

# Fundamentos de la Programación Orientada a Objetos

## Objetos y Clases

---

*Programación Orientada a Objetos*  
*Facultad de Informática*

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



## Conceptos fundamentales de la POO

---

- Objetos
- Clases
- Métodos
- Parámetros
- Tipos de datos

# Objetos

- Los objetos son/representan cosas
- Los objetos pueden ser simples o complejos
- Los objetos pueden ser reales o imaginarios



*Hola Mundo*

# Atributos

- Valores o características de los objetos
- Permiten definir el **estado** del objeto u otras cualidades



- Velocidad
- Aceleración
- Capacidad de combustible
- Marca
- Color
- Potencia
- Velocidad máxima
- Carburante



*variables*



*constantes*



## Métodos (u operaciones)

---

- Los métodos pueden devolver un valor al acabar su ejecución
  - Valor de retorno

## Clases

---

- Representan un tipo particular de objetos
  - Objetos con características y comportamiento similar
  - Categorías de objetos



# Clases

---

- Cada clase tiene asociado un código (definición de la clase), que determina
  - Los atributos que tienen los objetos de la clase
  - Los métodos que pueden ejecutar los objetos de la clase y cómo lo hacen
- Programar orientado a objetos consiste en escribir código de clases de objetos

## Definición de clases en C++

---

```
enum Carburante {  
    diesel, super, sinplomo  
};
```

```
class Coche {  
    char* marca;  
    double vel_max;  
    int potencia;  
    Carburante tipo_carburante;
```

```
    double velocidad;  
    double aceleracion;
```

```
public:  
    void arrancar() {  
        // instrucciones para arrancar el coche  
    };
```

```
    void frenar() {  
        // instrucciones para frenar el coche  
    };  
  
    void acelerar() {  
        // instrucciones para acelerar el coche  
    };  
  
    void girar_derecha(short grados) {  
        // instrucciones para girar a la derecha  
    };  
  
    // etc.  
}; // fin de definición de la clase Coche
```

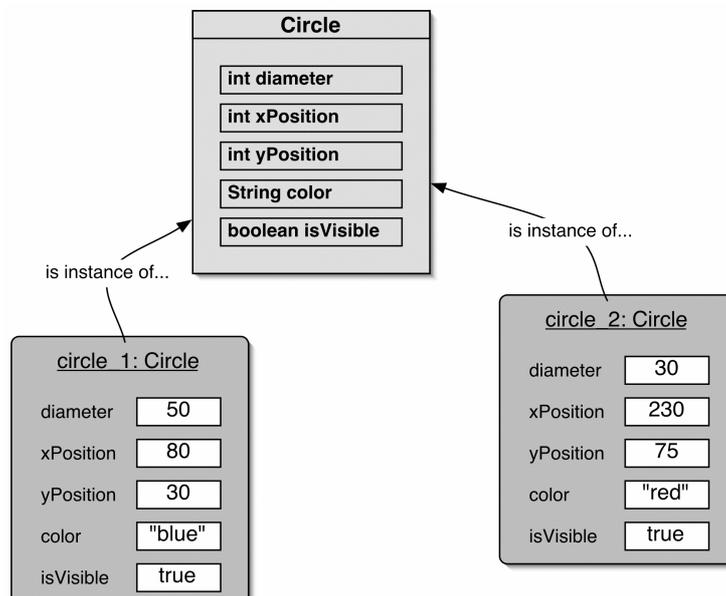
# Definición de clases en Java

```
class Coche {  
  // atributos:  
  String marca;  
  double vel_max;  
  int potencia;  
  String tipo_carburante;  
  
  double velocidad;  
  double aceleracion;  
  
  // métodos:  
  void arrancar() {  
    // instrucciones para arrancar el coche  
  };
```

```
  void frenar() {  
    // instrucciones para frenar el coche  
  };  
  
  void acelerar() {  
    // instrucciones para acelerar el coche  
  };  
  
  void girar_derecha(short grados) {  
    // instrucciones para girar a la derecha  
  };  
  
  // etc.  
}; // fin de definición de la clase Coche
```

## Clases

- De cada clase pueden crearse múltiples objetos
- Cada objeto tiene valores propios asignados a los atributos: **estado** del objeto



## Tipos de datos

---

- Indican la naturaleza de los datos
  - Que se pasan como parámetro o que devuelven los métodos
  - De los atributos de los objetos
  
- Hay algunos *básicos* (dependen del lenguaje de programación)
  - int
  - boolean
  - double
  - String
  - ...
  
- Y otros los definen las clases
  - Objetos de una clase determinada
    - Definida por el usuario
    - De una librería

## Un programa en POO

---

- Un programa consta de un conjunto de instancias o ejemplares de objetos (*object instances*) y un flujo de control principal (*main*)
  
- Durante la ejecución del programa:
  - Los objetos se crean y se destruyen
    - Gestión dinámica de la memoria
  - Se les solicita a los objetos que ejecuten métodos (operaciones)

## Un programa en C++

---

```
main() {
    Coche *c=new Coche(); // crea un objeto Coche
    c-> arrancar           // utiliza el objeto
    // ...
    dispose(c);           // elimina el objeto
}
```

## Un programa en Java

---

```
class Programa {
    public static void main(String args[]) {
        Coche c=new Coche(); // crea un objeto Coche
        c.arrancar();        // utiliza el objeto
        // ...
    } // se elimina el objeto cuando nadie lo puede utilizar
    // i automáticamente !
}
```

## Ejercicios

---

- Instalar BlueJ
  - Se puede encontrar en <http://www.bluej.org/>
- Abrir el ejemplo que tiene el nombre *shapes*
- Crear objetos
  - Seleccionar un objeto y ejecutar *new ClaseObjeto()*
- Ver atributos de objetos
- Invocar métodos en objetos
- Ver el código de las clases de objetos
- Crear un dibujo combinando varios objetos

## Bibliografía

---

- D.J. Barnes y M. Kölling, *Programación orientada a objetos con Java. Una introducción práctica usando BlueJ*. Pearson Educación, 2007
- T. Budd, *An introduction to Object-Oriented Programming (Third Edition)*. Pearson Education, 2001
- E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Paterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994
- B. Stroustrup, *The C++ Programming Language (Third Edition)*. Addison-Wesley, 1997
- Agustín Froufe. *Java 2. Manual de usuario y tutorial*. Ed. Ra-Ma
- J. Sánchez, G. Huecas, B. Fernández y P. Moreno, *Iniciación y referencia: Java 2*. Osborne McGraw-Hill, 2001.
- B. Meyer, *Object-Oriented Software Construction (Second Edition)*. Prentice Hall, 1997