

Tarea Semana 1

Nº Matrícula: 20106371k18

Fecha de entrega: viernes 8 de mayo, 14:00 Hrs.

Artefactos entregables:

- Para responder a los problemas 1 y 2, se deben agregar las respectivas respuestas en los lugares señalados del documento.
- Para responder al problema 3, se debe adjuntar el proyecto sin modificar su nombre, ni comprimir en un formato distinto a zip.

Problema 1: Considerando el siguiente programa, responda las siguientes preguntas:

- Explique, ¿qué se despliega al ejecutar el programa?
- Dibuje un modelo de memoria, que dé cuenta de las variables creadas en el programa.
- Explique, ¿A qué refiere la variable nivel en: `nivel = 50`; ?
 - ¿a una variable de la clase DciPokemon?
 - ¿a una variable local en el método principal?
 - ¿otra cosa?
 - ¿cómo influye en el programa?

```
public class DciPokemon {  
    public String nombre;  
    public int nivel;  
  
    public DciPokemon(String nombre, int nivel) {  
        this.nombre = nombre;  
        this.nivel = nivel;  
    }  
  
    public static void main(String[] args) {  
        DciPokemon p = new DciPokemon("Pikachu", 17);  
        int nivel = 100;  
        cambiar(p, nivel);  
        System.out.println("nombre: " + p.nombre + ", nivel: " + p.nivel);  
    }  
  
    public static void cambiar(DciPokemon poke, int nivel) {  
        poke.nivel = nivel;  
        nivel = 50;  
        poke = new DciPokemon("Gengar", 1);  
    }  
}
```

- Nombre: Pikachu, nivel: 100

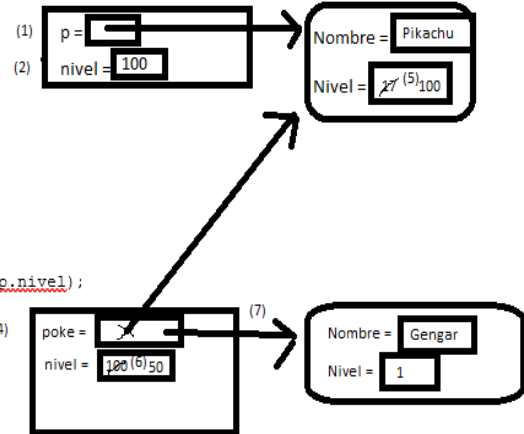
Tarea Semana 1

```
public class DciPokemon {
    public String nombre;
    public int nivel;

    public DciPokemon(String nombre, int nivel) {
        this.nombre = nombre;
        this.nivel = nivel;
    }

    public static void main(String[] args) {
        (1) DciPokemon p = new DciPokemon("Pikachu", 17);
        (2) int nivel = 100;
        (3) cambiar(p, nivel);
        System.out.println("nombre: " + p.nombre + ", nivel: " + p.nivel);
    }

    (4) public static void cambiar(DciPokemon poke, int nivel) {
        (5) poke.nivel = nivel;
        (6) nivel = 50;
        (7) poke = new DciPokemon("Gengar", 1);
    }
}
```



-
- 1. Como java prioriza los parámetros del método, ese "nivel" es una variable propia del método que se define con el valor otorgado cuando se llamó al método y se eliminará una vez este termine. 2.El como influye, es simple durante la ejecución del método este valor inicia como 100, ya que se llamo a este con ese parámetro, luego como ya hizo el cambio al objeto referenciado por "poke" en la línea anterior, el que se cambie más adelante no influye en nada, ya que ese valor no sale del método.

Problema 2: Considerando el siguiente programa, responda las siguientes preguntas:

- Explique, ¿qué problema presenta el programa? (hint: + de 1).
- ¿Qué se despliega al ejecutar por pantalla?, ¿Tuvo que realizar algún cambio? ¿Por qué?
- ¿Por qué los métodos "enojar" y "calmar" son definidos estáticos?

```
public class Gato {
    public String nombre;
    public static String sonido;

    public Gato(String nombre, String sonido) {
        this.nombre = nombre;
        this.sonido = sonido;
    }

    public void jugar() {
        System.out.println(sonido + " Soy " + nombre + " el gato!");
    }

    public static void enojar() {
        sonido = sonido.toUpperCase();
    }

    public static void calmar() {
```

Tarea Semana 1

```
        sonido = sonido.toLowerCase();  
    }  
  
    public static void main(String[] args) {  
        Gato a = new Gato("Pepe", "Meow!");  
        Gato b = new Gato("Goliat", "Buu!");  
        a.jugar;  
        b.jugar;  
        Gato.enojar();  
        a.calmar();  
        a.jugar;  
        b.jugar;  
    }  
}
```

- Al sonido ser declarado como estático, este es común para todos los objetos creados de la clase.
- Buu! Soy Pepe el gato!
Buu! Soy Goliat el gato!
Buu! Soy Pepe el gato!
Buu! Soy Goliat el gato!
- Porque ambos trabajan sobre la variable estática de la clase.

Problema 3: Para resolver este problema considere el proyecto IntelliJ disponible en el archivo ED_Tarea_S1.zip. Aquí encontrará una clase denominada ListaEnlazada, con tres métodos sin implementar: (1) agregarFrente(); (2) agregarFinal(); (3) agregarEnIndice(); y (4) imprimirLista();

Actividades:

1. Implemente cada uno de los métodos antes mencionados, considerando todos los casos de excepción. Recuerde que en clases se revisó sólo algunos excepciones y, por lo tanto, debe investigar otros casos que puedan hacer que su programa no funcione correctamente.
2. Observación 1: si bien se recomienda utilizar pruebas unitarias para evaluar el correcto funcionamiento de los métodos que implemente, no debe incluirlas en su entrega.
3. Observación 2: considerando el punto 1, cabe señalar que su código será evaluado automáticamente. Por lo tanto, sólo existirán métodos correctos o incorrectos.
4. Observación 3: considerando los puntos 1 y 2, es necesario recalcar que no debe modificar la estructura del código de la clase ListaEnlazada, salvo en el cuerpo de los métodos antes mencionados. Considere que si realiza algún cambio fuera de ellos, su código no podrá ser evaluado y, por lo tanto, recibirá puntuación mínima en este problema.