**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, la caja de comandos despliega la información del DciPokemon p, que corresponde al String “Pikachu” y el integer 100, mostrando así: nombre: Pikachu, nivel: 100.    -La variable en: nivel = 50; es, valga la redundancia, la variable int local del método cambiar, y ya que ésta no interactúa con ninguna otra variable, su valor no afecta a la impresión realizada por el programa. |

|  |
| --- |
| **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;  }  } |
| -Este programa presenta varios problemas, para empezar, podríamos mencionar el hecho de que el método main está dentro del objeto que, si bien no es un error como tal no es lo recomendado, para seguir, dentro de dicho método, las operaciones de a.jugar; y b.jugar; no son frases correctas, para que estas funcionen es necesario añadir el () luego del nombre del método, de modo que queden: a.jugar(); y b.jugar();. Luego se puede mecionar en hecho que la variable String sonido de la clase, es estática, por lo que, independente de los objetos Gato que se instancien, todos harán el sonido que se determine en el último orden, que en este caso sería Buu!. Para finalizar se puede añadir el hecho de que en el main se llama al método enojar, pero para el objeto Gato, ya que Gato no es una variable inicializada, la acción es inútil.  -Para ejecutar el programa y obtener una impresión, es necesario arreglar a.jugar; y b.jugar; para que el programa funcione correctamente, debido a que estas frases no son comandos, por esto se les debe añadir los () para que efectivamente ejecuten al método que se está tratando de llamar, luego de hacer esto la impresión obtenida es:  Buu! Soy Pepe el gato!  Buu! Soy Goliat el gato!  buu! Soy Pepe el gato!  buu! Soy Goliat el gato!  -Los métodos enojar y calmar son definidos estáticos para que los cambios que están realizando se vean reflejados en las variables de la clase sin la necesidad de instanciar a estas dentro del método. |

**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.