УДК 004.43:075.8

С.С. Федик
В.І. Макогон

СТРУКТУРИ ДАНИХ ТА ЇХ РЕАЛІЗАЦІЯ МОВОЮ С++

Вінницький національний технічний університет;

Анотація

У тезі розглянуто основні підходи до реалізації структур даних у мові програмування С++. Проаналізовано типи даних, масиви, структури, покажчики та динамічну пам'ять. Визначено особливості роботи з файлами та обробки даних у програмних системах. Показано роль структур даних у побудові ефективних програм.

Ключові слова: С++, структури даних, масиви, покажчики, динамічна пам'ять, програмування.

Abstract

The thesis examines the main approaches to implementing data structures in the C++ programming language. Data types, arrays, structures, pointers, and dynamic memory are analyzed. The features of file handling and data processing in software systems are determined. The role of data structures in building efficient programs is shown.

Keywords: *C++, data structures, arrays, pointers, dynamic memory, programming.*

Основна частина

Структури даних є важливою складовою програмування, оскільки вони визначають спосіб організації, зберігання та обробки інформації в програмі. Ефективність роботи програм значною мірою залежить від правильного вибору структур даних і способів їх реалізації [4]. У мові програмування С++ базовими структурами даних є змінні, масиви, рядки, а також структуровані типи, які дозволяють об'єднувати різні типи даних в єдину логічну структуру [3]. Масиви використовуються для зберігання однотипних елементів і забезпечують швидкий доступ до даних за індексом [2]. Ефективність реалізації програми значною мірою залежить від правильного вибору типу структури даних, оскільки різні способи розміщення елементів у пам'яті мають різну вартість за часом доступу та за обсягом використаних ресурсів [1][2].

Однією з найпростіших структур даних у С++ є масив, який являє собою сукупність елементів одного типу, розміщених у пам'яті послідовно [2]. Ім'я масиву інтерпретується як покажчик на перший елемент, тому доступ до елементів можливий не лише через індексацію, а й через адресну арифметику [2]. Наприклад, вирази arr[1], *(arr + 1) і 1[arr] є еквівалентними, хоча з практичної точки зору рекомендовано застосовувати індексну форму запису як більш читабельну [2]. Такий зв'язок масивів і покажчиків є принципово важливим для розуміння внутрішнього механізму роботи з пам'яттю в С++, а також для реалізації алгоритмів сортування, пошуку та обходу даних [2][3].

Ще одним важливим напрямом реалізації структур даних у С++ є робота з файлами, зокрема з бінарними файлами, у яких дані записуються у вигляді байтових послідовностей без текстового перетворення [1][3]. Для роботи з бінарними файлами в бібліотеці fstream використовуються функції read і write, які приймають покажчик на масив байтів і розмір блока, що зчитується або записується [1]. Коректне використання цих функцій передбачає відкриття файлу в режимі ios::binary, інакше потік може виконувати небажані перетворення даних [1]. Поєднання структур і файлового введення-виведення є технічно важливим, оскільки дозволяє безпосередньо серіалізувати записи користувацьких типів, зберігати масиви структур у файлах та відновлювати їх у пам'яті для подальшої обробки [1][2].

Окремий інтерес становлять бітові поля, які використовуються для щільної упаковки невеликих за обсягом даних у межах машинного слова [2]. У таких полях розмір кожного елемента задається в бітах, що дає змогу економити пам'ять, але водночас робить програму більш залежною від апаратної архітектури та менш ефективною під час побітового доступу порівняно зі звичайними байтовими операціями [2]. Крім того, для бітових полів не допускається взяття адреси окремого поля, оскільки

сучасні системи адресують пам’ять на рівні байтів, а не окремих бітів [2][3].

Важливу роль відіграють також покажчики, які забезпечують ефективне використання ресурсів. Використання покажчиків тісно пов’язане з динамічним виділенням пам’яті, що дає змогу створювати гнучкі структури даних під час виконання програми [2].Динамічна пам’ять у C++ реалізується за допомогою операторів new та delete, що дозволяє програмі керувати обсягом використовуваної пам’яті залежно від потреб [4]. Це особливо важливо при роботі з великими обсягами даних.

Окрім цього, у C++ широко використовуються структури (struct), які дозволяють описувати складні типи даних, що складаються з кількох полів різних типів. Це забезпечує зручну організацію інформації та підвищує читабельність програмного коду [3]. Також важливим аспектом є робота з файлами, які дозволяють нам зберігати та обробляти дані поза межами оперативної пам’яті. Використання файлових потоків забезпечує можливість читання та запису інформації, що є необхідним для багатьох прикладних задач [4]. Таким чином, використання структур даних у мові C++ дозволяє ефективно організовувати інформацію, оптимізувати використання пам’яті та забезпечувати гнучкість програмних рішень.

Висновок

Отже, структура даних є важливим елементом програмування мовою C++. Їх використання забезпечує ефективну організацію даних та підвищує продуктивність програм. Застосування масивів, покажчиків, структур та динамічної пам’яті дозволяє створювати гнучкі та оптимізовані програмні рішення і для цього треба знати структуру даних вона є необхідною умовою для розробки сучасного програмного забезпечення та вирішення складних обчислювальних задач [2,4].

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Гришанович Т. О., Глинчук Л. Я. Основи об’єктно-орієнтованого програмування : навч. посіб. Луцьк : Волинський національний університет імені Лесі Українки, 2022. 120 с.
2. Прокопенко О. В., Попов М. О., Чумак Г. Л. Мова програмування C/C++. Практикум : навч. посіб. Київ : Київський національний університет імені Тараса Шевченка, 2024. 375 с.
3. Іванов Є. О., Ліндер Я. М., Жереб К. А. Основи мови програмування C++ : посібник першокурсника.Київ, 2020.88 с.
4. Гаркуша І. М. Конспект лекцій з дисципліни “Програмування”. Частина 1. Дніпро : НТУ «Дніпровська політехніка», 2020. 103 с.

Федик Сергійович Святослав *– студент групи ПЗТ-24б, факультету інформаційних електронних систем, Вінницького національного університету, м. Вінниця, e-mail: svyatoslavfedik.gmail.com*

Макогон Віталій Іванович *- кандидат технічних наук, старший викладач кафедри інфокомунікаційних систем і технологій, Вінницький національний технічний університет, Вінниця, makogon.v.i@vntu.edu.ua.*

Fedyk Svyatoslav S. *– student of group PZT-24B, Faculty of Information Electronic Systems, Vinnytsia National University, Vinnytsia, e-mail: svyatoslavfedik@gmail.com*

Makogon Vitaly I. *– candidate of technical sciences, senior lecturer of the departmen of information communication systems and technologies, Vinnytsia National Technical University, Vinnytsia, e-mail: makogon.v.i@vntu.edu.ua.*