

Pauta Certamen 2, Programación II

Prof. Rodrigo Olivares

Instrucciones:

- El puntaje máximo del certamen es 100 %, siendo el 60 % el mínimo requerido para aprobar.
- Responda las preguntas en un único archivo, agregando el número de la pregunta, su nombre y RUT. Si no responde alguna pregunta, debe indicar en el mismo archivo que **no responde**. El nombre del archivo debe tener la forma <<apellido_nombre.ext>> y debe ser subido al aula virtual.
- El certamen es **individual**. Cualquier intento de copia, será sancionado con nota **1,0**.

1. 30pts (6pts c/u). **Comente** las siguientes declaraciones.

- (a) La herencia es un mecanismo que permite construir clases tipo Padre-Hija(s). Esta construcción es llevada a cabo sin la necesidad de conocer el comportamiento de la clase Padre.
R: Es **cierto** que la herencia es un mecanismo que permite construir clases tipo Padre-Hija(s), sin embargo esta construcción debe ser llevada a cabo conociendo el comportamiento de la clase Padre, pues es de vital importancia saber **qué** hace la clase Padre sin importar el **cómo**.
- (b) En la implementación de una clase abstracta, la sub-clase que la extiende debe manipular todos los métodos, sean o no abstractos.
R: La sub-clase que extiende a una clase abstracta no está obligada a implementar los métodos abstractos, sin embargo se deben re-definir nuevamente como **abstract**. Para los métodos no **abstract**, no existe obligación de re-definirlos.
- (c) La el uso de interfaces permite simular la herencia múltiple.
R: Ciertamente la herencia múltiple no existe en Java, sin embargo ésta puede ser simulada a partir de la implementación de diversas interfaces.
- (d) Una lista es un tipo de dato abstracto genérico, ideal para gestionar colecciones de datos.
R: Una lista es una interfaz que puede ser implementada por diversas clases. Es cierto que una lista puede ser un TDA y el comportamiento estará definido por la implementación (clase que la implementa).
- (e) La entidad es un componente fundamental en la programación orientada a objeto.
R: Una entidad es un tipo de dato abstracto (TDA) definida por el programador (clase) y cumple un rol fundamental en la programación orientada a objeto (permite realizar instancias/objetos). Estas entidades permiten matear un concepto de la vida real a una entidad lógica de programación. Esta entidad es indivisible (entidad atómica).

2. 70pts. Considere 3 dataset: *regiones.txt*, *provincias.txt* y *comunas.txt*: La información contenida es la siguiente:

- *regiones.txt*: identificador y nombre de la región.
- *provincias.txt*: identificador y nombre de la provincia, además del identificador de la región a la que pertenece.
- *comunas.txt*: identificador y nombre de la comuna, además del identificador de la provincia a la que pertenece.

De acuerdo a esto, debe:

20pts Construir las entidades que permitan mapear los dataset. Utilice **herencia** para "heredar" el compartamiento común (ver figura 1).

40pts Construya un clase que:

30pts Desarrolle los métodos de lectura de los dataset.

10pts Desarrolle los métodos necesarios para "buscar" la información en las listas de objetos.

10pts Construya la clase y el método principal para la ejecución del programa.

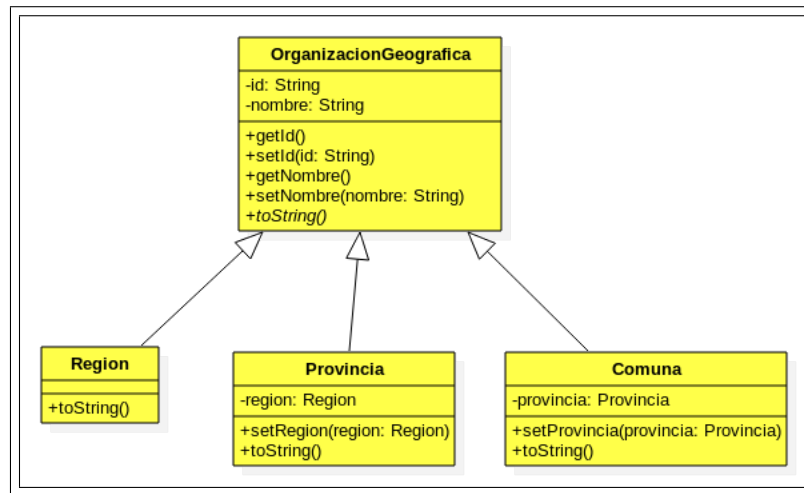


Figura 1: Diagrama de clase/entidades

****Revisar archivos java adjuntos.**

Para la pregunta 2.-, considere el uso de la siguiente clase:

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.List;

public class LecturaEscritura {

    public List<String> leer(String nombreArchivo) {

        File archivo;
        FileReader fr = null;
        List<String> lineas = null;

        try {
            archivo = new File(nombreArchivo);
            lineas = new ArrayList<String>();
            String linea;
            fr = new FileReader(archivo);
            BufferedReader br = new BufferedReader(fr);

            while ((linea = br.readLine()) != null) {
                lineas.add(linea);
            }
        } catch (IOException e) {
            System.out.println(e);
        } finally {
            try {
                if (fileReader != null) {
                    fileReader.close();
                }
            } catch (IOException e) {
                System.out.println(e);
            }
        }
        return lineas;
    }

    public void escribir(String nombreArchivo,
                        List<String> lineas) {

        FileWriter archivo;
        PrintWriter printWriter = null;

        try {
            archivo = new FileWriter(nombreArchivo, true);
            printWriter = new PrintWriter(archivo);

            for (String linea : lineas) {
                printWriter.println(linea);
            }
        } catch (IOException e) {
            System.out.println(e);
        } finally {
            printWriter.close();
        }
    }
}
```

¿Cómo será evaluado en la pregunta 1?

Tópico	Logrado	Medianamente logrado	No logrado
Herencia	6pts Comenta satisfactoriamente el mecanismo de herencia en la relación es-un .	3pts Comenta parcialmente el mecanismo de herencia en la relación es-un , dejando dudas respecto a la jerarquía Padre-Hijo.	1pts Comenta erróneamente el mecanismo de herencia en la relación es-un .
Clase abstracta	6pts Comenta satisfactoriamente el concepto de clase abstracta.	3pts Comenta parcialmente el concepto de clase abstracta, dejando dudas respecto a la manipulación de sus métodos.	1pts Comenta erróneamente el concepto de clase abstracta.
Interfaces	6pts Comenta satisfactoriamente la "simulación" de herencia múltiple.	3pts Comenta parcialmente la herencia múltiple, no utilizando interfaces.	1pts Comenta erróneamente la "simulación" de herencia múltiple.
TDA Lista	6pts Comenta satisfactoriamente el uso de TDA Listas para la colección de objetos.	3pts Comenta parcialmente el uso de TDA Listas para la colección de objetos, dejando dudas respecto a la manipulación de éstos.	1pts Comenta erróneamente el uso de TDA Listas.
TDA Bean	6pts Comenta satisfactoriamente el uso de TDA Beans como principal componente de la POO.	3pts Comenta parcialmente el uso de TDA Beans como principal componente de la POO, dejando dudas respecto a la utilidad de éstos.	1pts Comenta erróneamente el uso de TDA Beans.
Total máximo puntaje pregunta 1	30pts	15pts	5pts

¿Cómo será evaluado en la pregunta 2?

Tópico	Logrado	Medianamente logrado	No logrado
Construir entidades	20pts Aplica en forma correcta la herencia con el desarrollo de las entidades.	12pts No aplica de forma correcta la herencia, pero sí construye las entidades.	6pts No aplica de forma correcta la herencia y no construye las entidades.
Construir clase UbicacionImpl	40pts Construye satisfactoriamente la clase e implementa todos los métodos de lectura de los dataset y búsqueda de información en las listas de objetos.	25pts Construye la clase con lectura parcial de los dataset, con sólo algunos métodos de búsqueda de información en las listas de objetos.	12pts No construye la clase.
Construir clase principal	10pts Construye satisfactoriamente la clase principal y el método estático main, con las instancias de los objetos y la llamada a sus métodos.	5pts Construye la clase principal con el método estático main, pero no realiza correctamente las instancias de los objetos y las llamadas a los métodos.	0pts No construye la clase principal.
Total máximo puntaje pregunta 2	70pts	42pts	18pts