Testing II

Sintaxis básica de JAVA

Tipos de datos más comunes

- ➔ String: tiene como uso el almacenar cadenas de caracteres encerrados por comillas dobles como por ejemplo "Testing 2".
- ➔ int: tiene como uso el almacenar números enteros sin decimales, como por ejemplo 32 o -19
- ➔ float: tiene como uso el almacenar números decimales como por ejemplo 32.87 o -19.05
- ➔ char: tiene como uso el almacenar caracteres individuales encerrados por comillas simples como por ejemplo 'z' o 'Y'
- ➔ boolean: tiene como uso el almacenar dos valores, como Verdadero o Falso.

Declaración e inicialización de variables

Cuando decimos que queremos **“declarar” una variable** nos referimos a que reservaremos un espacio en la memoria indicando un tipo de dato y un identificador. Por ejemplo:

string materiaDH;
int edad;

En cambio, cuando vamos a inicializarla, es cuando asignamos un valor dado a la variable de interés. Por ejemplo:

string materiaDH = “Testing 2”;
int edad = 27;

Matrices (Arrays)

Las matrices o arrays son particularmente útiles para almacenar valores en una sola variable, en lugar de declarar variables separadas para cada valor.

Por ejemplo:

String[] materiasDH = {"Backend I", "Introducción a la informática", "Testing I"};

Estructuras condicionales

JAVA tiene estructuras condicionales que permiten establecer qué instrucciones deben de ejecutarse o no, en función del valor de una condición. Estas son:

- ➔ La estructura **if**: útil para definir que un cierto bloque de código se podrá ejecutar solo si una condición especificada es verdadera.
  - Ejemplo:

```
if (condición) {
  // sentencia ejecutable sólo si la condición del if es verdadera.
}
```

- ➔ La estructura **if-else**: útil para definir qué bloque de código se podrá ejecutar dependiendo si la condición resulta ser verdadera o falsa.
  - Ejemplo:

```
if (condición) {
        // sentencia ejecutable sólo si la condición del if es
verdadera.
} else {
        // sentencia ejecutable sólo si la condición del if es falsa.
}
```

- ➔ **Las estructuras anidadas entre if y else**: útil para cuando en una misma estructura coexisten dos o más if y/o else.
  - Ejemplo:

```
if (condición 1) {
        // sentencia ejecutable sólo si la condición 1 es verdadera.
} else {
```

```
    if (condición 2) {
        // sentencia ejecutable sólo si la condición 1 es falsa y la condición 2 es verdadera.
    } else {
        // sentencia ejecutable sólo si la condición 1 es falsa y la condición 2 es falsa.
    }
}
```

➔ La estructura **switch-case**: útil ante la necesidad de seleccionar entre varias alternativas posibles.

- Ejemplo:

```
switch (expresión) {
    case alternativa1:
        // sentencia ejecutable sólo si la alternativa1 es igual a la expresión.
        break;
    case alternativa2:
        // sentencia ejecutable sólo si la alternativa2 es igual a la expresión.
        break;
    default:
        // sentencia ejecutable sólo si ninguna de las alternativas es igual a la expresión.
}
```

Estructuras iterativas

JAVA cuenta con declaraciones conocidas popularmente como bucles que permiten ejecutar un conjunto de sentencias ejecutables un número fijo de veces.

Algunos de los tipos de estructuras iterativas son:

➔ **Bucle For**:

Tiene como función el ejecutar un cierto bloque de código dentro de su estructura un número predeterminado de veces.

- Ejemplo:

```
for (i = 0; i < 5; i++) {
    // sentencia ejecutable durante 5 veces consecutivas.
```

}

➔ **Bucle While:**

Tiene como función el ejecutar un cierto bloque de código dentro de su estructura de manera indefinida hasta que cierta condición sea falsa.

- Ejemplo:

```
while (i < 5) {
        // sentencia ejecutable 5 veces hasta que i ya no sea menor a 5
 i++;
}
```

- *Bucle Do-while:*

Es similar al bucle While aunque se diferencie en que la ejecución ocurre antes de la evaluación de la condición.

Ejemplo:

```
do {
        // sentencia ejecutable 5 veces hasta que i ya no sea menor a 5
 i++;
}
while  (i <5);
```