
Programación

Seguimiento 1

Johan Felipe Molina Aguirre

Facultad de Ingenierías y Ciencias Básicas 2024-2

Ingeniería de software

Armenia - Quindio

2024

# I. Seguimiento 1

## A. Desarrollo

1) Un paradigma de programación es una manera o estilo de programación de software. Existen diferentes formas de diseñar un lenguaje de programación y varios modos de trabajar para obtener los resultados que necesitan los programadores. Se trata de un conjunto de métodos sistemáticos aplicables en todos los niveles del diseño de programas para resolver problemas computacionales.

   Los lenguajes de programación adoptan uno o varios paradigmas en función del tipo de órdenes que permiten implementar como, por ejemplo, Python o JavaScript, que son multiparadigmas.

2) La programación orientada a objetos (POO) es un paradigma de programación que utiliza "objetos" para diseñar software. Los objetos son instancias de "clases", que pueden ser vistas como plantillas que definen propiedades (atributos) y comportamientos (métodos) que los objetos de esa clase pueden tener. Aquí están los conceptos fundamentales de la POO.

3) 3. Basado en el material de lectura del classroom:

   a. Resumen de las preguntas frecuentes de la página 9. “Frequently Asked Questions”.

   What is a virtual machine? Una máquina virtual (VM) es un software que emula una computadora real, permitiendo ejecutar programas como si fueran ejecutados en una máquina física. La Java Virtual Machine (JVM) es un tipo de VM que ejecuta programas Java, proporcionando independencia de la plataforma.

   b.

| Características | Java | Python | JavaScript |
|---|---|---|---|
| Paradigma | Orientado a Objetos | Orientado a objetos, funcional | Basado en prototipos, funcional |
| Tipado | Estático | Dinámico | Dinámico |
| Compilación | Compilado (byteCode) | Interpretado | Interpretado |
| Uso Principal | Aplicaciones empresariales, android | Desarrollo web, ciencia de datos | Desarrollo web (frontend y backend) |
| Sintaxis | Verbosa | Simple y legible | Intermedia |

| Rendimiento | Alto | Moderado | Moderado |
|---|---|---|---|
| Plataforma | Multi-plataforma (JVM) | Multi-plataforma | principalmente web |

c.

Explicaciones a partir de la página 20.
¿Cómo hacer un comentario en Java?
En Java, se pueden hacer comentarios de tres maneras:
Comentario de una sola línea: Usando // seguido del comentario. Por ejemplo:

```
int i = 0;  // Inicializa la variable de bucle
```

Comentario de múltiples líneas: Envolviendo el comentario entre /* y */. Por ejemplo:

```
/*
 * Este es un comentario de múltiples líneas.
 * Puede abarcar varias líneas.
 */
```

Comentario de documentación: Usando /** para comenzar y */ para cerrar. Este tipo de comentario se usa para documentar clases y métodos, y puede ser procesado por herramientas como Javadoc. Por ejemplo:

```
/**
 * Subir un archivo a un servidor web.
 *
 * @param file El archivo a subir.
 * @return <tt>true</tt> en caso de éxito,
 *         <tt>false</tt> en caso de fallo.
 * @author David Flanagan
 */
```

¿Qué es una palabra reservada?
Una palabra reservada en Java es una palabra que tiene un significado especial dentro del lenguaje y no puede ser utilizada como identificador (nombre de variable, clase, etc.). Ejemplos de palabras reservadas son int, for, if, while, entre otras. Estas palabras son parte de la sintaxis del lenguaje y se utilizan para definir la estructura y el flujo del programa.

¿Qué es un identificador?
Un identificador es el nombre que se le da a una variable, método, clase o cualquier otro elemento definible por el usuario en un programa Java. Los identificadores deben comenzar con una letra (A-Z o a-z), un signo de dólar ($), o un guión bajo (_), seguido de letras, dígitos, signos de dólar o guiones bajos. No pueden ser palabras reservadas.

4) ¿Qué es un literal?

Un literal es una representación fija de un valor en el código fuente. Los literales se utilizan para asignar valores a variables. Ejemplos de literales incluyen números (42), caracteres ('A'), cadenas de texto ("Hello, World!"), y valores booleanos (true y false). Los literales representan directamente los valores que se usan en el programa.

5) ¿Qué son los datos primitivos en java?

En Java, los datos primitivos son los tipos de datos más básicos y simples. No son objetos y representan valores simples. Los tipos de datos primitivos en Java se dividen en cuatro categorías principales: numéricos enteros, numéricos de punto flotante, caracteres y booleanos.

- Enteros: byte, short, int, long (diferentes tamaños y rangos para números enteros).
- Decimales: float, double (diferentes tamaños y precisiones para números con decimales).
- Booleano: boolean (verdadero o falso).
- Caracteres: char (un solo carácter).

6) En Java pueden evidenciarse diversos tipos de operadores, según la funcionalidad que estos guarden. Así pues, se denotan los siguientes:

• Operadores aritméticos
• Operadores de Asignación
• Operadores relacionales
• Operadores lógicos
• Operadores unarios

Operadores aritméticos
suma ( + ) 2 + 2 = 4
resta ( - ) 5 - 3 = 2
multiplicación ( * ) 2*3 = 6
división ( / ) 15 / 3 = 5
módulo ( % ) 2 % 2 = 0

Operadores Lógicos

AND ( && )  true AND false = false
OR  ( | )  true OR false = true
Not  ( ! )  !true = false

7)

Operador de Incremento y Decremento

i ++  incremento : Estos operadores se ven presentes en ciclos, y guardan la función de incrementar en una unidad un valor previamente establecido

```
int x = 0;
while (x <= 20) {
System.out.println(x);
    x++;
}
```

---

i – decremento: Estos operadores se ven presentes en ciclos, y guardan la función de decrementar en una unidad un valor previamente establecido.

```
int x = 20;
while (x <= 5) {
System.out.println(x);
    x--;
}
```

8) Operadores de comparación

Igualdad  ==
Desigualdad  !==
Mayor que  >
Menor que  <
Mayor o igual que  >=
Menor o igual que  <=

9) El punto nueve está en el archivo de intellij en el nombre **ternario**

10) El punto diez está en el archivo de intellij en el nombre **if_else**

11) El Switch es una herramienta que resulta útil al momento de trabajar con múltiples decisiones que no desean ser anidadas bajo sentencias if/else. Esta versatilidad se ve producto de diversos casos que pueden ser determinados por el usuario en favor de la variable a switchear, permitiendo así dimensionar más de una alternativa y definir valores por default o defecto, en caso de errores o datos no válidos.

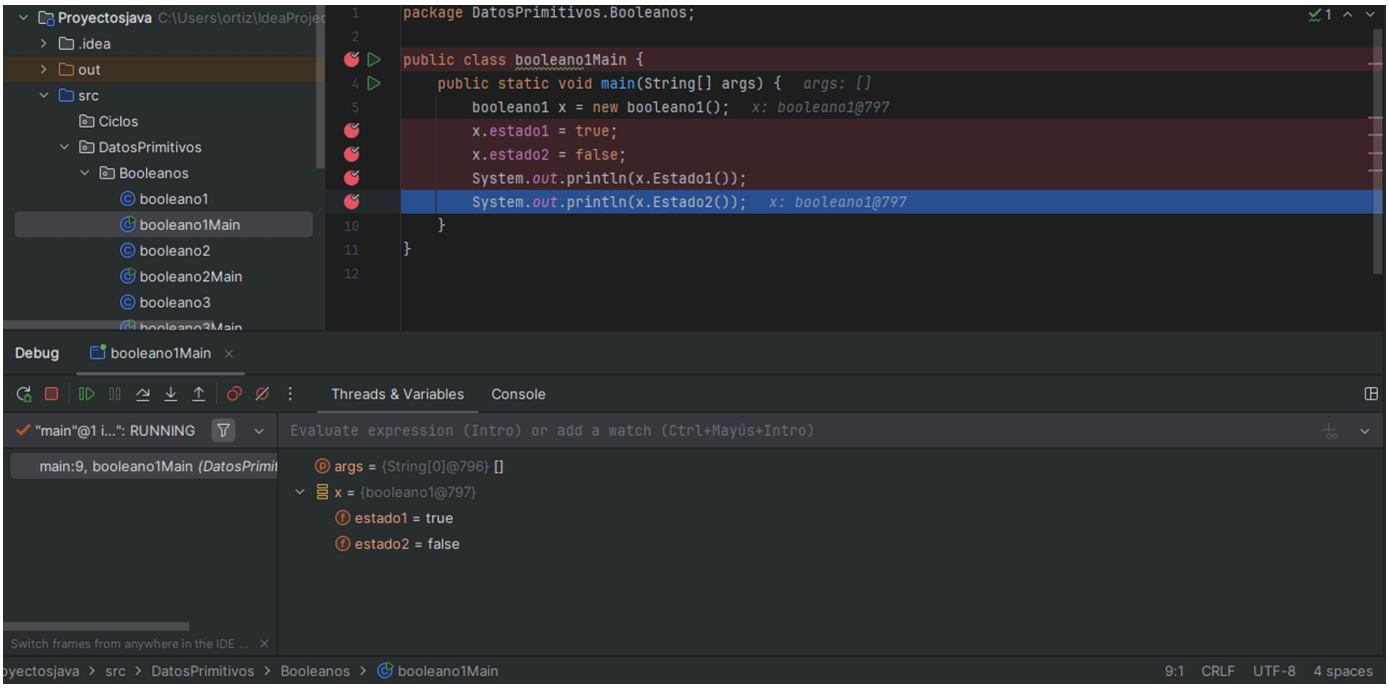
12) El punto doce está en el archivo de intellij en el nombre **ciclos**

13) El punto doce está en el archivo de intellij en el nombre **for_each**

14) Break es una sentencia cuya función principal se constituye en dar un cierre abrupto a la iteración en un ciclo, por ejemplo; mismo en el cual, al llegar a un valor de condición específico, la iteración culminará, dando fin al ciclo. Del mismo modo, la sentencia continua, permite iterar por completo un ciclo, de acuerdo al parámetro que se haya establecido para éste; no obstante, omite un valor condicional propio del siglo, dando lugar a la continuidad en la cadena de iteraciones hasta el último dato definido.

El punto catorce está en el archivo de intellij en el nombre **continueBreak**

15.




**16) Link del repositorio:** https://github.com/JohanM17/Seguimiento-1---Corte-1