ЛАБОРАТОРНА РОБОТА №12

Тема: **Структури та об’єднання**

Мета: **Ознайомитись з основами роботи зі структурами та об’єднаннями**

Час: 2 год.

Виконання роботи

- Надати викладачу, виконане завдання для самопідготовки в п. 4.2.
- Вивчити теоретичні відомості.
- Відповісти тестові завдання.
- Виконати самостійну роботу.

Завдання для самопідготовки

В процесі підготовки до заняття студент у обов'язковому порядку повинен виконати наступні завдання:

а) За допомогою конспекту лекцій і рекомендованої літератури розглянути суть таких питань:

1) Поняття та способи організації структур
2) Доступ до елементів структури
3) Обробка структур
4) Об’єднання

б) Занести в звіт такі дані:

1) номер практичної роботи;
2) тему і мету роботи;
3) короткий конспект основних теоретичних відомостей.

Теоретичні відомості

Структура - це сукупність змінних, об'єднаних одним ім'ям, що надає загальноприйнятий спосіб спільного зберігання інформації. Оголошення структури призводить до утворення шаблону, використовуваного для створення об'єктів структури. Змінні, що утворюють структуру, називаються членами структури. (Члени структури також часто називаються елементами або полями.)

Зазвичай всі члени структури пов'язані один з одним. Наприклад, інформація про ім'я та адресу, що знаходиться в списку розсилки, зазвичай представляється у вигляді структури. Наступний фрагмент коду оголошує шаблон структури, що визначає ім'я та адресу. Ключове слово struct повідомляє компілятору про оголошення структури.

```
struct addr {
char name [30];
char street [40]; char city [20];
char state [3];
unsigned long int zip;
};
```

Оголошення завершується крапкою з комою, оскільки оголошення структури - це оператор. Ім'я структури addr ідентифікує структуру даних і є специфікатором типу. Ім'я структури часто використовують як ярлик.

На даний момент насправді не створено жодної змінної. Визначено лише форма даних. Для оголошення справжньою змінної, відповідної даній структурі, слід написати:

```
struct addr addr_info;
```

У цьому рядку відбувається оголошення змінної addr_info типу addr. При оголошенні структури визначається змінна змішаного типу. До тих пір, поки не буде оголошена змінна даного типу, вона не буде існувати.

Коли оголошена структурна змінна, компілятор автоматично виділяє необхідну ділянку пам'яті для розміщення всіх її членів. Мал. показує розміщення addr_info в пам'яті.

При оголошенні структури можна одночасно оголосити одну або кілька змінних.

наприклад:

```
struct addr {
char name [30];
char street [40];
char city [20];
char state [3];
unsigned long int zip;
} Addr_info; binfo, cinfo;
```

оголошує структуру addr і оголошує змінні addr_info, binfo, cinfo даного типу.

Важливо зрозуміти, що кожна знову створювана структурна змінна містить свої власний копії змінних, що утворюють структуру. Наприклад, поле zip змінної binfo відокремлено від поля zip змінної cinfo. Фактично, єдиний зв'язок між binfo і cinfo полягає в тому, що вони обидві є екземплярами одного типу структури. Більше між ними немає зв'язку.

Якщо необхідна тільки одна структурна змінна, то немає необхідності в ярлику структури. Це означає, що

```
struct {
char name [30];
char street [40];
char city [20];
char state [3];
unsigned long int zip;
} Addr_info;
```

оголошує одну змінну addr_info з типом, визначеним попередньої структурою. Стандартний вид оголошення структури наступний:

```
struct ярлик {
тип ім'я змінної;
тип ім'я змінної;
тип ім'я змінної;
} Структурні змінні;
```

ярлик - це ім'я типу структури, а не ім'я змінної. Структурні змінні - це розділений комами список імен змінних. Слід пам'ятати, що або ярлик, або структурні змінні можуть бути відсутніми, але не обидва.

Доступ до окремих членів структури здійснюється за допомогою оператора. (Зазвичай називається «точкою») Наприклад, наступний фрагмент коду присвоює члену zip структурної зміною addr_info значення 12345:

```
addr_info.zip = 12345;
```

За ім'ям структурної змінної слід точка, а за нею ім'я члена, до якого відбувається звернення. До всіх членів структури доступ здійснюється точно таким же способом. Стандартний вид доступу наступний:

```
імя_структури.імя_члена
```

Отже, для виведення поля zip на екран треба написати:

```
printf ( "% ld", addr_info.zip);
```

Об'єднання - це такий формат даних, який подібний до структурі. Воно (об'єднання) здатне зберігати в межах однієї зарезервованої області пам'яті різні типи даних. Але в кожний певний момент часу в об'єднанні зберігається тільки один з цих типів даних і можливо використовувати лише значення цього елемента (компонента). Синтаксично визначають об'єднання так (дуже схоже на визначення структури):

Об'єднання (union) в C ++

```
union
{
short int name1;
int name2;
long int name3;
} MyUnion; // Об'єкт об'єднання
union
{
short int name1;
int name2;
long int name3;
} MyUnion; // Об'єкт об'єднання
```

Доступ до елементів об'єднання здійснюється так само, як і до елементів структур: Ім'я об'єкта об'єднання myUnion, точка. і ім'я елемента name1.

До даних, які зберігають елементи структури (наприклад short, int, long) ми можемо звертатися до будь-який момент (хоч до одного, хоч і до всіх відразу). А в об'єднанні можуть зберігатися дані або short, або int, або long.

Практична частина

Контрольний приклад

Приклад 1

```
#include <iostream>
using namespace std;

struct building // Створюємо структуру
{
    char * owner; // Тут буде зберігатися ім'я власника
    char * city; //назва міста
    int amountRooms; //кількість кімнат
    float price; // ціна
};

int main ()
{
    setlocale (LC_ALL, "rus");

    building apartment1; // Це об'єкт структури з типом даних,
ім'ям структури, building

    apartment1.owner = "Денис"; // Заповнюємо дані про власника
і т.д.
    apartment1.city = "Мелітополь";
    apartment1.amountRooms = 5;
    apartment1.price = 150000;
```

```
    cout << "Власник квартири:" << apartment1.owner << endl;
    cout << "Квартира знаходиться в місті" << apartment1.city <<
endl;
    cout << "Кількість кімнат:" << apartment1.amountRooms <<
endl;
    cout << "Вартість:" << apartment1.price << "$" << endl;

return 0;
}
```

Приклад 2.

```
#include <iostream>
using namespace std;

union
{
short int name1;
int name2;
long int name3;
} MyUnion;

int main ()
{
myUnion.name1 = 22;
cout << myUnion.name1 << endl;
myUnion.name3 = 222222222;
cout << myUnion.name3 << endl;

cout << myUnion.name1 << endl; // Знову звертаємося до name1
}
```

Самостійна робота

1. Є шкільний журнал, в якому внесені наступні дані (не менше 10): прізвища та ім'я учня, стать, дата народження, клас, зростання, останні оцінки з 5 предметів. Скласти програму, яка поверне наступні дані:
   - Середні бали по предмету, класу;
   - Середні бали зазначеного користувачем учня;
   - Список відмінників і двієчників;
   - Середнє зростання хлопчиків і дівчаток зазначеного користувачем класу;
   - Список випускників;
2. Є інформація про погоду за рік про середньомісячній температурі і кількості опадів в ряді країн Європи (не менше 4-х). Скласти програму, яка поверне наступні дані:

- Найбільш посушливий місяць кожної країни;
- Країну і місяць, які виявилися найбільш дощовим в Європі;
- Країну Європи з самим сухим і гарячим кліматом;
- Найхолодніша країна Європи;
- Кліматичні дані з розбивкою за порами року, старне, зазначеної користувачем.

3. Є графік руху поїздів (не менше 4-х), створений у вигляді таблиці, в якому вказані: номер поїзда, час прибуття і відправлення по певній станції, номер платформи. Розробити програму, яка допоможе диспетчерам визначати:
   - Кількість поїздів, які прямують через зазначену станцію і їх номери (з маршрутом);
   - Поїзди, які прибувають станцію після зазначеного часу (з зазначенням платформи);
   - Час стоянки зазначеного поїзда по певній станції (врахувати, що є швидкісні поїзди, які зупиняються тільки на великих станціях і час стоянки скорочено);
   - Час проходження з однієї станції до іншої. Станції задає диспетчер.
4. **Об'єднання.** Компанією було придбано ліфт для торгового центру. Маркетологами було прийнято рішення встановити інформаційне табло, на якому відображати назву обраного клієнтом поверху і що на ньому знаходиться. Створити програму для забезпечення роботи табло.

Вимоги до оформлення звіту

Звіт повинен містити:

– Короткий конспект теоретичних відомостей;
– Результати виконаних дій.