Seminario de Solución de problemas de Estructuras de Datos I

Investigación I: Estructuras

Sección D21

Ramiro Lupercio Coronel

Ismael Iván López Murillo

220975903

15 / 09 / 2020

Structs en C++

Una estructura es un tipo de dato compuesto que nos permite almacenar un conjunto de datos de diferente o igual tipo. Estos datos que se contienen en las estructuras pueden ser del tipo simple, como caracteres, enteros, flotantes, etc… y asimismo otros tipos de datos compuestos como los vectores, otras struct o listas.

A cada uno de los datos que contiene una estructura se le conoce como miembro y deben tener bien definido el tipo de dato.

Una estructura define un tipo de dato y no una variable, por lo que no existe una reserva de memoria cuando el compilador lo analiza.

Sintaxis:

```
struct nombre_estructura
{
        tipoDato1 nombreDato1;
        tipoDato2 nombreDato2;
        ….
        tipoDatoN nombreDatoN;
};
```

Al momento de querer declarar variables de un tipo de estructura se puede hacer de las siguientes formas:

1. Declarando la estructura y después en cualquier parte del programa declararla con el nombre de la estructura e identificador:
   struct ejemplo{ … };

struct prueba1, prueba2;

2. Declarándola como global al momento de definir la estructura. Se ponen los identificadores justo después de terminar la estructura pero antes del punto y coma.

```
struct ejemplo
{
        ….
} prueba1, prueba2;
```

cin

El comando cin es la forma estándar de lectura de datos en C++ que nos proporciona iostream. Este se puede leer como “console in” Este nos permite leer varios datos sin especificar su tipo explícitamente. Su sintaxis es

cin >> variable1 >> variable2 >> … >> variableN;

Para esto las variables ya deben de estar declaradas anteriormente y siempre se usa el operador “>>” para separarlas y usar este comando.

cout

El comando cout es la forma estándar de salida de datos en C++ que nos proporciona iostream. Este se puede leer como “console out”. Este nos permite mostrar varios datos, valores o cadenas sin especificar su tipo explícitamente. Su sintaxis es:

cout << “Cadena” << variable << variableN << endl;

Donde usamos “>>” para separar los valores y “endl” es un salto de línea. Se pueden mostrar diversos datos con un solo cout, pero es recomendable no hacerlo para simplicidad del código.

Tipos de Datos

En C++ existen diversos tipos de datos con los que podemos manejar, todos estos cuentan con un conjunto específico de valores que pueden tomar, una

cantidad de memoria que ocupan y una palabra reservada específica. La memoria suele depender del compilador, pero casi siempre es estándar.
Estos son los tipos de datos:

Nombre	Memoria	Tipo de dato
int	16 bits	Numérico, solo números enteros
long	32 – 64 bits	Numérico, solo números enteros
short	32 bits	Numérico, solo números enteros
float	16 bits	Numérico con punto flotante o decimal
double	32 bits	Numérico con punto flotante o decimal
char	8 bits	Almacena un carácter.
string o char *	Variable	Almacena una cadena de caracteres y su longitud es variable

Adicionalmente, para int existe el tipo “unsigned”, que nos permite tener más valores máximos positivos, pero no nos permite almacenar números negativos y long double, con el doble de espacio que un double normal.

Conclusión

Las struct son la base de las estructuras de datos, de una forma literal, si bien este tipo de dato puede ser reemplazado por clases, este contiene unas peculiaridades que en ocasiones nos sirven más para realizar estructuras de datos como lo pueden ser filas o colas.

Basta con conocer un poco sobre tipos de datos, entrada y salida de datos y el concepto de estructuras para empezar a programar estas estructuras de datos.

Son conceptos bastantes simples pero que nos ayudarán a obtener soluciones simples a problemas complejos facilitándonos demasiado el desarrollo.

Bibliografía:

- S, J. (n.d.). Definición de una Estructura. Consultado en septiembre 15, 2020, de http://decsai.ugr.es/~jfv/ed1/c/cdrom/cap7/cap71.htm
- Meza González, J.D. (Julio 2020). Entrada y salida de datos en C++. Uso de cin, cout, scanf y printf. Consultado en septiembre 15 2020, de https://www.programarya.com/Cursos/C++/Entrada-y-Salida-de-Datos
- Acosta, I. (2020, January 21). TIPOS DE DATOS EN C++. Consultado en septiembre 15, 2020, de https://geekelectronica.com/tipos-de-datos-en-c/