



Ejercicios

Clases en Java

PARADIGMAS DE PROGRAMACIÓN I

Carlos Rojas Sánchez

Licenciatura en Informática

Universidad del Mar

1. Ejercicio I
2. Ejercicio II
3. Ejercicio III
4. Ejercicio IV
5. Ejercicio V
6. Ejercicio VI
7. Ejercicio VII
8. Ejercicio VIII
9. Ejercicio IX
10. Ejercicio X

Ejercicio I

Se requiere un programa que modele el concepto de una persona. Una persona posee nombre, apellido, número de documento de identidad y año de nacimiento. La clase debe tener un constructor que inicialice los valores de sus respectivos atributos. La clase debe incluir los siguientes métodos:

- Definir un método que imprima en pantalla los valores de los atributos del objeto.
- En un método main se deben crear dos personas y mostrar los valores de sus atributos en pantalla.

Persona
nombre: String apellidos: String númeroDocumentoIdentidad: String añoNacimiento: int
«constructor»Persona(String nombre, String apellidos, String númeroDocumentoIdentidad, int añoNacimiento) imprimir()

```
Nombre = Pedro  
Apellidos = Pérez  
Número de documento de identidad = 1053121010  
Año de nacimiento = 1998
```

```
Nombre = Luis  
Apellidos = León  
Número de documento de identidad = 1053223344  
Año de nacimiento = 2001
```

Ejercicio II

Se requiere un programa que modele el concepto de un planeta del sistema solar. Un planeta tiene los siguientes atributos:

- Un nombre de tipo String con valor inicial de null.
- Cantidad de satélites de tipo int con valor inicial de cero.
- Masa en kilogramos de tipo double con valor inicial de cero.
- Volumen en kilómetros cúbicos de tipo double con valor inicial de cero.

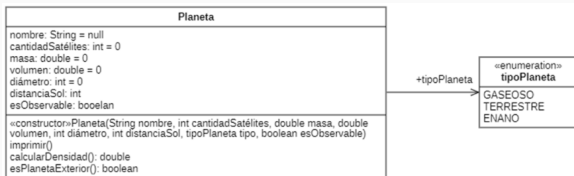
- Diámetro en kilómetros de tipo int con valor inicial de cero.
- Distancia media al Sol en millones de kilómetros, de tipo int con valor inicial de cero.
- Tipo de planeta de acuerdo con su tamaño, de tipo enumerado con los siguientes valores posibles: GASEOSO, TERRESTRE y ENANO.
- Observable a simple vista, de tipo booleano con valor inicial false.

La clase debe incluir los siguientes métodos:

- La clase debe tener un constructor que inicialice los valores de sus respectivos atributos.
- Definir un método que imprima en pantalla los valores de los atributos de un planeta.
- Calcular la densidad un planeta, como el cociente entre su masa y su volumen.

- Determinar si un planeta del sistema solar se considera exterior. Un planeta exterior está situado más allá del cinturón de asteroides. El cinturón de asteroides se encuentra entre 2.1 y 3.4 UA. Una unidad astronómica (UA) es la distancia entre la Tierra y el Sol = 149 597 870 Km.
- En un método main se deben crear dos planetas y mostrar los valores de sus atributos en pantalla. Además, se debe imprimir la densidad de cada planeta y si el planeta es un planeta exterior del sistema solar.

Clase Planeta



Clase Planeta

```
Nombre del planeta = Tierra
Cantidad de satélites = 1
Masa del planeta = 5.9736E24
Volumen del planeta = 1.08321E12
Diámetro del planeta = 12742
Distancia al sol = 150000000
Tipo de planeta = TERRESTRE
Es observable = true
Densidad del planeta = 5.514720137369484E12
Es planeta exterior = false
```

```
Nombre del planeta = Júpiter
Cantidad de satélites = 79
Masa del planeta = 1.899E27
Volumen del planeta = 1.4313E15
Diámetro del planeta = 139820
Distancia al sol = 750000000
Tipo de planeta = GASEOSO
Es observable = true
Densidad del planeta = 1.3267658771745964E12
Es planeta exterior = true
```

Ejercicio III

Se requiere un programa que modele el concepto de un automóvil.
Un automóvil tiene los siguientes atributos:

- Marca: el nombre del fabricante.
- Modelo: año de fabricación.
- Motor: volumen en litros del cilindraje del motor de un automóvil.
- Tipo de combustible: valor enumerado con los posibles valores de gasolina, bioetanol, diésel, biodiésel, gas natural.
- Tipo de automóvil: valor enumerado con los posibles valores de carro de ciudad, subcompacto, compacto, familiar, ejecutivo, SUV.

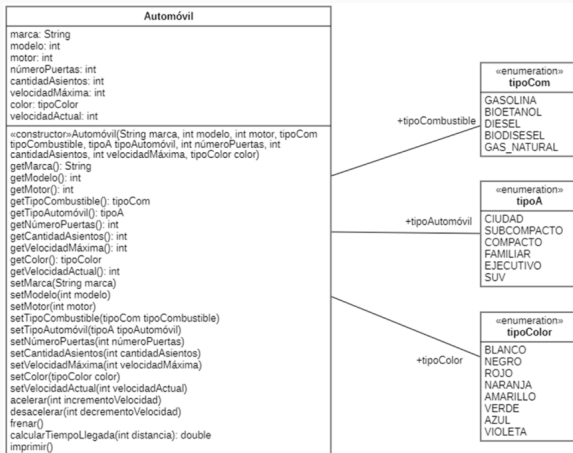
- Número de puertas: cantidad de puertas.
- Cantidad de asientos: número de asientos disponibles que tiene el vehículo.
- Velocidad máxima: velocidad máxima sostenida por el vehículo en km/h.
- Color: valor enumerado con los posibles valores de blanco, negro, rojo, naranja, amarillo, verde, azul, violeta.
- Velocidad actual: velocidad del vehículo en un momento dado.

La clase debe incluir los siguientes métodos:

- Un constructor para la clase Automóvil donde se le pasen como parámetros los valores de sus atributos.
- Métodos get y set para la clase Automóvil.
- Métodos para acelerar una cierta velocidad, desacelerar y frenar (colocar la velocidad actual en cero). Es importante tener en cuenta que no se debe acelerar más allá de la velocidad máxima permitida para el automóvil. De igual manera, tampoco es posible desacelerar a una velocidad negativa. Si se cumplen estos casos, se debe mostrar por pantalla los mensajes correspondientes.

- Un método para calcular el tiempo estimado de llegada, utilizando como parámetro la distancia a recorrer en kilómetros. El tiempo estimado se calcula como el cociente entre la distancia a recorrer y la velocidad actual.
- Un método para mostrar los valores de los atributos de un Automóvil en pantalla.
- Un método main donde se deben crear un automóvil, colocar su velocidad actual en 100 km/h, aumentar su velocidad en 20 km/h, luego decrementar su velocidad en 50 km/h, y después frenar. Con cada cambio de velocidad, se debe mostrar en pantalla la velocidad actual.

Clase Automóvil



```
Marca = Ford
Modelo = 2018
Motor = 3
Tipo de combustible = DIESEL
Tipo de automóvil = EJECUTIVO
Número de puertas = 5
Cantidad de asientos = 6
Velocidad máxima = 250
Color = NEGRO
Velocidad actual = 100
Velocidad actual = 120
Velocidad actual = 70
Velocidad actual = 0
No se puede decrementar a una velocidad negativa.
```

Ejercicio IV

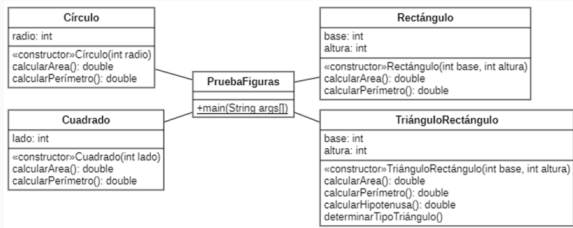
Se requiere un programa que modele varias figuras geométricas: el círculo, el rectángulo, el cuadrado y el triángulo rectángulo.

- El círculo tiene como atributo su radio en centímetros.
- El rectángulo, su base y altura en centímetros.
- El cuadrado, la longitud de sus lados en centímetros.
- El triángulo, su base y altura en centímetros.

Se requieren métodos para determinar el área y el perímetro de cada figura geométrica. Además, para el triángulo rectángulo se requiere:

- Un método que calcule la hipotenusa del rectángulo.
- Un método para determinar qué tipo de triángulo es:
Equilátero: todos sus lados son iguales.
Isósceles: tiene dos lados iguales.
Escaleno: todos sus lados son diferentes.

Clases sobre Figuras Geométricas



Clases sobre Figuras Geométricas

El área del círculo es = 12.566370614359172

El perímetro del círculo es = 12.566370614359172

El área del rectángulo es = 2.0

El perímetro del rectángulo es = 6.0

El área del cuadrado es = 9.0

El perímetro del cuadrado es = 12.0

El área del triángulo es = 7.0

El perímetro del triángulo es = 13.8309518948453

Es un triángulo escaleno

Ejercicio V

Clase Cuenta Bancaria

Se requiere un programa que modele una cuenta bancaria que posee los siguientes atributos:

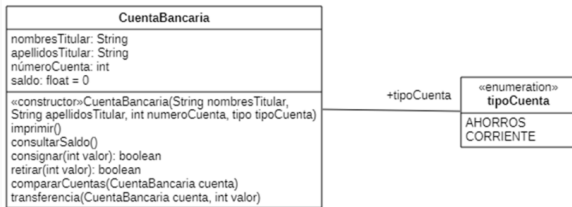
- Nombres del titular.
- Apellidos del titular.
- Número de la cuenta bancaria.
- Tipo de cuenta: puede ser una cuenta de ahorros o una cuenta corriente.
- Saldo de la cuenta.

Se debe definir un constructor que inicialice los atributos de la clase. Cuando se crea una cuenta bancaria, su saldo inicial tiene un valor de cero.

En una determinada cuenta bancaria se puede:

- Imprimir por pantalla los valores de los atributos de una cuenta bancaria.
- Consultar el saldo de una cuenta bancaria.
- Consignar un determinado valor en la cuenta bancaria, actualizando el saldo correspondiente.
- Retirar un determinado valor de la cuenta bancaria, actualizando el saldo correspondiente. Es necesario tener en cuenta que no se puede realizar el retiro si el valor solicitado supera el saldo actual de la cuenta.

Clase Cuenta Bancaria



Clase Cuenta Bancaria

```
Nombres del titular = Pedro  
Apellidos del titular = Pérez  
Número de cuenta = 123456789  
Tipo de cuenta = AHORROS  
Saldo = 0.0
```

```
Se ha consignado $200000 en la cuenta. El nuevo saldo es $200000.0
```

```
Se ha consignado $300000 en la cuenta. El nuevo saldo es $500000.0
```

```
Se ha retirado $400000 en la cuenta. El nuevo saldo es $100000.0
```

Ejercicio VI

Clase Película

Realizar un programa en Java que defina una clase Película con los siguientes atributos privados:

- Nombre: es el nombre de la película y es de tipo String.
- Director: representa el nombre del director de la película y es de tipo String.
- Género: es el género de la película, un tipo enumerado con los siguientes valores: ACCIÓN, COMEDIA, DRAMA y SUSPENSO.
- Duración: duración de la película en minutos, esta es de tipo int.
- Año: representa el año de realización de la película.
- Calificación: es la valoración de la película por parte de sus usuarios y es de tipo double.

Se debe definir un constructor público para la clase, que recibe como parámetros los valores de todos sus atributos. También se deben definir los siguientes métodos:

- Métodos get y set para cada atributo y con los derechos de acceso privados para los set y públicos para los get.
- Un método imprimir público que muestre en pantalla los valores de los atributos.
- Un método privado boolean `esPeliculaEpica()`, el cual devuelve un valor verdadero si la duración de la película es mayor o igual a tres horas, en caso contrario devuelve falso.

- Un método privado `String calcularValoración()`, el cual según la table siguiente devuelve una valoración subjetiva.

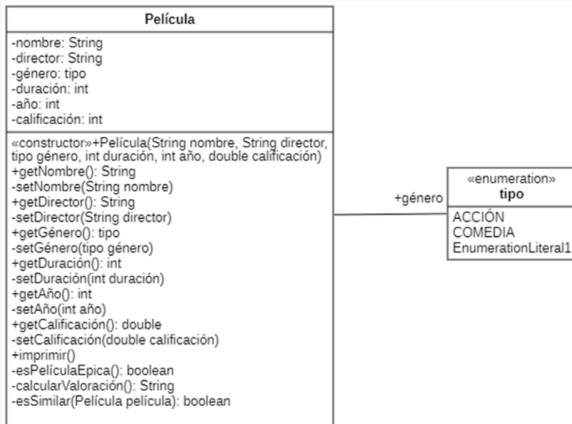
Calificación	Valoración	Calificación	Valoración
[0, 2]	Muy mala	(7, 8]	Buena
(2, 5]	Mala	(8, 10]	Excelente
(5, 7]	Regular		

Clase Película

- El método privado boolean `esSimilar()` compara la película actual con otra película que se le pasa como parámetro. Si ambas películas son del mismo género y tienen la misma valoración, devuelve verdadero; en caso contrario, devuelve falso.
- Un método `main` que construya dos películas; determinar si son películas épicas; calcular su respectiva valoración y determinar si son similares. Las dos películas están en la tabla siguiente.
- Mostrar los resultados de la ejecución del método `main`.

Objeto	Nombre	Director	Género	Duración	Año	Calif.
<code>película1</code>	Gandhi	Richard Attenborough	DRAMA	191	1982	8.1
<code>película2</code>	Thor	Kenneth Branagh	ACCIÓN	115	2011	7.0

Clase Película



Nombre: Gandhi
Director: Richard Attenborough
Género: DRAMA
Duración: 191
Año: 1982
Calificación: 8.3

Nombre: Thor
Director: Kenneth Branagh
Género: ACCIÓN
Duración: 115
Año: 2011
Calificación: 7.0

La película Gandhi es épica: true

La película Thor es épica: false

La película Gandhi y la película Thor son similares = false

Ejercicio VII

Se tiene un código que modela una clase Avión que posee dos atributos: nombre del fabricante del avión (tipo String) y número de motores del avión (tipo int). La clase tiene un constructor y métodos get y set para cada atributo. Además, tiene los siguientes métodos adicionales:

- Un método denominado imprimirFabricante(), que muestra en pantalla el nombre del fabricante de un avión.
- Un método denominado cambiarFabricante(Avión a), que recibe como parámetro un objeto de tipo Avión y le cambia el valor de su atributo fabricante a un valor predefinido "Loockhead"

En el método main se crean dos aviones: a1 (fabricante “Airbus” con cuatro motores) y a2 (fabricante “Lookheed” con cuatro motores). Luego, los datos de cada avión se imprimen por pantalla. Después, se realizan llamadas a los métodos setFabricante y cambiarFabricante, los cuales cambiarán los valores de sus atributos. ¿Cuál es el resultado final de la ejecución del método main? Determinar lo que se imprime en pantalla.

Avión
-fabricante: String -numeroMotores: int
«constructor»-Avión(String fabricante, int numeroMotores) +imprimirFabricante() +getFabricante(): String -setFabricante(fabricante: String) -cambiarFabricante(a: Avión) +getNumeroMotores(): int -setNumeroMotores(numeroMotores: int)

El fabricante del avión es: Airbus
El fabricante del avión es: Airbus
El fabricante del avión es: Douglas
El fabricante del avión es: Douglas
El fabricante del avión es: Lockheed

Ejercicio VIII

Realizar un programa en Java que permita realizar las siguientes conversiones de unidades de longitud:

- Metros a centímetros.
- Metros a milímetros.
- Metros a pulgadas.
- Metros a pies.
- Metros a yardas.

Instrucción	Descripción	Formato
<i>final</i>	Palabra clave para definir que una vez que a una variable se le ha asignado un valor, siempre contendrá el mismo valor (permanece constante). Se recomienda que los nombres de las constantes se escriban en mayúscula.	<i>final</i> tipo VARIABLE = valor;

Clase Conversor Metros

ConversorMetros
<ul style="list-style-type: none">-metros: double-FACTOR_METROS_CM: int = 100 {readOnly}-FACTOR_METROS_MILIM: int = 1000 {readOnly}-FACTOR_METROS_PULGADAS: double = 39.37 {readOnly}-FACTOR_METROS_PIES: double = 3.28 {readOnly}-FACTOR_METROS_YARDAS: double = 1.09361 {readOnly}
<ul style="list-style-type: none">«constructor»+ConversorMetros(double metros)+convertirMetrosToCentimetros(): double+convertirMetrosToMilímetros(): double+convertirMetrosToPulgadas(): double+convertirMetrosToPies(): double+convertirMetrosToYardas(): double

```
Metros = 3.5  
Metros a centímetros = 350.0  
Metros a milímetros = 3500.0  
Metros a pulgadas = 137.795  
Metros a pies = 11.479999999999999  
Metros a yardas = 3.827635
```

Ejercicio IX

Realizar un programa en Java que permita calcular el pedido que realiza un cliente en un restaurante. Los pedidos de un restaurante están conformados por las siguientes partes:

- Un primer plato.
- Un segundo plato.
- Una bebida.
- Un postre.

Cada uno de dichas partes tiene un nombre y un valor. Se requiere definir métodos sobrecargados para calcular el valor del pedido dependiendo si el cliente solicita:

- Un primer plato y una bebida.
- Un primer plato, un segundo plato y una bebida.
- Un primer plato, un segundo plato, una bebida y un postre.

Implementar un método main que utilice los tres métodos sobrecargados en tres diferentes pedidos.

Pedido
<pre>+calcularPedido(String primerPlato, double costoPrimerPlato, String bebida, double costoBebida) +calcularPedido(String primerPlato, double costoPrimerPlato, String segundoPlato, double costoSegundoPlato, String bebida, double costoBebida) +calcularPedido(String primerPlato, double costoPrimerPlato, String segundoPlato, double costoSegundoPlato, String postre, double costoPostre, String bebida, double costoBebida)</pre>

El costo de Sancocho y Gaseosa es = \$7000.0

El costo de Crema de verduras + Churrasco + Gaseosa es = \$13000.0

El costo de Crema de espinacas + Salmón + Gaseosa + Tiramisú es = \$22000.0

Ejercicio X

Realizar un programa en Java que permita modelar un artículo científico. Los artículos científicos contienen los siguientes metadatos: nombre del artículo, autor, palabras claves, nombre de la publicación, año y resumen. Se deben definir tres constructores sobrecargados:

- El primero inicializa un artículo científico con solo su título y autor.
- El segundo constructor, un artículo científico con su nombre, autor, palabras claves, nombre de la publicación y año. Debe invocar al primer constructor.

- El tercer constructor, un artículo científico con su nombre, autor, palabras claves, nombre de la publicación, año y resumen. Debe invocar al segundo constructor.

Se requiere un método que imprima los atributos de un artículo en pantalla. Realizar un método main que utilice el tercer constructor para instanciar un artículo científico e imprima los valores de sus atributos en pantalla.

ArtículoCientífico
<ul style="list-style-type: none">-título: String-autor: String-palabrasClaves: String-publicación: String-año: int-resumen: String
<ul style="list-style-type: none">«constructor»+ArtículoCientífico(String título, String autor)«constructor»+ArtículoCientífico(String título, String autor, String[] palabrasClaves, String publicación, int año)«constructor»+ArtículoCientífico(String título, String autor, String[] palabrasClaves, String publicación, int año, String resumen)+imprimir()

Título del artículo = La teoría especial de la relatividad

Autor del artículo = Albert Einstein

Palabras clave =

Física

Espacio

Tiempo

Publicación = Anales de Física

Año = 1913

Resumen = Las leyes de la física son las mismas en todos los sistemas de referencia inerciales.