UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE

PROGRAMACIÓN ORIENTADA A OBJETOS

NOMBRE: Alexandra lalaleo.

* Consulta *

Tipos de datos

En POO, los tipos de datos se dividen en: primitivos y referenciados.

Tipo de datos Primitivos.

Son tipo básicos o fundamentales que representan valores simples y no tienen métodos asociados. Estos tipos están definidos directamente por el lenguaje y ocupan un espacio fijo en la memoria.

Ejemplos:

- Enteros (int): Números enteros, como 10, -3.
- flotantes (float, double): Números decimales, como 3,14.
- Caracteres (char): Un único caracter, como 'A'.
- Booleanos (Boolean): Representan verdadero (true) y falso (false)

En Java: (ejemplo)

```
// Ejemplo de datos primitivos.

int edad = 25;                    // Número entero
double altura = 1.75;             // Número decimal
char inicial = 'A';               // Carácter único
boolean esMayorDeEdad = true;     // Valor boleano (verdadero/falso)
```

Tipo de datos Referenciados.

Son tipos que hacen referencia a un objeto o una estructura más compleja en memoria. Estos incluyen: clases, interfaces, arreglos y colecciones.

Ejemplos:

- Clase:

(java)

```
string nombre= " Juan";
```

Aquí, string. es un tipo referenciado, porque nombre hace referencia a un objeto en memoria

- Arreglos:

(java)

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

números es una referencia a un arreglo en memoria.

EJEMPLO DATOS PRIMITIVOS

(Ejemplo en Java)

```
int edad = 25;                    // Número entero
double temperatura = 36, 5;       // Número decimal
char inicial = "A";               // Carácter único.
```

EJEMPLO DATOS Referenciados.

(En Python)

```
nombre = " Juan"                  # 'nombre' es una referencia a un objeto cadena
números =[1; 2, 3]  # 'numeros' es una referencia a una lista.
```

TIPOS DE DATOS ESTÁTICOS.

Son aquellos cuyo tipo y tamaño se determinan en tiempo de compilación, es decir, el lenguaje requiere que se especifique explícitamente el tipo de dato que una variable almacenará, Una vez definido, no puede cambiar durante la ejecución del programa. Estos tipos son comunes en lenguaje de programación estáticamente tipados como C, C++, Java o Rust.

(Ejemplo En Java)

```
public class Ejemplo {
    public static void main (String[] args) {
        int edad = 25;                  // Tipo estático: entero.
        double salario = 5000.50;       // Tipo estático: decimal
        char inicial = 'J';             // Tipo estático: caracter
        boolean activo = true;          // Tipo estático: booleano.

        System.out.println ("Edad: " + edad);
        System.out.println ("Salario: " + salario);
        System.out.println ("Inicial: " + inicial);
        System.out.println ("Activo: " + activo);
    }
}
```

Tipos de Datos Estáticos:

Son aquellos cuyo tipo y tamaño se determinan en tiempo de compilación, es decir, el lenguaje requiere que se especifique explícitamente el tipo de dato que una variable almacenará. Una vez definido, no puede cambiar durante la ejecución del programa. Estos tipos son comunes en lenguajes de programación estáticamente tipados como C, C++, Java o Rust.

Tipos de estáticos Comunes:

1. Tipos PRIMITIVOS

Son los tipos definidos por el lenguaje para representar valores simples.

Ejemplos:

- Enteros (int): `int numero = 10;`
- Flotantes (float, double): `double temperatura = 36.5;`
- Booleanos (boolean): `boolean activo = true;`
- Caracteres (char): `char inicial = 'A';`

2. Tipos COMPUESTOS.

Agrupan varios o permiten representar datos más complejos.

Ejemplos:

- Arreglos:

(java)

```
int[] numeros = {1, 2, 3};  // Arreglo estático de enteros.
```

- Estructuras

C

```
struct Persona {
    char nombre[50];
    int edad;
};
```

5. Tipos Abstractos (Objetos)

Son instancias de clases en lenguajes orientados a objetos como Java o C++.

(Ejemplo en Java)

```
class Persona {
    String nombre;
    int edad;
}

Persona p = new Persona();
```

4. Enumeraciones.

Definen un conjunto fijo de constantes relacionadas.

(Ejemplo en Java)

```
enum Dias { lunes, martes, miercoles, jueves, viernes; }
Dias dia = Dias.Lunes;
```

TIPO DE DATOS DINÁMICOS

Son aquellos cuyo tipo no se declara explícitamente en el código y se determina en tiempo de ejecución.

(Ejemplo en Python)

```
# No se especifica el tipo, el intérprete lo determina automáticamente
dato = 10              # Inicialmente, es un entero
print(type(dato))      # Salida: <class 'int'>

dato = "Hola"          # Ahora, es una cadena de texto
print(type(dato))      # Salida: <class 'str'>
```

¿Que es un paradigma de la Programación Orientada a Objetos?

Un un modelo de la programación que se basa en el concepto de objetos los cuales pueden contener datos y código. En este paradigma, el diseño de software se organiza en torno a objetos, en lugar de usar funciones y lógica.

Algunos de los beneficios de la POO son: Rápido desarrollo, Alta calidad del código, bajo costo en fases de desarrollo.

¿Que es una clase?

Una clase es un modelo que define las caracteristicas y comportamientos de un conjunto de objetos similares. A partir de una clase se pueden crear multiples objetos, cada uno con sus propios valores, pero que comparten las caracteristicas definidas en la clase.

¿Que es un objeto?

Es una instancia de una clase creada con datos especificos. los objetos tienen los comportamientos de su clase. Por ejemplo, si se define la clase "Televisor", los objetos miTelevisor, TuTelevisor y elTelevisorDelVecino son televisores concretos que pertenecen a la clase Televisor.

¿Que es un atributo?

Un atributo es un elemento de datos que define una propiedad o el estado de un objeto o clase. Los atributos son información que se guardan en un objeto en un momento determinado y que se conoce como el estado del objeto.

¿Que es un método?

Es un bloque de código que contiene una serie de instrucciones que permiten realizar acciones con los objetos de una clase:

- Recuperar el estado de un objeto
- Modificar el estado de un objeto
- Enviar mensajes al objeto para que realice una acción
- Crear, construir o instanciar un objeto de una clase.
- Regresar un valor para realizar cálculos.

¿Que es un SCV y para que sirve?

Un sistema de control de versiones (VCS) es una herramienta de software que permite a los equipos de desarrollo:

- Hacer copias de seguridad del código fuente.
- Archivar el código fuente.
- Revisar y modificar el repositorio.
- Restaurar versiones anteriores del código.
- Colaborar e iterar rápidamente el código fuente.
- Evitar conflictos al trabajar en colaboración con otros equipos

3 UML

1.

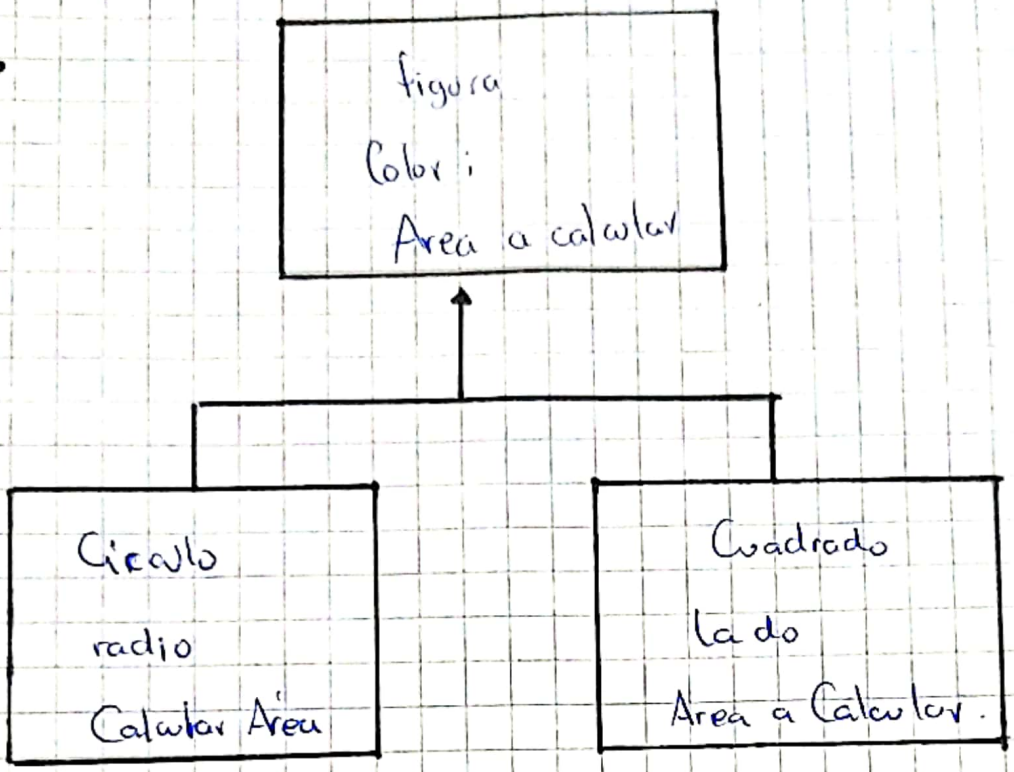
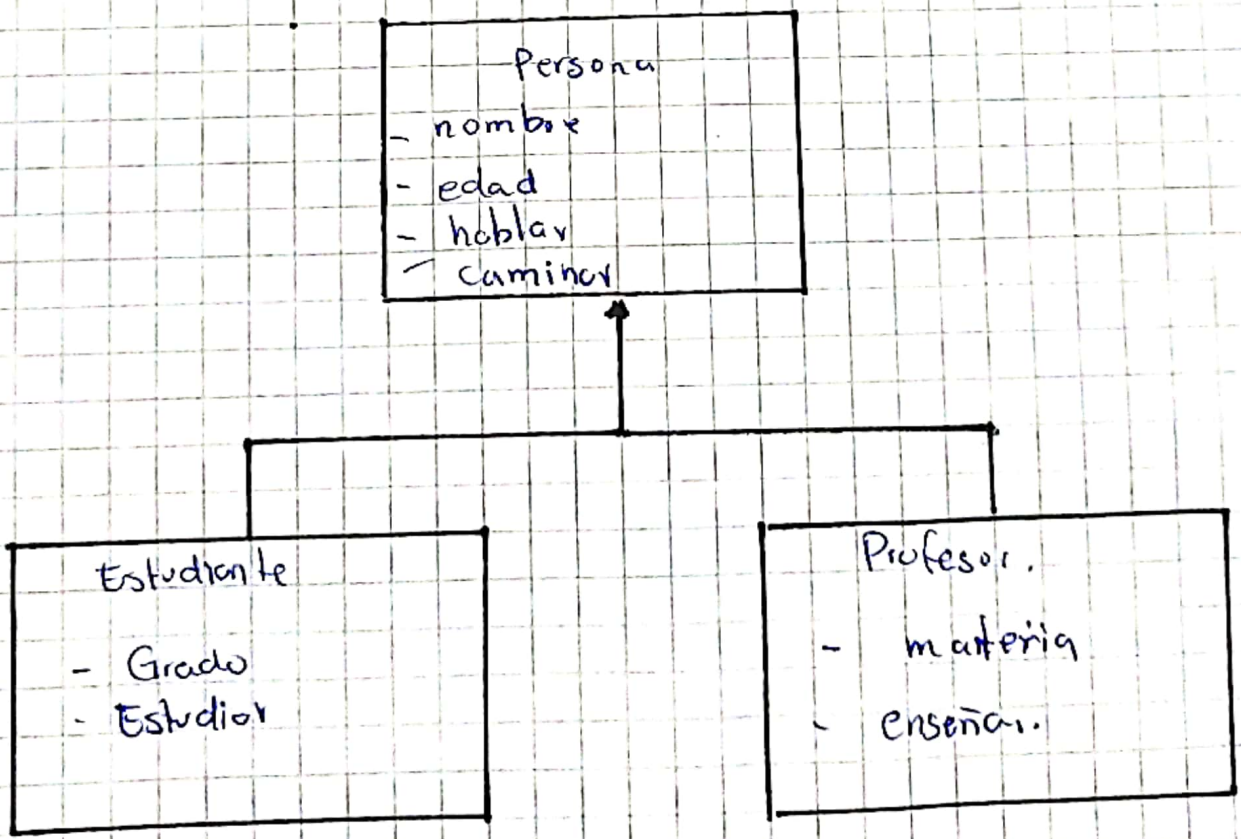


figura
Color:
Area a calcular
Circulo
radio
Calcular Area
Cuadrado
lado
Area a Calcular.

2.


Persona
- nombre
- edad
- hablar
- caminar
Estudiante
- Grado
- Estudiar
Profesor.
- materia
- enseña.