**Nº Matrícula: COMPLETAR**

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

Artefactos entregables:

* Para responder a los problemas 1 y 2, se deben agregan 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);  }  } |
| * Al ejecutar el programa se va a desplegar un mensaje el cual va a ser: “nombre: Pikachu, nivel: 100”, el cual va a aparecer en la ventana Run de una IDE, o si es que se abre a través de “command prompt” se vería después de ejecutar el programa.  * La variable nivel en: nivel = 50; es una variable de tipo int que se inicializo en el main (método principal) pero que al ser declarada en el método “cambiar” su valor que inicialmente es 100 pasa a ser 50, estando ahora el valor de “nivel” como 50 temporalmente hasta que se termine la ejecución del método “cambiar”, ahora todo eso no influye en el programa ya que como se dijo una vez terminado el método cambiar esta regresa a su valor original, además que en el periodo de tiempo en la que estuvo con el valor 50 la variable no fue utilizada. |

|  |
| --- |
| **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() {  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;  }  } |
| * El problema es que en la sección del main donde se encuentran: a.jugar y b.jugar, les falta en las parte del final un paréntesis, siendo la forma correcta de escribir a.jugar() y b.jugar() * Al principio sin hacer ningún cambio el programa no funcionaba, luego de agregar los paréntesis el programa se ejecutó correctamente, al ejecutar de nuevo el programa se mostraron cuatro mensaje los cuales fueron :   -Buu! Soy Pepe el gato!  -Buu! Soy Goliat el gato!  -buu! Soy Pepe el gato!  -buu! Soy Goliat el gato!  Como se puede observar todos los mensajes poseen el sonido Buu! Esto debido a que la variable Sonido es un static, lo que quiere decir que si la variable cambia en un objeto tambien se cambiara en los otros, si se quisiera quitar el static para que el sonido sea acorde a los valores que se les dio al instanciar inicialmente se tendrían que quitar los static de los metodos enojar y calmar, ademas de cambiar Gato.enojar(); por uno de los dos objetos ej: a.enojar; o b.enojar;   * Los métodos enojar y calmar son ambos estáticos debido a que la variable sonido que utilizan se definió como estático al inicio. |

**Problema 3**: Para resolver este problema considere el proyecto IntelliJ disponible en el archivo ED\_Tarea\_S1.zip. Acá 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.