CLASE 1

Ing. SILVESTRE ALEJANDRO
INFORMÁTICA III
IUA - 2023

Introducción personal

¿Quién soy?

¿Qué hago?

¿Qué me gusta hacer?

¿Cómo me gusta hacer?

¿Cómo me gusta trabajar?

Objetivo de la materia

1. Desarrollar e implementar en JAVA.
2. Comprender los prácticos.
3. “Aprender a aprender”.
4. Autogestionarse.
5. Trabajar en equipo.
6. Planificar y cumplir con tiempos de entrega.

Materia

1. Primer parcial teórico (con Luis Toledo).
2. Segundo parcial proyecto integrador (con Alejandro Silvestre)
3. Examen final (con Luis Toledo).

JAVA

Introducción

1. Historia del lenguaje Java y su creador, James Gosling.
2. Java como lenguaje de programación versátil y ampliamente utilizado en diversas áreas.
3. ¿Por qué es importante aprender JAVA?

Características de Java

1. Características claves de Java:
    a. Portable.
    b. Orientado a objetos.
    c. Robusto.
    d. Seguro.
    e. Alto rendimiento.
2. Ventajas de utilizar Java en proyectos de desarrollo de software:
    a. Documentación.
    b. Librerías.

Ambiente de Desarrollo

1. Requisitos para configurar el entorno de desarrollo de Java.
    a. JDK v17
2. Diferentes IDEs
    a. Eclipse.
    b. IntelliJ.
    c. NetBeans.
    d. VisualStudio.

Visual Studio

Sintaxis Básica

1. Introducción a la estructura de un programa Java.
    a. Main.
2. Declaración de:
    a. Una clase.
    b. Un método.
    c. Variables.
3. IMPORTANTE:
    a. Puntos y coma.
    b. Llaves
    c. Comentarios.

Clase

Las clases representan los prototipos de los objetos que tenemos en el mundo real.

Es decir, es una generalización de un conjunto de objetos. A su vez los objetos serán instancias de una determinada clase

Clase

```
class Escritorio {
    private Long base;
    private Long altura;
    private Long ancho;
    private String tipoMaterial;

    public Triangulo(Long base, Long altura, Long ancho, String tipoMaterial) {
        this.base = base;
        this.altura = altura;
        this.ancho = ancho;
        this.tipoMaterial = tipoMaterial;
    }
}
```

Objeto

Es un elemento de software que intenta representar un objeto del mundo real.

Un objeto tiene su estado (o estados) y su comportamiento. Esto se modela mediante propiedades (o variables) y métodos. Incluso un objeto puede contener a su vez a otro tipo de objeto.

“Es la instancia de una clase”

Método (o función)

Bloque de código que realiza una tarea específica. Nos sirve para encapsular tareas y reutilizar.

```
[modificador] tipo_de_retorno nombreDeLaFuncion([parámetros]) {
    // Cuerpo de la función (bloque de código)
    // Aquí va la lógica y las operaciones que realizará la función
    [return valor_de_retorno]; // Opcional, si la función devuelve un valor
}
```

Método (o función)

- **[modificador]:** Es un modificador de acceso que define la visibilidad de la función (public, private, protected, o sin modificador para acceso por defecto).
- **tipo_de_retorno:** Es el tipo de dato que la función devuelve. Puede ser un tipo primitivo, un objeto o void si la función no devuelve nada.
- **nombreDeLaFuncion:** Es el nombre con el que se invocará la función en el código.
- **[parámetros]:** Son los valores que la función puede recibir como entrada. Pueden ser cero o más parámetros separados por comas.
- **return valor_de_retorno:** Es la instrucción return que se utiliza para devolver un valor desde la función. Solo se utiliza si la función tiene un tipo de retorno distinto de void.

Comentarios

- Comentario de una línea.
  - //Mi primer comentario
- Comentario de multiples líneas.
  - /*
  - * Multiple comentario.
  - */
- Comentario de documentación:
  - /**
  - * Multiple comentario con documentación.
  - */

Tipos de datos

Tipos de datos primitivos:

- byte: 8 bits, permite almacenar valores enteros en el rango de -128 a 127.
- short: 16 bits, puede almacenar valores enteros en el rango de -32,768 a 32,767.
- int: 32 bits, puede almacenar valores enteros en el rango de -2^31 a 2^31 - 1.
- long: 64 bits, puede almacenar valores enteros en el rango de -2^63 a 2^63 - 1.
- float: 32 bits, se utiliza para representar números de punto flotante con precisión simple.
- double: 64 bits, se utiliza para representar números de punto flotante con precisión doble.
- char: 16 bits, se utiliza para representar caracteres individuales en Unicode.
- boolean: puede tener solo dos valores, "true" o "false", representa un valor de verdad.

Tipos de datos

Tipos de datos de referencia (objetos):

- String: representa una secuencia de caracteres y es una clase en Java.
- Long, Integer, Float, Boolean, Double, etc.
- Arrays: se utilizan para almacenar una colección de elementos del mismo tipo.
- Clases personalizadas: los usuarios pueden definir sus propios tipos de datos creando clases.

Operadores Aritméticos

- +: Suma dos operandos.
- -: Resta el segundo operando del primero.
- *: Multiplica dos operandos.
- /: Divide el primer operando entre el segundo.
- %: Devuelve el resto de la división entera del primer operando por el segundo.

Operadores de asignación

- =: Asigna el valor del operando de la derecha al operando de la izquierda.
- +=: Suma el valor del operando de la derecha al operando de la izquierda y asigna el resultado a la variable de la izquierda.
- -=: Resta el valor del operando de la derecha al operando de la izquierda y asigna el resultado a la variable de la izquierda.
- *=: Multiplica el valor del operando de la derecha al operando de la izquierda y asigna el resultado a la variable de la izquierda.
- /=: Divide el valor del operando de la izquierda entre el operando de la derecha y asigna el resultado a la variable de la izquierda.
- %=: Realiza la división entera del operando de la izquierda entre el operando de la derecha y asigna el resto como resultado a la variable de la izquierda.

Operadores de comparación

- ==: Comprueba si dos operandos son iguales.
- !=: Comprueba si dos operandos son diferentes.
- <: Comprueba si el operando de la izquierda es menor que el operando de la derecha.
- >: Comprueba si el operando de la izquierda es mayor que el operando de la derecha.
- <=: Comprueba si el operando de la izquierda es menor o igual que el operando de la derecha.
- >=: Comprueba si el operando de la izquierda es mayor o igual que el operando de la derecha.

Operadores lógicos

- && (AND lógico): Devuelve true si ambos operandos son true.
- || (OR lógico): Devuelve true si al menos uno de los operandos es true.
- ! (NOT lógico): Invierte el valor de verdad del operando.

Operadores de incremento y decremento

- ++: Incrementa en 1 el valor de la variable.
- --: Decrementa en 1 el valor de la variable.

Operadores ternarios

- condición ? expresión_verdadera : expresión_falsa: Evalúa la condición y devuelve la expresión_verdadera si es true, o la expresión_falsa si es false.

Estructuras de control condicionales

- **if:** Permite ejecutar un bloque de código si se cumple una condición especificada.
- **else if:** Permite agregar condiciones adicionales después del if para evaluar distintas posibilidades.
- **switch:** Permite evaluar múltiples casos y ejecutar un bloque de código dependiendo del valor de una variable.

Estructuras de control iterativas (bucles)

- **while:** Permite ejecutar un bloque de código mientras se cumpla una condición.
- **do-while:** Similar al bucle while, pero garantiza que el bloque de código se ejecutará al menos una vez antes de evaluar la condición.
- **for:** Se utiliza para realizar un bucle que se ejecuta un número específico de veces.
- **enhanced for (for-each):** Se utiliza para recorrer colecciones de elementos, como arrays o listas.

¿DUDAS?

BREAK

MANOS A LA OBRA

CHECK ENTORNO DESARROLLO

Entorno desarrollo

- ¿Computadora personal o del laboratorio?
- Definir IDE a utilizar.
- Versión de JAVA instalado.

Actividad

“Crear una calculadora simple en java.”

Importante:

a. Solo se debe poder hacer una operación (suma, resta, multiplicación y división de dos números)
b. Todas las funciones deben estar encapsuladas en métodos.
c. Se debe capturar el input de teclado para ingresar los números para los números y la operaciones.
d. Enviar el proyecto por email a asilvestre@iua.edu.ar