



Examen Final Evaluación 19 6 2023

Programación Primero DAM

Tiempo 1 Hora 30 minutos cada Parcial

PRIMERA EVALUACIÓN

Ejercicio 1 2.5 puntos

Dos números primos son primos gemelos si su diferencia es igual a 2. Es decir, una pareja de la forma $(p, p+2)$ siendo p un número primo es una pareja de números primos. Por ejemplo, las parejas (3, 5) y (11,13) son dos parejas de primos gemelos. Listar por pantalla todos los números gemelos entre 0 y 100.

Ejercicio 2 2.5 puntos

Calcular el coste del estacionamiento en un parking de modo que las 4 primeras horas se cobran a 3,50 euros y las siguientes a 1,50 hasta llegar a un tope de 35 euros para 24 horas. Si permanece aparcado más de un día se cobrarán 30 euros los días completos y 1,80 las horas sueltas.

Ejercicio 3 2.5 puntos

Programar un algoritmo recursivo que permita invertir un número. Ejemplo: Entrada: 1234 Salida: 4321

Ejercicio 4 2.5 puntos

Método que introduzca la estatura en cm de una persona y la convierta a pies y pulgadas.

1 pie (ft) = 12 in = 30,48 cm

1 pulgada (in) = 2,54 cm

Pies	Pulgadas	Metros
<input type="text" value="5"/>	<input type="text" value="10"/>	= <input type="text" value="1,79"/> m

SEGUNDA EVALUACIÓN

Ejercicio 1 2.5 puntos

Genera un array de 50 posiciones enteras que tomarán valores entre -100 y 100. Separe en dos matrices los valores positivos y negativos. Ordene ambos arrays por el método de la Burbuja.

Considere el cero positivo. A continuación, separe el array de positivos en dos arrays, uno de pares y otro de impares; muestre ambos.

Ejercicio 2 2.5 puntos

Programa que pida al usuario la introducción de una cadena y que compruebe si la misma es un Palíndromo, indicándoselo al usuario. Una frase es un Palíndromo si se lee igual al derecho que al revés.

Ejemplos: Ana lleva al oso la avellana. Sor Rebeca hace berros.

Ejercicio 3 2.5 puntos

Crea la clase Pizza con los atributos y métodos necesarios. Sobre cada pizza se necesita saber el tamaño - mediana o familiar - el tipo - margarita, cuatro quesos o funghi - y su estado - pedida o servida.

La clase debe almacenar información sobre el número total de pizzas que se han pedido y que se han servido. Siempre que se crea una pizza nueva, su estado es "pedida".

Testee la clase creando un arraylist de pizzas. Muestre la información de cada una. Indique el total de pedidas y el total de servidas

Ejercicio 4 2.5 puntos

Utilizando la clase Producto un supermercado nos pide que hagamos una pequeña aplicación que almacene los productos pasados por el escaner.

La aplicación debe almacenar Productos (clase), cada producto al crearse contiene una cantidad, un precio (estos dos generados aleatoriamente). El nombre del producto será básico (producto1, producto2, producto3, etc.). El precio ya viene con los impuestos incluidos.

Calcular el precio total de un arraylist de entre 1 y 8 productos (aleatorio).

Mostrar un ticket con todo lo vendido y el precio final como se hacen en los supermercados.

```
*****Cantidad****Precio*****Total
Producto1          5      3.5          17.5
Producto2          7      2.5          17.5
-----
Precio final                               35
```

TERCERA EVALUACIÓN

Ejercicio 1 3.5 puntos

Crear un proyecto llamado Concesionario como sigue:

Crear una superclase llamada Car. La clase Car tiene los siguientes campos y métodos:

```
int velocidad; double PrecioNormal; String color;  
double getPrecioVenta();
```

Crear una subclase de Car y llamarla Truck. La clase Truck tiene los siguientes campos y métodos:

```
int peso;  
double getPrecioVenta(); // Si peso > 2000, 10% descuento. En otro  
caso, 20% descuento.
```

Crear una subclase de Car y llamarla Ford. La clase Ford tiene los siguientes campos y métodos:

```
int año; int DescuentoFabricante;  
double getPrecioVenta(); // al precio de venta calculado de la clase  
Car, restar DescuentoFabricante;.
```

Crear una subclase de Car y llamarla Sedan. La clase Sedan tiene los siguientes campos y métodos:

```
int longitud;  
double getPrecioVenta(); // Si longitud > 5 metros, 5% descuento, en  
otro caso, 10% descuento.
```

a/ Crear la clase Testconcesionario que contiene el método main(). Realizar lo siguiente dentro del mismo:

Crear un objeto de la clase Sedan e inicializar todos los campos con valores apropiados.

Crear dos objetos de la clase Ford e inicializar todos los campos con valores apropiados.

Crear un objeto de la clase Car e inicializar todos los campos con valores apropiados.

Añadirlos a una lista y mostrar utilizando el bucle apropiado los precios de venta de todas las instancias y el precio de venta total.

b/ Añada dos clases nuevas, una que herede de Car llamada SUV y otra de Truck llamada VAN. La clase SUV tendrá un atributo double llamado potencia, y el método getPrecioVenta() será el heredado añadiendo 5 por la potencia si esta es mayor que 200, y 3 por la potencia si es menor que 200. La clase VAN tendrá un atributo double llamado cilindrada, y el método getPrecioVenta() será el heredado añadiendo 3 por la cilindrada si esta es menor que 3000, y 4 por la cilindrada si es mayor que 3000.

Vuelva a crear otra clase Testconcesionario2 añadiendo dos objetos de cada clase nueva y mostrando el precio de venta de cada uno y el total.

c/ Implemente la Interfaz Multable

```
int MULTA_MAXIMA=100;  
String Multa(int cantidad)  
String disminuirMulta (int cantidad);  
String aumentarMulta (int cantidad);
```

Los métodos indicarán por pantalla la nueva multa recibida. No se permitirá superar la multa máxima, indicándose dicha imposibilidad por consola.

Evidentemente tampoco la multa podrá ser inferior a cero.

Cree una lista imponiendo multas a todos los objetos creados en el test2, mostrandolas por consola

Vuelva a recorrer la lista aumentando o disminuyendo las multas y vuelva a mostrarlas por pantalla.

Ejercicio 2 3 puntos

Dada la Clase Felino.java del material auxiliar se pide:

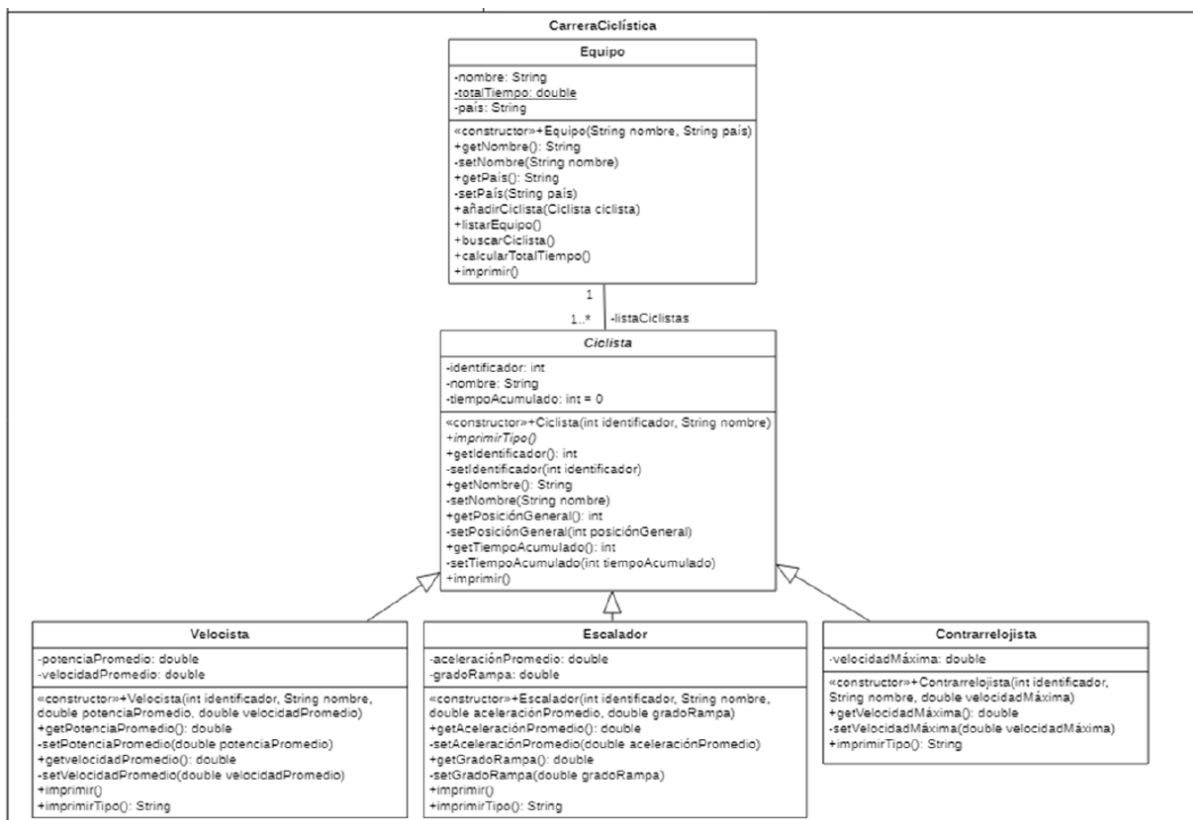
1. Crear la clase Leopardo que hereda de Felino y tiene dos atributos privados: pelajeManchado de tipo booleano que indica si el leopardo tiene o no manchas en el pelaje, y un atributo habitat de tipo String.
2. Crear un constructor para la clase Leopardo que recibe valor para todos los parámetros (incluidos los heredados). Hacer otro constructor por defecto sin parámetros que use el constructor con parámetros anterior, añada un método toString.
3. Modificar la clase Leopardo y la clase Felino para que se puedan serializar los objetos de estas clases.
4. Crear un método public void guardar (String fichero) en la clase Leopardo que guarde un objeto de la clase Leopardo en el fichero que se le pase por parámetro. Se deberán gestionar las excepciones pertinentes.
5. Crear un método public Leopardo leer (String fichero) de la clase Leopardo que lea un objeto de la clase Leopardo del fichero que se le pase por parámetro. Se deberán gestionar las excepciones pertinentes.

Crear una clase de prueba donde se cree un arraylist de 3 felinos que se guardaran como objetos en el fichero gatos, usando el metodo guardar.

Con posterioridad recupera desde dicho fichero los tres objetos usando el método leer y muéstralos por consola con el método toString.

Ejercicio 3 3.5 puntos

Dada la siguiente estructura jerárquica de clases



1/ Cambie el tipo de dato del atributo tiempoAcumulado a double

2/ escriba para la clase ciclista el método void calcularTiempoParcial() de manera que obtenga un valor double entre 5 y 20, de forma aleatoria, y se añada al atributo tiempoAcumulado.

3/ para los velocistas heredará y reescribirá este método de forma que además de hacer lo anteriormente indicado le añada un valor de 4 por la velocidad promedio y por último se añada al atributo tiempoAcumulado.

4/ en el caso de los escaladores exactamente igual y añadiendo un valor de 3 por la aceleración promedio y ese total se añadira al atributo tiempoAcumulado.

5/ por ultimo los contrarrelojistas calcularan el tiempo parcial con ese aleatorio mas un valor de 5 por la velocidad maxima y ese total se le sumara al atributo tiempoAcumulado.

6/ En el testeo creará DOS equipos de 3 ciclistas (uno de cada tipo) que participarán en una competición de TRES ETAPAS, DONDE EL TIEMPO DE CADA ETAPA SE IRÁ ACUMULANDO

7/ para cada etapa calculará el tiempo parcial de cada ciclista de cada equipo y mostrará la clasificación por equipos y la clasificación individual de ciclistas.