Tarea Semana 1

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);
    }
}
```

Corresponde a la variable local en el método "cambiar" y no tiene efecto alguno en el resto de variables del mismo nombre en la clase DciPokemon o el método Main.

Tarea Semana 1

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;
    }
}
```

Explicación:
1.- note que la variable sonido fue declarada como estática. Esto significa que esta variaba tiene sólo un sonido para la clase entera. Por el contrario, cada vez que un objeto es Gato es creado, este obtiene su propio nombre. Un uso común de una variable estática es que se utiliza para almacenar el número total de objetos de la clase que han sido creados. Se necesita sólo una variable por clase para almacenar algo como eso.

Tarea Semana 1

2.-
Buu! Soy Pepe el gato!
Buu! Soy Goliat el gato!
buu! Soy Pepe el gato!
buu! Soy Goliat el gato!

3.- Respecto a las funciones "enojar" y "calmar" que son ambas declaradas estáticas. Los métodos estáticos son llamados usando el nombre de la clase. La regla para estos, es saber si el método puede sólo modificar variables estáticas. ¿Por qué?. Porque si tenemos un método estáticos, por ejemplo, cambiarNombreABoby() y llamamos Gato.cambiarNombreABoby(), qué nombre debería cambiar? Gato a?, Gato b?. No sabemos. Por lo tanto, esta regla debe ser obedecida.

Problema 3: Para resolver este problema considere el proyecto IntelliJ disponible en el archivo Tarea_01.zip. Acá encontrará una clase denominada ListaEnlazada, con tres métodos sin implementar: (1) agregarFrente(); (2) agregarCola(); y (3) agregarEnIndice().

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.

Tarea Semana 1

```
public class ListaEnlazada {
  // variables miembro de la clase Lista Enlazada
  Nodo cabeza = null;
  Nodo cola = null;

  /*
   * Método que permite agregar un elemento
   * al frente de la Lista Enlazada
   */
  public void agregarFrente(int valor){
     // implementar
     Nodo nuevoNodo = new Nodo(valor);
     if(this.cabeza == null){
        cabeza = nuevoNodo;
        cola = nuevoNodo;
     }else {
        nuevoNodo.siguiente = cabeza;
        cabeza = nuevoNodo;
     }
  }

  /*
   * Método que permite agregar un elemento
   * al final la Lista Enlazada
   */
  public void agregarFinal(int valor){
     // implementar
     Nodo nuevoNodo = new Nodo(valor);
     if(this.cabeza == null){
        cabeza = nuevoNodo;
        cola = nuevoNodo;
     }else {
        cola.siguiente = nuevoNodo;
        cola = nuevoNodo;
     }
  }

  /*
   * Método que permite agregar un elemento
   * en un índice específico de la Lista Enlazada
   */
  public void agregarEnIndice(int index, int valor){
     Nodo nuevoNodo = new Nodo(valor);
     if(index<0){
        throw new IndexOutOfBoundsException();
     }else if ( index == 0) { // agregar en la cabeza
        agregarFrente(valor);
     }else{
        Nodo actual = cabeza;
        for (int i =0;i<index-1;i++){
           if ( actual == null ) {
              throw new IndexOutOfBoundsException();
           }
           actual = actual.siguiente;
        }
     if ( actual.siguiente == null ) { //añadir al final
        actual.siguiente = cola;
        cola = nuevoNodo;
```

Tarea Semana 1

```
    } else {
        nuevoNodo.siguiente = actual.siguiente;
        actual.siguiente = nuevoNodo;
    }
  }
  }

  /*
   * Método que permite imprimir los elementos
   * de la Lista Enlazada
   */
  public void imprimirLista(ListaEnlazada nombreLista){
    if(nombreLista.cabeza==null){
        System.out.print("[]");
    }else{
        Nodo actual = cabeza;
        String respuesta = "[";
        while(actual != null) {
            if(actual.siguiente != null) {
                respuesta = respuesta + actual.valor + ",";
            }else{
                respuesta = respuesta + actual.valor;
            }
            actual = actual.siguiente;
        }
        System.out.print(respuesta+"]");
    }
  }

  public static void main(String[] args) {

  }
}
```