

Orientación a Objetos 2 – Curso 2022

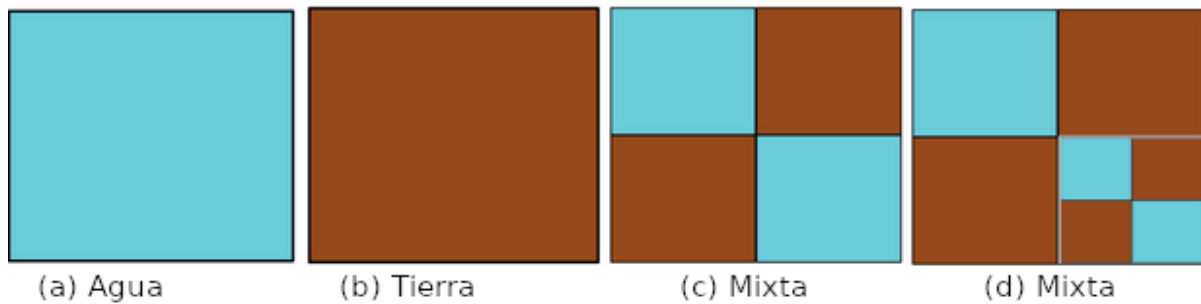
Práctica 2

Fecha de última edición: 7 marzo 2022

Ejercicio 1: Topografías

Un objeto Topografía representa la distribución de agua y tierra de una región cuadrada del planeta, la cual está formada por porciones de “agua” y de “tierra”. La siguiente figura muestra:

- (a) el aspecto de una topografía formada únicamente por agua.
- (b) otra formada solamente por tierra.
- (c) y (d) topografías mixtas.



Una topografía mixta está formada por partes de agua y partes de tierra (4 partes en total). Éstas a su vez podrían descomponerse en 4 más y así siguiendo.

La proporción de agua de una topografía sólo agua es 1. La proporción de agua de una topografía sólo tierra es 0. La proporción de agua de una topografía compuesta está dada por la suma de la proporción de agua de sus componentes dividida por 4. En el ejemplo, la proporción de agua es: $(1 + 0 + 0 + 1) / 4 = 1/2$. La proporción siempre es un valor entre 0 y 1.

1. Diseñe e implemente las clases necesarias para que sea posible:
 - a. crear Topografías,
 - b. calcular su proporción de agua y tierra,
 - c. comparar igualdad entre topografías. Dos topografías son iguales si tienen exactamente la misma composición. Es decir, son iguales las proporciones de agua y tierra, y además, para aquellas que son mixtas, la disposición de sus partes es igual.

Pista: notar que la definición de igualdad para topografías mixtas corresponde exactamente a la misma que implementan las listas en Java. <https://docs.oracle.com/javase/8/docs/api/java/util/AbstractList.html#equals-java.lang.Object->

2. Diseñe e implemente test cases para probar la funcionalidad implementada. Incluya en el set up de los tests, la topografía compuesta del ejemplo.

Ejercicio 2: Más Topografías

Extienda el ejercicio anterior para soportar (además de Agua y Tierra) el terreno Pantano. Un pantano tiene una proporción de agua de 0.7 y una proporción de tierra de 0.3. No olvide hacer las modificaciones necesarias para responder adecuadamente la comparación por igualdad.

Ejercicio 3: FileSystem

Un file system contiene un conjunto de directorios y archivos organizados jerárquicamente mediante una relación de inclusión. De cada archivo se conoce el nombre, fecha de creación y tamaño en bytes. De un directorio se conoce el nombre, fecha de creación y contenido (el tamaño es siempre 32kb). Modele el file system y provea la siguiente funcionalidad:

```
public class Archivo {
    /**
     * Crea un nuevo archivo con nombre <nombre>, de <tamano> tamaño
     * y en la fecha <fecha>.
     */
    public Archivo (String nombre, LocalDate fecha, int tamaño)

}

public class Directorio {
    /**
     * Crea un nuevo Directorio con nombre <nombre> y en la fecha <fecha>.
     */
    public Directorio(String nombre, LocalDate fecha)

    /**
     * Retorna el espacio total ocupado, incluyendo su contenido.
     */
    public int tamañoTotalOcupado()

    /**
     * Retorna el archivo con mayor cantidad de bytes en cualquier nivel del
     * filesystem contenido por directorio receptor
     */
    public Archivo archivoMasGrande()
```

```

/**
 * Retorna el archivo con fecha de creación más reciente en cualquier nivel
 * del filesystem contenido por directorio receptor.
 */
public Archivo archivoMasNuevo()
}

```

Tareas:

1. Diseñe y represente un modelo UML de clases de su aplicación, identifique el patrón de diseño empleado (utilice estereotipos UML para indicar los roles de cada una de las clases en ese patrón).
2. Diseñe, implemente y ejecute test cases para verificar el funcionamiento de su aplicación. En el archivo DirectorioTest.java del material adicional se provee la clase DirectorioTest que contiene tests para los métodos arriba descritos y la definición del método setUp. Utilice el código provisto como guía de su solución y extienda lo que sea necesario.
3. Implemente completamente en Java.

Ejercicio 4: Cálculo de sueldos

Sea una empresa que paga sueldos a sus empleados, los cuales están organizados en tres tipos: Temporarios, Pasantes y Planta. El sueldo se compone de 3 elementos: sueldo básico, adicionales y descuentos.

| | Temporario | Pasante | Planta |
|-----------|---|--|---|
| básico | \$ 20.000 + cantidad de horas que trabajo * \$ 300. | \$20.000 | \$ 50.000 |
| adicional | \$5.000 si está casado \$2.000 por cada hijo | \$2.000 por examen que rindió | \$5.000 si está casado \$2.000 por cada hijo \$2.000 por cada año de antigüedad |
| descuento | 13% del sueldo básico 5% del sueldo adicional | 13% del sueldo básico 5% del sueldo adicional | 13% del sueldo básico 5% del sueldo adicional |



Tareas:

1. Diseñe la jerarquía de Empleados de forma tal que cualquier empleado puede responder al mensaje #sueldo.
2. Desarrolle los test cases necesarios para probar todos los casos posibles.
3. Implemente en Java.