

Ejercicios

1. Analice el siguiente código que modela un club y sus respectivos miembros.

```
public class Club{
    //...
    public Club(){
        //...
    }
    public void join(Membership member){
        //...
    }
    public int numberOfMembers(){
        //...
    }
}

public class Membership{
    private String name;
    private int month;
    private int year;
    public Membership(String name, int month, int year){
        this.name = name;
        this.month = month;
        this.year = year;
    }
    public String getName() { return name; }
    public int getMonth(){ return month;}
    public int getYear(){ return year;}
    public String toString(){
        return "Name: " + name + " joined in month " + month + " of "
+ year;
    }
}
```

- A. Complete los métodos incompletos de la clase “Club”. (están marcados con //...).
- B. Agregue un método a la clase “Club” que retorne el numero de miembros que se unieron al club en un mes dado, usando la siguiente firma:
 - public int joinedInMonth(int month).
- C. Defina un método en la clase “Club” que retorne a todos los miembros que ingresaron al club en un determinado mes y año. El método debe tener la siguiente firma:
 - public ArrayList<Membership> select(int moth, int year).

2. Analice el siguiente código que modela un stock de productos.

```
import java.util.ArrayList;
public class StockManager{
    ArrayList<Product> stock;
    public StockManager(){
        stock = new ArrayList<Product>();
    }
    public void addProduct(Product item){
        stock.add(item);
    }
    public void delivery(int id, int amount){ //... }
    public Product findProduct(int id){ //...}
    public int numberInStock(int id){ //...}
    public void printProductDetails(){//...}
}
```

```

public class Product{
    private int id;
    private String name;
    private int quantity;
    public Product(int id, String name){
        this.id = id;
        this.name = name;
        quantity = 0;
    }
    public int getID(){ return id; }
    public String getName(){ return name; }
    public int getQuantity(){ return quantity;}

    public String toString(){
        return id + ": " + name + " stock level: " + quantity;
    }
    public void increaseQuantity(int amount){
        if(amount > 0) { quantity += amount; }
        else {
            System.out.println("Attempt to restock " + name +
                               " with a non-positive amount: " + amount); }
    }
    public void sellOne(){
        if(quantity > 0) { quantity--; }
        else {
            System.out.println("Attempt to sell an out of stock item:
" + name);
        }
    }
}

```

- A. Implemente el método “findProduct”. Este debe buscar un producto que coincida con el ID pasado como argumento, si el producto no existe el método debe retornar null.
- B. Implemente el método “numberInStock”. El mismo debe buscar el producto a través de su ID y devolver la cantidad de productos que aun quedan en stock.
- C. Implemente el método “delivery”. Este debe buscar el producto en base al ID, y incrementar la cantidad de ese producto en el stock. La cantidad a aumentar esta definida por la variable “amount”.