**Лабораторна робота №2. Складання простих програм з використанням масивів та покажчиків.**

***Мета:*** *навчитись складати й реалізовувати алгоритми та програми мовою С++ для обробки табличних величин одновимірних та двовимірних масивів, використовуючи покажчики та динамічну пам‘ять.*

***Завдання:***

1. Ознайомитися з теоретичними відомостями та врахувати їх при виконанні завдання і в подальшій роботі.
2. Запустити середовище програмування С++ Builder (або інше доступне при виконанні вдома)
3. Створіть новий документ. Збережіть під іменем **<Номер групи><Номер лабораторної><Прізвищеанглійською>.срр**. Наприклад,3102buts.cpp
4. У вікні редагування введіть текст програми, відповідно до наданої постановки задачі.
5. В першому рядку програми записати в коментарі номер групи та прізвище
6. Результати надсилати на електронну адресу викладача [**t.i.lumpova@gmail.com**](mailto:t.i.lumpova@gmail.com)у вигляді cpp-файлу з іменем у форматі

**<Номер групи> <Номер лабораторної><Прізвищеанглійською>**

Наприклад, 31-01Ivanov.cpp.

Тему в заголовку листа записати

**ООП<Номер групи>-ЛР <Номер лабораторної>-<Прізвище>**

**Строк відсилки ЛР ІПЗ-31 10.10.2023**

**ІПЗ-32 05.10.2023**

Всі запитання, що виникнуть, надсилайте на електронну адресу викладача, Тему в заголовку листа записати

**ООП<Номер групи>-Запитання-<Прізвище>**.

**Постановка задачі:**

1. Користувач вводить з клавіатури масив **А** розміром **(3×4)** (масив можна замість введення заповнити випадковими числами).

Програма виводить на екран введений масив у вигляді матриці (по рядках, елементи рядка відокремлюються один від одного табуляцією). В елементи динамічного масиву записується підрахована сума елементів рядку масиву **А**. Виводяться на екран елементи динамічного масиву з використанням покажчиків.

2. В цій же програмі створюється масив **В** розміром **(4х3),** розміри масиву задаються іменованою константою. Дані до цього масиву заносяться ініціалізацією. Дані елементів виводяться на екран.

3. Для масиву виконується розрахунок Ai,j= Ai,j+Bj,i Дані модифікованого масиву А виводяться на екран.

4. Підраховується кількість парних та непарних елементів масиву А, які записуються до динамічно створених масивів А1 та А2, відповідно, дані масивів виводяться на екран.

**Теоретичні відомості**

Типові помилки програмування.

1. Неправильне посилання на елемент двовимірного масиву **a[x][y]** як **a[x,y].** В дійсності С++ розцінює вираз **a[x,y]** як **a [y]**, сприймаючи вираз у квадратних скобках **x,y** як **y.**
2. Використання арифметичних дій з покажчиками, які не посилаються на елементи масиву.
3. Використання операцій віднімання або порівняння з покажчиками, які не посилаються на елементи одного і того ж масиву.
4. Вихід за межі масиву при виконанні арифметичних дій з покажчиками.
5. Присвоювання покажчика одного типу покажчику іншого типу (відмінного від void) викликає синтаксичну помилку.
6. Розіменування покажчика на void.
7. Хоч імена масивів є покажчиками на їх початок, а покажчики в арифметичних виразах можна модифікувати, імена масивів в цих виразах модифікувати не можна. Ім'я масиву є постійним покажчиком на початок масиву і вираз типу b+=3; є недопустимим, оскільки воно намагається модифікувати значення імені масиву за допомогою покажчика.

Хороший стиль програмування

1. В програмах повинна забезпечуватись перевірка правильності всіх значень, що вводяться, для унеможливлення впливу помилкової інформації на хід обчислення.
2. Деякі програмісти для спрощення розуміння програми включають імена змінних у прототипи функцій. Компілятор ігнорує ці імена.
3. Використовуйте нотацію масиву замість нотації покажчиків при маніпуляції з масивами. Програма буде зрозумілішою.

Нехай маємо

int b[5];

int bPtr;

Адреса першого елементу масиву b

bPtr= b;

Це еквівалентно

bPtr=&b[0];

Адрес &b[3] може бути записаний як bPtr+3 або \*(b+3)

В останньому випадку запис покажчик-зміщення застосовується до імені масиву як до покажчика.

**Заповнення масиву випадковими числами**

В С ++ для цього є спеціальні фyнкції **rand()**. Вони знаходяться в бібліoтечному файлі cstdlib, тому щоб їх застосовувати в програмі, необхідно підключити цей бібліотечний файл: #include <cstdlib>.

Окрім того, для ця функція може використовувати поточний системний час через використання функції time(), для якої необхідно підключити бібліотечний файл ctime: #include <ctime>.

Діапазон випадкових чисел, які генеруються таким чином – від 0 до 32767. Можливо вам знадобиться заповнити масив числами від 200 дo 300, від 0.1 дo 1, від -20 дo 20. Таку генерацію випадкових чисел можливо і нескладно реалізувати. У прикладі нижче розглянуто кілька випадків:

**#include <iostream>**

**#include <cstdlib>**

**#include <ctime>**

**using namespace std;**

**int main()**

**{**

**int randomDigits[10];**

**int randomDigits\_2[10];**

**int randomDigits\_3[10];**

**int randomDigits\_4[10];**

**float randomDigits\_5[10]; // для чисел з плаваючею крапкою**

**srand(time(NULL));**

**for (int i = 0; i < 10; i++)**

**{**

**randomDigits[i] = rand() % 7; // 0 ... 6**

**randomDigits\_2[i] = 1 + rand() % 7; // 1 ... 7**

**randomDigits\_3[i] = 200 + rand() % 101; // 200 ... 300**

**randomDigits\_4[i] = rand() % 41 - 20; // -20 ... 20**

**randomDigits\_5[i] = 0.01 \* (rand() % 101);// 0.01 ... 1**

**}**

**cout << "Масив з числами від 0 до 6: ";**

**for (int i = 0; i < 10; i++)**

**{**

**cout << randomDigits[i] << " ";**

**}**

**cout << endl << "Масив з числами від 1 до 7: ";**

**for (int i = 0; i < 10; i++)**

**{**

**cout << randomDigits\_2[i] << " ";**

**}**

**cout << endl << "Масив з числами від 200 дo 300: ";**

**for (int i = 0; i < 10; i++)**

**{**

**cout << randomDigits\_3[i] << " ";**

**}**

**cout << endl << "Масив з числами від -20 до 20: ";**

**for (int i = 0; i < 10; i++)**

**{ cout << randomDigits\_4[i] << " "; }**

**cout << endl << "Масив з числами від 0.01 дo 1: ";**

**for (int i = 0; i < 10; i++)**

**{ cout << randomDigits\_5[i] << " "; }**

**cout << endl;**

**}**

**Попередні відомості про Методику виконання курсової**.

**ЗМІСТ КУРСОВОЇ РОБОТИ**

Рекомендована наступна структура курсової роботи:

- титульний аркуш – 1 стор.;

- завдання на курсове проектування – 1 аркуш;

- календарний план – 1 аркуш;

- реферат – до 1 стор.;

- зміст – 1-2 стор.;

- перелік скорочень, умовних позначень, символів, одиниць і термінів – до 1 с.;

- вступ – 2-3 стор.;

- суть – 15-30 стор.:

теоретична частина (до 14 сторінок) ;

практична частина(до 14 сторінок) ;

- висновки – 1 стор.;

- список використаних джерел (перелік посилань) – 1 стор.;

- додатки.

Реферат

Реферат призначений для ознайомлення з курсовою роботою. Він має бути стислим, інформативним і містити відомості, які дозволяють прийняти рішення про доцільність читання всієї курсової роботи.

Реферат (Додаток 5) повинен бути не менше 0,5 сторінки, але й не більше 1 сторінки і містити:

– дані про кількість сторінок, ілюстрацій, таблиць, використаних джерел, додатків,

– перелік ключових слів, записаних великими літерами,

– стислий опис роботи.

У переліку ключових слів зазвичай наводять 5–15 слів або словосполучень, необхідних для розкриття суті роботи. Їх записують за абеткою мови роботи великими літерами в рядок із прямим порядком слів у називному відмінку однини та розділяють комами.

Стислий опис роботи зображує зміст роботи в такій послідовності:

– об’єкт дослідження або розроблення;

– мета роботи;

– методи та інструменти дослідження або розроблення;

– отримані результати та їх значення;

– інформація щодо впровадження;

– рекомендації щодо використання результатів роботи;

– висновки та пропозиції щодо розвитку об’єкта дослідження (розроблення) й доцільності продовження досліджень або розробок.

Якщо деякі з наведених відомостей відсутні, то інші викладають, зберігаючи вказану послідовність.

Вступ

У вступі студент обґрунтовує вибір теми, коротко викладає її актуальність, призначення розробки, для вирішення яких конкретних практичних завдань вона може бути використана. Тут потрібно сформулювати мету і завдання роботи, визначити основні підходи та ідеї, вибрати спосіб розв'язання задач.

У вступі слід акцентувати увагу на прикладній проблемі, яку треба вирішити в курсовій роботі. У цьому розділі недоцільно наводити означення відомих термінів об'єктно-орієнтованого програмування, давати надто детальні характеристики й описи використаного програмного забезпечення та іншу інформацію, що мало стосується теми курсової роботи.

Теоретична частина

Теоретична частинаскладається з наступних елементів:

• постановка задачі та її опис;

• виявлення особливостей, взаємозв‘язків і кількісних закономірностей, аналіз вхідної та вихідної інформації;

• побудова математичної (або інформаційно-логічної) моделі програмного додатку;

• опис алгоритму;

• аналіз існуючих вітчизняних та зарубіжних програмних засобів для вирішення даної проблеми.

Постановка задачі (далі - ПЗ)та її опис.

Потрібно розкрити такі аспекти:

* Мета розробки ПЗ, призначення ПЗ
* Вхідні дані.
* Вихідні дані, очікувані результати.
* Функції обробки інформації (коротка технологічна схема за принципом введення інформації → контроль інформації → збереження/накопичення → оброблення → формування результатів ), які відображують через які процеси вхідні дані трансформуються у вихідні.
* Вимоги до якості (коротко, потім вони розкриваються у функціональних та нефункціональних вимогах).
* Інші вимоги.

Функціональні вимоги – це перелік функцій, який повинен забезпечити застосунок, який розроблюється в КР, наприклад:

• відображення даних про перелік об'єктів у табличному виді;

• ввід даних про новий об'єкт;

• зміну даних про попередньо введені об'єкти;

• видалення попередньо введених об'єктів по одному;

• повне очищення всього переліку за однією командою;

• відображення результату роботи обох методів класу;

• збереження даних про перелік об'єктів у двійковому форматі;

• читання попередньо збережених даних у двійковому форматі і відображення їх у табличному виді;

• експорт даних про перелік об'єктів у текстовий файл;

• імпорт даних з текстового файлу і відображення їх у табличному виді;

• сортування переліку даних за кількома найважливішими параметрами;

• пошук запису у переліку за одним з найважливіших параметрів;

• фільтрування даних у переліку за одним з найважливіших параметрів.

Виявлення особливостей, взаємозв‘язків і кількісних закономірностей, аналіз вхідної та вихідної інформації

Побудова математичної (або інформаційно-логічної) моделі програмного додатку

Опис алгоритму

Аналіз існуючих вітчизняних та зарубіжних програмних засобів для вирішення даної проблеми

Практична частина

Практична частинаскладається з наступних елементів:

• UML-діаграма структури програмного продукту (діаграма класів);

• аналіз використаних конструктивних елементів програмного додатку;

• опис користувацького інтерфейсу;

• опис порядку тестування та вимоги до контрольного прикладу;

• результати розрахунків за допомогою створеної програми.

Висновки

Список використаних джерел

Додатки

* Код програми (обов‘язково)
* Тестовий набір даних та контрольний приклад (обов‘язково)
* Скріншот виконання програми (обов‘язково)