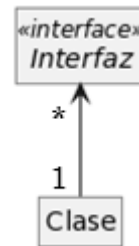


Nombre: Padrón:

TB025 – PARADIGMAS DE PROGRAMACIÓN – EXAMEN FINAL 2C-2024

1) Dado el siguiente diagrama de clases en UML:

- a) indique qué relación representa el mismo y su significado;
- b) explique cómo dicha relación podría aparecer en un programa.



2) Dado el siguiente programa:

```
public abstract class SuperHeroe {
    protected String nombre;
    public SuperHeroe(String nombre) {
        this.nombre = nombre;
    }
    public abstract void mostrarInformacion();
    public abstract void usarPoder();
}
```

```
public abstract class SuperHeroeDC extends SuperHeroe {
    public SuperHeroeDC(String nombre) {
        super(nombre);
    }
}
```

```
public class SuperHeroeDCHombre extends SuperHeroeDC {
    public SuperHeroeDCHombre(String nombre) {
        super(nombre);
    }
    @Override
    public void mostrarInformacion() {
        System.out.println(nombre + " es un superhéroe masculino de DC Comics.");
    }
    @Override
    public void usarPoder() {
        System.out.println(nombre + " utiliza sus gadgets especiales.");
    }
}
```

```
public class SuperHeroeDCMujer extends SuperHeroeDC {
    public SuperHeroeDCMujer(String nombre) {
        super(nombre);
    }
    @Override
    public void mostrarInformacion() {
        System.out.println(nombre + " es una superheroína femenina de DC Comics.");
    }
    @Override
    public void usarPoder() {
        System.out.println(nombre + " utiliza sus habilidades sobrehumanas.");
    }
}
```

```
public abstract class SuperHoeMarvel extends SuperHoe {
    public SuperHoeMarvel(String nombre) {
        super(nombre);
    }
}
```

```
public class SuperHoeMarvelHombre extends SuperHoeMarvel {
    public SuperHoeMarvelHombre(String nombre) {
        super(nombre);
    }
    @Override
    public void mostrarInformacion() {
        System.out.println(nombre + " es un superhéroe masculino de Marvel Comics.");
    }
    @Override
    public void usarPoder() {
        System.out.println(nombre + " utiliza su tecnología avanzada.");
    }
}
```

```
public class SuperHoeMarvelMujer extends SuperHoeMarvel {
    public SuperHoeMarvelMujer(String nombre) {
        super(nombre);
    }
    @Override
    public void mostrarInformacion() {
        System.out.println(nombre + " es una superheroína femenina de Marvel Comics.");
    }
    @Override
    public void usarPoder() {
        System.out.println(nombre + " utiliza sus habilidades sobrehumanas.");
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        SuperHoeDCHombre superHoeDCHombre = new SuperHoeDCHombre("Batman");
        superHoeDCHombre.mostrarInformacion();
        superHoeDCHombre.usarPoder();

        SuperHoeDCMujer superHoeDCMujer = new SuperHoeDCMujer("Wonder Woman");
        superHoeDCMujer.mostrarInformacion();
        superHoeDCMujer.usarPoder();

        SuperHoeMarvelHombre superHoeMarvelHombre = new SuperHoeMarvelHombre("Iron Man");
        superHoeMarvelHombre.mostrarInformacion();
        superHoeMarvelHombre.usarPoder();

        SuperHoeMarvelMujer superHoeMarvelMujer = new SuperHoeMarvelMujer("Black Widow");
        superHoeMarvelMujer.mostrarInformacion();
        superHoeMarvelMujer.usarPoder();
    }
}
```

- describa su estructura (mediante un diagrama de clases) y su funcionamiento;
- escriba la salida por pantalla al correrlo;
- indique si viola algún principio de diseño y, en caso afirmativo, indique cuál(es) y explique por qué;

3) Dada la siguiente expresión lambda:

$(\lambda x y. y \ x) \ ((\lambda x. x) \ ((\lambda z. w) \ u)) \ ((\lambda y. y) \ v)$

Redúzcala a su forma normal, utilizando u *orden normal* u *orden aplicativo*, aplicando las reglas *alfa*, *beta* y *eta* (usándolas solo cuando sea estrictamente necesario).

4) Utilizando terminología del paradigma de la Programación Funcional, explique la evaluación de la siguiente expresión escrita en Clojure e indique el resultado:

a)

`(- (reduce + (filter #(< % 0) (map #(int (Math/signum (float %))) [5 -4 3 0 -2 8 -3])))`

b) Justifique cuál es el error algorítmico que afecta negativamente la eficiencia del código anterior.

5) Dado el siguiente razonamiento:

1) $(\forall X) (\neg \text{riegan}(X) \supset \text{se_seca}(X))$
2) $(\forall X) (\text{se_seca}(X) \supset \text{muere}(X))$
3) $\text{riegan}(\text{jazmin}) \supset \text{riegan}(\text{rosal})$
4) $\neg \text{riegan}(\text{rosal})$

$\therefore \text{muere}(\text{jazmin})$

a) escriba el sistema clausal disyuntivo, lógicamente equivalente;

b) pruébelo mediante demostración por refutación, siguiendo una estrategia similar a la de Prolog.