**Parte 1- Repaso de conceptos**

**Nombre:** Daniel Anzules

**Conceptos básicos**

**¿Qué significa que una variable sea static en Java?**

Una variable static es una variable que pertenece a la clase y no a una instancia específica de esa clase. Esto significa que todas las instancias de la clase comparten la misma copia de esa variable.

**¿Qué es un método static y para qué se usa?**

Un método static es un método que pertenece a la clase, no a ninguna instancia. Esto significa que se puede llamar al método sin crear un objeto de la clase y no puede acceder directamente a variables o métodos de instancia no estáticos.

**¿Cuál es la diferencia entre una variable estática y una variable de instancia?**

Una variable static pertenece a la clase y es compartida por todas las instancias, es decir, todas las copias de un objeto acceden y modifican la misma variable estática. Se puede acceder a ella usando el nombre de la clase sin necesidad de crear un objeto.

Por otro lado, una variable de instancia pertenece a cada objeto individual, por lo que cada instancia tiene su propia copia y solo puede accederse a través de esa instancia.

**¿Cuál es la diferencia entre un método estático y uno de instancia?**

Un método estático pertenece a la clase y puede llamarse sin necesidad de crear un objeto. Este tipo de método solo puede acceder a variables y otros métodos estáticos, y no puede usar referencias como this o super porque no está asociado a una instancia específica.

En cambio, un método de instancia pertenece a un objeto particular y debe ser llamado desde una instancia. Los métodos de instancia pueden acceder tanto a variables y métodos estáticos como a los de instancia, y sí pueden usar this y super para referirse al objeto actual.

**¿Dónde se almacenan las variables estáticas en memoria?**

Las variables estáticas se almacenan en un área especial de memoria llamada "metaspace" (en Java 8+) o en el área de método (method area) dentro de la JVM.

**Aplicaciones y ejemplos**

**¿Cuándo debería usar una variable estática en mi programa?**

Debería usarse una variable estática cuando el dato que quieres almacenar debe ser compartido por todas las instancias de una clase, es decir, cuando el valor es común para todos los objetos o cuando quieres mantener un estado global. Por ejemplo, un contador que registra cuántos objetos de una clase han sido creados.

**¿Por qué algunos métodos como main son estáticos?**

El método main es estático porque es el punto de entrada de un programa Java y debe poder ejecutarse sin necesidad de crear un objeto de la clase. La JVM lo invoca directamente usando el nombre de la clase, y como no existe ningún objeto aun cuando inicia el programa, main debe ser estático para que pueda ejecutarse.

**¿Puedes darme un ejemplo simple que use una variable estática?**

Al llevar un contador de los objetos creados a partir de una misma clase.

class Contador {

static int cuenta = 0;

Contador() {

cuenta++; // Incrementa la variable estática cada vez que se crea un objeto

}

static void mostrarCuenta() {

System.out.println("Objetos creados: " + cuenta);

}

}

**¿Puedes mostrarme un ejemplo donde no se debe usar static?**

Cuando una variable representa datos que cambia, como una variable que representa el nombre o la edad de una persona, que varia dependiendo de una u otra persona.

class Persona {

String nombre; // Variable de instancia, cada persona tiene su propio nombre

static int edad; // Incorrecto: no debería ser estática porque cada persona tiene su propia edad

Persona(String nombre, int edad) {

this.nombre = nombre;

this.edad = edad;

}

}

**Acceso y restricciones**

**¿Cómo accedo a una variable o método estático desde otra clase?**

Se accede directamente usando el nombre de la clase, seguido del operador punto, sin necesidad de crear un objeto.

**¿Puedo acceder a variables de instancia desde un método estático?**

No se puede acceder directamente a variables o métodos de instancia desde un método estático, porque los métodos estáticos no pertenecen a ningún objeto en particular y no tienen acceso al estado de instancias específicas.

**¿Qué errores pueden ocurrir si intento usar mal una variable o método estático?**

* **Error de compilación:** Si se intenta usar this o acceder a una variable de instancia directamente dentro de un método estático, el compilador mostrará un error.
* **Comportamiento inesperado:** Si se declara como estática una variable que debería ser de instancia, todas las instancias compartirán ese dato, lo que puede generar resultados incorrectos o pérdida de datos específicos de cada objeto.
* **Confusión y dificultad para mantener el código**: Usar estático sin criterio puede complicar el diseño y el mantenimiento del programa.

**Comparación entre clases**

**¿Qué ocurre si creo muchos objetos de una clase que tiene una variable estática? ¿Qué valor tendrá esa variable?**

Todos los objetos comparten la misma variable estática, es decir, solo existe una copia en memoria para toda la clase. El valor de esa variable será el último valor que haya sido asignado, independientemente de cuál objeto lo haya cambiado.

**¿Cada objeto tiene su propia copia de una variable estática?**

No, las variables estáticas no se copian en cada objeto. Solo existe una copia compartida para toda la clase. En cambio, las variables de instancia sí tienen una copia individual en cada objeto.

**¿Qué ocurre si modifico el valor de una variable estática desde un objeto?**

Aunque se modifique la variable estática a través de un objeto, en realidad se está modificando la única copia que existe para toda la clase. Por lo tanto, ese cambio será visible desde todos los objetos.

**Parte 2 - Ejercicio**

1. **¿La variable de clase o variable estática counter qué representa (para qué se usa)?**

Se usa como una especie de contador de cuantas instancias de la clase Estudiante se han creado, es decir, el número de estudiantes.

**¿Cuál es la salida de ejecutar la clase DemoEstudiante?**

Nombre = Abigail

Codigo = 1

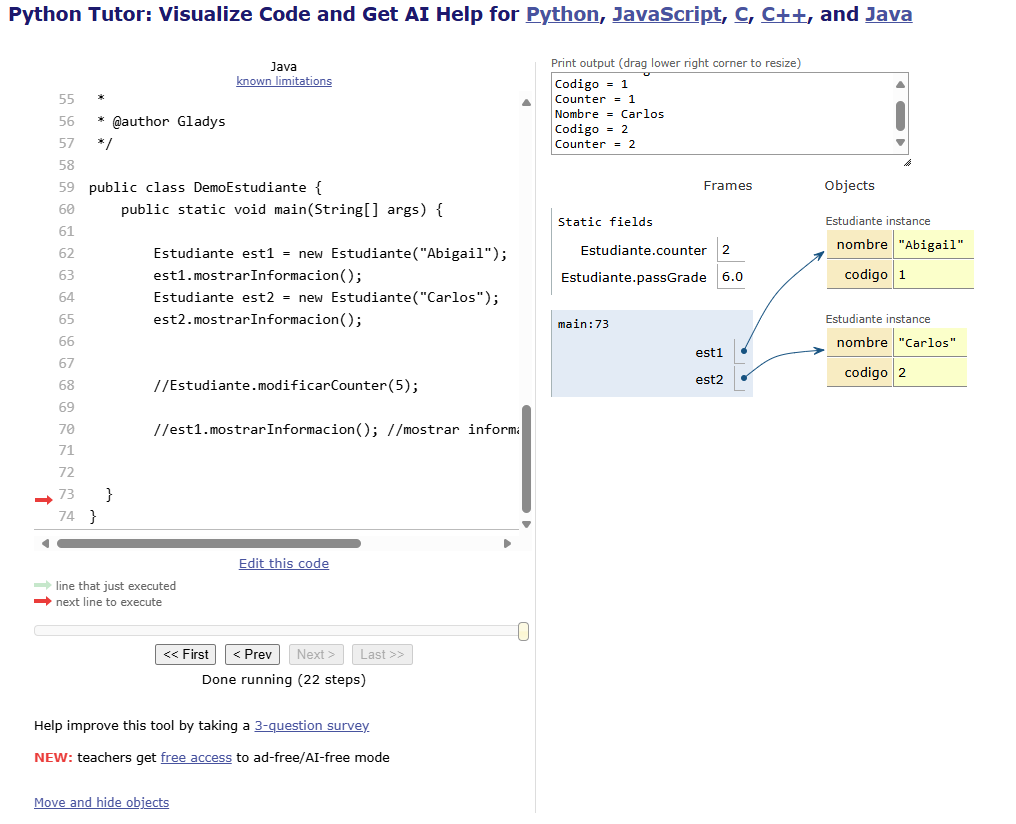
Counter = 1

Nombre = Carlos

Codigo = 2

Counter = 2

1. **Ejecute la clase Estudiante y DemoEstudiante en Python tutor – Analice dónde se crean las variables y agregue aquí una captura de la imagen**



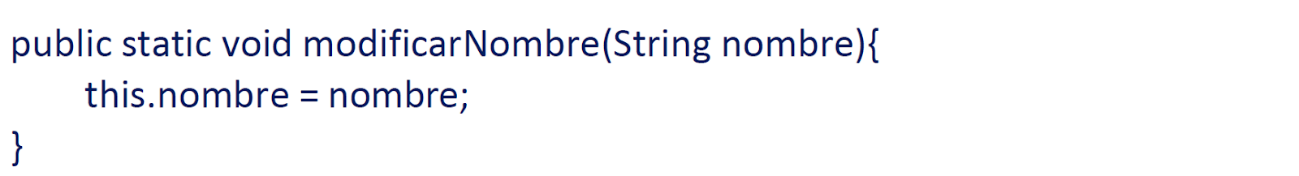
1. **En la clase DemoEstudiante descomente las siguientes líneas:** //Estudiante.modificarCounter(5);

//est1.mostrarInformacion();

**¿Cuál será la salida del programa? – Justifique su respuesta (¿por qué ocurre eso?)**

La salida será la parecida que antes de descomentar las líneas, mostrará la información del Estudiante 1 y luego del Estudiante 2, pero ahora volverá a mostrar la información del Estudiante 1 y su counter será 5 ya que la primera línea descomentada le asigna ese valor a la variable estática mediante el método modificarCounter, lo que cambiará el valor para todas las instancias de la clase, incluido el Estudiante 1.

1. **Supongamos que en la clase Estudiante se agrega el siguiente método:**

**¿La clase Estudiante compilará o no compilará?, ¿Por qué? – Justifique su respuesta.**

No, ya que es un método static y por ello solo puede acceder a variables estáticas, y nombre fue declarada inicialmente como una variable private, lo que impide compilar.

1. **Supongamos que en la clase Estudiante se agrega el siguiente método:**



**¿La clase Estudiante compilará? Si es que compila, ¿qué haría el método y cuál sería la salida si se lo invoca? Justifique su respuesta.**

Si, ya que hay un método definido con ese nombre, lo que haría será modificar el valor de la variable estática counter por el valor de 0.

1. **Supongamos que en la clase Estudiante se agrega el siguiente método:**



¿La clase **Estudiante** compilará? Si es que compila, ¿qué haría el método y cuál sería la salida si se lo invoca? - Justifique su respuesta

No compila, ya que el método test2 es un método estático y solo puede llamar a otros métodos estáticos, mientras que test1 es un método no estático, con ello en mente no compilara y no cambiara la salida.

1. Supongamos que en la clase **Estudiante** se agrega el siguiente método: public static void 

¿La clase **Estudiante** compilará? Si es que compila, ¿qué haría el método y cuál sería la salida si se lo invoca? Justifique su respuesta.

Si compila ya que los métodos modificarCounter y test3 son métodos estáticos, y test3 peude llamar al método modificarCounter sin ningún problema, en este caso le asgina el valor de cero a la variable estática counter.