



PROGRAMACIÓN
Examen 1^{er} Trimestre

Importante: Debes realizar sangría en el código (tabulaciones) para mejorar su legibilidad. No hacerlo supone una penalización de entre -0'25 y -0'5 dependiendo del valor del ejercicio.

Debes realizar un proyecto llamado Examen1Trimestre y crear un archivo nuevo por cada ejercicio (o mas si fuese necesario).

1. Codifica un programa en Java que pida números al usuario hasta que éste introduzca un valor negativo. A continuación el programa imprime cuántos números positivos o iguales a cero se han leído, el valor medio de todos ellos y el valor más alto introducido por el usuario y en qué posición se introdujo. (2 puntos).
2. Haz un programa que tenga los siguientes métodos estáticos:

Para realizar este ejercicio debes declarar las siguientes variables estáticas:

`int limiteInferior;`

`int limiteSuperior;`

- a. `creaIniArray()` -> Debe solicitar al usuario el tamaño del Array, iniciar un Arraye y devolverlo. Debe comprobar que el tamaño introducido sea mayor que 0, sino lo fuese debe descartarlo y solicitar otro número. (0.5 puntos)
- b. `esPrimoConLimite (int numero, int inf, int sup)` -> Este método debe comprobar si el número está entre los límites deseados y además si es primo, en caso afirmativo devuelve true, en caso negativo devuelve false. (1.5 puntos)

- c. pideLimites() -> Solicita los límites al usuario, debe guardarlos en las variables estáticas limiteSuperior y limiteInferior. Debe comprobar que efectivamente el limiteSuperior sea mayor que el limiteInferior y que sean mayores o iguales a 0 . En caso contrario vuelve a solicitarlos. (0.75 puntos)
- d. rellenarArray(int [] miArray) -> Debe solicitar al usuario números para rellenar el array, por cada número debe llamar a la función esPrimoConLimite y solo meter el número en el array si es primo y está entre los límites. Sino debe solicitar otro. (1 punto).
- e. mostrarArray(int [] miArray) -> Muestra el array por pantalla (0,25 puntos).

Haz un main con los pasos necesarios para mostrar el funcionamiento del programa.

3. Realiza un programa que tenga las siguientes clases:

- a. Clase Titular (0.5 puntos) Atributos privados y métodos públicos
 - i. String nombre
 - ii. String apellidos
 - iii. Int edad
 - iv. Constructor con todos los atributos.
 - v. Get y Set de todos los atributos.
 - vi. Método muestraTitular(), que muestra por pantalla los datos del titular.
- b. Clase CuentaCorriente (3.5 putos) Atributos privados y métodos públicos
 - i. Titular elTitular (Sino sabes hacerlo con composición hazlo con atributos pero pierdes 0.5 puntos)
 - ii. String numCuenta
 - iii. Double saldo

- iv. Constructor con todos los atributos
- v. Constructor sin saldo que inicia el saldo a 15.5
- vi. Get para todos los atributos
- vii. Set para el saldo
- viii. Tenga un método ingresar que incremente el saldo en una cantidad que le pasa por teclado el usuario. Si se intenta meter una cantidad negativa el método informará de que no es posible hacer la operación.
- ix. Método ingresar que ingrese una cantidad pasada por parámetro al método. Si se intenta meter una cantidad negativa el método informará de que no es posible hacer la operación.
- x. Método imprimirCuenta que imprima la info de la cuenta así como la de su titular.
- xi. Método retirarSaldo que pide al usuario que introduzca el saldo a retirar, se debe restar lo retirado del saldo pero solo si no deja el saldo por debajo de 0. Si se intenta sacar una cantidad negativa el método informará de que no es posible hacer la operación.
- xii. Método retirarSaldo igual que el anterior pero se pasa por parámetro la cantidad a retirar.
- xiii. Tenga un método que compruebe si el saldo es positivo o negativo y que diga el numero de cuenta y si su saldo es positivo o negativo (se considera 0 como positivo).
- xiv. Tenga un método para compararCuentas, que compare la cuenta this con otra que se pasa por parámetro. Debe ver que cuenta tiene mayor saldo y mostrarla mediante el numero de cuenta, si tuviesen el mismo saldo debe informar de ello.
- xv. Método esIgual que compara la cuenta this con otra pasada por parámetro y dice si son iguales, dichas cuentas serán iguales si su numCuenta es el mismo.

En el main del programa habrá que crear varios Titulares y CuentasCorrientes usando los diferentes constructores y habrá que usar todos los métodos y comprobar que funcionan.