

Лабораторна робота 03. Розробка структур даних

Мета: навчитися надавати в табличній формі опис структур даних

Постановка задачі.

Потрібно створити додаток/застосунок, який забезпечує введення, накопичення, коригування та оброблення інформації по викладачам і студентам гіпотетичного навчального закладу, а також пошуку інформації по викладачах та студентах. Введення може виконуватися в інтерактивному режимі або з файлу. Оброблення передбачає формування списків студентів по групах, викладачів по групах та предметах звітності з успішності.

Завдання лабораторної роботи

1. Повторити матеріал лекцій №4-5 та ознайомитися з теоретичним матеріалом, наданим до цієї лабораторної роботи.
2. Розробити структуру збереження/подання даних в програмі по викладачах, студентах, списках груп, предметах, шкалу оцінювання, звітів з успішності, які в програмі представляють опис відповідних інформаційних об'єктів.

Звіт з виконання практичної роботи повинен містити:

Титульний лист

Завдання

Звіт (безпосередньо виконання завдання)

Теоретичні відомості.

Структура — це сукупність різнотипних елементів, яким присвоюється одне ім'я (воно може бути відсутнім), що займає одну ділянку пам'яті. Елементи, що складають структуру, називаються *полями*.

Змінна типу структура, як і будь-яка змінна, повинна бути описана. Цей опис складається з двох кроків: опису шаблону (тобто складу) або типу структури та опису змінних структурного типу.

В подальшому при вивченні C++ будемо створювати структури даних як користувацький тип даних. **Користувацькі типи даних** — типи, які групують декілька окремих змінних разом. Одним з найпростіших користувацьких типів даних є структура. **Структура дозволяє згрупувати змінні різних типів даних в єдине ціле.**

Синтаксис опису структури в C++ має вигляд:

```
struct [<ім'я структури>]
{ <тип 1> ім'я поля 1;
  <тип 2> ім'я поля 2 ...;
} p1, p2 ...;
```

де **struct** — службове слово;

<ім'я структури> — ім'я типу структури (може бути відсутнім);

<тип 1>, <тип 2> — імена стандартних або визначених типів;

ім'я поля 1, ім'я поля 2,... — імена полів структури;

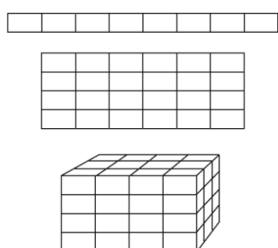
p1, p2 ...; — імена змінних типу структура.

Програмісти у своїй професійній діяльності широко використовують структури даних для зручного зберігання інформації. «Структури даних – це контейнери, які зберігають дані у певному форматі».

Структури не є єдиним агрегованим типом даних в мові C++. Є ще **масиви** — типи даних, які дозволяють отримати доступ до всіх змінних одного і того ж типу даних через використання одного ідентифікатора. «Масив — це найпростіша структура даних, що широко використовується. Інші структури даних, такі як стеки та черги, є похідними від масивів». Масив являє собою сукупність однотипних змінних, розташованих у послідовно пронумерованих суміжних комірках пам'яті. Номер елемента масиву задається індексом. Індексція елементів масиву в C++ починається з нуля. Найменший індекс відноситься до першого елемента масиву, а найбільший — до останнього. У залежності від способу зв'язування індексів з комірками пам'яті, масиви розділяються на три категорії: статичний, фіксований автоматичний, динамічний.

Таким чином, масив — це велика кількість неперервної області пам'яті, доступ до якої здійснюється за допомогою індексів (їх кількість вказує на розмірність масиву).

Його легко уявити у вигляді сукупності комірок, в яких програма зберігає значення (рисунок нижче).



Масиви є найбільш широко використовуваними структурами даних і передбачені у всіх високорівневих мовах програмування. Масив складається з елементів одного типу, званого базовим, тому структура масиву однорідна.

Базовий тип, у свою чергу, може бути як скалярним, так і структурованим або абстрактним, тобто елементами масиву можуть бути числа, символи, рядки, структури, списки, дерева, у тому числі і інші масиви. Число елементів масиву фіксовано, тому обсяг займаної масивом пам'яті залишається незмінним. З кожним елементом масиву пов'язаний один (для одновимірного масиву) або декілька індексів (для багатовимірного масиву). Вони однозначно визначають місце елемента в масиві і забезпечують прямий та швидкий доступ до цього елемента.

Індекси масиву відносяться до певного порядкового типу, тому індекси можна обчислювати. Це забезпечує, з одного боку, гнучкість обробки елементів масиву, з іншого боку, створює небезпеку виходу за межі масиву, якщо не передбачені відповідні засоби контролю (що і відбувається у мовах Cі та C++).

В залежності від числа індексів розрізняють одновимірні і багатовимірні

Методичні вказівки.

Структуру подання даних доцільно подавати в такий таблиці

№№	Рівень	Назва атрибуту	Тип даних	Довжина	Примітка

--	--	--	--	--	--

В графах "№№" записується порядковий номер атрибуту;

"Рівень" записується рівень атрибуту;

"Тип даних" визначається тип даних атрибуту:

Ц – цілий,

Д – дійсний,

С – символьний/текстовий;

"Довжина" визначається максимальна кількість знаків значення атрибуту;

"Примітка" надаються важливі відомості щодо атрибуту, наприклад, 1) якщо атрибут представляє собою масив з z елементів, то записується "масив з z елементів";

2) якщо числовий атрибут z має певні межі, то вказується діапазон значень, у вигляді $[X_1, X_2]$ для $X_1 \leq z \leq X_2$, де X_1, X_2 – конкретні числові значення;

3) якщо атрибут повинен заповнюватися за певним шаблоном, то вказується цей шаблон, наприклад для номеру телефону можна записати таким чином "номер телефону подається в міжнародному форматі у вигляді +380 XX XX XXXXX"

Приклад заповнення таблиці "Оцінювання роботи продавця мережі супермаркетів"

№№	Рівень	Назва атрибуту	Тип даних	Довжина	Примітка
1	1	Прізвище	С	25	
2	1	Ім'я	С	15	
3	1	Місце роботи			
4	2	Номер магазину	Ц	2	[1,12]
5	2	Номер відділу	Ц	2	[1,10]
6	1	Щомісячна оцінка	Ц	1	[1,5], масив з 12 елементів
7	1	Телефон	С	16	Задається у вигляді +380 XX XX XXXXX

В даному прикладі, в структуру вкладена інша структура. Вкладені структури можуть також містити масиви та дані різного типу.

Переліки атрибутів в структурах збереження/подання даних по викладачах, студентах, списках груп, предметах, шкалу оцінювання, звітів з успішності в

В структурах по викладачах та студентах обов'язковим є визначення коду особи, прізвища, імені та по-батькові, для студентів – подання оцінок по предмету, для викладачів – переліку кодів предметів.

Звіт з успішності може бути декілька, наприклад, середня оцінка по групах по предметах, кількість відмінників в групах. В звіті з успішності повинна подаватися середня оцінка по предметах.

З точки зору практичної реалізації інформацію по предмету доцільно в об'єктах опису викладачів, студентів, звітів з успішності (те що описують

структури) доцільно зберігати за кодом предмету, а окремо зробити об'єкт "Список предметів", де надати назви предметів. В цьому випадку, назва предмета буде вибиратися з цього об'єкту в разі потреби, наприклад при виведенні інформації, яка повинна містити цю назву. Окрім того, якщо в середині структури мають міститися масиви (елементів або вкладених структур), то доцільно в таблиці зробити окрему графу, де визначається розмір масиву.

В наданому нижче прикладі ці положення проілюстровано, стрілкою вказано зв'язок між об'єктами.

№№	Рівень	Назва атрибуту	Тип даних	Довжина	Розмір масиву	Діапазон значень та інші обмеження
1	1	ПІБ				
2	2	Прізвище	С	25		
3	2	Ім'я	С	15		
4	1	Предмети			6	
5	2	Код предмету	Ц	2		[1,20]
6	2	Середня оцінка по групі			3	
7	3	Код групи	С	8		[1,10]
8	3	Щомісячна оцінка	Ц	2		[0,100]
9	1	Телефон	С	16		Задається у вигляді +380 XX XX XXXXX

Список предметів

№№	Рівень	Назва атрибуту	Тип даних	Довжина	Примітка
1	1	Код предмету	Ц	2	[1,20]
2	1	Назва предмету	С	20	