

Fundamentos de Programación

Relación de Problemas Tema 3



1. Crear una clase llamada **generica** con dos atributos, uno de ellos será un valor entero y el otro se corresponderá con su raíz cuadrada. Esta clase tendrá los siguientes métodos públicos:

- **void informacion();** /* Pide el valor del número entero por teclado */
- **void operacion();** /* Calcula el valor de la raíz cuadrada del número introducido, y lo almacena en el atributo correspondiente. Si no es posible calcular la raíz cuadrada, el resultado de este método será -1 */
- **void resultado();** /* Muestra por pantalla el valor de la raíz cuadrada del número que se ha leído por teclado, o bien un mensaje de texto indicando que ha ocurrido un error */

Implementar un pequeño programa que haga uso de esta clase.

2. Crear una clase llamada **generica** con dos atributos, el primero será un valor entero, y el segundo indicará si dicho valor es un número par o impar. Esta clase tendrá los siguientes métodos públicos:

- **void informacion();** /* Pide el valor del número entero por teclado */
- **void operacion();** /* Determina si el número introducido es par o impar, almacenando el resultado en el atributo correspondiente */
- **void resultado();** /* Muestra por pantalla el resultado obtenido tras la utilización del método operación() */

Implementar un pequeño programa que haga uso de esta clase.

3. Crear una clase llamada **TipoFecha** con los siguientes atributos:

- **dia** /* Contiene el día indicado por teclado */
- **mes** /* Contiene el mes indicado por teclado */
- **agno** /* Contiene el año indicado por teclado */

Esta clase tendrá un método privado, que será:

- **int bisiesto();** /* Determina si el año introducido por teclado es o no bisiesto */

Además contará con los siguientes tres métodos públicos:

- **void informacion();** /* Lee por teclado una fecha en formato dd mm aaaa, y la almacena en los atributos correspondientes */
- **void diasiguiente();** /* Muestra por pantalla la fecha del día siguiente. Al realizar la implementación de este método deberemos tener en cuenta la característica peculiar del mes de febrero cuando el año es bisiesto (por simplificar la condición, un año es bisiesto si es divisible entre 4). Se supondrá que la fecha introducida por teclado es correcta */
- **void fechatexto();** /* Escribe por pantalla la fecha, que ha sido introducida por teclado, con el siguiente formato “Huelva, a dd de mm de aaaa”, donde dd y aaaa son número y mm es el nombre del mes */

Implementar un pequeño programa que haga uso de esta clase.

4. Crear una clase llamada **TipoFecha** con un atributo entero que haga referencia al guarismo de un día del mes de Marzo de 2003. Esta clase tendrá dos métodos públicos, que serán los siguientes:

- **void informacion();** /* Pide por teclado el guarismo de un día del mes de Marzo de 2003 y lo almacena en el atributo de la clase */
- **void nombredia();** /* Sabiendo que el día 1 de Marzo de 2003 fue sábado, determina a qué día correspondió el guarismo leído desde teclado. Por ejemplo, si el usuario introduce el día 19, este método muestra por pantalla “El día 19 de Marzo de 2003 fue miércoles” */

Implementar un pequeño programa que haga uso de esta clase.

5. Crear una clase llamada **generica** con un atributo que haga referencia a un número entero. Esta clase contendrá dos métodos públicos:

- **void informacion();** /* Solicita un número por teclado y lo almacena en el atributo de la clase */
- **int operacion();** /* Devuelve 1 si el número solicitado por teclado está entre 10 y 100, y 0 en caso contrario */

Implementar un pequeño programa que haga uso de esta clase. El programa deberá mostrar por pantalla si el número solicitado está comprendido en el intervalo [10,100]

6. Crear una clase llamada **TipoInversion**, que contenga dos atributos:

- **Cinicial**, que hará referencia al capital inicial que deposita un inversor.
- **Interes**, que se referirá al interés al que ha colocado su inversión.

Esta clase tendrá tres métodos públicos:

- **void informacion();** /* Pide por teclado y almacena en los atributos correspondientes, el capital depositado y el interés al que se ha colocado, por parte de un inversor */
- **float CapitalFinal();** /* Devuelve el capital final al término de 2 años de inversión. Teniendo en cuenta la fórmula siguiente: $\text{Capital_Final} = \text{Capital_Inicial} \times (1 + \text{Interes})^2$ */
- **float CapitalInicial();** /* Devuelve el capital inicial invertido */

Implementar un pequeño programa que haciendo uso de esta clase determine si después de dos años de inversión, el capital se habrá doblado o bien será mayor o menor que el inicial.

7. Crear una clase llamada **generica** que contenga dos atributos haciendo referencia a valores enteros. Esta clase contará con dos métodos públicos, que serán los siguientes:

- **void informacion();** /* Solicita por teclado dos números enteros, y los almacena en los atributos de la clase */
- **void operacion();** /* Solicita al usuario la operación a realizar con los dos números introducidos por teclado, de modo que si el usuario pulsa '+', 'S' o 's' el método mostrará un mensaje indicando el resultado de la suma de ambos números; pero si el usuario indica '-', 'R' o 'r' el método mostrará el resultado de la resta. Para cualquier otra pulsación se deberá indicar un mensaje de error, indicando que la operación solicitada no es válida */

8. Crear una clase llamada **generica** que contenga un atributo que haga referencia a un número entero. Esta clase contará con dos métodos públicos:

- **void informacion();** /* Solicita al usuario el número del cuál quiere conocer su tabla de multiplicar */
- **void tabla();** /* Muestra por pantalla la tabla de multiplicar correspondiente al valor almacenado en el atributo de la clase */

Implementar un pequeño programa que utilice esta clase.

9. Crear una clase llamada **generica** que contenga un atributo que haga referencia a la cantidad de números de los cuales queremos conocer su cuadrado. Esta clase contará con dos métodos públicos:

- **void informacion();** /* Solicita al usuario la cantidad, N, de números a los que se le va a calcular su cuadrado */
- **void cuadrados();** /* Muestra por pantalla el cuadrado de los N primeros números enteros */

Implementar un pequeño programa que utilice esta clase.

10. Crear una clase llamada **TipoEstadistica** que contenga los siguientes atributos:

- **nalum** /* Contiene el número de alumnos examinados */
- **susp** /* Almacena el número de suspensos */
- **apr** /* Contiene el número de aprobados */
- **not** /* Contiene el número de notables */
- **sob** /* Almacena el número de sobresalientes */

Además esta clase dispondrá de los siguientes métodos públicos:

- **void informacion();** /* Pregunta al usuario el número de alumnos que se han presentado a un examen */
- **void elaborarEstadistica();** /* Pide por teclado las notas, que deben estar comprendidas en el rango [0,10], de los diferentes alumnos presentados al examen */
- **void resultado();** /* Muestra por pantalla el número de suspensos, aprobados, notables y sobresalientes que ha habido en el examen */

Implementar un pequeño programa que utilice esta clase.

11. Crear una clase llamada **generica** con un atributo de tipo entero, y con los siguientes métodos públicos:

- **void informacion();** /* Pregunta al usuario de cuántos números quiere conocer su cuadrado, almacenando dicho número en el atributo correspondiente (llamado N, por ejemplo) */
- **void cuadrados();** /* Muestra por pantalla los cuadrados de los N primeros números enteros divisibles entre 3 */

Implementar un pequeño programa que utilice esta clase.

12. Crear una clase llamada **generica** con un atributo de tipo entero, y con los siguientes métodos públicos:

- **void informacion();** /* Pregunta al usuario cuántos números quiere sumar, almacenando dicho número en el atributo correspondiente (llamado N, por ejemplo) */
- **void suma();** /* Muestra por pantalla la suma de los N primeros números enteros, exceptuando de dicha suma aquellos números que sean primos */

Implementar un pequeño programa que utilice esta clase.

13. Crear una clase llamada **generica** con dos atributos:

- **x** /* Es la base, es decir el número al que le queremos calcular una potencia */
- **n** /* Es el exponente al que queremos elevar el número x */

Esta clase deberá tener los siguientes métodos públicos:

- **void informacion();** /* Pide por teclado la base y el exponente, para almacenarlos en los atributos correspondientes */
- **float elevar();** /* Devuelve el resultado de calcular x^n , donde x puede ser cualquier número entero, y n puede ser cualquier número entero positivo, negativo o nulo */

Implementar un pequeño programa que utilice esta clase y muestre por pantalla el resultado de elevar un número a una potencia.

14. Crear una clase llamada **generica** con un atributo de tipo entero, y con los siguientes métodos públicos:

- **void informacion();** /* Pide por teclado un número entero que debe ser positivo, para almacenarlo en el atributo de la clase */
- **long int factorial();** /* Calcula el factorial del número almacenado en el atributo de la clase */

Implementar un pequeño programa que utilice esta clase y muestre por pantalla el factorial del número solicitado por teclado.

15. Crear una clase llamada **generica** con dos atributos de tipo entero:

- **inf** /* Hace referencia al límite inferior de un intervalo */
- **sup** /* Es el límite superior de un intervalo */

Esta clase deberá tener los siguientes métodos públicos:

- **void informacion();** /* Pide al usuario los límites inferior y superior de un intervalo de números enteros, comprobando que el inferior es menor que el superior y almacenándolos en los atributos correspondientes */
- **void PrimosIntervalo();** /* Muestra por pantalla todos los números primos comprendidos en el intervalo [inf, sup] */

Implementar un pequeño programa que utilice esta clase.

16. Crear una clase llamada **generica** que contenga los siguientes atributos que hacen referencia a la fórmula que aparece a continuación:

$$E(x) = 1 + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$$

- **x** /* Valor de tipo entero, para el cual obtendremos el valor de la fórmula */
- **n** /* Valor entero que determina el número de términos que compondrán la fórmula */
- **E** /* Atributo que almacenará el resultado de evaluar la fórmula con los valores de los atributos x y n */

Esta clase deberá tener los siguientes métodos públicos:

- **void informacion();** /* Pide al usuario los valores x y n para evaluar la fórmula, almacenándolos en los atributos correspondientes */
- **void Ex();** /* Calcula el resultado de la fórmula para los valores x y n indicados por el usuario, y lo almacena en el atributo E */
- **void resultado();** /* Muestra por pantalla el resultado de la fórmula para los valores x y n */

Implementar un pequeño programa que utilice esta clase.