6 ПРОГРАМУВАННЯ КОРИСТУВАЛЬНИЦЬКИХ ТИПІВ

Мета: здобути навички програмування структур, об'єднань та перерахування.

6.1 Короткі теоретичні відомості

C++ надає можливість моделювання нових типів даних на базі вже існуючих типів. Як приклад доцільності застосовування таких типів розглянемо оголошення

```
double A[5][7];
```

Тут оголошується змінна A як матриця (двовимірний масив) дійсних чисел розмірності 5 рядків і 7 стовпчиків. Якщо у програмі оголошується кілька аналогічних масивів або якщо програма містить функції, параметрами яких є матриця 5×7, зручніше є створити окремий тип “матриця” для оголошення змінної A, щоб уникнути помилок і покращити читабельність тексту програми. У мові C++ для створювання типів користувача на базі вже існуючих типів використовується службове слово typedef (від англійської “type definition” – означення типу). Отже оголошення описаного типу matrica може мати вигляд

```
typedef double matrix [5][7];
```

Цей тип є синонімом дійсного двовимірного масиву розмірності 5×7, і його можна використовувати як звичайний тип для оголошення змінної A:

```
matrix A;
```

Це оголошення означає, що змінна A має тип matrix, тобто є дійсним двовимірним масивом.

Структура – це складений тип даних, змінні (поля) якого можуть містити різноманітну й різнотипну інформацію.

Опис структури має синтаксис

struct *<ім’я_типу_структури>*

{ <тип1> <поле1>;

<тип2> <поле2>;

. . .

<типN> <полеN>;

} <список_змінних>;

Ім’я типу структури задається програмістом виходячи зі змісту даних, які зберігатимуться у структурі. Список змінних наприкінці оголошення структури може бути відсутнім, у цьому разі після фігурної дужки має бути крапка з комою.

Наведемо приклад оголошення структури з ім’ям sportsman, яка поєднає в собі інформацію про змагання спортсменів: прізвище і вік спортсмена, країна, кількість його очок, утворюючи відповідні поля:

```
struct sportsman
{       char prizv[15]; char kraina[10]; // Прізвище та країна,
        int vik; float ochki; // вік та кількість очок.
};
```

Як наслідок цього оголошення створено новий тип sportsman. Тепер у програмі цей тип може використовуватись нарівні зі стандартними типами для оголошення змінних. Оголошені змінні типу sportsman матимуть чотири поля: два рядки (prizv і kraina), ціле (vik) і дійсне (ochki) числа.

```
sportsman Sp; /* Змінна Sp типу sportsman може зберігати інформацію про одного спортсмена. */

sportsman Mas[15]; /* Масив Mas типу sportsman може містити інформацію про 15 спортсменів. */
```

При оголошуванні структур їхні поля можна ініціалізовувати початковими значеннями. Наприклад, оголошення змінної Sp типу sporsman й ініціалізація її даними про 20-річного спортсмена з України за прізвищем Бойко, який набрав 75.3 очок, матиме вигляд:

```
sporsman Sp={"Бойко", "Україна", 20, 75.3};
```

Поля структури можуть бути якого завгодно типу, у тому числі й масивом, покажчиком чи структурою, окрім типу тієї ж самої структури (але можуть бути покажчиком на неї).

Доступ до полів структури відбувається за допомогою оператора “.” для статичного оголошення та «->» у випадку динамічного.

Як і звичайними масивами простих типів, так само можна оперувати масивами структур, елементи якого мають структурований тип.

Доступ до полів структури аналогічний доступу до звичайних змінних, плюс використання індексу номеру елементу у квадратних дужках.

Об’єднання – формат даних, який може містити різні типи даних, але лише один тип водночас. Можна сказати, що об’єднання є окремим випадком структури, всі поля якої розташовуються за однією й тією самою адресою. Формат опису об’єднання є такий самий як і у структури, лише замість ключового слова struct використовується слово union. Але, тоді як структура може містити, скажімо, елементи типу і int, і short, і double, об’єднання може містити чи то int, чи short, чи double:

```
union prim
```

```
{       int x;
        short y;
        double z;
};
```

Обсяг пам’яті, яку займає об’єднання дорівнює найбільшому з розмірів його полів. Об’єднання застосовують для економії пам’яті у тих випадках, коли відомо, що понад одного поля водночас не потрібно. Часто об’єднання використовують як поле структури.

Ім’я типу задається в тому разі, якщо у програмі потрібно визначати змінні цього типу. Змінним перераховного типу можна присвоювати кожне значення зі списку констант, зазначених при оголошуванні типу. Імена перераховних констант мають бути унікальними. Перераховні константи подаються й опрацьовуються як цілі числа і можуть ініціалізуватися у звичайний спосіб. Окрім того вони можуть мати однакові значення. За відсутності ініціалізації перша константа обнулюється, а кожній наступній присвоюється значення на одиницю більше, аніж попередній.

```
enum week { Sat=0, Sun=0, Mon, Tue, Wed, Thu, Fri } rob_den;
```

Тут описано перерахування week з відповідною множиною значень і оголошено змінну rob_den типу week. Перераховні константи Sat та Sun мають значення 0, Mon – значення 1, Tue – 2, Wed – 3, Thu – 4, Fri – 5.

До перераховних змінних можна застосовувати арифметичні операції й операції відношення, наприклад:

```
enum days {Sun, Mon, Tue, Wed, Thu, Fri, Sat};
days day1 = Mon, day2 = Thu;
int diff = day2 – day1;
cout << "Різниця у днях: " << diff << endl;
if(day1 < day2) cout << "day1 настане раніш, аніж day2 \n";
```

Арифметичні операції над змінними перераховних типів зводяться до операцій над цілими числами. Проте, не зважаючи на те, що компіляторові відомо про цілочисельну форму подання перераховних значень, варто використовувати цей факт надто обережно.

6.2 Завдання

У період сессії студенти спеціальності ІПЗ (до 20 чоловік) отримали бали за стобальною шкалою за дисциплінами: “Основи програмування”, “Дискретна математика”, “Вища математика”. Організувати введення даних та обчислити:

- середній бал кожного студента;

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

6.3 Хід роботи

6.3.1 Постановка задачі

Дано:n — ціле — кількість студентів у групі;

ІПЗ — вектор студентів - назва групи;

Студент — структура, що містить Прізвище, Ім'я та дисципліни студента);

Прізвище — рядковий тип даних;

Ім'я — рядковий тип даних;

Дисципліна — структура, що містить назву та оцінку студента;

Назва — рядковий тип даних;

Оцінка — цілочисельний тип даних.

Визначити:

Ss- середній бал студентів;

Sd –вектор середнього балу з предметів;

Surname – вектор прізвищ відмінників;

k – кількість відмінників.

6.3.2 Метод реалізації інформаційного процесу

По-перше організуємо структури:

Дисципліна{назва, оцінка};

Успішність — вектор з трьох дисциплін;

Студент {Призвище, Ім'я, Успішність};

Група — вектор зі студентів.

Середній бал прораховуємо за формулою:

$$S=\sum_{i=0}^{n-1}\frac{показник_i}{n} \tag{6.1}$$

Якщо у студента всі предмети більше або дорівнюють 90 балів, то він є відмінником та заносимо його до вектору відмінників.

6.3.3 Програмування

Лістинг 6.1 — Вихідний код програми

```
#include<iostream>
#include <Windows.h>

using namespace std;

struct Discipline // Сутність дисципліна
{
	char name[30]; // Назва
	unsigned short int mark; //Оцінка
};

typedef Discipline Shedule[3];// Розклад з трьома дисциплінами
struct Student // Сутність студента
{
	char firstName[20]; //Ім'я студента
	char lastName[20]; // Прізвище
	Shedule shedule;
	unsigned short int averageMark;
};


int main()
{
	SetConsoleCP(1251);
	SetConsoleOutputCP(1251);
	unsigned short int n;
	cout << "Введіть кількість студентів групи - ";
	cin >> n;
	Student* IPZ = new Student[n];

	/*for(int i=0;i<n;i++)
	{
		cout << "Введення даних студента № " << i<<endl;
		cout << "Введіть прізвище: ";cin >> IPZ[i].lastName;
		cout << "Введіть Ім'я    : ";cin >> IPZ[i].firstName;
		strcpy_s(IPZ[i].shedule[0].name,"Основи програмування");
		strcpy_s(IPZ[i].shedule[1].name, "Дискретна математика");
		strcpy_s(IPZ[i].shedule[2].name, "Вища математика");
		for (int j = 0;j < 3;j++)
		{
			cout << "Введіть оцінку з дицилпліни " << IPZ[i].shedule[j].name <<" :
";
			cin >> IPZ[i].shedule[j].mark;
		}
	}*/
	IPZ[0] = { "Іван","Іванов",{{"Основи програмування",85},{"Дискретна математика",91},
{"Вища математика",75}} };
	IPZ[1] = { "Петро","Ребро",{ { "Основи програмування",91 },{ "Дискретна
математика",93 },{ "Вища математика",98 } } };
	IPZ[2] = { "Ірина","Шевченко",{ { "Основи програмування",65 },{ "Дискретна
математика",91 },{ "Вища математика",78 } } };



	//Визначення середнього балу студента
```

```
	for (int i = 0;i < n;i++)
	{
		for (int j = 0;j < 3;j++)IPZ[i].averageMark += IPZ[i].shedule[j].mark;
		IPZ[i].averageMark /= 3;
	}

	cout << "Прізвище   | Ім'я  | Основи програмування | Дискретна математика | Вища
математика | Середній бал\n";
	for (int i = 0;i < n;i++)
	{
		cout << IPZ[i].lastName << " | " << IPZ[i].firstName << " | ";
		for (int j = 0;j < 3;j++) cout << IPZ[i].shedule[j].mark << " | ";
		cout << IPZ[i].averageMark;
		cout << endl;
	}

	//Визначення середнього балу з дисциплін
	Shedule Sd;
	strcpy_s(Sd[0].name, "Основи програмування");
	strcpy_s(Sd[1].name, "Дискретна математика");
	strcpy_s(Sd[2].name, "Вища математика");
	for (int j = 0;j < 3;j++)
	{
		Sd[j].mark = 0;
		for (int i = 0;i < n;i++) Sd[j].mark += IPZ[i].shedule[j].mark;
		Sd[j].mark /= n;
	}

	cout << " Результат обчислення середнього балу за дисциплінами \n";
	for (int j = 0;j < 3;j++) cout << Sd[j].name << " : "<<Sd[j].mark<<endl;

	//Визначення відмінників
	cout << " Визначення відмінників навчання \n";
	for (int i = 0;i < n;i++)
	{
		bool fl = true;
		for (int j = 0;j < 3;j++)
			if (IPZ[i].shedule[j].mark < 90)
			{
				fl = false;
				break;
			}
		if (fl) cout << IPZ[i].lastName << endl;

	}
	delete[] IPZ;
	system("pause");
	return 0;
}
```

Зверніть увагу на те, що коментарем вказаний блок введення даних користувачем, а у прикладі наведено безпосередня ініціалізація перших трьох студентів.

6.3.4 Обчислення, обробка і аналіз результатів

У ході виконання даної роботи отримано наступні результати:

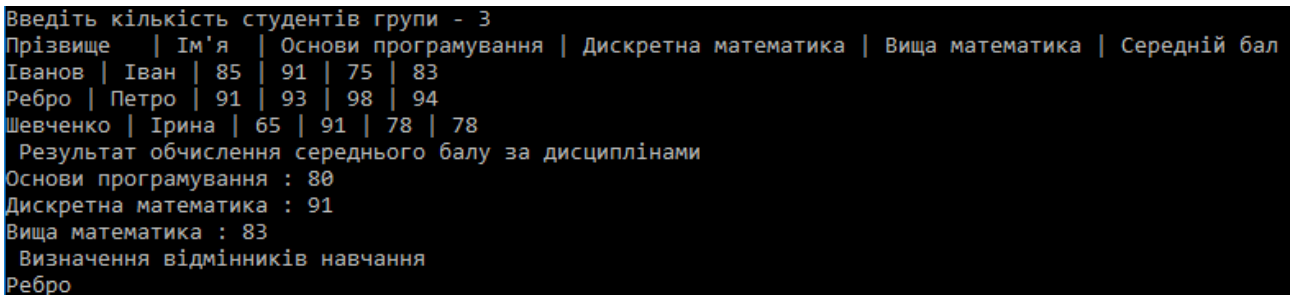

Рисунок 6.1 — Результат обчислень

Прізвище	Ім'я	Успішність				
		Основи програмування	Дискретна математика	Вища математика	Середній бал	
Іванов	Іван	85	91	75	84	не відмінник
Ребро	Петро	91	93	98	94	відмінник
Шевченко	Ірина	65	91	78	78	не відмінник
Середній бал дисципліни		80	92	84		

Рисунок 6.2 — Результат обчислення у електронній таблиці

Порівнюючи результати, отримані двома різними способами з високою вірогідністю можна стверджувати, що обчислення виконано правильно, так як отримані значення співпали (окрім балів округлення).

6.4 Програми та обладнання.

У даному підрозділі студент описує обладнання, програмні продукти та складові, що були використані при опрацюванні даної лабораторної роботи.

6.5 Висновки.

У даному підрозділі студент робить висновки за опрацюванням даної лабораторної роботи з урахуванням поставленої мети.

Завдання № 6.1

Задача 1.1 Номер телефону, наприклад (212) 767-8900, можна умовно розділити на три частини: код міста (212), номер телефонної станції (767) і номер абонента (8900). Напишіть програму з використанням структури, що дозволяє окремо зберігати ці три частини телефонного номера. Назвіть структуру phone. Створіть дві структурні змінні типу phone. Ініціалізацію однієї з них зробіть самі, а значення для іншої запросить з клавіатури. Потім виведіть вміст обох змінних на екран. Результат роботи програми повинен виглядати приблизно так: Введіть код міста, номер станції і номер абонента: 612 239 1212 Мій номер (612) 767-8900 Ваш номер (612) 239-1212

Задача 1.2 Розташування точки на площині можна задати за допомогою двох координат: x і y. Наприклад, якщо точка має координати (4, 5), то це означає, що вона відстоїть на 4 одиниці праворуч від вертикальної осі і на 5 одиниць вгору від горизонтальної осі. Сума двох точок визначається як точка, що має координати, які дорівнюють сумі відповідних координат складових. Напишіть програму, яка використовує для інтерпретації точки на площині структуру з назвою point. Визначте три змінні типу point, і дві з них ініціалізується за допомогою значень, що вводяться з клавіатури. Потім надайте третій змінній значення суми перших двох змінних і виведіть результат на екран. Результат роботи програми може виглядати наступним чином: Введіть координати точки p1 3 4 Введіть координати точки p2 5 7 Координати точки p1 + p2 рівні 8 11.

Задача 1.3 Створіть структуру з ім'ям Volume, що містить три поля типу float, для зберігання трьох вимірів приміщення. Визначте три змінні типу Volume, ініціалізуйте безпосередньо та з клавіатури, обчисліть об’єм кожногог приміщення, і виведіть результат на екран. Для підрахунку об'єму слід перемножити три дійсні змінні структури. Виведіть яка з трьох змінних має найбільший об'єм.

Задача 1.4 Створіть структуру з двома полями — цілим та дійсним числами. Оголосити дві змінні даної структури. Першу ініціалізувати напряму, а іншу ввести з клавіатури. Знайти, яке з двох змінних має максимальне ціле значення, та мінімальне дійсне значення.

Задача 1.5 Створіть структуру з двох полів {логічний та дійсний}. Оголосіть дві змінні. Якщо логічні поля двох змінних дорівнюють true – обчисліть змінну R як добуток дійсних полів цих змінних, а якщо обидва дорівнюють false - сума, у іншому випадку — R – максимальне значення.

Задача 1.6 Створіть структуру з трьох дійсних полів. Оголосіть три змінні даної структури. Дві введіть напряму, а третю через клавіатуру. Визначте, яка змінна має максимальну сумму полів.

Задача 2.1 Створити структуру даних:

```
"company": {
  "companyName": "data",
  "directorName": {
    "firsName": "data",
    "lastName": "data",

  },
```

"countofEmpoyers:”data”

}

Створити чотири компанії у векторі, знайти всі компанії, що мають більше десяти співробітників.

Задача 2.2 Описати структуру з іменем STUD, яка містить поля: NAME – прізвище та ініціали; GROUP – група; SES – оцінки з п'яти предметів (масив з п'яти елементів). Написати програму, що реалізує наступні дії окремими функціями: – введення з клавіатури даних в масив ST, що складається з N змінних типу STUD; – виведення на екран прізвищ і номерів груп для всіх студентів, середній бал яких більший за 4.0; якщо таких немає, то вивести повідомлення.

Задача 2.3 Описати структуру з іменем NOTE, яка містить поля: NAME – прізвище, ім'я; TEL – номер телефону; BDAY – день народження (масив із трьох чисел). Написати програму, що окремими функціями виконує наступні дії: – введення з клавіатури даних в масив BLOCKNOTE, що складається з N змінних типу NOTE; – виведення на екран інформації про людей, чиї дні народження припадають на місяць, значення якого введено з клавіатури; якщо таких людей немає, то вивести відповідне повідомлення.

Задача 2.4 Описати структуру з іменем NOTE, яка містить поля: NAME – прізвище, ім'я; TEL – номер телефону; BDAY – день народження (масив із трьох чисел). Написати програму, що окремими функціями виконує наступні дії: – введення з клавіатури даних в масив BLOCKNOTE, що складається з N змінних типу NOTE; – виведення на екран інформації про людей, чиї дні народження припадають на місяць, значення якого введено з клавіатури; якщо таких людей немає, то вивести відповідне повідомлення.

Задача 2.5 Сворити наступну структуру даних:

"citizen": {

"name": {

"firsName": "data",

"lastName": "data"

},

"dateOfBirth": {d,m,y},

}

Створити вектор з елементів даної структури. Прорахувати поточний вік громадян. Вивести на екран всіх громадян, ім'я яких починається на літеру 'A'.

Таблиця 6.1 — Варіанти завдань

№ варіанту	Задача № 1	Задача №2
1.	1.	1.
2.	2.	2.
3.	3.	3.
4.	4.	4.
5.	5.	5.
6.	6.	1.
7.	1.	2.
8.	2.	3.
9.	3.	4.
10.	4.	5.
11.	5.	1.
12.	6.	2.
13.	1.	3.
14.	2.	4.
15.	3.	5.
16.	4.	1.
17.	5.	2.
18.	6.	3.
19.	1.	4.
20.	2.	5.
21.	3.	1.
22.	4.	2.
23.	5.	3.
24.	6.	4.
25.	1.	5.
26.	2.	1.
27.	3.	2.
28.	4.	3.
29.	5.	4.
30.	6.	5.