



Trabajo Práctico I: POO Primeros Conceptos

Presentación de la materia. POO. Primeros conceptos

Programación Orientada a Objetos - Motivación Una nueva forma de pensar. Paradigmas. El Objeto. Objetos y Clases. Componentes de un objeto: Identidad, Estado, Comportamiento. Clases: Constructores
Instanciación de clases. Manipulación de objetos. El método main(). Abstracción de la realidad al modelar con objetos UML. Diagrama de Clases. Ejemplo de un programa en Java: Implementación TDA Pila en objetos

Objetivos de la práctica

- **Parte1:** Conceptos básicos de programación en Java: estructura de un programa, variables, tipos y conversiones, entrada y salida de datos por pantalla, estructuras de control.
- **Parte 2:** Pensando en objetos: identificación, definición de estado y comportamiento

PARTE 1

Ejercicio: 1

La clase **CuentaBancaria**, tiene los siguientes atributos y propiedades:

CuentaBancaria
- double saldo
- String titular
+ void depositar(double cantidad)
+ void extraer(double cantidad)
+ double getSaldo()
+ String getTitular()
+ void setTitular(String titular)

Suponiendo que “unacuenta” es una instancia de la clase CuentaBancaria, escribe **intuitivamente en lápiz y papel**, como harías para:

- depositar \$1250 en unacuenta
- extraer \$450 en unacuenta
- averiguar el saldo de unacuenta
- colocar como titular de unacuenta a “Juan Perez”
- recueprar en el String llamado “aux” al titular de unacuenta



Ejercicio: 2

Disponemos de las clases imagen, persona y lugar que tienen los siguientes atributos y métodos.

Lugar
- String ciudad
- double latitud
- double longitud
- String pais
+ String getCiudad()
+ double getLatitud()
+ double getLongitud()
+ String getPais()
+ void setCiudad(String ciudad)
+ void setLatitud(double latitud)
+ void setLongitud(double longitud)
+ void setPais(String pais)

Persona
- int edad
- Lugar lugarNacimiento
- String nombre
+ int getEdad()
+ Lugar getLugarNacimiento()
+ String getNombre()
+ void setEdad(int edad)
+ void setLugarNacimiento(Lugar lugarNacimiento)
+ void setNombre(String nombre)

Imagen
- int cantidaPersonas
- String comentarios
- Date fecha
- Lugar lugar
- Persona[] personasenfoto
- Persona propietario
+ void agregarComentario(String comentario)
+ void etiquetarPersona(Persona participante)
+ int getCantidaPersonas()
+ String getComentarios()
+ String getFecha()
+ Lugar getLugar()
+ Persona[] getPersonasenfoto()
+ Persona getPropietario()
+ void setFecha(Date fecha)
+ void setLugar(Lugar lugar)
+ void setPropietario(Persona propietario)

Las instancias p1, p2, p3 son de tipo Persona, l1, l2 y l3 son de tipo Lugar, i1, i2, e i3 son de tipo Imagen.

Escribe **intuitivamente en papel y lápiz** los mensajes y/o el código correspondientes para:

- Etiquetar a las personas p1 y p2 en la imagen i1.
- Agregar el comentario “que hermosa foto” en la imagen i3.
- Guardar en el entero min, la edad de la menor de las 3 personas.
- Averiguar si el propietario de la imagen i2 es la persona p3.
- Recuperar en un String llamado “cadena” los mensajes de la imagen i3.
- Recuperar en un String llamado “paisauxiliar” el país correspondiente a la imagen i2.
- Indicar que la imagen i1 ha sido tomada en el lugar l2.

Ejercicio: 3

El objetivo de este ejercicio es que te familiarices con el entorno de trabajo. Crea y transcribe las clases Empleado y Categoria. Trata de interpretar, desde el código, el funcionamiento de las clases.



Universidad Nacional de Mar del Plata
Facultad de Ingeniería - Ingeniería en Informática
**Programación III – TP I: POO Primeros
Conceptos**



```
public class Empleado
{
    private String nombre;
    private int antigüedad, hstrababajadas;
    private Categoria cat;

    public String getNombre()
    {
        return nombre;
    }

    public void setNombre(String nombre)
    {
        this.nombre = nombre;
    }

    public int getAntigüedad()
    {
        return antigüedad;
    }

    public void setAntigüedad(int antigüedad)
    {
        this.antigüedad = antigüedad;
    }

    public int getHstrababajadas()
    {
        return hstrababajadas;
    }

    String detalle()
    {
        String retorno =
            "Nombre: " + this.nombre + "\nAntigüedad: " + this.antigüedad + " años\nHoras trabajadas: " +
            this.hstrababajadas + "\nTipo de categoria: " + this.cat.getNombrecategoria() + "\nSueldo por hora: " +
            this.cat.getSueldoporhora() + "\n Sueldo a cobrar: " + this.getsueldo() + "\n";

        return retorno;
    }
}

public class Categoria
{
    private String nombrecategoria;
    private double sueldoporhora;

    public String getNombrecategoria()
    {
        return nombrecategoria;
    }

    public double getSueldoporhora()
    {
        return sueldoporhora;
    }

    public Categoria(String nombrecategoria, double sueldoporhora)
    {
        this.nombrecategoria = nombrecategoria;
        this.sueldoporhora = sueldoporhora;
    }

    public void setHstrababajadas(int hstrababajadas)
    {
        this.hstrababajadas = hstrababajadas;
    }

    public Categoria getCat()
    {
        return cat;
    }

    public void setCat(Categoria cat)
    {
        this.cat = cat;
    }

    public double getsueldo()
    {
        double aux;
        aux = this.cat.getSueldoporhora() * this.hstrababajadas;
        if (this.antigüedad > 5)
        {
            if (this.antigüedad < 10)
                aux *= 1.25;
            else
                aux *= 1.35;
        }
        return aux;
    }
}
```



Inciso a)

Realiza un diagrama de clases

Inciso b)

Crea la clase Prueba, y en el método main instancia 6 empleados y 3 categorías de manera que tengan las siguientes características:

Empleados:

- “Juan Perez” - Categoría Principiante – horas trabajadas 100 – antigüedad 4 años
- “Roberto Gonzalez” - Categoría Principiante – horas trabajadas 120 – antigüedad 8 años
- “Sandra Lopez” - Categoría Principiante – horas trabajadas 120 – antigüedad 14 años
- “German Gutierrez” - Categoría Operario – horas trabajadas 110 – antigüedad 16 años
- “Vicente Hernandez” - Categoría Experto – horas trabajadas 100 – antigüedad 9 años
- “Carolina Gomez” - Categoría Experto – horas trabajadas 115 – antigüedad 20 años

Categorías:

- Principiante - sueldo por hora \$ 80
- Operario - sueldo por hora \$ 100
- Experto sueldo por hora \$ 130

Muestra por pantalla los sueldos de cada empleado.



Ejercicio: 4

Escribe la clase Rectangulo

Rectangulo
- int x1
- int x2
- int y1
- int y2
+ Rectangulo()
+ Rectangulo(int ancho, int alto)
+ Rectangulo(int x1, int y1, int x2, int y2)
+ int ancho()
+ int area()
+ int getX1()
+ int getX2()
+ int getY1()
+ int getY2()
+ int perimetro()
+ void setX1(int x1)
+ void setX2(int x2)
+ void setY1(int y1)
+ void setY2(int y2)

El constructor sin parámetros construirá un rectángulo con las características:

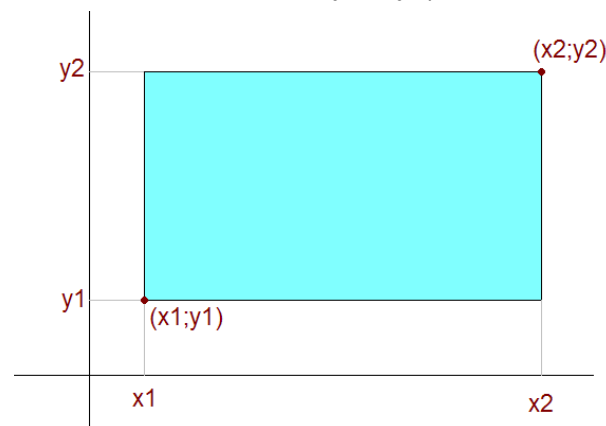
$x1 = -1$; $y1 = -1$; $x2 = 1$; $y2 = 1$.

El constructor con dos parámetros enteros construirá un rectángulo con la esquina inferior izquierda en (0,0) y de ancho y alto correspondiente a los parámetros.

El constructor con cuatro parámetros asignará a cada atributo su parámetro correspondiente.

(Los atributos $x1$, $y1$, $x2$, $y2$ hacen referencia a las coordenadas de dos vértices opuestos de un rectángulo.

Se asume que $x1 < x2$; $y1 < y2$)





Ejercicio: 5

Escribe la clase **Guerrero**, que representa un personaje en un juego de estrategia.

Guerrero
- String nombre - double vitalidad - double armadura - double x - double y
<<todos los getters y setters>> + void mover(double inc_x, double inc_y) + void recibeDano double cantidad)

void mover(double inc_x, double inc_y)	Incrementa las coordenadas "x" e "y" del guerrero de acuerdo a los parámetros inc_x e inc_y respectivamente.
void recibeDano double cantidad)	Al recibir daño el la armadura del guerrero lo absorbe, una vez que la armadura llega a cero el daño restante lo se resta a la vitalidad. Por ejemplo si la armadura y la vitalidad valen 1000 cada una y se recibe un daño de 800, entonces quedara la armadura en 200 y la vitalidad en 1000. Si lego se recibe un daño de 300, la armadura quedará en cero y la vitalidad en 900.

Ejercicio: 6

Escriba una clase de nombre Socio, que contenga los métodos y atributos necesarios para calcular la cuota que se debe abonar en un club de golf. La cuota es de 500\$. Tendrán un 50% de descuento las personas mayores de 65 años y un 25% los menores de 18 años que representen al club en torneos oficiales.

Condiciones: definir las constantes que considere y el siguiente método para realizar el cálculo
public double calcularCuota(...)

Escribe también una clase Prueba con un método main en el cual se setearán los datos de varios socios con diferentes situaciones y se mostrarán los cálculos por pantalla.

PARTE 2



Para cada una de los siguientes problemas plantee una solución identificando objetos que intervienen, su estado y su comportamiento. No es necesario escribir código Java.

Analizar y especificar un problema simple construyendo un modelo del mundo del problema e identificando los requerimientos funcionales.

Proponer una solución al problema:

1. Refinar el modelo del mundo del problema hasta poderlo expresar en términos de clases, tipos básicos de datos y agrupamientos lineales simples.
2. Asignar responsabilidades a cada clase, expresándolas en términos de métodos y parámetros y, además, explicar las colaboraciones entre las clases.

Ejercicio: 7

Plantee la solución identificando objetos que intervienen, su estado y su comportamiento para las siguientes estructuras de datos: Lista, Fila

Ejercicio: 8

Un país tiene que controlar el gasto público de las ciudades con más de 100.000 habitantes. Para ello, tiene información del monto recaudado por cada ciudad a través de cinco diferentes tipos de impuestos (denominados aquí, imp1, imp2, imp3, imp4 e imp5) e información acerca de gastos realizados en mantenimiento de la ciudad. Dicho país necesita un sistema que le informe cuáles son las ciudades que gastan mas de lo que recaudan (condición de déficit), y las provincias que tienen más de la mitad de las ciudades en condición de déficit.

Ejercicio: 9

Una agenda personal que permita registrar reuniones en las que el usuario va a participar. En tal agenda debe registrarse dónde ocurrirá la reunión, quiénes van a participar de ella, el tema que van a tratar y la duración de la misma. Piense la solución contemplando o no el solapamiento de reuniones. En caso de que no se permitan solapamientos, a quien corresponde el comportamiento de detectarlos y corregirlos.



Ejercicio: 10 **Variables e Instancias**

Analiza los siguientes códigos e indica, cuantas variables y cuantas instancias hay de la clase Automovil, y realiza un pequeño esquema de las áreas de memoria Stack y Heap.
(Supongamos que el Garbage Colector no ha pasado aún, indica cuales instancias no tienen referencias en caso de que las haya)

A) Automovil a1; Automovil a2; Automovil a3; Automovil a4; a1=new Automovil(); a1.setPatente("ABC-123"); a2=new Automovil(); a2.setPatente("ZZZ-987"); a3=a1; a4=a2;	B) Automovil a1; Automovil a2; Automovil a3; Automovil a4; a1=new Automovil(); a1.setPatente("ABC-123"); a2=a1; a3=null; a4=a3;
C) Automovil a1=null; Automovil a2=null; Automovil a3=null; Automovil a4=null; a1=new Automovil(); a1.setPatente("ABC-123"); a2=new Automovil(); a2.setPatente("ZZZ-987"); a1=a3; a2=a1;	D) Automovil a1=null; Automovil a2=null; Automovil a3=null; Automovil a4=null; a1=new Automovil(); a1.setPatente("ABC-123"); a2=new Automovil(); a2.setPatente("ZZZ-987"); a3=a1; a4=a3; a1=null a3.setPatente("DDD-456")
E) Automovil a1=null; Automovil a2=null; a1=new Automovil(); a1.setPatente("ABC-123"); a2=new Automovil(); a2.setPatente("ZZZ-987");	F) Automovil a1=null; Automovil a2=null; a1=new Automovil(); a1.setPatente("ABC-123"); a2=new Automovil(); a2.setPatente("ZZZ-987");



```
a1=new Automovil();  
a1.setPatente("FFF-444");  
a2=new Automovil();  
a2.setPatente("HHH-555");
```

```
a1=new Automovil();  
a1.setPatente("FFF-444");  
a2=new Automovil();  
a2.setPatente("HHH-555");  
a1=null;  
a2=null;
```