

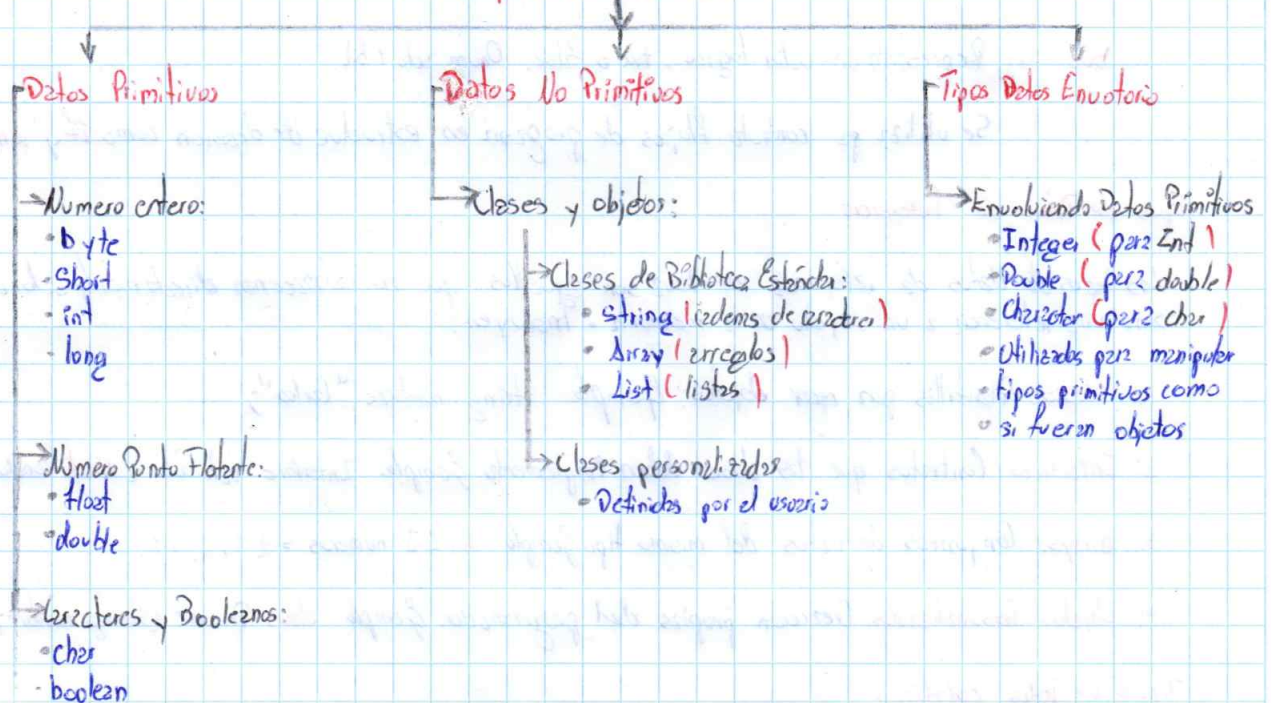
UNIVERSIDAD DE LAS FUERZAS ARMADAS

Nombre: Peltoxi Tipón Juan Fabricio

NRC: 1323

Java es un lenguaje de programación multiplataforma; eso quiere decir que las variables deben declararse especificando su tipo de dato antes de ser utilizados, lo que ayuda a prevenir errores y garantiza un manejo correcto de la información.

Tipos de Datos en Java



Tipos de Datos Primitivos y Referenciados

Números Enteros

- Representan valores numéricos sin decimales.
- Se diferencian por la cantidad de memoria que ocupan y el rango de valores que pueden almacenar.
 - **byte**. Ocupa 8 bits. Util para ahorrar memoria en grandes arreglos. Rango: -128 a 127
 - **short**. Ocupa 16 bits. Util en sistemas con limitaciones de memoria. Rango: -32.768 a 32.767
 - **int**. Ocupa 32 bits. Es el tipo entero más utilizado. Rango: -2.147.483.648 a 2.147.483.647
 - **long**. Ocupa 64 bits. Se usa cuando se necesitan números enteros muy grandes.

Número en punto flotante.

Representan valores numéricos con decimales. Se utilizan para cálculos que requieren mayor precisión.

float. Ocupa 32 bits. Tiene menos precisión, pero es útil para aplicaciones con limitaciones de memoria.

double. Ocupa 64 bits. Es el más preciso para valores decimales y el más utilizado.

Caracteres

char. Representan un solo carácter Unicode. Ocupa 16 bits.

Útil para almacenar letras, símbolos o números que no se utilizarán en los cálculos.

Booleanos

boolean. Representa un valor lógico, true o false. Ocupa solo 1 bit.

Se utilizan para controlar flujos de programación en estructuras de decisión como **if** y **while**.

Tipos de Datos de Referencia

Los tipos de datos de referencia en Java son aquellos que no almacenan directamente valores sino una referencia a un objeto en la memoria. Incluyen:

- 1: **Clases:** Plantillas para crear objetos. Ejemplo `String nombre = "Carlos";`
- 2: **Interfaces:** Contratos que las clases deben implementar. Ejemplo `Interface Vehiculo { void conducir(); }`
- 3: **Arrays:** Conjuntos de datos del mismo tipo. Ejemplo `int[] numeros = { 1, 2, 3 };`
- 4: **Objetos personalizados:** Creación propia del programador. Ejemplo `class Persona { String nombre; int edad; }`

Tipos de Datos Estáticos

Tipos primitivos

Representan valores básicos y no son objetos.

int (números enteros)
float (números decimales)
char (caracteres)
boolean (valores lógicos)

Ejemplo

`int edad = 25;` // Tipo primitivo
`String nombre = "Juan";` // Tipo referencia

Tipos de referencia

String (cadena de texto)
Array (arreglo)

Datos Dinámicos

Clases como object:

Todas las clases en Java heredan de Object - por lo que puedes usar para almacenar cualquier tipo de datos

```
Object dato = "Texto";  
dato = 42;
```

Colectores dinámicos (de la biblioteca estándar)

Estructuras como `ArrayList`, `HashMap` o `LinkedList` permiten manejar datos dinámicos en tiempo de ejecución

```
ArrayList<Object> lista = new ArrayList<>();  
lista.add("Texto");
```

Polimorfismo:

Utiliza herencia, puedes manejar objetos de diferentes clases bajo un mismo tipo base

```
class Animal {}  
class Perro extends Animal {}  
class Gato extends Animal {}
```

1. ¿Qué es el paradigma de la programación orientada a objetos?

Es un enfoque de programación basado en la creación y manipulación de objetos, que son instancias de clases. Este paradigma organiza el código en estructuras de llamadas clases, que contienen tanto datos (atributos) como funciones (métodos) que trabajan con esos datos. Los principales principios de la programación orientada a objetos son:

- **Encapsulación**

Protege los datos al hacerlos accesibles solo a través de métodos definidos.

- **Herencia**

Permite que una clase reutilice propiedades y comportamientos de otra clase.

- **Polimorfismo:**

Capacidad de un método para comportarse de diferentes maneras dependiendo del objetivo que lo invoque.

- **Abstracción**

Ocultar detalles complejos, mostrando solo lo necesario. Este paradigma mejora la reutilización del código, facilita la resolución de problemas complejos y es ampliamente utilizado en lenguajes Java, Python y C++.

2. ¿Qué es una clase, un objeto, un atributo y un método?

- **Clase.** Es una plantilla que define un conjunto de atributos y métodos. Es como un plano similar para construir objetos con características.

- **Objeto** Es una instancia de una clase, es decir, una representación concreta y única de esa plantilla.

- **Atributo** Representa las características de un objeto como variables que almacenan información específica.

- **Método** Es una función que realiza una acción relacionada con el objeto.

3. ¿Qué es un sistema de control de versionamiento y para qué sirve?

Es una herramienta que gestiona los cambios realizados en un proyecto, especialmente en el código fuente, a lo largo del tiempo. Permite a los desarrolladores rastrear las modificaciones, colaborar en equipo y revertir cambios si es necesario.

• Funciones principales

- Almacena un historial completo de las versiones del proyecto
- Identifique quién realizó cambios y cuándo
- Funciona cambios realizados por diferentes desarrolladores
- Facilita la creación de diferentes ramas para probar nuevas funcionalidades sin afectar el código principal