

Ciclo 2 Programación Básica

Reto 1 – Requerimiento Asesor Ventas Constructora



Descripción del problema:

Una constructora requiere adicionar una funcionalidad al sistema de información que tiene actualmente, orientado al acompañamiento que realizan los asesores comerciales a los potenciales compradores. El requerimiento consiste en que el sistema calcule los intereses generados si se utiliza una proyección con interés compuesto, una proyección con interés simple y la diferencia de estos, sobre los proyectos que tienen a la venta. Las ecuaciones para determinar el total de intereses son las siguientes:

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

Ecuación 1 - Cálculo del interés simple

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

Ecuación 2 - Cálculo del interés compuesto

$$\text{compararInversion} = \text{interesCompuesto} - \text{interesSimple}$$

Ecuación 3 - Diferencia entre intereses

Para integrar esta nueva funcionalidad con el sistema de información de la firma constructora, se solicita crear una clase llamada `VentaProyecto`, 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 `VentaProyecto`, 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 del precio establecido por la constructora para la venta que se está asesorando.
pInteres	double	Interés establecido por la entidad con la que se realiza el financiamiento.
pTiempo	int	Número de meses de financiamiento de la inversión.

Salida:

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

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:



```
VentaProyecto ventaConstruccion = new VentaProyecto();  
System.out.println(ventaConstruccion.calcularInteresSimple());  
System.out.println(ventaConstruccion.calcularInteresCompuesto());  
System.out.println(ventaConstruccion.compararInversion(36,200000,5.0));
```

Caso de Prueba 1:

Entradas			Observación
pTiempo	pMonto	pInteres (%)	
36	200000	5.0	Valores enviados a través del método <code>compararInversion</code>

Salida

0.0

0.0

La diferencia entre la proyección de interés compuesto e interés simple es: \$598363.0

Caso de Prueba 2:

Entradas			Observación
pTiempo	pMonto	pInteres (%)	
12	150000	2.0	Valores entregados al constructor de la clase <code>VentaProyecto</code>

Salida

36000.0

40236.0

La diferencia entre la proyección de interés compuesto e interés simple es: \$4236.0

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.p61;

public class VentaProyecto {

    // -----
    // 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 clase, los nombres de los métodos y las respectivas firmas (tipos de datos) para completar exitosamente el reto en iMaster.