Universidad FASTA
Programación B

Primer Parcial (Práctica) – 30 de junio de 2023

Apellido y nombre:

La empresa de Almacenamiento y Logística "**ACOVACHA y DESPACHA S.A.**" desea disponer de un sistema informatizado para gestionar los costos del gran depósito en el que almacenan sus productos, previo a su envío. El depósito está subdividido en áreas en las que contienen dos tipos de compartimentos distintos (estanterías y espacios refrigerados) en los que se ubican las distintas mercaderías que la empresa distribuye.

Los distintos tipos de compartimentos tienen como atributos en común, su identificador, la descripción, la capacidad total (en kg o m^3 según sea estantería o espacio refrigerado) y la capacidad ocupada. De los estantes, a su vez, se almacena la cantidad de niveles que tiene el estante. De los espacios refrigerados, se conoce la temperatura a la que refrigera y los kilowatts que consume por día.

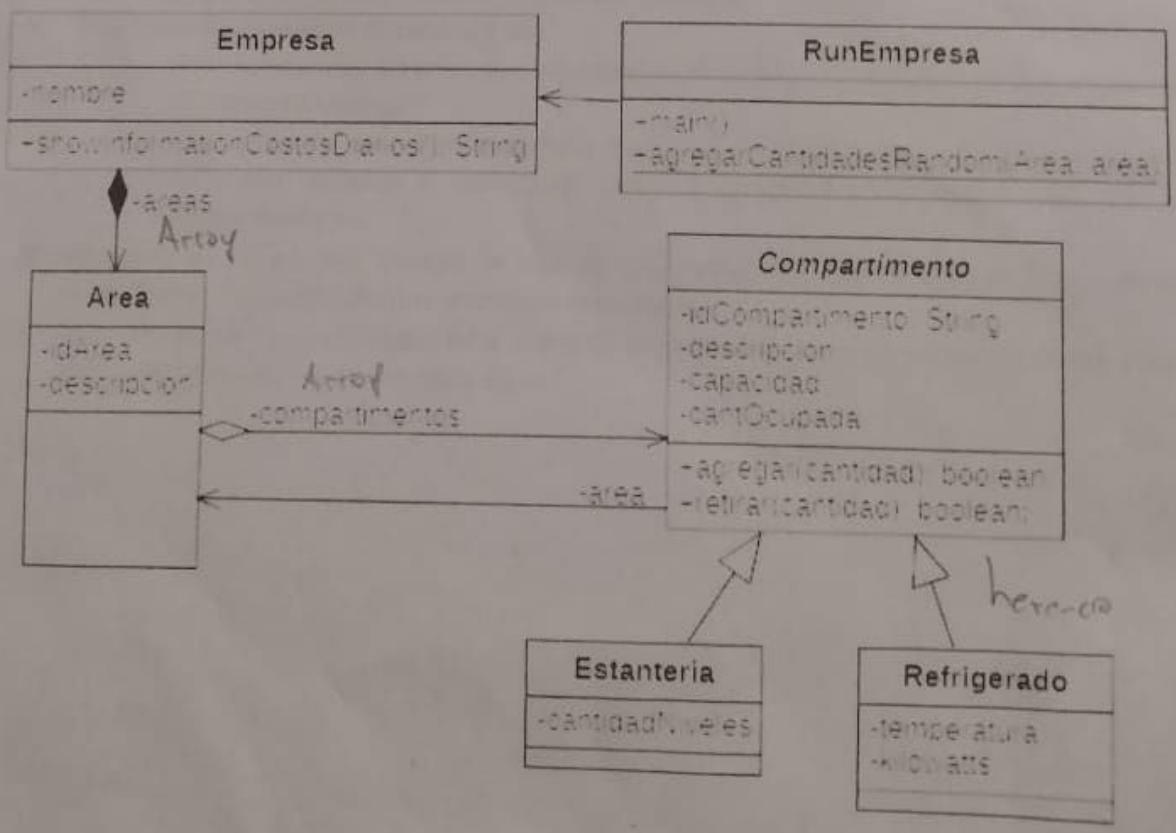
A un compartimento se pueden incorporar o retirar cantidades, debiéndose controlar **que no se agreguen cantidades superiores a la capacidad máxima ni que se retiren cantidades mayores a las que tiene ocupada**.

Se desea conocer cual es el costo diario de mantener la mercadería en cada compartimento. Para ello, se tiene una fórmula que calcula un **costo base** (común a cualquier tipo de compartimento) **más** un **costo específico** de cada tipo de compartimento:

Costo base	COSTO_DIARIO_COMPARTIMENTO * $\frac{\text{capacidad ocupada compartimento}}{\text{capacidad total compartimento}}$
Costo estantería	COSTO_DIARIO_ESTANTERIA + Cantidad niveles estante * COSTO_DIARIO_NIVEL
Costo espacio refrigerado	COSTO_KILOWATT_DIA * Kilowatt * factorTemperatura donde • Si temperatura <= -5°C → factorTemperatura = 1.5 • Si -5°C < temperatura <= 3° C → factorTemperatura = 1.2 • Si temperatura > 3° C → factorTemperatura = 1

Los valores de las constantes son:

- COSTO_DIARIO_COMPARTIMENTO = $1.000
- COSTO_DIARIO_NIVEL = $10
- COSTO_DIARIO_ESTANTERIA = $100
- COSTO_KILOWATT_DIA = $10



Primer Parcial (Práctica) – 30 de junio de 2023

La clase que contiene el main podría ser similar a lo que se muestra a continuación:

```
package model;

import java.util.Random;

public class RunEmpresa {
	public RunEmpresa() {
	}
	public static void main(String[] args) {
		Empresa empresa = RunEmpresa.creaEmpresa();
		empresa.showInformacionCostosDiarios();
	}

	public static void agregarCantidadesRandom(Area area) {
		Random = new Random();
		//Ciclo con los compartimentos del Area.
			//A cada compartimento, agregarle el valor randomValue
			int randomValue = random.nextInt(10000 + 1);
		//Fin Ciclo
	}

	public Empresa void creaEmpresa() {
	//* Crear la empresa, las areas y sus compartimentos
	//* Para cada área, invocar al método agregarCantidadesRandom(area)
	//* Invocar al método showInformacionCostoDiario de la empresa;
	}
}
```

Resolver:

- Codificar todas las clases del diagrama de clases sugerido.
- En la clase que contiene el método *main*
  - Crear la empresa, las áreas que componen el depósito y, para cada área, crear distintos tipos de compartimentos.
  - Una vez hecho esto, para cada Area invocar el método *agregarCantidadesRandom*() de la clase que contiene a *main*(). Allí deberán agregarse cantidades a cada uno de los compartimentos.
- Finalmente, mostrar por consola, la lista de compartimentos agrupados por Área, con toda la información disponible de los mismos y su costo diario.
- También mostrar el costo diario total (suma de los costos de todos los compartimentos) y los datos del compartimento de mayor costo diario.

Universidad FASTA
Programación B

Primer Parcial (Práctica) – 30 de junio de 2023

Apellido y nombre: _____

```
================================================================================================================================
Acovacha y Despacha S.A
================================================================================================================================
S03 - Sector 3
      E01 - Estante   Aceites (Estanteria ) Capacidad: 1000    Ocupado: 245  Cant. niveles: 4  Costo: 385.00
      E02 - Estante  Limpieza (Estanteria ) Capacidad: 4000    Ocupado: 13   Cant. niveles: 2  Costo: 123.25
      E03 - Estante  Vinoteca (Estanteria ) Capacidad: 3000    Ocupado: 1671 Cant. niveles: 1  Costo: 667.00

S02 - Sector 2
      E04 - Estante   Electro (Estanteria ) Capacidad: 5000    Ocupado: 4686  Cant. niveles: 3 Costo: 1067.20
      E05 - Estante  Mascotas (Estanteria ) Capacidad: 1000    Ocupado: 97    Cant. niveles: 2 Costo: 217.00
      E06 - Estante  Jardines (Estanteria ) Capacidad: 3000    Ocupado: 2860  Cant. niveles: 2 Costo: 1073.33

F01 - Frigorifico
      C01 - Camara Carniceria (Refrigerado) Capacidad: 5000    Ocupado: 3977      Temperatura: 0     KW: 10 Costo: 945.40
      C02 - Camara de Lacteos (Refrigerado) Capacidad: 3000    Ocupado: 1709      Temperatura: 6     KW: 30 Costo: 869.67
      C03 - Camara Congelados (Refrigerado) Capacidad: 3000    Ocupado: 2326      Temperatura: -5    KW: 20 Costo: 1175.33
      E07 - Estante de Quesos (Estanteria ) Capacidad: 1000    Ocupado: 567   Cant. niveles: 2 Costo: 687.00
--------------------------------------------------------------------------------------------------------------------------------
Costo total: 7210.18      Compartimento mayor costo: Camara Congelados: 1175.33
```

Observaciones:

- Para aprobar, **necesariamente** debe:
  - **Utilizar adecuadamente los principios de la POO**
  - **Implementar polimorfismo en métodos propios del modelo**. Es decir, hay que usar polimorfismo más allá de métodos como *toString()*, *equals()*, etc., que son propios de Java pero no de las clases del dominio.
- Se valorará positivamente el uso de constantes, el uso correcto de constructores, la eficiencia, la reutilización de código, el uso de nombres significativos en Clases, atributos, métodos y variables, la prolijidad y la correcta indentación.
- Se sugiere usar *double* para los atributos y variables que contengan importes. Para mostrarlos, se puede utilizar la clase DecimalFormat:

```
DecimalFormat decimalFormat = new DecimalFormat("#0.00");
decimalFormat.format(valorDecimal);
```