

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ПРОГРАМУВАННЯ

МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ КОМП'ЮТЕРНИХ ПРАКТИКУМІВ

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для студентів,
які навчаються за спеціальністю 171 «Електроніка»,
спеціалізацією «Електронні системи»*

Київ
КПІ ім. Ігоря Сікорського
2017

Програмування: Методичні вказівки до виконання комп'ютерних практикумів [Електронний ресурс]: навч. посіб. для студ. спеціальності 171 «Електроніка», спеціалізації «Електронні компоненти і системи» / КПІ ім. Ігоря Сікорського ; уклад.: О.О.Абакумова, Д.Р.Ушаков – Електронні текстові данні (1 файл: 331 кбайт). – Київ : КПІ ім. Ігоря Сікорського, 2017. – 50 с.

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № X від DD.MM.YYYY р.)
за поданням Вченої ради факультету електроніки (протокол № X від DD.MM.YYYY р.)*

Електронне мережне навчальне видання

ПРОГРАМУВАННЯ

МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ

КОМП'ЮТЕРНИХ ПРАКТИКУМІВ

Укладачі: *Абакумова Олена Олегівна, канд. ф. наук*
Ушаков Дмитро Ростиславович

Відповідальний редактор *Хижняк Т.А., доцент кафедри промислової електроніки*
КПІ ім. Ігоря Сікорського, канд. техн. наук, доцент

Рецензенти: *Казміренко В.А., доцент кафедри електронної інженерії*
КПІ ім. Ігоря Сікорського, канд. техн. наук, доцент

Методичні вказівки містять роз'яснення щодо виконання 7 комп'ютерних практикумів, передбачених робочою навчальною програмою кредитного модуля «Програмування».

Кожна робота містить 20 варіантів завдань однакового ступеню складності. Наводяться необхідні теоретичні відомості та приклад програмної реалізації типового завдання мовою програмування С++. Для самоперевірки пропонуються контрольні питання. Для самостійної роботи студентів до кожної роботи наводиться список рекомендованої літератури.

© КПІ ім. Ігоря Сікорського, 2017

ЗМІСТ

Зміст.....	3
Вступ.....	4
Комп'ютерний практикум №1 «Динамічні змінні»	5
Комп'ютерний практикум №2 «Структури».....	26
Комп'ютерний практикум №3 «Класи».....	34
Комп'ютерний практикум №4 «Наслідування».....	44
Комп'ютерний практикум №5 «Перевантаження операторів».....	51
Додатки.....	58

ВСТУП

Мова програмування C++ призначена для розробки високопродуктивного програмного забезпечення та дуже популярна серед програмістів. C++ можна назвати універсальною мовою програмування, оскільки вона забезпечує концептуальний фундамент (синтаксис та стиль), на який спираються інші мови програмування. Вивчивши C++, студенти отримують фундаментальні знання, які дозволять їм у подальшому опанувати будь-які аспекти сучасного програмування.

Дані методичні вказівки складені у відповідності до програми кредитного модуля «Інформатика-2. Програмування та алгоритмічні мови». Розглянуто 7 практикумів, присвячених ключовим аспектам мови програмування C++, таким як цикли, масиви, функції, класи, наслідування та перевантаження. Основні необхідні теоретичні відомості для виконання роботи підкріплюються прикладами програмної реалізації типового завдання мовою C++ та питаннями для самоконтролю. Для самостійної роботи студентів до кожної роботи наводиться список рекомендованої літератури.

Основна мета посібника – надати студентові допомогу в опануванні теоретичних положень курсу та в оволодінні основними прийомами програмування мовою C++, що вивчаються навчальною дисципліною.

Для студентів напряму підготовки: 6.050802 «Електронні пристрої та системи» всіх форм навчання.

КОМП'ЮТЕРНИЙ ПРАКТИКУМ №1

Тема: Багатомірні масиви

Мета роботи – розробка програмного забезпечення з реалізації алгоритмів обробки багатомірних динамічних масивів даних: зберігання, сортування, пошук.

Завдання: Задано двомірний масив дійсних чисел m на n . Пам'ять для елементів масиву необхідно виділити динамічно. Елементи масиву ініціалізуються або вводяться з клавіатури. Необхідно реалізувати вивільнення пам'яті після завершення роботи з масивом.

1. Знайти найбільший елемент масиву. Вивести на екран вихідний масив і новий, в якому елементи стовпчика, що містить найбільший елемент, помножено на цей елемент.
2. Знайти найменший елемент масиву. Вивести на екран вихідний масив і новий, в якому елементи рядка, що містить найменший елемент, помножено на цей елемент.
3. Знайти номери рядка й стовпця, які містять найбільший елемент масиву. Вивести на екран вихідний масив і новий, елементи якого по стовпчиках впорядковано за зростанням.
4. Знайти номери рядка й стовпця, які містять найменший елемент масиву. Вивести на екран вихідний масив і новий, елементи якого по рядках впорядковано за спаданням.
5. Визначити добуток елементів у рядках, що не містять нульових елементів. Вивести на екран вихідний масив і новий, в якому всі додатні елементи замінено відповідними від'ємними.
6. Визначити суму елементів в стовпчиках, що не містять від'ємних елементів. Вивести на екран вихідний масив і новий, в якому всі від'ємні елементи замінено відповідними додатними.

7. Визначити номери рядків без нульових елементів. Вивести на екран вихідний масив і новий, в якому елементи вихідного масиву помножено на задане дійсне число.
8. Визначити номери стовпчиків без нульових елементів. Вивести на екран вихідний масив і новий, в якому до елементів першого рядка додано значення середнього арифметичного його елементів.
9. Визначити номер рядка, в якому знаходиться найбільша кількість заданих елементів. Вивести на екран вихідний масив і новий, елементи якого впорядковано за зростанням.
10. Визначити стовпчик з найбільшою кількістю нульових елементів. Вивести на екран вихідний масив і новий, елементи якого впорядковано за спаданням.
11. Знайти максимальний елемент першого рядка масиву. Вивести на екран вихідний масив і новий, в якому всі додатні елементи помножено на знайдене число.
12. Знайти мінімальний елемент останнього стовпчика масиву. Вивести на екран вихідний масив і новий, в якому всі від'ємні елементи помножено на знайдене дійсне число.
13. Знайти кількість нульових елементів в кожному стовпчику масиву. Вивести на екран вихідний масив і новий, в якому всі нульові елементи замінено заданим дійсним числом.
14. Знайти кількість ненульових елементів в кожному рядку масиву. Вивести на екран вихідний масив і новий, в якому елементи рядка, що містить найбільшу кількість ненульових елементів, помножено на знайдене дійсне число.
15. Знайти кількість від'ємних елементів в кожному стовпчику масиву. Вивести на екран вихідний масив і новий, в якому елементи стовпчика, що

містить найбільшу кількість від'ємних елементів, замінено заданим цілим числом.

16. Знайти кількість додатних елементів в кожному рядку масиву. Вивести на екран вихідний масив і новий, в якому елементи рядка, що містить найменшу кількість додатних елементів, помножено на задане ціле число.

17. Знайти кількість елементів, що перевищують значення середнього арифметичного елементів масиву. Вивести на екран вихідний масив і новий, в якому всі елементи, менші за значення середнього арифметичного, замінено нулями.

18. Визначити стовпчик з найбільшою кількістю ненульових елементів. Вивести на екран вихідний масив і новий, в якому до елементів стовпчика з найменшою кількістю ненульових елементів додано число 5.

19. Знайти максимальний елемент головної діагоналі масиву. Вивести на екран вихідний масив і новий, в якому елементи першого стовпчика помножено на знайдене число.

20. Знайти мінімальний елемент головної діагоналі масиву. Вивести на екран вихідний масив і новий, в якому всі елементи головної діагоналі замінено на знайдене число.

Теоретичні відомості

У C++ можна використовувати багатомірні масиви. Найпростіший багатомірний масив – двомірний. Двомірний масив, по суті, являє собою список одномірних масивів.

Визначення двомірного масиву

Загальний синтаксис оголошення двомірного масиву наступний:

тип_даних ім'я_масиву [розмір1][розмір2];

Наприклад,

```
int b[10][10]; // оголошення двомірного масиву цілих чисел розміром 10x10  
double s[3][4]; // оголошення двомірного масиву дійсних чисел розміром 3x4
```

Зверніть увагу! При оголошенні двомірного масиву у C++ кожна розмірність укладається у власну пару квадратних дужок.

Доступ до елементів двомірного масиву

Для роботи з масивом його елементи нумерують. Номер позиції елемента всередині масиву називають його *індексом*. У C++ індексація масивів починається з нуля.

У двомірному масиві позиція будь-якого елемента визначається двома індексами. Якщо уявити двомірний масив у вигляді таблиці, то перший індекс (за угодою) означає рядок, а другий – стовпчик. З цього випливає, що, якщо доступ до елементів масиву надати у порядку, в якому вони реально зберігаються у пам'яті, то правий індекс буде змінюватися швидше, ніж лівий.

Для звернення до певного елемента двомірного масиву вказують ім'я масиву та індекси елемента, укладені у квадратні дужки:

```
ім'я_масиву [індекс1][індекс2];
```

Наприклад,

```
int b[3][5]; // доступ до елемента двомірного масиву з координатами 3, 5  
float list[2][3]; // доступ до елемента двомірного масиву з координатами 2, 3
```

Ініціалізація двомірного масиву

Багатомірні масиви можуть отримувати початкові значення у своїх оголошеннях так само, як одномірні масиви, за допомогою списку ініціалізації.

Формат ініціалізації двомірного масиву базується на тому факті, що двомірний масив, по суті, являє собою список одномірних масивів. Ініціалізуючі значення для кожного підмасиву укладають у фігурні дужки,

розділяючи комами, а потім всі ці підмасиви також укладають у дужки та розділяють комами.

Наприклад,

```
int b[2][3]={
    {1, 2, 3},
    {5, 6, 7},
}; // ініціалізація двомірного масиву цілих чисел розміром 2x3
```

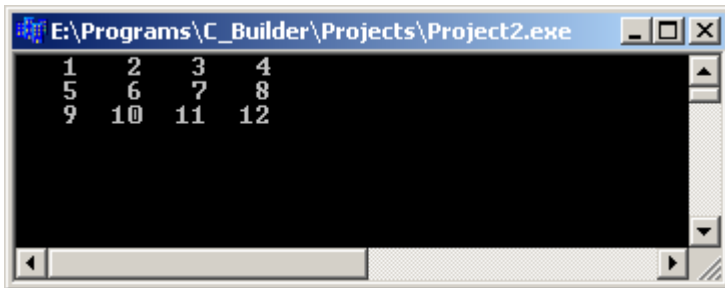
Якщо початкових значень у підмасиві не вистачає, то решті елементів автоматично присвоюється нульове початкове значення.

Приклад програми

Завдання: Задано двомірний масив дійсних чисел 3 на 4. Ініціалізувати елементи масиву послідовністю чисел від 1 до 12.

```
#include <iostream>
#include <iomanip>
#include <conio.h>
using namespace std;
void main()
{
    int i, j, mas[3][4];
    for(i=0; i < 3; ++i)
    {
        for(j=0; j < 4; ++j)
        {
            mas[i][j] = (i*4)+j+1;
            cout << setw(4)<<mas[i][j];
        }
        cout << endl;
    }
}
```

```
    }  
    getch ( ) ;  
}
```



Контрольні питання

1. Що називають масивом?
2. Які масиви називають двомірними?
3. Наведіть синтаксис оголошення двомірного масиву.
4. Наведіть синтаксис звернення до певного елемента двомірного масиву.
5. В який спосіб можна задавати початкові значення елементам масиву?
6. Як ініціалізувати двомірний масив за допомогою списку ініціалізації?
7. Як ініціалізувати двомірний масив за допомогою циклу for?
8. Які початкові значення отримають елементи двомірного масиву у наступному випадку ініціалізації: `array[2][3] = {1, 2, 3, 4, 5, 6};`
9. Які початкові значення отримають елементи двомірного масиву у наступному випадку ініціалізації: `array[2][3]= {{1, 2}, {4}};`
10. Чим визначається обсяг пам'яті, що виділяється для багатомірного масиву?
11. Які операції над багатомірними масивами підтримуються C++?

Рекомендована література

- Архангельский А.Я.** C++Builder 6. Справочное пособие. Книга 1. Язык C++. -- М.: Бином-Пресс, 2002 г. — 544 с.: ил. – с. 176;
- Х. Дейтел, П. Дейтел** Как программировать на C++. Пер. с англ. – М.: ЗАО

«Издательство БИНОМ», 2000 г. – 1008 с. – С. 290;

Р. Лафоре Объектно-ориентированное программирование в C++. Классика Computer Science. 4-е изд. – СПб.: Питер, 2008. – 928 с. – 10 000 экз. – ISBN 978-5-94723-302-5. – С. 267;

С. Прата Язык программирования C++. Лекции и упражнения. Учебник: Пер. с англ./ Стивен Прата – СПб.: ООО «ДиаСофтЮП», 2005. – 1104 с. – ISBN 5-93772-123-3. – С. 125;

К. Рейсдорф, К. Хендерсон Borland C++Builder. Освой самостоятельно – М.: «Издательство БИНОМ», 1998. – 704 с.: ил. – 6 000 экз. – ISBN 5-7989-0099-1. – С. 45;

Г. Шилдт C++: руководство для начинающих, 2-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 672 с.: ил. – Парал. тит. англ. – 3 000 экз. – ISBN 5-8459-0840-X. – С. 163;

КОМП'ЮТЕРНИЙ ПРАКТИКУМ №2

Тема: Структури

Мета роботи – розробка програмного забезпечення з реалізації алгоритмів із використанням структурного типу даних

Завдання: Написати програму, яка реалізує роботу зі структурними змінними.

1. У довідці аеропорту зберігається розклад рейсів на поточну добу. Для кожного рейсу вказаний час відправлення, авіакомпанія, номер рейсу, пункт призначення. Вивести інформацію про всі літаки, що вилітають у вказаний пункт призначення пізніше заданого часу.

2. У списку абітурієнтів вказані прізвища та набрані бали з трьох предметів: математики, фізики, української мови. Вивести список всіх абітурієнтів, сумарний бал яких вище за прохідний. Прохідний бал ввести з

клавіатури.

3. У списку студентів вказані прізвища та екзаменаційні оцінки з трьох дисциплін: математики, англійської мови, філософії. Виключити зі списку всіх студентів, які мають середній бал нижче 4. Вивести оновлений список.
4. У довідці аеропорту зберігається розклад рейсів на поточну добу. Для кожного рейсу вказаний час відправлення, авіакомпанія, номер рейсу, пункт призначення. Вивести інформацію про всі літаки вказаної авіакомпанії, що вилітають пізніше заданого часу.
5. У списку студентів вказані їхні імена, прізвища та екзаменаційні оцінки з трьох дисциплін: фізики, математики, хімії. Вивести даний список у порядку спадання сумарного балу студентів.
6. У відділі кадрів зберігається список викладачів університету, в якому вказані П.І.Б., вік, посада, факультет, на якому працює викладач. Вивести список докторів наук заданого факультету.
7. В ДАІ зберігається інформація про зареєстровані авто: ПІБ власника, район реєстрації, марка автомобіля, рік випуску, колір, державний номер. Вивести інформацію про всі автомобілі певної марки, що зареєстровані у заданому районі.
8. У відділі кадрів підприємства зберігається картотека на працівників, що містить таку інформацію: П.І.Б. працівника, стать, вік, адреса, розмір посадового окладу. Вивести інформацію про всіх працівників підприємства, старших за 50 років, що не є киянами.
9. В оргкомітеті студентської олімпіади є відомості про її учасників: ПІБ, факультет, курс, група, оцінка роботи у балах. Вивести інформацію про учасників з певного факультету, які отримали бал, не нижчий заданого.
10. В ДАІ зберігається інформація про зареєстровані авто: ПІБ власника, марка автомобіля, рік випуску, колір, державний номер. Вивести інформацію

про всі автомобілі заданої марки в порядку зростання року випуску.

11. Каталог бібліотеки містить інформацію про наявні книжки: прізвище автора, назва книги, видавництво, рік видання, кількість сторінок. Вивести інформацію про всі книжки певного автора, які видано пізніше заданої дати.

12. У відділі кадрів підприємства зберігається картотека на працівників, в якій зазначено: П.І.Б. працівника, стать, вік, місце проживання, розмір посадового окладу. Вивести інформацію про всіх працівників, які мешкають у Києві й отримують зарплату вищу за задану.

13. У довідці залізничного вокзалу зберігається розклад руху потягів на поточну добу. Для кожного потягу вказаний його номер, час відправлення, пункт призначення, час у дорозі. Вивести інформацію про всі потяги до вказаного пункту призначення, час у дорозі яких не перевищує заданого.

14. У фонді зайнятості ведеться облік вакантних робочих місць. Кожний запис містить назву організації, вакантну посаду, розмір посадового окладу, потрібний стаж роботи. Клієнт вводить інформацію про посаду та стаж роботи і повинен отримати інформацію про наявні вакансії.

15. У списку студентів вказані прізвища та екзаменаційні оцінки з трьох дисциплін: фізики, математики, іноземної мови. Вивести список студентів, які мають середній бал нижчий 4.

16. У відділі кадрів зберігається список викладачів університету, в якому зазначено П.І.Б. викладача, його вік, посаду, факультет, на якому працює викладач. Вивести список доцентів, молодших за 55 років.

17. У прас-листі сервісного центру зберігається інформація про комплектуючі деталі до мобільних телефонів, що випускаються компанією: номер моделі телефону, номер самої деталі та її вартість. Вивести список комплектуючих до певної моделі телефону, чия вартість лежить у заданому ціновому діапазоні.

18. У деканаті є відомості про результати першої атестації студентів кафедри: ПІБ, група, атестація з математики, фізики та іноземної мови. Вивести список всіх студентів, які мають незадовільні атестації з трьох дисциплін.

19. Відомість успішності студентів містить інформацію про П.І.Б. студента, курс, номер групи, номер залікової книжки, середній бал за останню сесію. Вивести списки студентів за групами. В кожній групі прізвища студентів розмістити в порядку спадання середнього бала.

20. В електронному каталозі смартфонів, наявних у продажу, зазначено: марка смартфона, колір, формфактор, операційна система, вага, ціна. Необхідно вивести список усіх смартфонів НТС, вартість яких лежить у заданому ціновому діапазоні.

Теоретичні відомості

Структура – це складений тип даних, побудований з використанням даних інших типів. Структура є об'єднанням декількох змінних. Ці змінні можуть мати різні типи: `int`, `char`, `float` тощо. Саме різноманітністю типів змінних структури відрізняються від масивів, в яких всі змінні повинні мати однаковий тип. Змінні, що входять до складу структури, називають *полями* структури.

Визначення структури

Структура створюється за допомогою ключового слова **`struct`**.

Загальний формат оголошення структури має наступний вигляд:

`struct ім'я_структури`

`{`

`// поля структури`

`};`

Оголошення полів структури укладають у фігурні дужки. Визначення структури закінчується крапкою з комою (;).

Наприклад,

```
struct student  
{  
  char fio[30];      //прізвище, ім'я, по-батькові  
  int data;         // рік народження  
  int rik;          // рік вступу  
  char grupa[10];  // група  
  int kurs;        // курс  
};
```

Визначення структури не створює ніяких змінних. Воно задає внутрішню організацію структурних змінних. Структуру можна вважати новим типом даних. Тож, визначення структури необхідно для того, щоб на його основі створювати змінні складеного типу.

Визначення структурних змінних

Визначення структурної змінної за своїм синтаксисом ідентичне визначенню змінної стандартного типу. Адже ми можемо вважати структуру новим типом даних.

Наприклад,

```
student st1, st2;    //визначення змінних st1 і st2 типу student
```

Інший спосіб визначення структурних змінних

Структурні змінні можна визначати також безпосередньо у визначенні структури. Для наведених вище прикладів це буде виглядати так:

```
struct  
{  
  char fio[30];  
  int data;
```

```
int rik;  
char grupa[10];  
int kurs;  
} st1, st2;
```

Доступ до полів структури

Коли структурна змінна визначена, доступ до її полів можливий за допомогою *операції крапка* (.). Операцію крапка називають *операцією доступу до поля структури*.

Поле структури ідентифікується за допомогою трьох складових: імені структурної змінної, операції крапки (.) та імені поля.

Наприклад,

```
st1.rik = 2006;           // полю rik змінної st1 присвоюється значення 2006  
st2.fio = "Ivanov";     // полю fio змінної st2 присвоюється значення Ivanov
```

Зверніть увагу! На першому місці у виразі стоїть не ім'я структури, а ім'я структурної змінної.

З полями структурної змінної можна поводитися так само, як із звичайними простими змінними.

Наприклад,

виведення значення поля на екран відбувається за допомогою оператора cout:

```
cout << "Rik= " << st1.rik;
```

присвоювання – за допомогою оператора присвоювання (=):

```
st2.rik = st1.rik;
```

Також можливо присвоювати значення однієї структурної змінної іншій структурній змінній:

```
st2 = st1;
```

В даному випадку значення кожного поля змінної st1 буде присвоєне відповідному полю змінної st2.

Зверніть увагу! Операція присвоювання може бути виконана лише над

змінними, що мають один і той самий тип.

Приклад програми

Завдання: У сервісному центрі зберігаються квитанції про здану у ремонт апаратуру. В кожній квитанції вказано: торгівельну марку приладу, дату прийому замовлення, стан готовності замовлення (виконано - не виконано). Необхідно вивести інформацію про виконання замовлень на поточну дату.

```
#include <iostream>
#include <conio.h>
#include <iomanip>
using namespace std;
struct
{
    char marka[20];
    int day;
    int month;
    char stan[5];
}remont[6]={{ "LG",13,4,"no"}, {"Sony",2,5,"no"},
            {"Sharp",10,1,"yes"}, {"LG",15,3,"yes"},
            {"Samsung",19,2,"yes"}, {"Sony",12,1,"no"} };
void main()
{
    int i;
    cout << "Database:\n";
    for (i=0;i<6;i++)
    cout << setw(12) << remont[i].marka
        << setw(5) << remont[i].day
        << "/" << remont[i].month
        << setw(6) << remont[i].stan << endl;
```

```

int d, m;
cout << "\nEnter date";
cout << "\nEnter day: ";
cin >> d;
cout << "Enter month: ";
cin >> m;
cout << "\nNew base:\n";
for (i=0;i<6;i++)
    if (remont[i].month < m || (remont[i].month == m &&
remont[i].day <= d))
        cout << setw(12) << remont[i].marka
            << setw(5) << remont[i].day
            << "/" << remont[i].month
            << setw(6) << remont[i].stan << endl;
getch();
}

```

```

E:\Programs\C_Builder\Projects\Project2.exe
Database:
   LG   13/4   no
  Sony   2/5   no
  Sharp 10/1   yes
   LG   15/3   yes
 Samsung 19/2   yes
  Sony  12/1   no

Enter date
Enter day: 16
Enter month: 3

New base:
  Sharp 10/1   yes
   LG   15/3   yes
 Samsung 19/2   yes
  Sony  12/1   no

```

Контрольні питання

1. Що таке структура?
2. Як називають змінні, що входять до складу структури?

3. Яке ключове слово використовують для визначення структури?
4. Наведіть загальний формат оголошення структури.
5. Як отримати доступ до полів структури?
6. Для чого використовують тег (ім'я) структури?
7. Як оголосити структурну змінну?
8. Як ініціалізувати структурну змінну?

Рекомендована література

- Архангельский А.Я.** C++Builder 6. Справочное пособие. Книга 1. Язык C++. -- М.: Бином-Пресс, 2002 г. — 544 с.: ил. – с. 179;
- Х. Дейтел, П. Дейтел** Как программировать на C++. Пер. с англ. – М.: ЗАО «Издательство БИНОМ», 2000 г. – 1008 с. - С. 408;
- Р. Лафоре** Объектно-ориентированное программирование в C++. Классика Computer Science. 4-е изд. – СПб.: Питер, 2008. – 928 с. – 10 000 экз. – ISBN 978-5-94723-302-5. – С. 142;
- С. Прата** Язык программирования C++. Лекции и упражнения. Учебник: Пер. с англ./ стивен прата – СПб.: ООО «ДиаСофтЮП», 2005. – 1104 с. – ISBN 5-93772-123-3. – С. 139;
- К. Рейсдорф, К. Хендерсон** Borland C++Builder. Освой самостоятельно – М.: «Издательство БИНОМ», 1998. – 704 с.: ил. – 6 000 экз. – ISBN 5-7989-0099-1. – С. 69;
- Г. Шилдт** C++: руководство для начинающих, 2-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 672 с.: ил. – Парал. тит. англ. – 3 000 экз. – ISBN 5-8459-0840-X. – С. 421;
- Б. Эккель** Философия C++. Введение в стандартный C++. 2-е изд. – СПб.: Питер, 2004. – 572 с.: ил. – 3 500 экз. – ISBN 5-94723-763-6. – С. 137.

Тема: Класи

Мета роботи – розробка програмного забезпечення з реалізації алгоритмів роботи з класами

Завдання: Написати програму для тестування класу.

1. Клас `Int` імітує стандартний тип даних `int`. Єдине поле цього класу повинне мати тип `int`. Передбачити конструктори класу та методи, що будуть ініціалізувати його об'єкти нулем, цілим значенням, виводити значення на екран, а також додавати, віднімати, множити і ділити два цілих числа.
2. Клас `Float` імітує стандартний тип даних `float`. Єдине поле цього класу повинне мати тип `float`. Передбачити конструктор класу та методи, що будуть ініціалізувати об'єкти класу, виводити значення на екран, а також додавати і віднімати два дійсних числа.
3. Клас `Vector` імітує роботу із векторами у двовимірному просторі. Полями цього класу повинні бути чотири поля типу `int`, призначені для зберігання координат точок початку і кінця вектора. Передбачити конструктор класу та методи для виведення інформації на екран, а також для знаходження координат вектора та порівняння (`=` чи `≠`) двох векторів на рівність.
4. Клас `Date` моделює роботу з календарем. Полями цього класу повинні бути три поля типу `int`: рік, місяць і день. Передбачити конструктор класу, методи ініціалізації та виведення дати у форматі `dd/mm/yy`, а також декремента поточної дати на один день. При цьому необхідно врахувати можливий перехід до попереднього місяця та року. [Лафоре, 258]
5. Клас `Time` моделює роботу з годинником. Полями цього класу повинні бути три поля типу `int`, призначені для зберігання годин, хвилин, секунд. Один з конструкторів має ініціалізувати годинник нульовими значеннями, а інший – заданим набором значень. Передбачити методи класу для виведення

часу на екран у форматі 23:59:59, а також інкремента хвилин поточного часу. При цьому необхідно врахувати можливий перехід до наступної години.

6. Клас Complex імітує роботу з комплексними числами. Полями цього класу повинні бути два поля типу `int`, призначені для зберігання дійсної й уявної частини комплексного числа. Передбачити конструктор за замовчуванням, методи класу для віднімання та множення двох комплексних чисел, а також виведення комплексного числа на екран.

7. Клас RatNum імітує роботу зі звичайними дробами. Полями цього класу повинні бути два поля типу `int`: чисельник і знаменник дроби. Передбачити конструктор класу та методи, що будуть ініціалізувати об'єкти класу, ділити і множити два дроби, а також виводити на екран значення у форматі `a/b`.

8. Клас Account моделює роботу з поточним банківським рахунком. Полями цього класу повинні бути прізвище власника рахунку та сума на рахунку. Передбачити конструктор класу та методи для перевірки існування певного рахунку, зняття певної суми з рахунку, за умови наявності необхідної суми, а також виведення інформації на екран.

9. Клас Vector2D імітує роботу із векторами у двовимірному просторі. Полями цього класу повинні бути два поля типу `int`, призначені для зберігання координат вектора. Передбачити конструктор класу та методи для виведення інформації на екран, а також для знаходження довжини вектора та добутку вектора на число.

10. Клас Complex імітує роботу з комплексними числами. Полями цього класу повинні бути два поля типу `int`, призначені для зберігання дійсної й уявної частини комплексного числа. Передбачити конструктор та методи класу, що будуть ініціалізувати об'єкти класу, виводити значення на екран, а також порівнювати (`=` чи `≠`) два комплексні числа.

11. Клас Time моделює роботу з годинником. Полями цього класу повинні бути три поля типу int, призначені для зберігання годин, хвилин, секунд. Один з конструкторів має ініціалізувати годинник нульовими значеннями, а інший – заданим набором значень. Передбачити метод класу для виведення часу на екран у форматі 11:59:59, а також метод додавання значень двох об'єктів типу Time, які передаються в якості аргументів. [Лафоре, 258,861]

12. Клас Float імітує стандартний тип даних float. Єдине поле цього класу повинне мати тип float. Передбачити конструктор класу та методи, що будуть ініціалізувати об'єкти класу, виводити значення на екран, а також множити і ділити два дійсних числа.

13. Клас Vector імітує роботу із векторами у двовимірному просторі. Полями цього класу повинні бути чотири поля типу int, призначені для зберігання координат точок початку і кінця вектора. Передбачити конструктор та методи класу, що будуть ініціалізувати об'єкти класу, виводити значення на екран, а також знаходити координати вектора та перевіряти, чи є два вектори колінеарними.

14. Клас Date моделює роботу з календарем. Полями цього класу повинні бути три поля типу int: рік, місяць і день. Передбачити конструктор класу, методи ініціалізації та виведення дати у форматі dd.mm.yyyy, а також інкремента поточної дати на один день. При цьому необхідно врахувати можливий перехід до наступного місяця та року.

15. Клас Time моделює роботу з годинником. Полями цього класу повинні бути три поля типу int, призначені для зберігання годин, хвилин, секунд. Один з конструкторів має ініціалізувати годинник нульовими значеннями, а інший – заданим набором значень. Передбачити методи класу для виведення часу на екран у форматі 11:59:59, а також інкремента секунд поточного часу. При цьому необхідно врахувати можливий перехід до наступної хвилини.

16. Клас Complex імітує роботу з комплексними числами. Полями цього класу повинні бути два поля типу `int`, призначені для зберігання дійсної й уявної частини комплексного числа. Передбачити конструктор за замовчуванням, методи класу для додавання та ділення двох комплексних чисел, а також виведення комплексного числа на екран.

17. Клас RatNum імітує роботу зі звичайними дробами. Полями цього класу повинні бути два поля типу `int`: чисельник і знаменник дробу. Передбачити конструктор класу та методи, що будуть ініціалізувати об'єкти класу, додавати і віднімати два дроби, а також виводити на екран значення у форматі `a/b`.

18. Клас Account моделює роботу з поточним банківським рахунком. Полями цього класу повинні бути прізвище власника рахунку та сума на рахунку. Передбачити конструктор класу та методи для перевірки існування певного рахунку, збільшення рахунку на задану суму, а також виведення інформації на екран.

19. Клас Vector3D імітує роботу із векторами у тривимірному просторі. Полями цього класу повинні бути три поля типу `int`, призначені для зберігання координат вектора. Передбачити конструктор класу та методи для виведення інформації на екран, а також додавання та скалярного добутку двох векторів.

20. Клас Employee моделює роботу з даними про співробітників фірми. Клас повинен включати поле типу `int` для зберігання ідентифікаційного номера співробітника й поле типу `float` для зберігання розміру його окладу. Передбачити конструктор класу та методи, призначені для введення й відображення інформації про співробітника. Передбачити запит даних на трьох співробітників. [Лафоре, 258]

Теоретичні відомості

Клас – це базова C++-одиниця інкапсуляції. Класи використовують для створення об'єктів.

Клас можна представити як деякій шаблон, що визначає формат *об'єкта*. Клас включає як дані, так і функції для виконання дій над цими даними. У C++ специфікацію класу використовують для побудови об'єктів. Об'єкти – це *екземпляри* класу. Важливо розуміти, що клас – це логічна абстракція, котра реально не існує до тих пір, поки не буде створений об'єкт цього класу.

Функції та змінні, що складають клас, називають його *членами*. Змінну, оголошену у класі, називають *даним-членом*, а функцію, оголошену у класі, називають *функцією-членом*. Члени класу можуть мати три рівні доступу: *закритий* (private), *відкритий* (public) чи *захищений* (protected). Рівні доступу до членів класу визначають спосіб роботи користувачів із класом.

Оголошення класу

Оголошення класу містить оголошення членів даних і функцій-членів класу.

Клас оголошується за допомогою ключового слова **class**.

Загальний формат оголошення класу має наступний вигляд:

```
class ім'я_класу {  
    private:  
        // закриті дані і функції  
    public:  
        // відкриті дані і функції  
    protected:  
        // захищені дані і функції  
};
```

Наприклад:


```

class MyClass {
private:                // закриті дані і функції
    int date;
public:                // відкриті дані і функції
    void memfunc(int d)
        {date = d;}
};

```

Ім'я класу стає ім'ям нового типу даних, яке можна використовувати для створення об'єктів класу. Всі операції програма виконує над об'єктами. Об'єкти визначаються у функції main(). Визначення об'єкта схоже на визначення змінної: воно означає виділення пам'яті, необхідної для зберігання об'єкта.

Доступ до членів класу

Дані-члени класу та функції-члени мають *область дії класу*. В області дії дані-члени класу безпосередньо доступні усім функціям-членам цього класу й на них можна посилатися просто за іменем. Поза областю дії до елементів класу можна звертатися або через ім'я об'єкта, або посиланням на об'єкт, або за допомогою вказівника на об'єкт.

Доступ до функцій (методів) класу можливий лише через конкретний об'єкт цього класу. При цьому використовують *операцію крапка* (.), яка пов'язує метод з ім'ям об'єкта.

Конструктор класу

Класи в C++ мають спеціальну функцію, яку називають *конструктором*.

Конструктор – це функція, яка автоматично викликається при створенні (реалізації) об'єкта класу.

Конструктор використовують для ініціалізації змінних-членів класу, виділення необхідної пам'яті та виконання інших дій, необхідних перед

початком використання класу. Ім'я конструктора повинне співпадати з ім'ям класу. Це слугує характерною ознакою конструктора. Крім того, для конструктора не вказують тип значення, що повертається, адже конструктор не може повертати ніякого значення.

Клас може мати більше одного конструктора. Це можливе завдяки переважанню функцій. Наприклад, можна визначити конструктор без аргументів (*конструктор за замовчуванням*) та *конструктор з параметрами*, що приймає один чи кілька аргументів для ініціалізації членів-даних.

Деструктор класу

Деструктор – це спеціальна функція-член класу, що спрацьовує під час знищення динамічно розміщеного об'єкту класу і звільняє займану ним пам'ять.

Ім'я деструктора співпадає з ім'ям класу. Перед ім'ям деструктора записується символ тильда «~». Так само як і для конструктора, для деструктора не вказують тип значення, що повертається.

Наприклад:

```
class MyClass {  
public:  
    MyClass ( ); // конструктор класу  
    ~MyClass ( ); // деструктор класу  
    . . .  
};
```

Деструктори необхідні, якщо конструктор або інші функції-члени класу динамічно розподіляють пам'ять, створюючи в ній об'єкти. Тоді деструктор повинен ці об'єкти видалити. В інших випадках зазвичай можна обійтися без деструктора.

Якщо деструктор явним чином в класі не оголошений, компілятор сам генерує необхідні коди звільнення пам'яті.

Приклад програми

Завдання: Написати програму для тестування класу `Int`, що імітує стандартний тип даних `int`. Єдине поле цього класу повинне мати тип `int`. Передбачити конструктори класу та методи, що будуть ініціалізувати його об'єкти нулем, цілим значенням, виводити значення на екран, а також додавати два цілі числа. [Лафоре, 258, 860]

```
#include <iostream>
#include <conio.h>
using namespace std;
class Int                // оголошення класу
{
private:
    int i;
public:
    Int()                // конструктор за замовчуванням
    { i = 0; }           // ініціалізація Int нулем
    Int(int i1)          // конструктор з одним параметром
    { i = i1; }         // ініціалізація Int
    void add(Int i2,Int i3) //додавання двох значень типу Int
    { i = i2.i + i3.i; }
    void display()      //виведення Int
    { cout << i; }
};
void main()
{
    Int Int1(5);        // створення та ініціалізація першої змінної типу Int значенням 5
    Int Int2(12);       // створення та ініціалізація другої змінної типу Int значенням 12
```

```

Int Int3;          // створення та ініціалізація нулем третьої змінної типу Int
cout << "\nInt1 = "; Int1.display(); //виведення першої змінної
cout << endl;
cout << "\nInt2 = "; Int2.display(); //виведення другої змінної
cout << endl;
cout << "\nInt3 = "; Int3.display(); //виведення третьої змінної
cout << endl;
Int3.add(Int1, Int2); //додавання двох змінних типу Int
cout << "\nInt3 = "; Int3.display(); //виведення рез-ту додавання
cout << endl;
getch();
}

```

```

E:\Programs\C_Builder\Projects\Project2.exe
Int1 = 5
Int2 = 12
Int3 = 0
Int3 = 17
_

```

Контрольні питання

1. Що представляє собою клас?
2. Чим клас відрізняється від структури?
3. Як визначити новий клас?
4. Наведіть загальний формат оголошення класу.
5. Поясніть принцип інкапсуляції даних.
6. Які рівні доступу до членів класу Вам відомі?
7. У чому відмінність між відкритими і закритими членами класу?
8. Як визначити об'єкти класу?

9. Який оператор використовують для доступу до членів класу через об'єкт?
10. Що таке конструктор класу?
11. Назвіть особливості конструктора.
12. Як і коли викликається конструктор класу?
13. Чи може клас мати більше одного конструктора?
14. Що таке деструктор?
15. Назвіть особливості деструктора.

Рекомендована література

- Архангельский А.Я.** C++Builder 6. Справочное пособие. Книга 1. Язык C++. -- М.: Бином-Пресс, 2000. — 544 с.: ил. – с. 185;
- Х. Дейтел, П. Дейтел** Как программировать на C++. Пер. с англ. – М.: ЗАО «Издательство БИНОМ», 2000. – 1008 с. - С. 405;
- Р. Лафоре** Объектно-ориентированное программирование в C++. Классика Computer Science. 4-е изд. – СПб.: Питер, 2008. – 928 с. – 10 000 экз. – ISBN 978-5-94723-302-5. – С. 217;
- Д. Либерти** Освой самостоятельно C++ за 21 день. 3-е изд. – М.: Издательский дом «Вильямс», – 820 с. – С. 135;
- С. Прата** Язык программирования C++. Лекции и упражнения. Учебник: Пер. с англ./ стивен прата – СПб.: ООО «ДиаСофтЮП», 2005. – 1104 с. – ISBN 5-93772-123-3. – С. 428;
- К. Рейсдорф, К. Хендерсон** Borland C++Builder. Освой самостоятельно – М.: «Издательство БИНОМ», 1998. – 704 с.: ил. – 6 000 экз. – ISBN 5-7989-0099-1. – С. 103;
- Г. Шилдт** C++: руководство для начинающих, 2-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 672 с.: ил. – Парал. тит. англ. – 3 000 экз. – ISBN 5-8459-0840-X. – С. 349;

КОМП'ЮТЕРНИЙ ПРАКТИКУМ №4

Тема: Наслідування

Мета роботи – розробка програмного забезпечення з реалізації алгоритмів із використанням механізму наслідування

Завдання: Написати програму для тестування похідного класу.

1. На основі базового класу «Геометрична фігура», в якому зберігаються основні параметри (наприклад, ширина та висота) двовимірного об'єкта, створити похідний клас «Прямокутник». У похідному класі передбачити функцію, яка визначає, чи є прямокутник квадратом та функцію, що обчислює площу прямокутника.
2. На основі базового класу «Відрізок», в якому зберігається довжина відрізка, створити похідний клас «Коло». У похідному класі передбачити функції, що обчислюють довжину кола та площу круга.
3. На основі базового класу «Точка», в якому зберігаються координати точки на площині (x та y), створити похідний клас «Вектор». У похідному класі передбачити функції для обчислення довжини вектора та суми двох векторів.
4. На основі базового класу «Особа», в якому зберігаються П.І.Б. особи, її стать та вік, створити похідний клас «Викладач», що містить інформацію про факультет, кафедру, де працює викладач, та займану посаду. У похідному класі передбачити також функції для виведення інформації щодо викладача на екран та визначення окладу в залежності від посади.
5. На основі базового класу «Геометрична фігура», в якому зберігаються основні параметри (наприклад, ширина та висота) двовимірного об'єкта, створити похідний клас «Паралелепіпед». У похідному класі передбачити функції, що обчислюють площу поверхні та об'єм паралелепіпеда.

6. На основі базового класу «Число» створити похідний клас «Дії». У похідному класі передбачити функції для додавання, віднімання, множення двох цілих чисел.
7. На основі базового класу «Відрізок», в якому зберігається довжина відрізка, створити похідний клас «Куля». У похідному класі передбачити функції, що обчислюють площу поверхні та об'єм кулі.
8. На основі базового класу «Геометрична фігура», в якому зберігаються основні параметри (наприклад, ширина та висота) двовимірного об'єкта, створити похідний клас «Паралелепіпед». У похідному класі передбачити функцію, що визначає, чи є паралелепіпед кубом та функцію, яка обчислює його площу поверхні.
9. На основі базового класу «Особа», в якому зберігаються П.І.Б. особи, її стать та вік, створити похідний клас «Автовласник», що містить інформацію про марку автомобіля, рік випуску, колір, державний номер. У похідному класі передбачити також функції для виведення інформації щодо автовласника на екран та визначення приналежності автомобіля із заданим державним реєстраційним номером конкретній особі.
10. На основі базового класу «Точка», в якому зберігаються координати точки на площині (x та y), створити похідний клас «Вектор». У похідному класі передбачити функцію, що визначає, чи є два вектори рівними, та функцію для обчислення скалярного добутку двох векторів.
11. На основі базового класу «Геометрична фігура», в якому зберігаються основні параметри (наприклад, ширина та висота) двовимірного об'єкта, створити похідний клас «Трапеція». У похідному класі передбачити функцію, що обчислює площу трапеції.
12. На основі базового класу «Відрізок», в якому зберігається довжина відрізка, створити похідний клас «Куб». У похідному класі передбачити

функції, що обчислюють площу поверхні та об'єм куба.

13. На основі базового класу «Число» створити похідний клас «Дії». У похідному класі передбачити функції для множення та ділення двох цілих чисел, а також функцію піднесення до степеня.

14. На основі базового класу «Особа», в якому зберігаються П.І.Б. особи, її стать та вік, створити похідний клас «Працівник», що містить інформацію про посаду та розмір посадового окладу. У похідному класі передбачити також функції для виведення інформації щодо працівника на екран та визначення надбавки до окладу в залежності від вислуги років.

15. На основі базового класу «Геометрична фігура», в якому зберігаються основні параметри (наприклад, ширина та висота) двовимірного об'єкта, створити похідний клас «Циліндр». У похідному класі передбачити функції для обчислення площі бічної поверхні циліндра та його об'єм.

16. На основі базового класу «Відрізок», в якому зберігається довжина відрізка, створити похідний клас «Кільце». У похідному класі передбачити функцію, що обчислює площу кільця.

17. На основі базового класу «Точка», в якому зберігаються координати точки на площині (x та y), створити похідний клас «Вектор». У похідному класі передбачити функцію, що визначає, чи є два вектори колінеарними, та функцію для обчислення добутку вектора на число.

18. На основі базового класу «Геометрична фігура», в якому зберігаються основні параметри (наприклад, ширина та висота) двовимірного об'єкта, створити похідний клас «Конус». У похідному класі передбачити функції для обчислення площі бічної поверхні конуса та його об'єм.

19. На основі базового класу «Особа», в якому зберігаються П.І.Б. особи, її стать та вік, створити похідний клас «Студент», який зберігає результати сесії. У похідному класі передбачити також функції для виведення

інформації щодо студента на екран, для визначення середнього балу студента та розміру його стипендії в залежності від середнього балу.

20. На основі базового класу «Хлопець», в якому зберігаються П.І.Б. та вік хлопця, створити похідний клас «Студент», що містить інформацію про місце навчання. У похідному класі передбачити функцію для визначення, чи є студент повнолітнім, та функцію для виведення інформації щодо певного студента на екран.

Теоретичні відомості

Однією з найбільш потужних властивостей класів у С++ є можливість їхнього розширення шляхом *наслідування*. Наслідування дозволяє одному класу наслідувати характеристики іншого.

У стандартній термінології мови С++ клас, що наслідується, називають *базовим*. Клас, який наслідує базовий клас, називають *похідним*.

Похідний клас наслідує всі члени, визначені в базовому класі, й додає до них власні унікальні елементи. Т. б., похідний клас являє собою спеціалізовану версію базового класу.

Базовий клас при наслідуванні залишається незмінним.

Оголошення похідного класу

Загальний формат оголошення класу, що наслідує базовий клас, має такий вигляд:

```
class ім'я_похідного_класу : специфікатор_доступу ім'я_базового_класу  
{  
    // тіло похідного класу  
};
```

При цьому використовується *операція двокрапка (:)*, що встановлює відносини між класами.

Статус доступу членів базового класу в похідному класі визначається специфікатором доступу (`private`, `public`, `protected`), що використовується для наслідування базового класу.

Доступ до базового класу

Для об'єктів похідного класу можуть бути використані відкриті методи базового класу.

Проте, наслідування не працює у зворотному напрямку. Базовому класу та його об'єктам недоступні похідні класи.

Приклад програми

Завдання: На основі базового класу `TwoShape`, в якому зберігаються основні параметри (ширина та висота) двовимірного об'єкта, створити похідний клас `Triangle`, який інкапсулює трикутники.

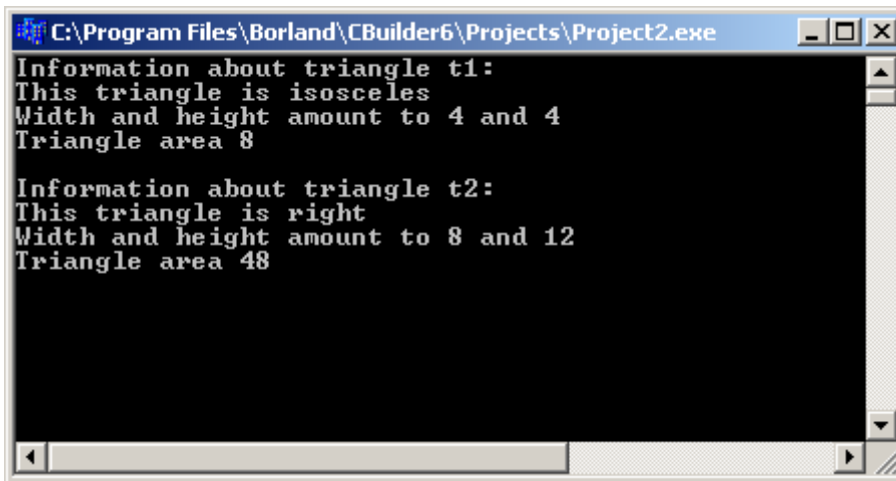
```
#include <iostream>
#include <cstring>
#include <conio.h>
using namespace std;
// оголошення базового класу
class TwoShape {
public:
    double width;
    double height;
    void showDim()
    { cout << "Width and height amount to "
      << width << " and " << height << "\n"; }
};
// оголошення похідного класу
class Triangle : public TwoShape {
```

```

public:
    char style[20];
    double area()
    { return width* height/2; }
    void showStyle()
    {cout << "This triangle is" << style << "\n"; }
};

void main()
{
Triangle t1, t2;
t1.width=4.0;
t1.height =4.0;
strcpy(t1.style," isosceles ");
t2.width=8.0;
t2.height =12.0;
strcpy(t2.style," right ");
cout << "Information about triangle t1:\n";
t1.showStyle();
t1.showDim();
cout<<"Triangle area "<<t1.area()<<"\n"<<"\n";
cout << "Information about triangle t2:\n";
t2.showStyle();
t2.showDim();
cout<<"Triangle area "<<t2.area()<<"\n";
getch();
}

```



```
C:\Program Files\Borland\CBuilder6\Projects\Project2.exe
Information about triangle t1:
This triangle is isosceles
Width and height amount to 4 and 4
Triangle area 8

Information about triangle t2:
This triangle is right
Width and height amount to 8 and 12
Triangle area 48
```

Контрольні питання

1. У чому полягає механізм наслідування?
2. Як називають клас, що породжує інші класи?
3. Як називають клас, що є нащадком іншого класу?
4. Які рівні наслідування базового класу Вам відомі?
5. Наведіть загальний формат оголошення похідного класу.
6. Як змінюється базовий клас при наслідуванні?
7. Як називають випадок створення класу на основі двох чи більше класів?
8. Який клас називають абстрактним базовим класом?

Рекомендована література

Архангельский А.Я. С++Builder 6. Справочное пособие. Книга 1. Язык С++. – М.: Бином-Пресс, 2002. – 544 с.: ил. – 4 000 экз. – ISBN 5-9518-0007-2. – С. 196;

Х. Дейтел, П. Дейтел Как программировать на С++. Пер. с англ. – М.: ЗАО «Издательство БИНОМ», 2000. – 1008 с. – С. 551;

Р. Лафоре Объектно-ориентированное программирование в С++. Классика Computer Science. 4-е изд. – СПб.: Питер, 2008. – 928 с. – 10 000 экз. – ISBN

978-5-94723-302-5. – С. 361;

Д. Либерти Освой самостоятельно С++ за 21 день. 3-е изд. – М.: Издательский дом «Вильямс», – 820 с. – С. 135;

К. Рейсдорф, К. Хендерсон Borland C++Builder. Освой самостоятельно – М.: «Издательство БИНОМ», 1998. – 704 с.: ил. – 6 000 экз. – ISBN 5-7989-0099-1. – С. 122;

Г. Шилдт С++: руководство для начинающих, 2-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 672 с.: ил. – Парал. тит. англ. – 3 000 экз. – ISBN 5-8459-0840-X. – С. 465;

КОМП'ЮТЕРНИЙ ПРАКТИКУМ №5

Тема: Перевантаження операторів

Мета роботи – розробка програмного забезпечення з реалізації алгоритмів із використанням можливостей перевантаження операторів

Завдання: Написати програму, яка перевантажує оператори.

1. Оператор «*» для множення звичайних дробів, об'єктів класу RatNum.
2. Оператор «+» для виконання операції додавання одномірних масивів (множин), об'єктів класу Massif.
3. Оператор «+» для конкатенації (об'єднання) рядків, об'єктів класу String. [Лафоре, 323]
4. Оператор «-» для віднімання комплексних чисел, об'єктів класу Complex.
5. Оператор «<» для порівняння звичайних дробів, об'єктів класу RatNum.
6. Оператор інкременту «++» дня для об'єктів класу Date, членами якого є рік, місяць, день. Необхідно передбачити правильний перехід до наступного місяця та року.

7. Оператор «==» для порівняння одномірних масивів, об'єктів класу Massif, на рівність.
8. Оператор «+» для додавання комплексних чисел, об'єктів класу Complex.
9. Оператор «*» для знаходження добутку векторів, об'єктів класу Vector.
10. Оператор «+» для додавання звичайних дробів, об'єктів класу RatNum.
11. Оператор декременту «--» секунд для об'єктів класу Timer, членами якого є хвилина і секунда. Необхідно передбачити правильний перехід до наступної хвилини.
12. Оператор «-» для виконання операції віднімання одномірних масивів (множин), об'єктів класу Massif.
13. Оператор «*» для множення комплексних чисел, об'єктів класу Complex.
14. Оператор «+» для додавання векторів, об'єктів класу Vector.
15. Оператор «==» для порівняння рядків, об'єктів класу String. [Лафоре, 327]
16. Оператор декременту «--» дня для об'єктів класу Date, членами якого є рік, місяць, день. Необхідно передбачити правильний перехід до попереднього місяця та року.
17. Оператор «/» для ділення комплексних чисел, об'єктів класу Complex.
18. Оператор інкременту «++» секунд для об'єктів класу Timer, членами якого є хвилина і секунда. Необхідно передбачити правильний перехід до наступної хвилини.
19. Оператор «-» для віднімання звичайних дробів, об'єктів класу RatNum.
20. Оператор «+» для виконання операції додавання довжин (у метрах та

сантиметрах), об'єктів класу Distance. . [Лафоре, 320]

Теоретичні відомості

Перевантаження операторів виконують для того, щоб мати можливість використовувати стандартні вбудовані операції C++ для виконання дій над об'єктами класів.

Перевантажуючи оператори, програміст визначає їхнє призначення для конкретного класу. Це дає можливість органічно інтегрувати нові класи в існуюче програмне середовище, оскільки після перевантаження операції над об'єктами нових класів виглядають аналогічно операціям над змінними вбудованих типів.

Оператори перевантажують за допомогою *операторних функцій* з ім'ям **operator**.

Загальний формат оголошення операторної функції має такий вигляд:

```
тип ім'я _класу :: operator # (список_аргументів)
{
    // тіло функції
};
```

На перевантаження операторів накладається ряд обмежень. Не можна змінювати пріоритет оператора. Не можна змінювати кількість операндів, що приймаються оператором. Також операторні функції не можуть мати аргументів за замовчуванням.

Майже усі C++-оператори можна перевантажувати (у тому числі й оператор індексації масиву «[]», виклику функції «()», а також оператор «->»).

Зверніть увагу! Оператори, перевантаження яких заборонено, такі: оператор «.*», оператор «.»», оператор «::», оператор «?».

Приклад програми

Завдання: Написати програму, яка перевантажує оператор «+» для додавання координат об'єктів тривимірного простору.

```
#include <iostream>
#include <conio.h>
#include <locale>
using namespace std;
class ThreeD {
private:
    int x, y, z; // 3D координати
public:
    ThreeD () // конструктор за замовчуванням
    { x = y = z = 0; } // ініціалізація координат нулем
    ThreeD(int i, int j, int k) // конструктор з параметрами
    { x = i; y = j; z = k; }
    ThreeD operator+(ThreeD op2);
    void show() ;
};
// Перевантаження оператора «+»
ThreeD ThreeD::operator+(ThreeD op2)
{
    ThreeD temp;
    temp.x = x + op2.x;
    temp.y = y + op2.y;
    temp.z = z + op2.z;
    return temp;
}
// Відображення координат X, Y, Z
```



```

void ThreeD::show()
{
cout << x << ", ";
cout << y << ", ";
cout << z << "\n";
}

void main()
{
setlocale (LC_ALL, "Russian");
ThreeD a(1, 2, 3), b(10, 10, 10), c;
cout << "Вихідні координати об'єкта a: "; a.show();
cout << "Вихідні координати об'єкта b: "; b.show();
cout << "Вихідні координати об'єкта c: "; c.show();
cout << endl;

c = a + b; // додавання об'єктів a та b
cout << "Нові координати об'єкта c=a+b: "; c.show();
cout << endl;

c = a + b + c; // додавання об'єктів a, b та c
cout << "Нові координати об'єкта c=a+b+c: "; c.show();
cout << endl;

getch();
}

```

```

E:\Programs\C_Builder\Projects\Project2.exe
Тшї | фэ | ъююЁфшэрЄш юс' || ЪЄр а: 1, 2, 3
Тшї | фэ | ъююЁфшэрЄш юс' || ЪЄр b: 10, 10, 10
Тшї | фэ | ъююЁфшэрЄш юс' || ЪЄр c: 0, 0, 0

⇒от | ъююЁфшэрЄш юс' || ЪЄр c = a + b: 11, 12, 13
⇒от | ъююЁфшэрЄш юс' || ЪЄр c = a + b + c: 22, 24, 26

```

Контрольні питання

1. Що таке перевантаження?
2. Як реалізувати перевантаження оператору?
3. Яке призначення ключового слова `operator`?
4. Скільки параметрів має операторна функція-член для реалізації бінарного оператора?
5. Скільки аргументів необхідно для перевантаження унарної операції?
6. Які існують обмеження на перевантаження операторів?
7. Назвіть оператори, перевантаження яких заборонено.
8. Чи можливо перевантажити операцію «*» для виконання ділення?

Рекомендована література

Х. Дейтел, П. Дейтел Как программировать на C++. Пер. с англ. – М.: ЗАО «Издательство БИНОМ», 2000. – 1008 с. - С. 497;

Р. Лафоре Объектно-ориентированное программирование в C++. Классика Computer Science. 4-е изд. – СПб.: Питер, 2008. – 928 с. – 10 000 экз. – ISBN 978-5-94723-302-5. – С. 312;

С. Прата Язык программирования C++. Лекции и упражнения. Учебник: Пер. с англ./ стивен прата – СПб.: ООО «ДиаСофтЮП», 2005. – 1104 с. – ISBN 5-93772-123-3. – С. 483;

Г. Шилдт C++: руководство для начинающих, 2-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 672 с.: ил. – Парал. тит. англ. – 3 000 экз. – ISBN 5-8459-0840-X. – С. 430;

Комплексні числа

• **Порівняння**

Два комплексних числа рівні між собою тоді і тільки тоді, коли рівні їх дійсні та уявні частини:

$a + bi = c + di$ означає, що $a = c$ та $b = d$.

• **Додавання**

$$(a + bi) + (c + di) = (a + c) + (b + d)i.$$

• **Віднімання**

$$(a + bi) - (c + di) = (a - c) + (b - d)i.$$

• **Множення**

$$(a + bi) * (c + di) = ac + bci + adi + bdi^2 = ac + bci + adi - bd = (ac - bd) + (bc + ad)i.$$

• **Ділення**

$$\frac{a + bi}{c + di} = \frac{(a+ib)(c-id)}{(c+id)(c-id)} = \left(\frac{ac+bd}{c^2+d^2} \right) + \left(\frac{bc-ad}{c^2+d^2} \right) i$$

Операції над множинами

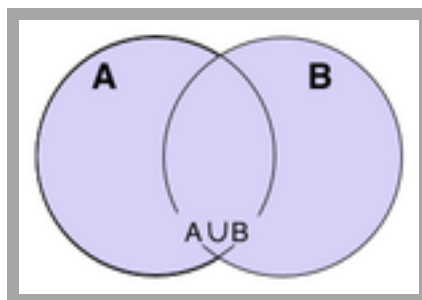
Операція «+» над множинами створює нову множину, яка міститиме в собі усі елементи, що зустрілися у множинах-доданках.

Наприклад:

$$\{0,2,4,5,6\} + \{0,1,2,3,4\} = \{0,1,2,3,4,5,6\}.$$

$$\{1, 2\} + \{\text{червоний, білий}\} = \{1, 2, \text{червоний, білий}\}$$

$$\{1, 2\} + \{1, 2\} = \{1, 2\}$$



Об'єднання множин A та B

Операція « \leftarrow » над множинами створює нову множину, яка міститиме в собі лише елементи, що не входять до множини-від'ємника.

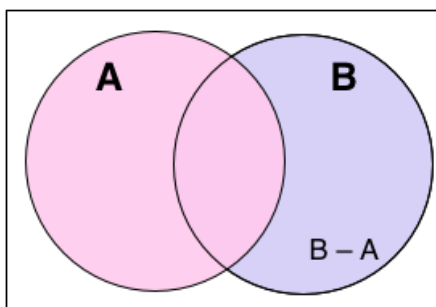
Наприклад:

$$\{\text{One, Two, Tree, Four}\} - \{\text{Five, Four, Six}\} = \{\text{One, Two, Tree}\}.$$

$$\{1, 2\} - \{\text{червоний, білий}\} = \{1, 2\}$$

$$\{1, 2, \text{зелений}\} - \{\text{червоний, білий, зелений}\} = \{1, 2\}$$

$$\{1, 2\} - \{1, 2\} = \emptyset$$



Різниця множин A та B

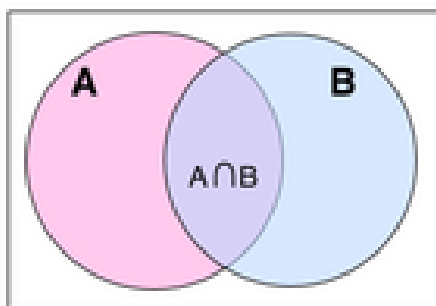
Операція « $*$ » над множинами створює нову множину, яка міститиме в собі лише ті елементи, що входять до обох множин-множників.

$$\{\text{'a', 'b', 'c', 'D', 'E', 'F'}\} * \{\text{'A', 'B', 'c', 'd'}\} = \{\text{'c'}\}.$$

$$\{1, 2\} * \{\text{червоний, білий}\} = \emptyset$$

$$\{1, 2, \text{зелений}\} * \{\text{червоний, білий, зелений}\} = \{\text{зелений}\}$$

$$\{1, 2\} * \{1, 2\} = \{1, 2\}$$



Перегин множин A та B

Векторне числення

Числа $a_x = x_2 - x_1$, $a_y = y_2 - y_1$ називають **координатами вектора** \vec{a} з початком у точці $A(x_1; y_1)$ і кінцем у точці $B(x_2; y_2)$.

Довжину вектора \vec{a} з координатами $(a_x; a_y)$ обчислюють формулою $|\vec{a}| = \sqrt{a_x^2 + a_y^2}$.

Два **вектори рівні** тоді і тільки тоді, коли їхні відповідні координати рівні.

Сумою векторів $\vec{a}(a_x; a_y)$ і $\vec{b}(b_x; b_y)$ називають вектор $\vec{c}(a_x + b_x; a_y + b_y)$.

Скалярним добутком векторів \vec{a} і \vec{b} називають число, що дорівнює сумі добутків відповідних координат, тобто

$$\vec{a} \cdot \vec{b} = a_x \cdot b_x + a_y \cdot b_y.$$

Добутком вектора $\vec{a}(a_x; a_y)$ **на число** λ називають вектор з координатами $(\lambda a_x; \lambda a_y)$.

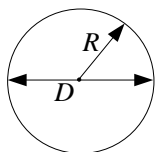
Два вектори є **ортогональними** тоді і тільки тоді, коли їхній скалярний добуток дорівнює нулю.

Два **вектори** \vec{a} і \vec{b} **колінеарні** тоді і тільки тоді, коли їх відповідні координати пропорційні, тобто

$$\frac{a_x}{b_x} = \frac{a_y}{b_y}.$$

Геометрія

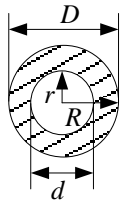
Коло і круг



$\ell = 2\pi R = \pi D$ – довжина кола;

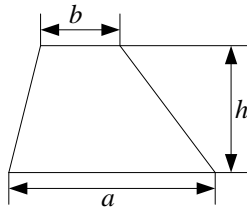
$S = \pi R^2 = \frac{\pi D^2}{4}$ – площа круга.

Кільце



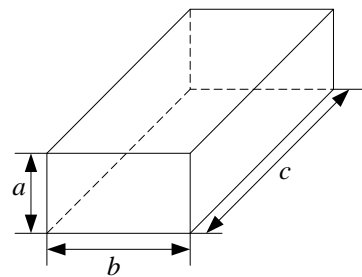
$$S = \pi(R^2 - r^2) = \frac{\pi D^2}{4} - \frac{\pi d^2}{4}$$

Трапеція



$$S = \frac{(a+b)h}{2}$$

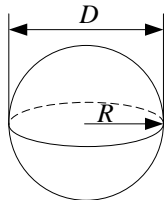
Паралелепіпед



$$S = 2(ab + ac + bc)$$

$$V = abc$$

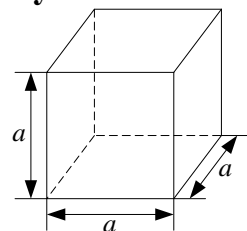
Куля



$$S = 4\pi R^2 = \pi D^2$$

$$V = \frac{4}{3}\pi R^3$$

Куб



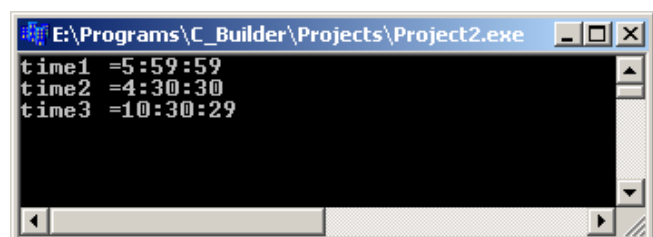
$$S = 6a^2$$

$$V = a^3$$

ДЛЯ ПОДАТОК

`//ex5_11.cpp`

```
//Робота класу Time
#include <iostream>
#include <conio.h>
using namespace std;
class Time
{
private:
    int hrs,mins,secs;
public:
    Time() : hrs(0), mins(0), secs(0) //конструктор без аргументів
    {}
    //конструктор з трьома аргументами
    Time(int h, int m, int s) : hrs(h), mins(m), secs(s)
    {}
    void display() const //формат 11:59:59
    {cout <<hrs <<":" <<mins <<":" <<secs;}
    void add (Time t1, Time t2) //додати дві змінні типу Time
    {
        secs =t1.secs +t2.secs; //додати секунди
        if(secs >59 ) //якщо забагато секунд, додати 1 хвилину
        { secs -=60; mins++; }
        mins +=t1.mins +t2.mins; // додати хвилини
        if(mins >59 ) // якщо забагато хвилин, додати 1 годину
        { mins -=60;hrs++; }
        hrs +=t1.hrs +t2.hrs; //додати години
    }
};
void main()
{
    const Time time1(5,59,59); //створення та ініціалізація двох змінних
    const Time time2(4,30,30);
    cout <<"time1 ="; time1.display(); cout <<endl; // вивести на екран
    cout <<"time2 ="; time2.display(); cout <<endl;
    Time time3; //створити ще одну змінну
    time3.add(time1,time2); //додати дві змінні
    cout <<"time3 =";time3.display(); //вивести результат
    cout <<endl;
    getch();}
```



```
E:\Programs\C_Builder\Projects\Project2.exe
time1 =5:59:59
time2 =4:30:30
time3 =10:30:29
```


//ex7_14

```
// Перевантаження префіксної версії оператора інкремента «++»
#include <iostream>
using namespace std;
class ThreeD {
privat:
    int x, y, z; // 3-D координати
public:
    ThreeD()
    { x = y = z = 0; }
    ThreeD(int i, int j, int k)
    { x = i; y = j; z = k; }
    ThreeD operator++(); // префіксна версія оператору ++
    void show() ;
};
// Перевантаження оператора «++»
ThreeD ThreeD::operator++()
{
    x++; // інкремент координат x, y, z
    y++;
    z++;
    return *this; // повертає інкрементований об'єкт
}
// Відображення координат X, Y, Z
void ThreeD::show()
{
    cout << x << " ";
    cout << y << " ";
    cout << z << "\n";
}
void main()
{
    ThreeD a(1, 2, 3);

    cout << "Вихідні координати об'єкта a: ";
    a.show();
    ++a; // інкремент об'єкта a
    cout << "Значення ++a: ";
    a.show();
    getch();
}
```

```
//ex3_13.cpp
```

```
TotalSeconds=hours*3600+minutes*60+seconds;
```