

## **GUÍA DE LABORATORIO N° 05**

**Escuela Profesional:** Ingeniería de Sistemas.

**Ciclo y Turno:** IV – Mañana

**Docente:** Ing. Ivan Petrlik / Ing. Gustavo Coronel

**Asignatura:** Metodología de la Programación.

**Semestre Académico:** 2017 - II

**Fecha:** 25 al 29 de Setiembre del 2017

### **CLASES, OBJETOS Y CONTROLES**

#### **INTRODUCCION**

Una de las grandes utilidades de la serialización de objetos con el manejo de arraylist y manejo de archivos binarios es su capacidad de guardar cualquier tipo de datos y su manipulación está sujeta al uso de una clase principal con el manejo de un arreglo de objetos para manipular la información a través de posiciones.

Se implementaran aplicaciones en entorno visual con JFrame, utilizando como herramienta de desarrollo NetBeans 6.8 y además para demostrar los el uso de archivos binarios y arreglos de objetos.

#### **I. OBJETIVOS**

1. Desarrollo de ejercicios de Manejo de Cadenas usando Swing WT.
2. Codificar, compilar y ejecutar los ejercicios de aplicación.
3. Conocer la estructura de un programa visual en Java.
4. Se espera que el estudiante asocie los conocimientos nuevos con la “nueva plantilla” que se propone.

#### **II. EQUIPOS Y MATERIALES**

- ③ Computadora personal.
- ③ Programa NetBeans IDE 6.7.1 o 6.8 correctamente instalado.
- ③ Notas de los ejercicios resueltos en la clase.

#### **III. METODOLOGIA Y ACTIVIDADES**

- Codificar los ejercicios desarrollados en el aula.
- Presentar avances y ejecución de cada uno de los ejercicios al docente o jefe de práctica encargado para la calificación correspondiente.
- Guardar la carpeta de sus archivos a sus memorias.
- Apagar el computador y dejarla en buen estado al retirarse del laboratorio dejar todo en orden.

#### **IV. OBSERVACION**

- 🔗 El estudiante deberá crear una carpeta de trabajo con el nombre “Laboratorio\_12”, a donde deberá direccional su proyecto a crear.
- 🔗 Se recomienda que el estudiante haya planteado los ejercicios de la Guía de ejercicios 12 para que pueda comprender las soluciones que se proponen.

#### **V. NUEVAS LIBRERIAS**

Ver diapositiva GUI en Java-AWT & SWING.pdf

## VI. EJERCICIOS.

### Proyecto 01: Distancia entre dos puntos

#### Enunciado

La Facultad de Matemáticas de la UNI está desarrollando una serie de aplicativos para que refuercen el aprendizaje de las matemáticas a los alumnos del nivel primario.

En esta oportunidad necesita de un programa que dado dos puntos del plano cartesiano, determine a que cuadrante pertenece cada punto, y cuál es la distancia entre ambos puntos.

#### Análisis

Se puede identificar dos tipos de objetos:

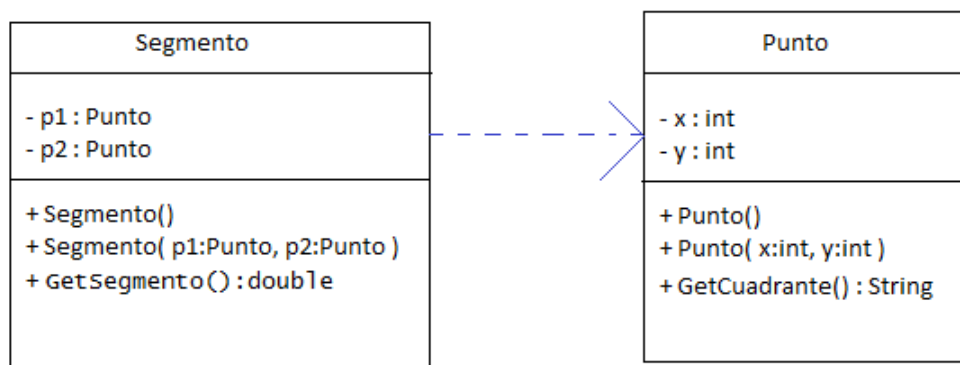
1. **Punto:** Que representa cualquier punto del plano cartesiano.
  - Cada punto tiene una abscisa (x) y una ordenada (y).
  - Cada punto también conoce a que cuadrante pertenece, por lo tanto debería tener un método que retorne el cuadrante al que pertenece.
2. **Segmento:** Que representa el fragmento de recta comprendido entre dos puntos.
  - Cada segmento esta con constituido por dos puntos (p1 y p2).
  - Cada segmento conoce cuál es la distancia entre sus dos puntos, por lo tanto debería tener un método que retorne el valor la distancia entre los dos puntos.

Podemos afirmar que un objeto de tipo Segmento está constituido por dos objetos de tipo Punto.

#### Diagrama de Clases

A continuación tenemos el diagrama de clases.

##### DIAGRAMA DE CLASES



Para acceder a las variables privadas de cada clase se deben implementar sus propiedades respectivas.

## Implementación

1. Crear un nuevo proyecto de nombre **POO\_Lab05\_Proj01**.
2. Crear una carpeta de nombre modelo.
3. Crear la clase **Punto** en la carpeta modelo.

```
class Punto {
    private int x;
    private int y;

    // Implementar los métodos getters y setters

    public String getCuadrante() {
        String cuadrante = "EJE";

        // Aquí debe escribir el código para encontrar
        // el cuadrante al que pertenece el punto.

        return cuadrante;
    }
}
```

4. Crear la clase **Segmento** en la carpeta modelo.

```
class Segmento {
    private Punto p1;
    private Punto p2;

    public Segmento() {
        p1 = null;
        p2 = null;
    }

    public Segmento(Punto p1, Punto p2) {
        this.p1 = p1;
        this.p2 = p2;
    }

    // Implementar los métodos getters y setters

    public double getDistancia() {
        // variable para el valor de retorno
        double d = 0.0;

        // Aquí debe escribir el código para
        // encontrar la distancia entre los dos puntos.

        // Retorno de resultado
        return d;
    }
}
```

5. Construya la siguiente interfaz de usuario.

SEGMENTO

Punto 1

X 0

Y 0

Punto 2

X 3

Y 4

Procesar

Resultado

Cuadrante del punto 1: EJE

Cuadrante del punto 2: I

Distancia entre los dos puntos: 5

## Proyecto 02: Rentabilidad de Curso

La institución educativa EduTech necesita un software para estimar la rentabilidad de un curso. Los datos del curso son:

- Horas que dura el curso
- El precio de venta del curso por alumno
- La cantidad de vacantes
- El pago por hora al profesor

Además se considera los siguientes gastos:

- Gastos administrativos, el 3% del ingreso bruto
- Gastos de publicidad, el 5% del ingreso bruto
- Gastos logísticos, el 4% del ingreso bruto

El software debe permitir obtener un reporte de los ingresos, los gastos y la rentabilidad que se espera obtener.

## Proyecto 03: Calcular el Área de un Rombo

La Universidad Nacional de Ingeniería está desarrollando software para facilitar el aprendizaje las matemáticas de los alumnos de primaria.

En esta oportunidad requiere un software que permita calcular el área de un rombo, la formula es la siguiente:

$$S = \frac{D * d}{2}$$

Donde:

- D : Diagonal Mayor
- d : Diagonal Menor
- S : Área del Rombo