

Лабораторна робота №6

Створення та застосування структур

Мета роботи

Ознайомитись зі створенням та застосуванням структур та отримати практичні навички роботи з їх полями.

Хід роботи

- 1) Ознайомитись з методичними вказівками до лабораторної роботи та матеріалами лекції 2_3_ за темою Комбіновані типи даних
- 2) сформувати структуру згідно з варіантом завдання;
- 3) розробити схему алгоритму програми, що демонструє роботу зі створеною структурою;
- 4) за схемою алгоритму написати програму на мові C++;
- 5) зробити висновки;
- 6) підготувати звіт про виконання лабораторної роботи, що включає наступні пункти: номер, тема, мета та хід лабораторної роботи, завдання, схема алгоритму програми, лістинг програми з коментуванням кожної інструкції, результат роботи програми (скріншот), висновки.

Теоретичні відомості та рекомендації до виконання

Як і масиви, структури дозволяють зберігати не одне значення, а декілька, проте на відміну від масивів у структурі для кожної змінної зазначено власний тип даних та унікальне ім'я. Таким чином, структури є групою змінних різного типу даних, що поєднані під одним ім'ям.

Як і масиви структури слугують для зберігання даних.

При виконанні завдання до лабораторної роботи студент повинен ознайомитися з особливостями створення структур та навчитись застосовувати їх в програмах на мові C++.

Виконання лабораторної роботи починається ознайомленням з теоретичними відомостями про оголошення та визначення структур і роботу з їх полями. Потім студент приступає до розробки схеми алгоритму програми згідно із завданням за варіантом. За схемою алгоритму виконується написання програми на мові C++.

У висновках до звіту студент повинен розкрити призначення комбінованих типів даних та застосування структур. Висновок має бути поданий в формі пояснення сфери застосування вивченого матеріалу згідно із розумінням його студентом.

Завдання для самостійного виконання

Розробити схему алгоритму та написати програму, яка створює на основі структури за варіантом таблицю (табл. 1) та дозволяє заповнити її поля значеннями, що вводяться з консолі.

За запитом користувача програма повинна виводити відповідний рядок або всю отриману таблицю (Запити сформулювати по кожному полю самостійно, згідно з тематикою варіанта).

Варіанти завдань

Таблиця 1

№ вар.	Ім'я структури (призначення)	Перше поле	Друге поле	Третє поле
1	Одяг	вид	розмір	вартість
2	Муз. композиція	назва пісні	рейтинг	виконавець
3	Телепередача	час показу	назва	рейтинг
4	Фільм	жанр	тривалість	бюджет картини
5	Студентська група	П.І.Б.	вік	середній бал
6	Книжка	автор	назва	кількість сторінок
7	Їжа	назва блюда	Час приготування	калорійність
8	Напій	назва	калорійність	присмак
9	Обладнання	тип	виробник	енергоспоживання
10	Канцелярські знаряддя	вид	вартість	шифр товару
11	Військова техніка	клас	витрата палива	марка
12	Автомобілі	виробник	рік випуску	марка
13	Човни	тип	швидкість	місткість
14	Рослини	вид	середній час життя	число підвидів
15	Тварини	вид	ареал проживання	число особин
16	Комп'ютери	марка	швидкодія	рейтинг
17	Картини	художник	вартість	жанр
18	Будівлі	клас будівлі	число поверхів	район
19	Фірми	діяльність	назва	прибуток
20	Озера	країна	назва	площа
21	Планети	назва	маса	діаметр
22	Шкільні предмети	назва	час викладання	учитель
23	Мобільний телефон	фірма	модель	ціна
24	Мультиплікація	компанія	назва	тривалість
25	Країна	назва	населення	політ. режим
26	Ріки	назва	протяжність	число притоків
27	Літаки	виробник	місткість	дальність польоту
28	Оргтехніка	фірма	модель	вартість
29	Косметика	назва	ціна	номер товару
30	Робітники	професія	стаж роботи	сер. заробіток

Приклад виконання завдання

Розробити схему алгоритму та написати програму, яка створює на основі структури таблицю та дозволяє заповнити її поля значеннями, що вводяться з консолі

№ вар.	Ім'я структури (призначення)	Перше поле	Друге поле	Третє поле
N	Співробітник	номер	ім'я	прізвище

Розробимо схему алгоритму роботи програми створення та заповнення таблиці на основі структури

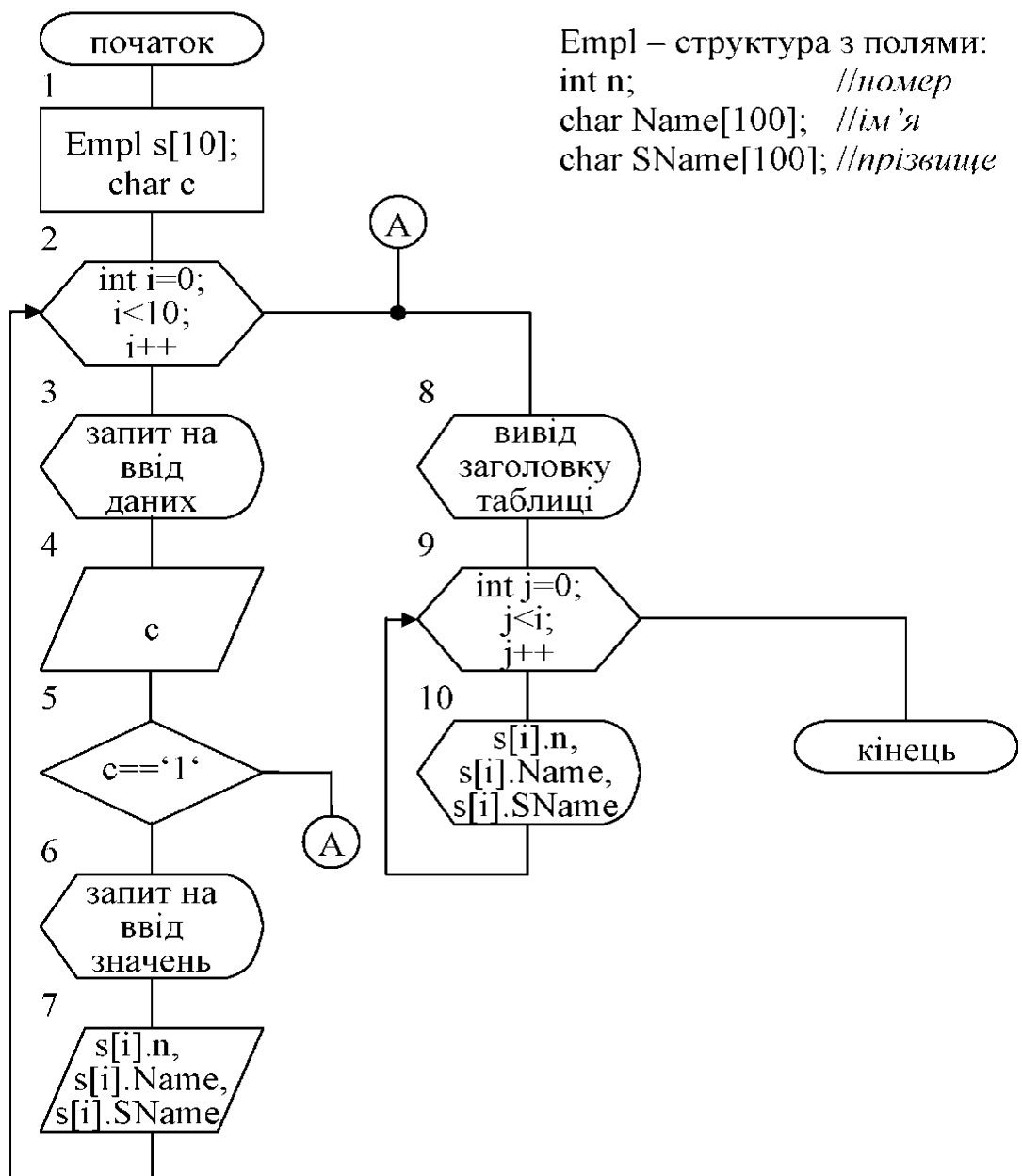


Рис. 1. Схема алгоритму програми роботи з таблицею

За схемою алгоритму напишемо програму на мові C++:

```
#include<stdio.h>                                //підключення бібліотеки вводу/виводу

struct Empl                                     //оголошення та визначення структури співробітників
{
    int n;                                       //цілочисельне поле номера
    char Name[100];                             //поле символьний масив для імені
    char SName[100];                           //поле символьний масив для прізвища
};

void main()                                    //оголошення та визначення головної функції
{
    Empl s[10];                                //оголошення масиву екземплярів структури
    char c;                                    //оголошення змінної зупинки заповнення таблиці
    printf("You start table filling!\r\n");      //вивід запиту до користувача
    for(int i=0; i<10; i++)                    //цикл заповнення таблиці
    {
        //вивід запиту про закінчення заповнення
        printf("Continue? 1 – No\r\n");
        scanf("%s", &c);                       //отримання значення від користувача
        if(c=='1')                             //якщо введено '1'
            break;                             //вихід з оператора
        //вивід запиту на ввід значення поля номера
        printf("Enter first field value (Number): ");
        scanf("%i", &s[i].n);                  //отримання значення від користувача
        //вивід запиту на ввід значення поля імені
        printf("Enter second field value (Name): ");
        scanf("%s", s[i].Name);                //отримання значення від користувача
        //вивід запиту на ввід значення поля прізвища
        printf("Enter third field value (Second Name): ");
        scanf("%s", s[i].SName);               //отримання значення від користувача
    }
    //вивід заповненої таблиці
    printf("Table:\r\n");                      //вивід константного рядка
    printf("N\t\tName\t\tSName\r\n");        //вивід константного рядка
    for (int j=0; j<i; j++)                    //цикл виводу полів таблиці
        //вивід значень рядка таблиці
        printf("%i\t\t%s\t\t%s\r\n", s[j].n, s[j].Name, s[j].SName);
}
```

Результат виконання програмного коду буде наступним:

```

You start table filling!
Continue? 1 - No
0
Enter first field value (Number): 1
Enter second field value (Name): Sergey
Enter third field value (Second Name): Ivanov
Continue? 1 - No
0
Enter first field value (Number): 2
Enter second field value (Name): Dmitriy
Enter third field value (Second Name): Petrov
Continue? 1 - No
0
Enter first field value (Number): 3
Enter second field value (Name): Oleg
Enter third field value (Second Name): Sidorov
Continue? 1 - No
1
Table:
N          Name          SName
1          Sergey        Ivanov
2          Dmitriy       Petrov
3          Oleg          Sidorov

```

Приклад Завдання 2

1. Запустити середовище програмування C++ Builder .

Задача. Ввести в комп'ютер відомість успішності студентів, які здали сесію з дисциплін: «Математика», «Фізика» і «Програмування», та обчислити:

- середній бал кожного студента;
- середній бал групи за кожним предметом;
- вивести на екран прізвища відмінників з програмування.

```

#include <iostream.h>
#include <string.h>
void main()
{ const n = 5; //n - количество студентов
  int i;
  float sm, sf, sp;
  /* sm, sf.sp - сумма оценок группы соответственно по математике,
  физике, программированию */
  struct stud
  { char fam[25];
    int mat, fiz, prg;
    float sb; }
  ved[n]; //ved[n] - массив студентов (ведомость)
  sm = sf = sp = 0;
  //---- ввод и обработка информации о студентах и их успеваемости
  for (i = 0; i < n; i++)
  { cout << "***** Введите информацию о "<< (i+1) << " студенте\n";
    { cout << "Введите фамилию и инициалы\n";
      cin >> ved[i].fam;

```

```

        cout << "Оценки по матем., физике и программир.\n";
        cin >> ved[i].mat >> ved[i].fiz >> ved[i].prg;
//----- вычисление среднего балла студента за сессию
        ved[i].sb = (float(ved[i].mat + ved[i].fiz + ved[i].prg))/3;
//----- суммирование оценок в группе по предметам
        sm += ved[i].mat;
        sf += ved[i].fiz;
        sp += ved[i].prg;
    }
}
//----- вывод результатов вычислений
    cout << '\n***** Результаты сессии\n';
    cout.precision(3);
    for (i = 0; i < n; i++)
        cout << i+1<< " "<< ved[i].fam << " матем. = " << ved[i].mat << "
физика = "<< ved[i].fiz<< "програм. = " <<ved[i].pig << " ср. балл = "
<< ved[i].sb << "\n";
        cout << "\n\nСредний балл группы по математике = "<< sm << "/n";
        cout << "\nСредний балл группы по физике = "<< sf << "/n";
        cout << "\nСредний балл группы по программированию = " << sp <<
"/n";
    cout << "\n\n***** Отличники по программированию: \n";
    for (i = 0; i < n; i++)
        if (ved[i].prg == 5)
            cout << ved[i].fam << "\n";
    cin>>i;
}

```

2. Розробіть структуру, що містить відомості про результати екзаменів по предмету:

Прізвище,

Ім'я,

По батькові,

Число балів по першому предмету,

Число балів по другому предмету,

Число балів по третьому предмету.

Розробіть функцію, що визначає результат вступу абітурієнта по заданому прохідному балу. Введіть відомості про **N** абітурієнта і виведіть результати зарахування за зразком (дані, введені користувачем, виділені жирним шрифтом):

Інформація про 1 абітурієнта:

Прізвище: **Петров**

Ім'я: **Федор**

По батькові: **Степанович**

Число балів з першого предмету: **84**

Число баллов з другого предмету: **76**

Число баллов з третього предмету: **95**

Результат: зарахований

Питання для підготовки до захисту лабораторної роботи

- 1) Що називають складеними типами даних?
- 2) Дайте визначення поняттю структура? Наведіть приклади використання структур?
- 3) Як описуються дані типу структура?
- 4) Як оголошується структура?
- 5) Як називаються змінні, що входять до структури?
- 6) Як отримати доступ до змінних структури з основної програми?
- 7) Як сформувати у пам'яті комп'ютера на основі структури таблицю?