

Лабораторна робота № 10.

Тема: Практичне застосування структур.

Мета: Оволодіти практичними навичками зі створення та використання структур. Навчитись створювати масиви структур та здійснювати операції з полями структур.

Зміст роботи.

Написати програму мовою C++ з використанням структур:

Завдання 1

Створити структуру даних про студентів деякого вузу, які здавали в сесію іспити. Структура має містити: ПІБ, групу, отримані бали з 5 дисциплін (назви вибрати на свій розсуд). Заповнити структуру для трьох студентів. Для кожного студента розрахувати середній бал успішності. Вивести на екран всю інформацію про студентів.

Завдання 2

В автосалоні підготовлено список пропонуваного до продажу автомобілів. Про кожен з них містить таку інформація: марка, модель, ціна (\$), кількість, колір. Створити масив структур автомобілів із 5 елементів. Інформацію вивести на екран.

Завдання 3.

Модифікувати код завдання №1, представивши успішність кожного студента у вигляді масиву отриманих балів, без вказання назв дисциплін.

Теоретичні відомості

Структура – це складний (агрегований) тип даних, що є об'єднанням фіксованого числа компонентів одного або декількох типів.

Складові частини структури називаються членами структури або полями або елементами. У мові C++ структура схожа на клас, тобто має об'єктно-орієнтовані властивості.

Як правило члени структури називаються полями.

Структура оголошується за синтаксисом:

```
struct тип_структури {  
    тип_поля1 ім'я_поля1;  
    тип_поля2 ім'я_поля2;  
    .....  
    тип_поляk ім'я_поляk; };
```

Тут **struct** – службове слово, за яким компілятор ідентифікує оголошення структури;

тип_структури - ім'я, яке позначає у програмі структуру даної форми;

у *фігурних дужках* міститься список полів, кожне з яких є змінною. Структура – це тип. А змінна цього типу оголошується в мові C++ як тип_структури ім'я_структурної_змінної.

Для такої змінної виділяється пам'ять, розмір якої залежить від полів структури. Поля структури послідовно розташовуються в пам'яті. Структурна змінна ініціалізується через надання значення її елементам.

Звертання до полів структурної змінної здійснюється через «крапку».

ім'я_структурної_змінної . ім'я_поля

Приклад оголошення структури:

```
struct anketa {  
    char prizm[20];  
    int vik;  
    double zarplata; };
```

Оголошення структурної змінної і вказівника на структуру: struct anketa kozak;

Ініціалізація структури при оголошенні:

```
struct anketa kozak = {"Kovalenko", 45, 15670.50};
```

Звертання до полів структури: **kozak.prizm, kozak.vik**

Масиви структур

Масивами структур, елементи якого мають структурований тип, можна оперувати, як звичайними масивами простих типів. Приклад:

```
struct Date
```

```
{  
    int d; /* день */  
    int m; /* місяць */  
    int y; /* рік */  
} Date;
```

```
Date arr[100];
```

Доступ до полів структури аналогічний доступу до звичайних змінних, при цьому у квадратних дужках вказують індекс номеру елемента:

```
arr[25].d=24; arr[12].m=12;
```

Контрольні запитання.

1. Що таке структура?
2. Для представлення якої інформації можна використовувати структури?
3. Наведіть синтаксис оголошення типу структури.
4. Як звернутися до поля структури?
5. Яким може бути тип поля структури?
6. Які є способи ініціалізації структури?