
    System.out.println("###################");
    System.out.println("# The BlueJ Line");
    System.out.println("# Ticket");
    System.out.println("# " + price + " cents.");
    System.out.println("###################");
    System.out.println();

    // Update the total collected with the balance.
    total = total + balance;
    // Clear the balance.
    balance = 0;
}
```

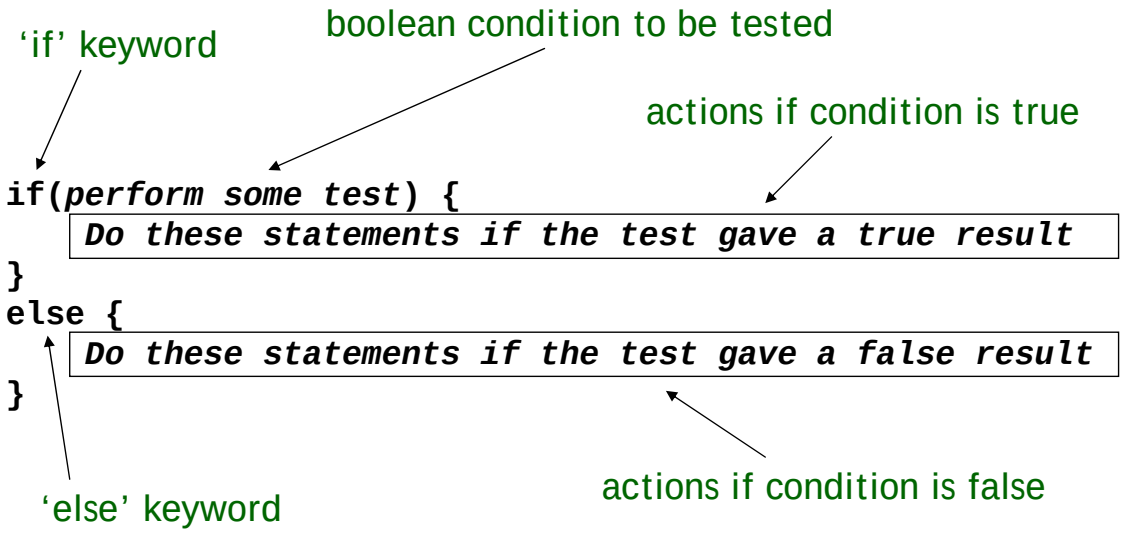
Problemas con el ejemplo de máquina expendedora

- Tiene varios fallos
  - No comprueba la cantidad introducida
  - No da cambio
  - No comprueba si la inicialización es correcta
- Hagamos una máquina más sofisticada…

Sentencia condicional

```
public void insertMoney(int amount)
{
    if(amount > 0) {
        balance = balance + amount;
    }
    else {
        System.out.println("Use a positive amount: " +
                           amount);
    }
}
```

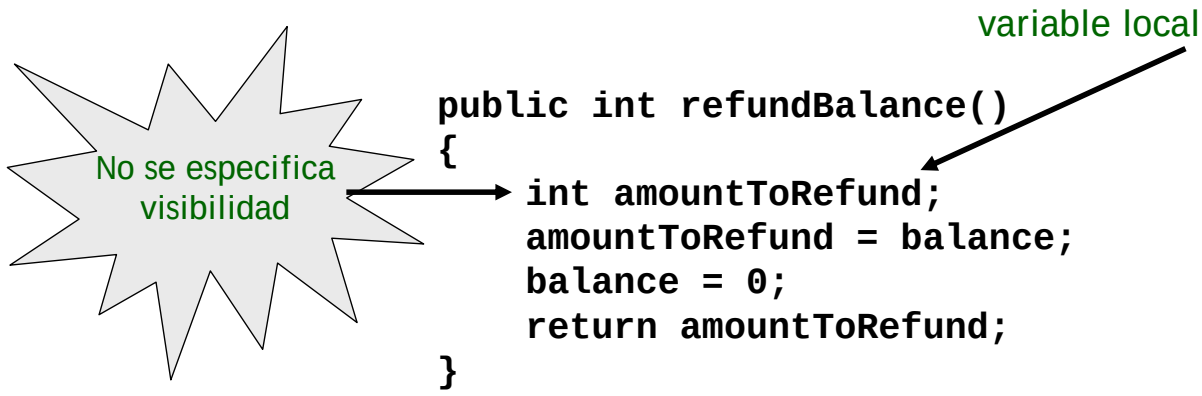
Sentencia condicional



Variables locales

- Las atributos son un tipo de variable
  - Almacenan valores a lo largo de la vida del objeto
  - Se pueden utilizar en el código de la clase
- Los métodos pueden incluir otras variables de vida más limitada
  - Existen sólo durante la ejecución del método
  - Sólo se pueden utilizar dentro del método

Variables locales



Resumen

- El cuerpo de una clase consta de
  - Atributos (campos): almacenan valores que determinan el estado del objeto
  - Constructores: inicializan los objetos
  - Métodos: implementan el comportamiento de los objetos
- Variables
  - Campos (atributos): duran toda la vida del objeto
  - Parámetros: para recibir valores en un método o constructor
  - Variables locales: para almacenar temporalmente algún valor
- Los objetos toman decisiones mediante sentencias condicionales (if)
  - El resultado (true o false) determina uno de dos cursos de acción posibles