

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»  
Інститут телекомунікацій, радіоелектроніки та електронної техніки  
кафедра «Радіоелектронні пристрої та системи»



Звіт з лабораторної роботи №18

з дисципліни «Програмування»

Підготував:  
ст. групи АП-11  
Чаус В. О.

Прийняла:  
Чайковський І.Б.

**Львів – 2024**

**Тема:**

Структури та об'єднання даних.

**Мета:** ознайомитися з поняттями структури та об'єднання даних , навчитися їх використовувати у процесі програмування.

**Теоретичні відомості:**

Структура – це сукупність змінних, об'єднаних під одним ім'ям. За допомогою структур зручно розміщувати в суміжних полях пов'язані між собою елементи інформації.

Перед будь-яким використанням структур треба оголосити структурний тип.

Оголошення структурного типу має такий вигляд:

```
struct ім'я _ структурного _ типу { тип _ поля  
ім'я _ поля ;
```

```
...
```

```
тип _ поля ім'я _ поля ;  
};
```

Елементами структури вважаються змінні, декларовані в списку, що обмежується фігурними дужками.

Оголошення структури створює шаблон, який можна використовувати для створення її об'єктів (тобто примірників цієї структури). Змінні, з яких складається структура, називаються членами (члени структури ще називаються елементами або полями.)

Як правило, члени структури пов'язані один з одним за змістом.

Наприклад, елемент списку розсилки, що складається з імені та адреси логічно представити у вигляді структури


**Хід роботи:**

1. Ознайомитися з теоретичними відомостями.
2. Здійснити виконання прикладів, представлених у теоретичних відомостях, після чого представити скріни їх коду та результати виконання у звіті.
3. Написати програму для виведення нижчепредставленої інформації шляхом використання структури. Ім'я, вага, висота, вік – вводяться з клавіатури (вказати довільні дані).
4. Оформити звіт.

**Виконання роботи:**

2.

```
main.c
1  #include <stdio.h>
2
3  int main(void) {
4      struct {
5          int a;
6          int b;
7      } x, y;
8      x.a = 10;
9      y = x;
10     printf("%d", y.a);
11     return 0;
12 }
13
```



10

main.c

```
1  #include <stdio.h>
2  #include <string.h>
3
4  struct student {
5      char name[30];
6      int kurs;
7      int age;
8  };
9
10 int main() {
11     struct student stud1;
12     printf("Введіть ім'я: ");
13     fgets(stud1.name, sizeof(stud1.name), stdin);
14     stud1.name[strcspn(stud1.name, "\n")] = '\0';
15     printf("Введіть вік: ");
16     scanf("%d", &stud1.age);
17     printf("Введіть курс: ");
18     scanf("%d", &stud1.kurs);
19     printf("Студент %s\n", stud1.name);
20     printf("Курс %d\n", stud1.kurs);
21     printf("Вік %d\n", stud1.age);
22     return 0;
23 }
24
```

▼ ↗ ⚙ 📄

Введіть ім'я: test  
Введіть вік: 18  
Введіть курс: 1  
Студент test  
Курс 1  
Вік 18

main.c

```
1  #include <stdio.h>
2
3  struct student {
4      char name[30];
5      int kurs;
6      int age;
7  };
8
9  int main() {
10     struct student stud[10];
11     int i, n;
12     printf("Кількість студентів: ");
13     scanf("%d", &n);
14     for(i = 0; i < n; i++) {
15         printf("Введіть ім'я: ");
16         scanf("%s", stud[i].name);
17
18         printf("Введіть вік: ");
19         scanf("%d", &stud[i].age);
20
21         printf("Введіть курс: ");
22         scanf("%d", &stud[i].kurs);
23     }
24     for(i = 0; i < n; i++) {
25         printf("Студент %s\n", stud[i].name);
26         printf("Курс %d\n", stud[i].kurs);
27         printf("Вік %d\n", stud[i].age);
28     }
29
30     return 0;
31 }
```

Кількість студентів: 2  
Введіть ім'я: test1  
Введіть вік: 18  
Введіть курс: 1  
Введіть ім'я: test2  
Введіть вік: 17  
Введіть курс: 1  
Студент test1  
Курс 1  
Вік 18  
Студент test2  
Курс 1  
Вік 17

3.

```
1  #include <stdio.h>
2
3  struct zavd {
4      char name[50];
5      float weight;
6      float height;
7      int age;
8  };
9
10 int main() {
11     struct zavd emp;
12     printf("Ім'я працівника: ");
13     scanf("%s", emp.name);
14     printf("Вага працівника: ");
15     scanf("%f", &emp.weight);
16     printf("Висота працівника: ");
17     scanf("%f", &emp.height);
18     printf("Вік працівника: ");
19     scanf("%d", &emp.age);
20     printf("\nІнформація про працівника %s\n", emp.name);
21     printf("-----\n");
22     printf("Ім'я: %s\n", emp.name);
23     printf("Вага: %.2f\n", emp.weight);
24     printf("Висота: %.2f\n", emp.height);
25     printf("Вік: %d\n", emp.age);
26
27     return 0;
28 }
29
```

Ім'я працівника: TEST  
Вага працівника: 59  
Висота працівника: 180  
Вік працівника: 18

Інформація про працівника TEST  
-----  
Ім'я: TEST  
Вага: 59.00  
Висота: 180.00  
Вік: 18

### **Контрольні запитання:**

1. Структура - це складний тип даних в програмуванні, який дозволяє комбінувати різні типи даних під одним ім'ям. Вона може включати в себе змінні різних типів, об'єднуючи їх в логічний блок.
2. Оголошення структури зазвичай відбувається шляхом визначення нового типу даних, який включає в себе різні поля або члени. У багатьох мовах програмування це здійснюється за допомогою ключового слова, такого як "struct" (у мові C або C++) або "class" (у мові C++ або Java).
3. Синтаксис об'єднання даних (зазвичай відомий як "структури" або "об'єкти" в об'єктно-орієнтованих мовах програмування) включає в себе оголошення полів або членів, їх типи та доступ до них через оператори членства.
4. До структур можна застосовувати багато операцій, але є деякі обмеження. Наприклад, неможливо виконати арифметичні операції безпосередньо над самою структурою (наприклад, додавання двох структур). Також, в деяких мовах програмування можуть бути обмеження на операції порівняння між структурами, які потрібно визначити явно.