

Tecnología de Videojuegos

Práctica 2: Introducción a la POO en Java

UAH, Departamento de Automática, ATC-SOL
<http://atc1.aut.uah.es>

Semana de laboratorio 2

Objetivos:

- Practicar la sintaxis básica de Java relacionada con POO
- Afianzar los conceptos elementales de POO
- Desarrollar el pensamiento algorítmico

Comentario inicial

En Java, a diferencia de C, un mismo proyecto puede contener más de una función `main()`. Más concretamente, cada clase puede tener una función `main()`, por eso es necesario indicar qué clase se quiere empezar a ejecutar cuando se invoca la máquina virtual de Java. En las prácticas puede integrar la función `main()` dentro de la clase (y por lo tanto el mismo archivo), o puede crear un archivo nuevo dentro del mismo paquete y ubicar ahí el `main()`. Se sugiere integrar el `main()` dentro de la clase, dado que es la práctica más habitual.

Ejercicio 1

Implemente la clase `Coche` que represente coches. Incluya los atributos `marca`, `modelo`, `marcha` y `color`; y los métodos que simulen las acciones de arrancar el motor, describir el coche, cambiar de velocidad (subir y bajar marcha), acelerar, frenar y parar el motor (simplemente muestre un texto por pantalla o modificando algún atributo).

Muestre el funcionamiento de la clase creando tres instancias de la clase representando coches diferentes, llamando a sus métodos y visualizando el resultado por pantalla.

Ejercicio 2

Cree una clase `Polinomio` que implemente el siguiente polinomio:

$$f(x) = ax^3 + bx^2 + cx + d \tag{1}$$

Con los atributos `a`, `b`, `c` y `d`, de tipo `double`, y los métodos siguientes:

1. `public void setA(double a)`: Establece el valor del coeficiente a.
2. `public void setB(double b)`: Establece el valor del coeficiente b.
3. `public void setC(double c)`: Establece el valor del coeficiente c.
4. `public void setD(double d)`: Establece el valor del coeficiente d.
5. `public void setABCD(double a, double b, double c, double d)`: Establece el valor de los coeficientes a, b, c y d.
6. `public double compute(double x)`: Calcular el valor del polinomio para la x dada.

Ejercicio 3

Diseñe e implemente una clase Calculadora que calcule la suma, resta, división y multiplicación de dos números que se pasarán como argumento a los métodos. ¿Es necesario utilizar atributos? Discútalos con el profesor del laboratorio.

Ejercicio 4

Implemente una clase TableroAjedrez que visualice en modo texto un tablero de ajedrez (8x8), usando 'N' para identificar casillas negras, y 'B' las casillas blancas. Implementar un método adicional que imprima el tablero, pero indicando además a qué casillas se puede mover un alfil dada una determinada posición en forma (fila, columna).

Utilice los siguientes métodos:

1. `public void imprimirTablero()`: Imprime un tablero sin fichas.
2. `public void imprimirTablero(int fila, int columna)`: Imprime el tablero y las posiciones a las que se puede mover un alfil en la ubicación dada.

Optativo: Diseñe (e implemente) una nueva clase que permita visualizar el tablero y los movimientos válidos de varias figuras (por ejemplo, torre, caballo y alfil) atendiendo a los principios de modularidad, encapsulamiento y reutilización. Discuta el diseño con el profesor del laboratorio.

Ejercicio 5

Repetir el ejercicio 3 usando una interfaz que contenga las funciones de la calculadora e implementar dicha interfaz en una clase.

Ejercicio 6

Hacer un proyecto que integre el cálculo de polinomio del ejercicio 2 y la calculadora del ejercicio 5, de tal manera que todos los cálculos en la clase `Polinomio` se realicen invocando a la calculadora. El proyecto debería estar dividido en las siguientes clases con sus correspondientes paquetes:

1. `p2.ej6.calculadora.InterfazCalculadora`: Interfaz que define las operaciones de la calculadora.
2. `p2.ej6.calculadora.Calculadora`: Clase que implementa la interfaz anterior.
3. `p2.ej6.polinomio.Polinomio`: Clase desarrollada en el ejercicio 2 modificado para usar la clase `Calculadora`.
4. `p2.ej6.Main`: Clase que contiene la función `main()`.