

Tarea 4: Scanner y Controles de flujo

Sánchez Segura Cristian Alonso (314217547)

14 de abril de 2020

1. Preguntas generales

1. **¿Para qué sirve los parámetros del main "String arg []"?**
Para recibir valores ingresados desde la línea de comandos.
2. **Escribe 5 métodos de Scanner que nos ayuden a tomar los datos de entrada. Agrega su definición del API de JAVA.**
 - a) `nextByte()` : Devuelve un valor de tipo byte.

```
public byte nextByte()
```
 - b) `nextDouble()` : Devuelve un valor de tipo double.

```
public double nextDouble()
```
 - c) `nextInt()` : Devuelve un valor de tipo int

```
public int nextInt()
```
 - d) `nextLong()` : Devuelve un valor de tipo long

```
public long nextLong()
```
 - e) `next()` : Devuelve un valor de tipo String

```
public String next()
```
3. **¿En que momento se dispara "java.util.InputMismatchException"?**
Lanzado por un escáner para indicar que el valor recuperado no coincide con el patrón para el tipo esperado, o que el valor está fuera del rango para el tipo esperado.
4. **Responde las siguientes preguntas. Deja tu resultado y el resultado de la ejecución, explica cual fue tu error.**
 - ¿Cuántas veces se itera en los siguientes códigos?
 - a) Se itera 4 veces
 - b) Se itera 1 vez
 - c) Se itera 4 veces
 - d) Se itera indefinidamente
 - ¿Cual es el valor final de i en cada caso?
 - a) Valor final: 5
 - b) Valor final: 5
 - c) Valor final: 4
 - d) Valor final i: 0, Valor final j: 0
 - ¿Cual es el alcance de la variable i, j en cada caso?
 - a) La variable i tiene alcance global (en el contexto de su código).
 - b) La variable i tiene alcance global
 - c) La variable i tiene alcance solo dentro del for
 - d) Las variables i,j tienen alcance solo dentro del for
 - Si hay error de sintaxis o semántica, ¿Como se corrige?
 - a) No tiene errores

- b) No tiene errores
- c) Tiene un error de semántica, afuera del for no se puede acceder a las variables que se definan dentro del for. Se soluciona haciendo la variable global.
- d) Tiene dos errores de semántica, el primero es igual al inciso anterior y el segundo es la condición para terminar con el for, nunca hay un incremento en el valor de las variables por tanto, nunca se cumple la condición. Se soluciona haciendo el incremento de las variables y haciendo global la variable j.

- a)

```
int i = 0;
while( i++ < 4){
    System.out.println(i);
}
System.out.println("valor_final "+i);
```
- b)

```
int i = 3;
while( i++ < 4){
    System.out.println(i);
}
System.out.println("valor_final "+i);
```
- c)

```
for(int i = 0 ; i < 4 ; i++){
    System.out.println(i);
}
System.out.println("valor_final "+i);
```
- d)

```
for(int i = 0, j = 0; i < 4 && j < 3; i+=j){
    System.out.println(i);
}
System.out.println("valor_final "+j);
```

2. Preguntas del codigo

1. **¿Por qué utilizaste esa estructura de control para el manejo del menú?**
Utilice switch y while porque me parece la mas fácil de usar y de entender.
2. **Si utilizaste una estructura diferente al switch, ¿Por qué esa implementación es mejor?** Se podría utilizar if-else pero como son muchas las opciones que hay que evaluar, sería escribir código de mas.
3. **¿Cómo puedes mejorar la robustez de tu código o que medidas tomas-te para robustecerlo? Aclaro, no estoy pidiendo la implementación de excepciones en java.** Se puede mejorar utilizando excepciones para cada caso, así te aseguras que si el usuario ingresa algún valor inválido no termine el programa.

3. Referencias

Berenguer, M. C. (2019, noviembre 12). Cómo funciona el método `main()` en Java. Recuperado 13 de abril, 2020,

<https://javautodidacta.es/metodo-main-en-java/>

Oracle. (s. f.). Java Platform SE 7. Recuperado 13 de abril, 2020,

<https://docs.oracle.com/javase/7/docs/api/>

Oracle. (2018, octubre 6). `InputMismatchException` (Java Platform SE 7). Recuperado 13 de abril, 2020,

<https://docs.oracle.com/javase/7/docs/api/java/util/InputMismatchException.html>