

La naturaleza es el tesoro más preciado.



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA



Nombre: Sebastian Clavijo  
Tema: Consulta  
NRC: 1323

Fecha: 17/11/2024

Deber N°: 1

Docente: Luis Saramillo

## Consulta 1

### 1. Tipos de datos primitivos

Son los tipos de datos más básicos en Java.

No tienen métodos asociados. Ejemplos: int, char, double, boolean, etc.

Ejemplo: Crear e imprimir un valor de tipo double

```
public class Main{  
    public static void main(String[] args){  
        double precio = 99.99; // Declaración  
        // y asignación de un valor decimal  
        System.out.println("El precio es: $" +  
            precio); // Imprimir el valor  
    }  
}
```

### 2. Tipos de datos de referencia

Estos tipos hacen referencia a objetos o instancias de clases. Incluye String, Arrays, y clases personalizadas.

Ejemplo: Crear e imprimir un string

```
public class Main {  
    public static void main(String[] args) {  
        String mensaje = "Hola, Mundo"; //  
        Declaracion de un String  
        System.out.println(mensaje); //  
        Imprimir el string  
    }  
}
```

### 3. Tipos de datos Estáticos

En Java, un dato estático pertenece a la clase en lugar de a una instancia específica de la clase. Se usa la palabra clave static.

Ejemplo: Declarar un entero estático

```
public class Main {  
    static int contador = 10; // variable estática  
  
    public static void main (String[] args) {  
        System.out.println("El valor del contador  
        es: " + contador);  
    }  
}
```

La naturaleza es el tesoro más preciado.

#### 4. Tipos de Datos Dinámicos

En Java, aunque los tipos de datos deben ser definidos en tiempo de compilación, se puede crear objetos o estructuras de datos cuyos contenido cambie dinámicamente, como las colecciones (ArrayList, HashMap)

Ejemplo: Usar un ArrayList

```
import java.util.ArrayList;  
  
public class Main {  
    public static void main(String[] args){  
        ArrayList<String> lista = new ArrayList<>();  
        // Declaración de ArrayList  
        lista.add("Uno");  
        lista.add("Dos");  
        lista.add("Tres");  
  
        System.out.println("Contenido de la lista:  
        + lista);  
    }  
}
```

La naturaleza es el tesoro más preciado.



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA



Nombre: Sebastian Clavijo

Fecha: 17/11/2024

Tema:

Deber N°:

NRC: 1323

Docente: Luis Jaramillo

## Preguntas

1. Que es el paradigma de programación orientada a objetos?

Es una forma de organizar y estructurar el código pensado en objetos del mundo real. En lugar de enfocarse solo en funciones o procedimientos, se basa en objetos que tienen propiedades (datos) y comportamientos (funciones). Esto ayuda a que el código sea mas fácil de entender, reutilizar y mantener.

2. Que es una clase, un objeto, un atributo, un método.

Clase

Es un molde o un plano que define las características y un comportamiento de un grupo de objetos. Es una plantilla para crear un objeto

### Objeto

Es una instancia concreta de una clase. Cuando usas la clase para crear algo específico, obtienes un objeto.

### Atributo

Es una propiedad o característica que describe al objeto. Es como una variable dentro del objeto.

### Método

Es una función dentro de una clase que define qué puede hacer un objeto. Representa las acciones o comportamientos del objeto.

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

Es una herramienta que permite gestionar los cambios realizados en el código fuente o en cualquier archivo a lo largo del tiempo. Es ampliamente usado en desarrollo de software y proyectos colaborativos para rastrear, organizar y coordinar modificaciones.

### ¿Para qué sirve?

#### 1. Seguridad

Actúa como una copia de seguridad del proyecto, ya que los cambios se almacenan en un servidor central o repositorio local.

La naturaleza es el tesoro más preciado.

## 2. Historial de cambios

Permite ver quién hizo qué cambios, cuándo, y por qué.

Esto es útil para solucionar problemas o revertir a versiones anteriores si algo falla.

## 3. Control de versiones

Guarda versiones o snapshots de los archivos, lo que permite comparar, fusionar o restaurar versiones anteriores fácilmente.

## 4. Realiza tres UML por persona de dos objetos hijos un objeto padre.

1: Vehículo (Objeto padre) -> Coche y Motocicleta (Objeto hijo)

Vehículo

- marca: String

- modelo: String

- año: int

+ acelerar(): void

+ frenar(): void

/  
Coche

\  
Motocicleta

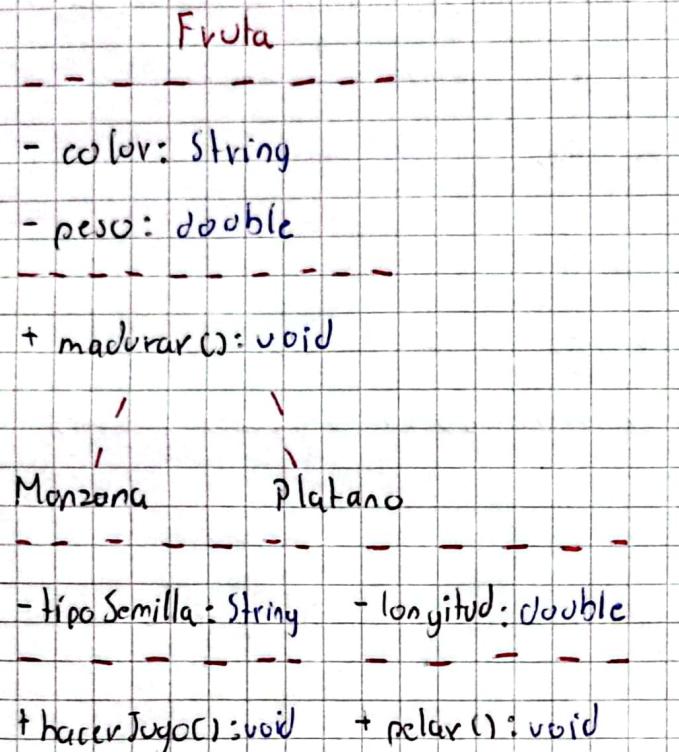
- puertas: int

- tipoCoche: String

+ abrirmaletero(): void

+ hacercaballito(): void

2. Fruta (Objeto padre) -> Manzana y Platano (Objetos hijos)



3. Animal (Objeto padre) -> Pajaro y Pez (Objetos hijos)

