

Deber #1

Nombre: Muminacho Alexis
APC:

Tipos de datos primitivos y referencia

► **primitivos**: es una categoría básica de datos que permiten almacenar valores simples directamente en la memoria por la cual poseen ocho tipos de datos primitivos.

1. **Tipo primitivo char**: se utiliza para representar caracteres individual no palabras completas, por ejemplo la palabra "caracter" estaría compuesta por ocho caracteres 'o' = 'c' 'a' 'r' 'a' 'c' 't' 'e' 'r'
2. **Tipo boolean** se utiliza para representar valores lógicos: true "verdadero" o false "falso", las variables de este tipo son específicamente útiles para evaluar condiciones y determinar el resultado de operaciones lógicas.
3. **Tipo byte** estos se representan pequeños números enteros (8 bit) puede contener números entre 128-127.
4. **Tipo short** Utilizado para representar números enteros más grandes que byte "aunque no demasiado".
5. **Tipo int** Es uno de los más utilizados ya que suele ser la opción predeterminada al trabajar con números enteros.
6. **Tipo long** Es usado para representar números enteros realmente grandes, de orden -2^{63} a 2^{63} hasta 2^{64} .
7. **Tipo float**. Es útil cuando queremos representar números decimales.
8. **Tipo double**. Es el tipo de dato numérico más grande, con este podemos casi representar casi cualquier número.

Tipos de datos por referencia

- **Tipo String** Es una clase utilizada para manejar cadenas de caracteres en aplicaciones, lo que permite almacenar y trabajar con el texto.

► **Tipo Arrays**: Es una estructura de datos diseñada para almacenar múltiples elementos del mismo tipo en una única variable.

► **Tipo clases**: Son tipos de datos que definen un conjunto de variables y métodos que representan un objeto del mundo real.

- Tipo de dato estático

Son aquellas en las que el tamaño ocupado en memoria se define antes de que el programa se ejecute y no puede modificarse dicho tamaño, su principal característica es que ocupan una sola casilla de memoria, dentro de este grupo de datos se encuentra:

- 1. Enteros
- 2. Reales
- 3. Caracteres

- 4. Booleano
- 5. Enumerado
- 6. Subrangos

- Tipo de datos dinámicos

Es aquella estructura en la que el tamaño asignado en memoria se determina antes de la ejecución del programa y no es posible modificarlo mientras el programa este en funcionamiento.

Deber

Nombre Muirminacho Alexis

NRC

Fecha

1. Qué es un paradigma de programación orientada a objetos

Un paradigma de programación orientada a objeto es una forma particular de concebir y organizar el desarrollo de programas basada en principios y fundamentos específicos para solucionar problemas computacionales. Este enfoque determina como se ordena y estructuran las instrucciones y estructuras dentro de un programa además de guiar el diseño y la manera de enfrentar los problemas

2. Qué es una clase, objeto, atributo, metodo

- **Clase:** Una clase es una plantilla para definir objetos con características similares, de la cual se crean instancias específicas llamadas objetos y son fundamentales para simplificar el programa
- **Objeto:** Un Objeto es una representación de algo concreto con propiedades "atributos" y metodos "funciones" que permiten modificar estas propiedades. Los objetos se crean a partir de una clase, que actua como modelo general y contienen unicamente la información necesaria
- **Atributo:** Los atributos son todas aquellas características que engloban a la abstracción de un objeto, generalmente se definen como privados para limitar el acceso externo y aumentar la seguridad
- **Metodos:** Son funciones asociadas a una clase que establecen como se comportan sus objetos y facilitan su interacción, pueden recibir parametros y devolver valores.

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

Un sistema de control de versionamiento es una herramienta que permite seguir y administrar las modificaciones en el código fuente de un proyecto, su proposito es ayudar a los equipos de desarrollo a colaborar de forma eficaz, al proporcionar un Historial de cambios, la opción de deshacer modificaciones cuando sea necesario y facilitar el trabajo en conjunto, específicamente en entornos de desarrollo rápidos y en metodologías

4) Ejemplo de 3 UML

