

Tarea Semana 1

Nº Matrícula: 19458012616

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;
    }

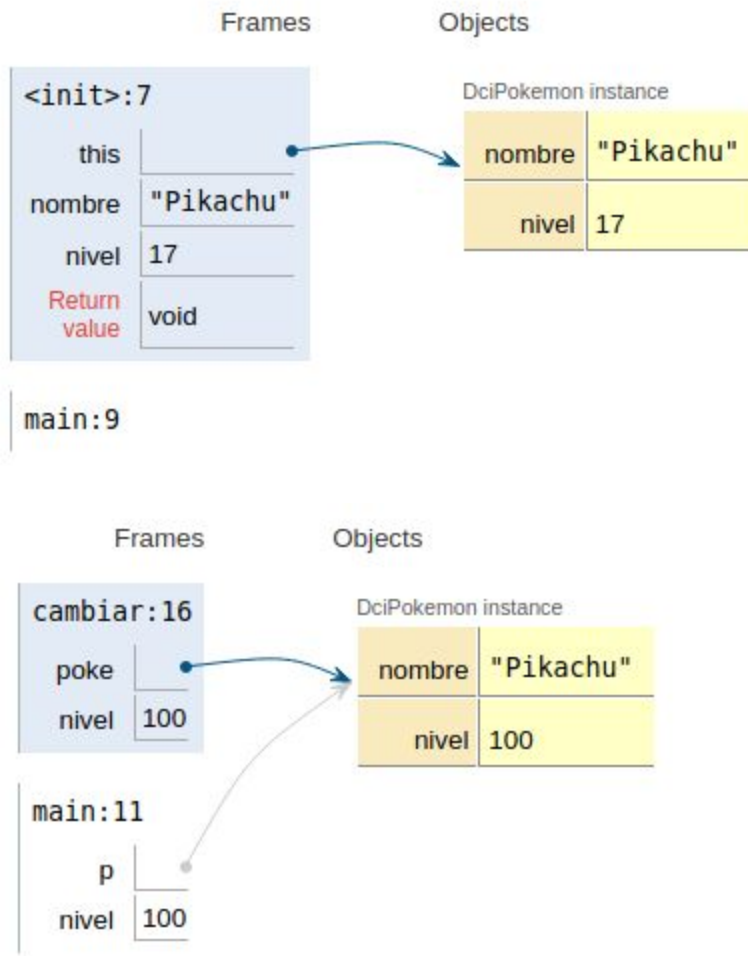
    public static void main(String[] args) {
        DciPokemon p = new DciPokemon("Pikachu", 17);
        int nivel = 100;
        cambiar(p, nivel);
        System.out.println("nombre: " + p.nombre + ", nivel: " + p.nivel);
    }

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

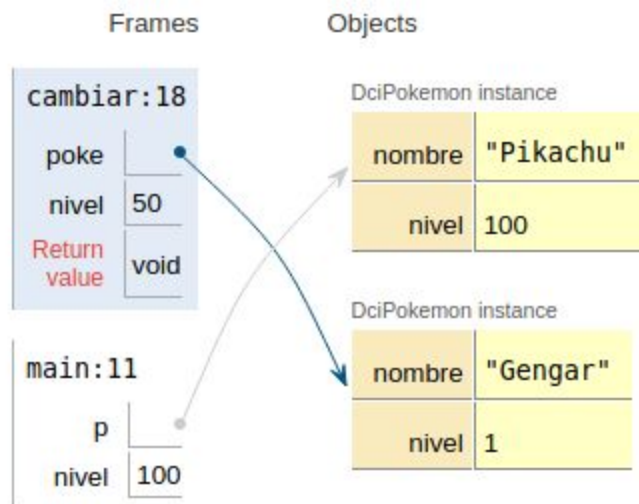
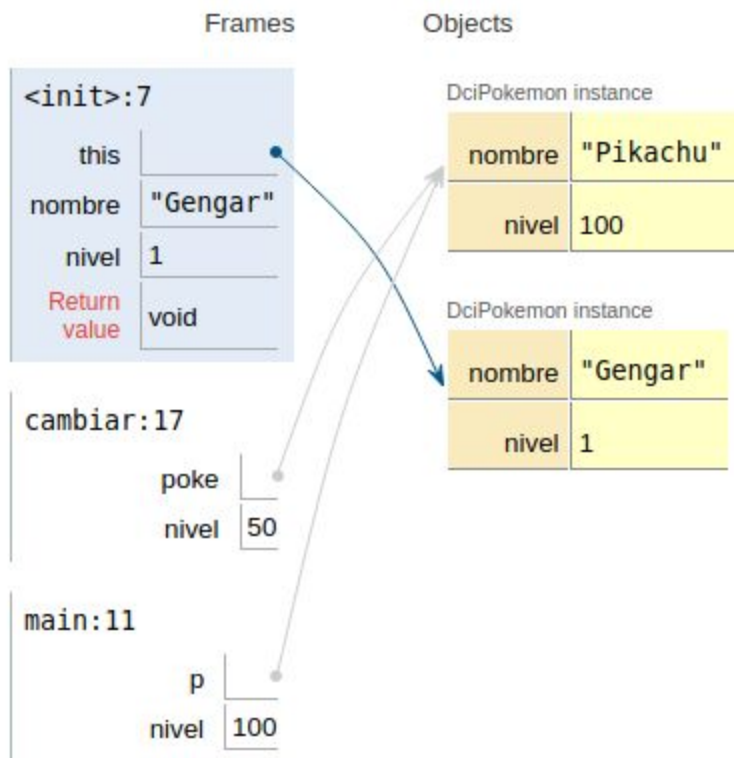
- Se despliega el siguiente mensaje por consola: - nombre: Pikachu, nivel: 100

Tarea Semana 1

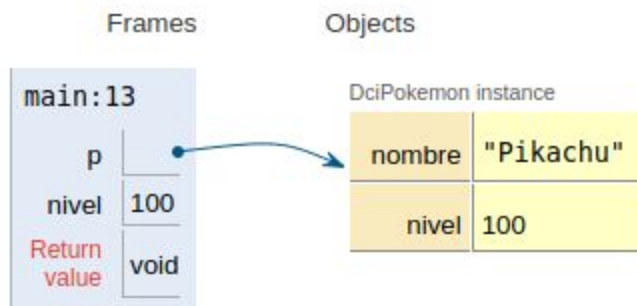
- Modelo de Memoria:



Tarea Semana 1



Tarea Semana 1



- Se refiere a la variable nivel del objeto poke que sólo está presente en el método cambiar, por lo que afecta al resto del programa sólo si se instancia este método dentro la clase principal, en otro caso no afecta ya que en ningún momento se asignó el valor de esa variable al valor del atributo de clase.

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();  
    }  
}
```

Tarea Semana 1

```
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;  
}  
}
```

- Los problemas presentes en el program son:
 - El atributo sonido está definido de forma estática, por lo que en el constructor no le logra asignar un valor al atributo, ya que el constructor no es estático.
 - No se instancia correctamente el método jugar, en a.jugar y b.jugar ya que tiene problemas de sintaxis, siendo a.jugar() y b.jugar() la forma correcta.
 - Debido al problema de la incorrecta asignación de valor en el constructor del atributo sonido esto afecta directamente a la instancia de objetos tipo gato y a sus métodos.
- Nada, debido a que tiene problemas de sintaxis el programa no logra ejecutarse correctamente y por consiguiente no se muestra por pantalla. Los cambios realizados para un correcto funcionamiento fue cambiar el atributo sonido de estático a no estático, para que el constructor logre asignar valores al atributo, y reemplazar la instancia de los métodos a.jugar y b.jugar por a.jugar() y b.jugar() ya que de esa es la forma correcta la sintaxis.
- Los métodos enojar y calmar fueron definidos estáticos para que cada uno se instancie dentro del main sin necesidad de crear un objeto Gato, y así poder anidar el método principal dentro de la misma clase Gato.

Tarea Semana 1

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 algunas 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.