



Semana 11: Objetos

Programación

2020 - 2021

- Ejercicio 1.** Crear una clase `Estudiante` que represente a un estudiante de primer año. Sus atributos son: `nombre`, `apellido`, `nota programación`, `nota álgebra`, `nota cálculo`, `nota física`, `nota técnicas` y `nota humanidades`. Crear un método `init` que reciba valores para cada uno de los atributos y compruebe que los rangos de los valores de las notas introducidas es el adecuado (si son incorrectos la nota se inicializa con un 0). Escribir un programa que cree un objeto de este tipo, solicite al usuario los valores de los campos por teclado e imprima por pantalla la información del estudiante.
- Ejercicio 2.** Crear un nuevo tipo de datos denominado `TrianguloRectangulo` para representar un triángulo rectángulo, tendrá dos atributos: `base` y `altura`. Crear un método `init` para recibir los valores de los campos que compruebe que son correctos. Crear también un programa que pida la base y la altura de un triángulo rectángulo y que guarde los datos en un objeto triángulo rectángulo. A continuación, pedir al usuario que pulse 0 para calcular el área del triángulo o 1 para calcular el perímetro. Imprimir por pantalla el resultado de calcular la elección del usuario.
- Ejercicio 3.** Crear una clase `Dado`, con campos `nombre` y `tiradas`, para simular una partida de juego de dados. Crear un método `init` que recibe el nombre del jugador y un entero `n` que representa el número de tiradas (debe ser mayor que cero). Asignará el nombre al campo correspondiente y creará una lista de `n` posiciones rellena con números entre el 1 y el 6. Se simulará una partida de dados, preguntando a cada jugador su nombre y el número de tiradas del juego. El ganador será el que tenga el mayor número de dados iguales. En caso de empate ganará el jugador cuyos dados sumen más.
- Ejercicio 4.** Crear una clase que represente un intervalo temporal:
- Un intervalo temporal se representa por un instante inicial (horas y minutos) y un instante final (horas y minutos)
 - Los objetos de esta clase deben contener también información sobre si el intervalo pertenece a la mañana (6:00 a 12:00), la tarde (12:00 a 18:00), la noche (18:00 a 0:00) o la madrugada (0:00 a 6:00). Cabe destacar que un intervalo temporal puede pertenecer a varias de las categorías anteriores. Por ejemplo el intervalo que va desde las 21:00 a las 23:15 pertenece únicamente a la noche, pero el que va de las 11:00 a las 19:00 pertenece a la mañana, la tarde y la noche.
 - Crear un método `init` que reciba valores para las horas y los minutos del intervalo. Debe comprobar que los valores recibidos son correctos de la siguiente forma:
 - Si el valor de la hora inicial no es correcto, copiará el valor de la hora final y viceversa. Si ninguno de los dos es correcto, se pondrá un 0 en ambos.
 - Si los minutos iniciales o finales no son correctos, se pondrá un 0 en el incorrecto.
 - El instante inicial debe ser anterior al instante final, en caso contrario se intercambiarán.
 - Además, el método `init` debe rellenar automáticamente los valores de mañana/tarde/noche/madrugada.

Crear un programa que le pregunte al usuario cuántos intervalos temporales quiere y los guarde en una lista. Para finalizar debe imprimir los intervalos como sigue:

Intervalo temporal: [11:59-12:18]
Pertenece a: mañana tarde

Ejercicio 5. Crear clases para representar los objetos necesarios para el proyecto final. Para todos ellos crear métodos `init` que comprueben que los valores recibidos son correctos.

Normas de entrega

Los ejercicios se deben subir a Aula Global antes de las 7:00 de la mañana del próximo día con clase de laboratorio). Se deberá subir un fichero comprimido **zip** con un fichero para cada programa, nombrados `ejercicio1.py`, `ejercicio2.py`, etc. El nombre del fichero será “s11-iniciales-del-alumno.zip” (por ejemplo Lucía Pérez Gómez subirá un archivo llamado `s11-lpg.zip`).