Ciclo 2. Programación básica

Reto 1 – Requerimiento crédito beca universitaria

Descripción del problema

Una operadora de becas estudiantiles requiere adicionar una funcionalidad al sistema de información que tiene actualmente. Esta nueva funcionalidad implica que el sistema calcule los intereses generados sobre un monto determinado si se utiliza una proyección con interés compuesto, una proyección con interés simple y una proyección con la diferencia de estos. Los montos mencionados están relacionados con las becas universitarias (de pregrado y posgrado) que se van a otorgar por parte de la operadora a los potenciales beneficiarios. Las ecuaciones para determinar los intereses son las siguientes:

$$interesSimple = monto * \frac{interes}{100} * tiempo$$

Ecuación 1 - Cálculo del interés simple

$$interesCompuesto = monto * \left[\left(1 + \frac{interes}{100}\right)^{tiempo} - 1\right]$$

Ecuación 2 - Cálculo del interés compuesto

$$compararInversion = interesCompuesto - interesSimple$$

Ecuación 3 - Diferencia entre intereses

Para integrar esta nueva funcionalidad al sistema de información de la operadora, se solicita crear una clase llamada `BecaUniversitaria`, la cual debe tener el método `compararInversion()`. Dicho método, podrá recibir como parámetros las entradas `(int pTiempo, double pMonto, double pInteres)`, o no recibir ningún parámetro, y realizar la comparación a partir de los atributos de la clase. El método `compararInversion()` debe utilizar los métodos `calcularInteresSimple()` y `calcularInteresCompuesto()`, los cuales deben retornar el total de interés simple y compuesto a partir de las ecuaciones dadas, en formato `double`, sin recibir parámetros de entrada.

Tener en cuenta que si no se pasan argumentos al constructor de la clase `BecaUniversitaria`, sus atributos deben ser inicializados en cero.

La comparación entonces, implementada en el método `compararInversion()`, debe retornar una cadena (`String`) de la siguiente forma:

"La diferencia entre la proyección de interés compuesto e interés simple es: $" + diferencia

O bien, si solamente fue instanciada la clase, y los parámetros para la proyección no fueron enviados, `compararInversion()` retornaría una cadena de la siguiente forma:

"No se obtuvo diferencia entre las proyecciones, revisar los parámetros de entrada."

Especificación de entradas

Las entradas podrían llegar a través del constructor, o a través del método `compararInversion()`.

Nombre	**Tipo**	**Descripción**
pMonto	double	Valor en dólares de la beca estudiantil.
pInteres	double	Interés establecido por la entidad con la que se realiza el financiamiento.
pTiempo	int	Número de meses de financiamiento del crédito de estudios.

Salida:

Nombre	**Tipo**	**Valores Esperados**
Retorno del método `compararInversion()`	String	*"La diferencia entre la proyección de interés compuesto e interés simple es: $" + diferencia*
		"No se obtuvo diferencia entre las proyecciones, revisar los parámetros de entrada."

Casos de prueba (ejemplos):

Para todos los casos de prueba presentados a continuación, los valores de retorno de los intereses deben ser redondeados con `Math.round()`. Adicionalmente, los casos de prueba arrojan tres valores, porque en cada uno se realizan los siguientes tres llamados:

```
BecaUniversitaria becaUniversitaria = new BecaUniversitaria();
System.out.println(becaUniversitaria.calcularInteresSimple());
System.out.println(becaUniversitaria.calcularInteresCompuesto());
System.out.println(becaUniversitaria.compararInversion(60,13000,1.4));
```

Caso de prueba 1:

Entradas			Observación
pTiempo	**pMonto**	**pInteres (%)**	
60	13000	1.4	Valores enviados a través del método `compararInversion`

Salida
0.0 0.0 La diferencia entre la proyección de interés compuesto e interés simple es: $6018.0

Caso de prueba 2:

Entradas			Observación
pTiempo	**pMonto**	**pInteres (%)**	
48	10000	2.0	Valores entregados al constructor de la clase `BecaUniversitaria`

Salida
9600.0 15871.0 La diferencia entre la proyección de interés compuesto e interés simple es: $6271.0

Caso de prueba 3:

Entradas			Observación
pTiempo	**pMonto**	**pInteres (%)**	
Ninguno	Ninguno	Ninguno	Caso de prueba sin entradas

Salida
0.0 0.0 No se obtuvo diferencia entre las proyecciones, revisar los parámetros de entrada.

Esqueleto:

```
package co.edu.utp.misiontic2022.reto1.p45;

public class BecaUniversitaria {

    // ------------------------------------------------------------------
    // Atributos
    // ------------------------------------------------------------------

    // ...

    // ------------------------------------------------------------------
    // Constructores
    // ------------------------------------------------------------------

    // ...

    // ------------------------------------------------------------------
    // Métodos
    // ------------------------------------------------------------------

    public double calcularInteresSimple( ){
        // ...
    }

    public double calcularInteresCompuesto( ){
        // ...
    }

    public String compararInversion (int pTiempo, double pMonto, double
pInteres){
        // ...
    }

    public String compararInversion ( ){
        // ...
    }

    //Sección principal: Instanciación de clase y uso de métodos.
```

```
    //Importante: En la plataforma de iMaster se sube esta clase SIN MAIN.
    //Se indica esta sección para probar localmente antes de someter a revisión.
    public static void main(String[] args) {

        // Llamados para verificar con los casos de prueba el funcionamiento de
la clase

    }
}
```

Importante: conservar los nombres de la clases, los nombres de los métodos y las respectivas firmas (tipos de datos) para completar exitosamente el reto en iMaster.

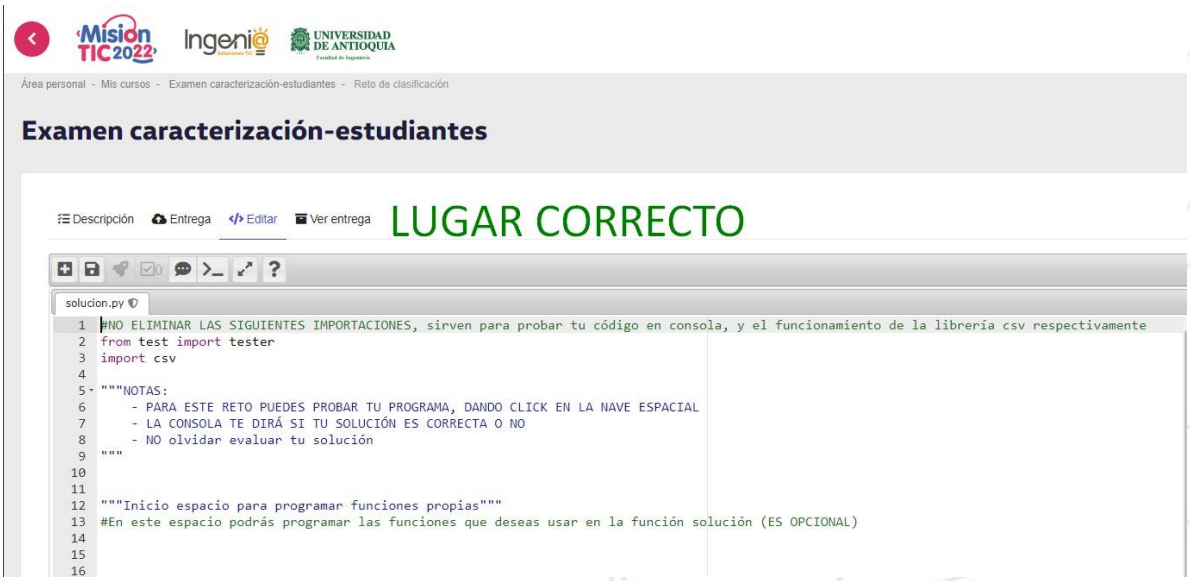
NOTA ACLARATORIA

Se recomienda desarrollar la prueba en un IDE.

Para esto se puede copiar y pegar el esquema de solución proporcionado en el VPL a su IDE preferido, recuerde que al final debe copiar y pegar el código del IDE a la herramienta VPL, pero **NO** deberá subir archivos, es decir:

Modo incorrecto:

Modo correcto:



TRIPULANTE
¡MUCHOS ÉXITOS EN EL DESARROLLO DEL RETO 1!