

REGISTROS (struct) en c++:

ESTRUCUTURA DE DATOS:

Índice

Introducción a las estructuras en C++

Creación de una estructura

Inicialización y asignación de datos

Modificación de datos

Eliminación de registros

Ejemplo completo de uso de estructuras

Consideraciones adicionales

Introducción a las estructuras en C++

Una estructura en C++ es *una colección de variables de diferentes tipos, agrupadas bajo un mismo nombre*. Las estructuras permiten organizar datos relacionados de manera más lógica.

Definición de una estructura

Las estructuras se definen utilizando la palabra clave struct seguida del nombre de la estructura y un bloque de miembros entre llaves {}.

Creación de una estructura

Aquí te muestro cómo definir una estructura básica:

```
#include <iostream>
#include <string>

// Definición de una estructura llamada Persona
struct Persona {
    std::string nombre;
    int edad;
    std::string direccion;
};
```

En este ejemplo, la estructura Persona tiene tres miembros: nombre, edad, y direccion.

Inicialización y asignación de datos

Inicialización de una estructura

Puedes inicializar una estructura al momento de declararla de varias maneras:

Inicialización directa:

```
Persona persona1 = {"Juan Pérez", 30, "Calle Falsa 123"};
```

Inicialización con el operador =:

```
Persona persona2;
persona2.nombre = "Ana Gómez";
persona2.edad = 25;
persona2.direccion = "Avenida Siempre Viva 742";
```

Uso de constructor (si decides definir uno dentro de la estructura):

```
struct Persona {
    std::string nombre;
    int edad;
    std::string direccion;

    // Constructor
    Persona(const std::string& n, int e, const std::string& d) : nombre(n), edad(e),
direccion(d) {}
};
```

```
Persona persona3("Carlos López", 40, "Boulevard de los Sueños Rotos");
```

Asignación de datos

Los datos en los miembros de una estructura pueden asignarse directamente utilizando el operador `.` (punto):

```
Persona persona4;  
persona4.nombre = "Laura Martínez";  
persona4.edad = 28;  
persona4.direccion = "Plaza Mayor 12";
```

Modificación de datos

Para modificar los datos en una estructura, puedes simplemente reasignar los valores a los miembros de la estructura:

```
persona4.edad = 29; // Cambiar edad  
persona4.direccion = "Plaza Mayor 24"; // Cambiar dirección
```

Eliminación de registros

En C++, las estructuras son simples tipos de datos que no requieren una gestión explícita de memoria (como liberación de memoria). No hay un concepto de "eliminación" en el sentido de liberación de memoria como en lenguajes que manejan dinámicamente los registros. Cuando una estructura se sale del ámbito, su memoria es liberada automáticamente.

Ejemplo completo de uso de estructuras

Aquí tienes un ejemplo completo que muestra cómo definir, inicializar, modificar y usar una estructura en C++:

```
#include <iostream>
#include <string>

// Definición de la estructura
struct Persona {
    std::string nombre;
    int edad;
    std::string direccion;

    // Constructor
    Persona(const std::string& n, int e, const std::string& d) : nombre(n), edad(e),
direccion(d) {}
};

// Función para imprimir los datos de una Persona
void imprimirPersona(const Persona& p) {
    std::cout << "Nombre: " << p.nombre << std::endl;
    std::cout << "Edad: " << p.edad << std::endl;
    std::cout << "Dirección: " << p.direccion << std::endl;
}

int main() {
    // Crear e inicializar instancias de Persona
    Persona persona1("Juan Pérez", 30, "Calle Falsa 123");
    Persona persona2("Ana Gómez", 25, "Avenida Siempre Viva 742");

    // Imprimir datos
    std::cout << "Datos de Persona 1:" << std::endl;
    imprimirPersona(persona1);

    std::cout << std::endl;

    std::cout << "Datos de Persona 2:" << std::endl;
    imprimirPersona(persona2);

    // Modificar datos
    persona1.edad = 31;
    persona2.direccion = "Avenida Siempre Viva 743";

    std::cout << std::endl;
    std::cout << "Datos de Persona 1 después de modificación:" << std::endl;
    imprimirPersona(persona1);

    std::cout << std::endl;

    std::cout << "Datos de Persona 2 después de modificación:" << std::endl;
    imprimirPersona(persona2);

    return 0;
}
```

Explicación:

Definición de Persona: Se define una estructura con un constructor para inicializar los datos.

Inicialización: Se crean instancias de Persona con datos iniciales.

Impresión: Se imprime la información de las personas usando la función imprimirPersona.

Modificación: Se actualizan algunos campos y se vuelven a imprimir.

Este ejemplo muestra cómo trabajar con estructuras en C++, desde su definición hasta la modificación de datos y el uso en programas reales.