Universidad de las Fuerzas Armadas

Nombre : Bonilla Stefano Mendoza Marquez
Fecha : 14/11/2024
NRC : 1323

DATOS PRIMITIVOS Y DE REFERENCIA

DATOS PRIMITIVOS SON LOS TIPOS DE DATOS BASICOS PROPORCIONADOS DIRECTAMENTE POR EL LENGUAJE DE PROGRAMACION. SON INDIVISIBLES LO QUE SIGNIFICA QUE NO PUEDES DESCOMPONERSE EN OTROS TIPOS.

- ENTEROS (INT): NUMEROS COMO 1, 2, -5.
- PUNTO FLOTANTE (FLOAT) NUMEROS DECIMALES COMO 3.14 O -0.001.
- BOOLEANOS (BOOL) VALORES DE VERDAD, TIPICAMENTE TRUE O FALSE
- CARACTERES (CHAR) UN SOLO CARACTER COMO 'A' O '#'

TIPOS DE REFERENCIA A DIFERENCIA DE LOS DATOS PRIMITIVOS, LOS TIPOS DE REFERENCIA NO CONTIENE LOS DATOS DIRECTAMENTE, SINO UNA DIRECCION DE MEMORIA QUE APUNTA A LOS DATOS ESTE TIPO DE DATOS ES USADO GENERALMENTE POR ESTRUCTURAS

- STRING SECUENCIAS DE CARACTERES
- ARRAYS LISTA DE DATOS DEL MISMO TIPO
- OBJETOS INSTANCIA DE CLASES QUE PUEDEN CONTENER MULTIPLES ATRIBUTOS Y METODOS

DATOS ESTATICOS.

LOS TIPOS DE DATOS ESTATICOS SON FUERTEMENTE TIPADO O LENGUAJES DE PROGRAMACION ESTATICAMENTE TIPADO, ESTOS LENGUAJES DE PROGRAMACION ESTOS LENGUAJES REQUIEREN QUE SE ESPECIFIQUE EL TIPO DE DATO AL DECLARAR UNA VARIABLE Y ESTE TIPO NO PUEDE CAMBIAR DURANTE LA EJECUCION DEL PROGRAMA.

- DECLARACION EN T. DE COMPILACIÓN EL TIPO DE DATO DEBE DECLARARSE ANTES DE USARSE COMO EN JAVA O C++
- SEGURIDAD DE TIPOS PERMITE DETECTAR ERRORES DEL TIPO EN TIEMPO DE COMPILACIÓN
- EFICIENCIA AL ESTAR TIPO DETERMINADO, EL COMPILADOR PUEDE OPTIMIZAR MEJOR EL USO DE MEMORIA Y RENDIMIENTO

.JAVA

```
PUBLIC CLASS OPERACION{
    PUBLIC STATIC INT SUMAR (INT X1, INT X2){
        INT A = X1+X2;
    RETURN S;
    }

    PUBLIC STATIC INT (RESTAR INT X1, INT X2){
        INT R = X1 - X2;
    RETURN R;
    }
}
```

.JAVA

```
INT NUMERO = 10
```

SE DECLARA QUE "NUMERO" ES UN ENTERO Y NO PODRA CAMBIAR A OTRO TIPO.

DATOS DINAMICOS

LOS TIPOS DE DATOS DINAMICOS ESTAN ASOCIADOS A LENGUAJES DE PROGRAMACION DINAMICO TIPADOS DONDE EL TIPO DE UNA VARIABLE PUEDE ALMACENAR DISTINTOS TIPOS DE DATOS EN DIFERENTES MOMENTOS

- FLEXIBILIDAD: NO ES NECESARIO ESPECIFICAR EL TIPO DE DATO AL DECLARAR UNA VARIABLE; EL ALGUNO LO INFIERE EL TIEMPO DE EJECUCIÓN.

- MAYOR ADAPTABILIDAD: DEBIDO EN LENGUAJES DONDE LOS DATOS PUEDEN VARIAR AMPLIAR Y EL CAMBIO DE ESTRUCTURA ES FRECUENTE

- MENOR SEGURIDAD DE TIPOS: COMO EL TIPO NO SE VERIFICA HASTA TIEMPO DE EJECUCIÓN, ES MÁS FÁCIL COMETER ERRORES RELACIONADOS CON EL TIPO DE DATOS.

```
PUBLIC CLASS EJEMPLO DINAMICO {
    PUBLIC STATIC VOID MAIN (STRING[] ARGS){

        INT NUMERO = 5
        SYSTEM.OUT.PRINT("NUMERO ES:" + NUMERO);

        SYSTEM.OUT.PRINTLN("TEXTO (TIPO E.): " + TEXTO)

    }
}
```

PREGUNTAS.

1.- Que es paradigma de la programación orientada a objetos?

La POO es un enfoque de programación que organiza el software en "objetos" los cuales representan entidades del mundo real y contienen datos y comportamientos.

Este paradigma facilita la reutilización del código, la organización, y el mantenimiento promoviendo conceptos como encapsulamiento, herencia y polimorfismo.

2.- Que es una clase, un objeto, un atributo y un método?

- Clase: es un molde o plantilla que define características y comportamientos de un tipo de objeto. Por ejemplo una clase "coche" podría definir atributos como "color" y "marca" y métodos como "acelerar".

- Objeto: es una instancia de una clase, es decir una entidad concreta creada a partir de una clase. Si la clase es "coche" un objeto sería "mi coche" con valores específicos.

- Atributo: son las propiedades o características de una clase u objeto, en la clase "coche" un atributo podría ser "color"

- Método: son las funciones propias que puede realizar un objeto. En el ejemplo del "coche", un método sería "arrancar".

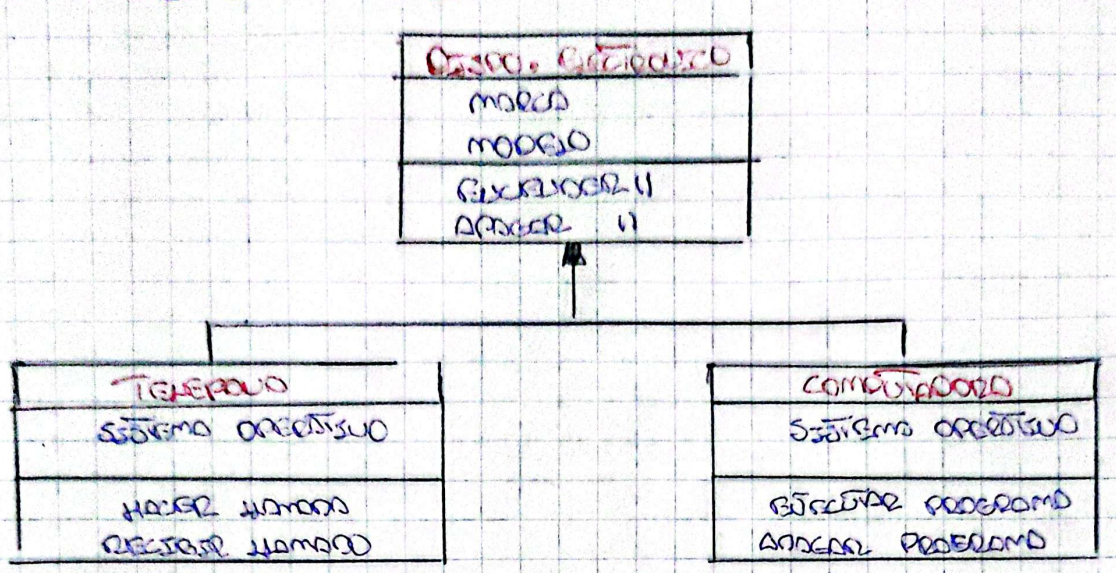
3.- Que es un sistema de control de versiones y para que sirve?

Es una herramienta que permite controlar y registrar cambios en el código fuente de un proyecto a lo largo del tiempo. Permite el trabajo en equipo, permite recuperar versiones anteriores y mantiene un historial de modificaciones.

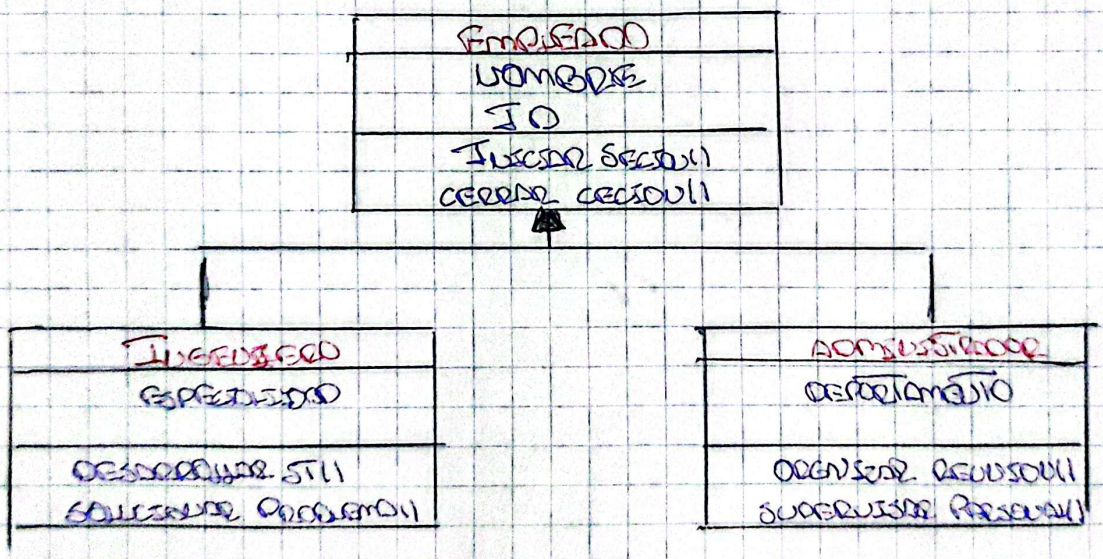
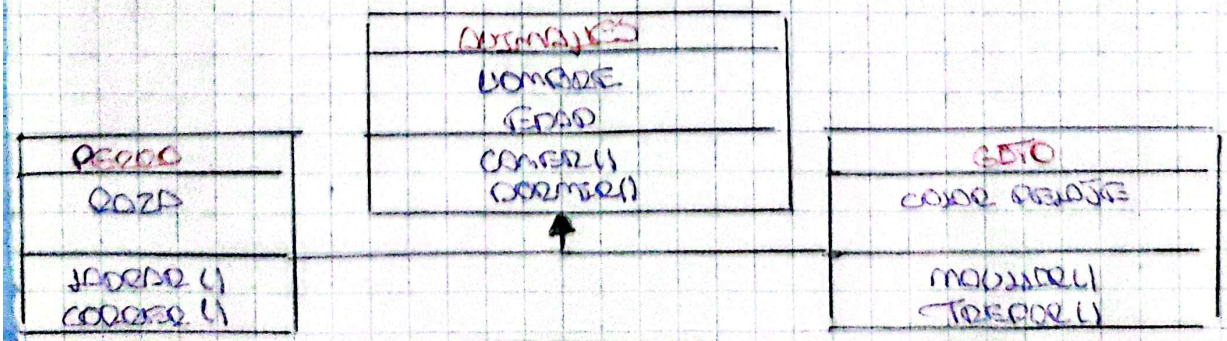
El más popular es Git

4. Hacer 3 uml de 2 clases hijas y una clase padre

Ejemplo 1.



Ejemplo 2.



Ejemplo 3