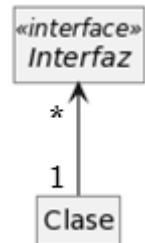


TB025 – PARADIGMAS DE PROGRAMACIÓN – EXAMEN FINAL 2C-2024

- 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 SuperHeroeMarvel extends SuperHeroe {
    public SuperHeroeMarvel(String nombre) {
        super(nombre);
    }
}

public class SuperHeroeMarvelHombre extends SuperHeroeMarvel {
    public SuperHeroeMarvelHombre(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 SuperHeroeMarvelMujer extends SuperHeroeMarvel {
    public SuperHeroeMarvelMujer(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) {
        SuperHeroeDCHombre superheroeDCHombre = new SuperHeroeDCHombre("Batman");
        superheroeDCHombre.mostrarInformacion();
        superheroeDCHombre.usarPoder();

        SuperHeroeDCMujer superheroeDCMujer = new SuperHeroeDCMujer("Wonder Woman");
        superheroeDCMujer.mostrarInformacion();
        superheroeDCMujer.usarPoder();

        SuperHeroeMarvelHombre superheroeMarvelHombre = new SuperHeroeMarvelHombre("Iron Man");
        superheroeMarvelHombre.mostrarInformacion();
        superheroeMarvelHombre.usarPoder();

        SuperHeroeMarvelMujer superheroeMarvelMujer = new SuperHeroeMarvelMujer("Black Widow");
        superheroeMarvelMujer.mostrarInformacion();
        superheroeMarvelMujer.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:

```
(λx y.y x) ((λx.x) ((λz.w) u)) ((λy.y) v)
```

Redúzcalo a su forma normal, utilizando un *orden normal* o *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) (∀X) (¬riegan(X) ⊃ se_seca(X))
2) (∀X) (se_seca(X) ⊃ muere(X))
3) riegan(jazmin) ⊃ riegan(rosal)
4) ¬riegan(rosal)
∴ muere(jazmin)
```

- a) escriba el sistema clausal disyuntivo, lógicamente equivalente;
- b) pruébelo mediante demostración por refutación, siguiendo una estrategia similar a la de Prolog.