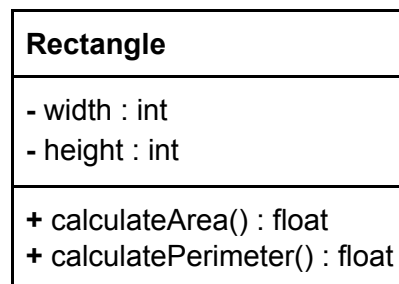


Guia de objetos (Modelado e Implementación)

Sobre esta guía

Lea atentamente cada uno de los enunciados y antes de llevar a código cada ejercicio realice en papel o con la ayuda de algún software, el correspondiente diagrama de UML, detallando los atributos, métodos y constructores que considere necesarios. Una vez terminado escriba el código y ejecute las pruebas que se detallan.

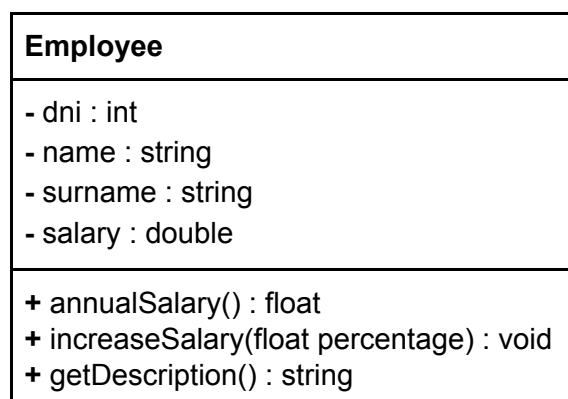
1. Usando como base el siguiente Diagrama



Modele el objeto Rectángulo con sus propiedades, ancho y alto. Ambas con valor 1,0 por defecto. Este objeto debe exponer los getters y setters necesarios. También debe contar con un método para calcular el área y perímetro del mismo. Ejecute las siguientes pruebas:

- Inicializar un objeto Rectángulo estableciendo ancho y alto.
- Imprimir por pantalla el alto y ancho.
- Imprimir por pantalla el área y perímetro.
- Modificar el alto y el ancho de la instancia.
- Imprimir por pantalla nuevamente el cálculo del área y perímetro.
- Inicializar un objeto Rectángulo con los valores por defecto y verificar.

2. Usando como base el siguiente Diagrama



Modele el objeto Empleado que posee las siguientes características, dni, nombre, apellido y salario. El mismo debe contar con la posibilidad de calcular el salario anual. A su vez se requiere otro método que permita aumentar el salario dependiendo del porcentaje que se le pase por parámetro. Considere crear un método que facilite imprimir por pantalla las características del objeto de la siguiente forma:

Empleado[dni=?, nombre=?, apellido=?, salario=?]

- a. Inicialice un empleado Carlos Gutiérrez, con dni 23456345 y salario inicial de 25000.
- b. Inicialice un empleado Ana Sánchez, con dni 34234123 y salario inicial de 27500.
- c. Imprima ambos objetos por pantalla.
- d. Aumente el salario del empleado Carlos en un 15% e imprima en pantalla el salario anual del mismo.

3. Modele el objeto Ítem de Venta con las siguientes propiedades, identificador, descripción, cantidad y precio unitario. Cree los métodos para calcular el precio total teniendo en cuenta el precio unitario y cantidad. Un método que permita imprimir por pantalla los atributos del objeto de la siguiente forma:

ItemVenta[id=?, descripción=?, cantidad=?, pUnitario=?, pTotal=?]

- a. Inicialice el objeto con los atributos necesarios
- b. Imprima por pantalla el objeto inicializado.

4. Modele el objeto que representa la cuenta de un banco, con identificador, nombre y balance. Considere los getters, setters y constructores necesarios. Tenga en cuenta los siguientes métodos.

- a. El método crédito que representa un depósito de dinero en la cuenta. Este método debe devolver el balance luego de la operación.
- b. El método débito que representa una sustracción de dinero de la cuenta. Este método debe devolver el balance luego de la operación. Si el dinero en la cuenta no es suficiente para cubrir la sustracción, se debe imprimir por pantalla un aviso.
- c. Un método que imprima por pantalla las características del objeto.

- Realice las siguientes operaciones:

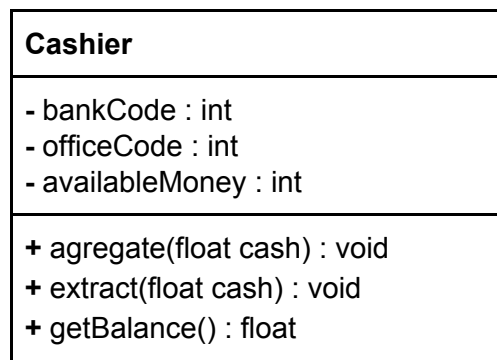
- a. Inicialice una cuenta con un monto inicial de 15000.
- b. Realice una operación de crédito de 2500.
- c. Realice una operación de compra de 1500.
- d. Realice una operación de compra de 30000.
- e. Imprima por pantalla los valores de la cuenta y verifique que el balance sea correcto

5. Modele el objeto Hora, teniendo en cuenta sus atributos, hora, minuto y segundo. Tenga en cuenta el rango de valores aceptados para cada uno de estos.

- a. Hora: 0 ... 23
 - b. Minuto: 0 ... 59
 - c. Segundo: 0 ... 59
- Considere el siguiente comportamiento:
 - a. Un método que imprima la hora bajo el siguiente formato hh:mm:ss usando zero a la izquierda ejemplo 13:04:22 .
 - b. Un método que avance en 1 segundo y devuelva la instancia del objeto.
 - c. Un método que retroceda en 1 segundo y devuelva la instancia del objeto.
 - d. Instanciar el objeto y probar los métodos creados.

6. Usando como base el siguiente Diagrama

★ (En conjunto con lo realizado en el ejercicio 4)



Se pretende desarrollar una aplicación que simule el funcionamiento de un cajero automático. Reutilizando lo hecho en el punto 4, modele el objeto Cajero, teniendo en cuenta sus atributos, getters y setters. Además del constructor y atributos que se estimen necesarios.

Por otro lado, existirá una clase con el método main encargada de la captura y presentación de datos y administración de la cuenta. Al iniciarse la aplicación se mostrará el siguiente menú.

1. Crear cuenta vacía
2. Crear cuenta con saldo inicial
3. Ingresar dinero en cuenta desde el Cajero.
4. Sacar dinero desde Cajero
5. Ver saldo
6. Salir

- La opción 1 crea un objeto Cuenta con saldo 0, la opción 2 solicita una cantidad y crea un objeto Cuenta con ese saldo inicial.
- En la opción 3 se solicita una cantidad y la ingresa en el objeto creado en las opciones 1 y 2 (debe haber pasado antes por estas opciones), mientras que la opción 4 se solicita una cantidad y la extrae del objeto creado en las opciones 1 o 2 (también debe haber pasado antes por estas opciones).
- Finalmente, la opción 5 muestra el saldo, mientras que la 6 finaliza el programa, lo que provocará que el objeto Cuenta se destruya y se pierda el saldo.
- El menú vuelve a presentarse en pantalla mientras no se elija la opción de salir