

Programación en Lenguaje Java

Práctica 2.1. Creación de objetos



Michael González Harbour
Mario Aldea Rivas

Departamento de Matemáticas,
Estadística y Computación

Este tema se publica bajo Licencia:

[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Práctica 2-1. Creación de objetos

Objetivos:

- Familiarizarse con los procesos de compilación y ejecución de programas
- Practicar con la creación de objetos y la invocación de sus métodos

Parte 1: compilar y ejecutar un programa desde la línea de comando

Parte 2: compilar y usar una clase desde `bluej`

Parte 3: crear un programa main que crea un objeto y usa sus métodos

Parte avanzada: Usar una clase externa para hacer una gráfica

Parte 1: compilar y ejecutar un programa desde la línea de comando

Compilar y ejecutar la clase `Hola` del capítulo 1 de los apuntes desde el intérprete de órdenes

Hacer una captura de pantalla que muestre la compilación y la ejecución, y meterla en el informe

Parte 2: compilar y usar una clase desde bluej

Crear en bluej un nuevo proyecto

Crear y compilar en él la clase Piedra suministrada

Con el ratón:

- crear un par de objetos de la clase Piedra
- usar sus métodos

Hacer dos capturas de pantalla que muestren, cada una, el resultado de usar el método posY() sobre ambos objetos

- una captura después de crear los objetos
- una segunda captura después de haber invocado el método avanzaTiempo() para ambos objetos con $t=0.5$ s

Parte 3: crear un programa main que crea un objeto y usa sus métodos

Crear en el proyecto una nueva clase con un método `main` que:

- crea un objeto de la clase `Piedra` que se suministra
- muestra en pantalla los resultados de invocar `posX()` y `posY()`
- invoca el método `avanzaTiempo()` para $t=0.5$ s
- vuelve a mostrar en pantalla los resultados de invocar `posX()` y `posY()`

Hacer una captura de pantalla que muestre los resultados

Crear y usar objetos

Crear objetos de una clase:

- Sintaxis

```
Clase nombreObjeto=new Clase(parámetros);  
// los parámetros son los del constructor
```

- Ejemplo

```
Piedra p=new Piedra(0.0,100.0,30.0,20.0);  
// los parámetros indican la pos. y vel. iniciales
```

Invocar un método del objeto:

- Sintaxis

```
nombreObjeto.nombreMétodo(parámetros);  
// los parámetros son los que requiera el método
```

- Ejemplo

```
p.avanzaTiempo(0.5);  
// avanza el tiempo 0.5 segundos
```

Parte avanzada

Crear una gráfica de la evolución de la piedra. Para ello:

Crear una nueva clase con un `main` que haga:

- crea un objeto de la clase `Grafica`
- crea un objeto de la clase `Piedra`
- inserta en la gráfica las posiciones `x` e `y` de la piedra
- repite 10 veces:
 - inserta en la gráfica las posiciones `x` e `y` de la piedra
 - avanza el tiempo 0.5 segundos
- pinta la gráfica

Hacer una captura de la gráfica para el informe

Hacer una gráfica sencilla

Métodos de la clase `Grafica`, del paquete `fundamentos`

new <code>Grafica</code> (String titulo, String tituloX, String tituloY)	Constructor que pone los títulos de la ventana y de los ejes X e Y
void <code>inserta</code> (double x, double y)	Inserta el punto (x,y) en la gráfica actual
void <code>pinta</code> ()	Pinta la gráficas

Ejemplo:

```
Grafica g = new Grafica ("Piedra", "x", "y");  
g.inserta(x1,y1);  
g.inserta(x2,y2);  
...  
g.pinta();
```

Nota: acordarse de importar el paquete `fundamentos`:

```
import fundamentos.*; // al principio de la clase
```


Entregar

Parte obligatoria

- La captura de pantalla de la parte 1
- Las dos capturas de pantalla de la parte 2
- La captura de pantalla de la parte 3

Parte avanzada, si se ha hecho

- El código de las clase principal
- Una captura de pantalla con la gráfica