

Universidad de las Fuerzas Armadas - ESPE

- Nombre: Claudio Dahik Villalba Paladines

Materia: POO NRC: 1325

Actividad: Tarea 1

Tipos de datos primitivos

Son los tipos de datos más básicos, generalmente definidos por el lenguaje de programación y no pueden descomponerse en otros tipos. Ocupan un espacio fijo en la memoria y, al usarlos, se manipulan directamente.

- byte: -128 a 127
- short: -32768 a 32767
- int: -2^{31} a $2^{31} - 1$.
- long: -2^{63} a $2^{63} - 1$. Se puede usar con el sufijo L.
- float: 6-7 cifras decimales. Requiere F o f.
- char: Representa un carácter único entre comillas simples.
- boolean: Solo tiene dos valores posibles: true o false,

Ejemplo:

```
public class TiposPrimitivos {  
    public static void main (String [] args) {  
        byte = 25;  
        short = 2024;  
        int población = 100000;  
        long distancia = 9876543210L;  
        float precio = 19.99f;  
        char inicial = 'A';  
        boolean esMayor = true;  
    }  
}
```

Tipos de datos Referenciados

Son tipos de datos más complejos que se refieren a una ubicación de memoria donde se almacenan los datos. No se manipulan directamente, sino a través de referencias o punteros. Estos tipos suelen permitir estructuras de datos más complejas y flexibles. Ejemplos de tipos de datos referenciados incluyen:

Clases: Son la base de la programación orientada a objetos. Los objetos de clases personalizadas se almacenan como referencias.

```
String nombre = "Andrés";  
Persona persona = new Persona();
```

Arreglos: Son estructuras de datos que almacenan múltiples elementos del mismo tipo. Se accede a ellos mediante índices.

```
int [] numeros = {1,2,3,4};
```

Interfaces: Son tipos de datos abstractos que definen un comportamiento (métodos), pero no lo implementan.

Clases Predefinidas: Java incluye muchas clases listas para usar, como String, Integer, Double, ArrayList, etc.

```
String saludo = "Hola";  
ArrayList<String> lista = new ArrayList<>();
```

Objetos de Envoltorio (Wrappers): Proporcionan una representación como objeto para los tipos primitivos (por ejemplo, Integer para int, Double para double).

```
Integer numero = 10;
```

Datos Estáticos

Los datos estáticos están asociados a la clase en lugar de instancias específicas de la clase. Estos datos existen durante toda la vida del programa y son compartidos por todas las instancias de la clase.

- Se declaran con la palabra clave static.
- Se inicializan una sola vez al cargar la clase en memoria.
- Son compartidos por todas las instancias de la clase, lo que significa que cualquier cambio afecta a todas las referencias.
- No necesitan una instancia de la clase para ser accesibles.

Datos Dinámicos

Los datos dinámicos están asociados a las instancias de una clase. Cada objeto tiene su propia copia de estos datos, y su tiempo de vida está determinado por el ciclo de vida de objeto.

- Se crean y destruyen con la instancia del objeto.
- Cada instancia de la clase tiene su propia copia.
- Se acceden a través de las referencias de los objetos.

1) ¿Qué es la programación orientada a objetos?

Es un enfoque de programación que organiza el diseño de software en torno a objetos. Estos objetos representan entidades del mundo real o abstracto y encapsulan tanto datos (atributos) como el comportamiento (métodos) relacionado con esos datos.

1. Abstracción:

Permite enfocarse en los aspectos esenciales de un objeto, ignorando los detalles irrelevantes para el contexto. Se logra mediante la definición de clases que representan conceptos del mundo real.

2. Encapsulamiento:

Consiste en ocultar los detalles internos de los objetos y exponer solo lo necesario a través de métodos públicos. Esto permite proteger los datos y controlar el acceso a ellos.

3. Herencia:

Es el mecanismo que permite que una clase (subclase) herede atributos y comportamientos de otra clase (superclase), promoviendo la reutilización de código.

4. Polimorfismo:

Permite que diferentes clases implementen el mismo método de manera específica. Esto facilita que el mismo mensaje desencadene comportamientos distintos según el objeto que lo reciba.

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

Clase: Es un modelo o plantilla que define las propiedades (atributos) y comportamientos (métodos) de un tipo de objetos.

Atributo: Es una variable dentro de una clase que define las características de un objeto.

Ejemplo: nombre y edad en la clase Persona son atributos.

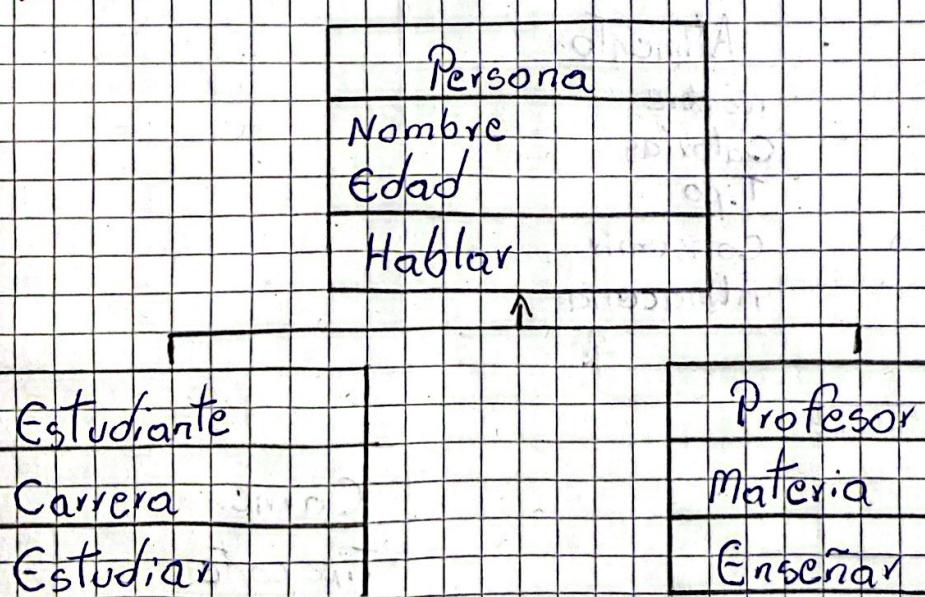
Método: Es una función dentro de una clase que define las acciones o comportamientos que un objeto puede realizar.

Ejemplo: saludar () en la clase Persona es un método.

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

Un sistema de versionamiento es una herramienta que permite gestionar los cambios realizados en el código fuente de un proyecto a lo largo del tiempo. Su objetivo principal es facilitar el control de versiones, el trabajo en equipo y el seguimiento de la evolución del proyecto.

4) Realizar 3 UML



Electrodoméstico
marca
modelo
Encender
Apagar

Refrigerador
Capacidad
Enfriar

Microondas
Potencia
Calentar

Alimento.
Nombre
Calorías
Tipo
Consumir
Almacenar

Fruta
Sabor
Preparar jugo

Carne
Tipo corte
Cocinar