

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний технічний університет  
«Харківський політехнічний інститут»

**МЕТОДИЧНІ ВКАЗІВКИ**  
**до лабораторної роботи**  
**«Використання показчиків на функцію**  
**в програмах мовою С++»**  
**з курсу «Програмування»**  
для студентів напрямку 6.040302 – Інформатика  
і курсу «Програмування та алгоритмічні мови»  
для студентів напрямку 6.040303 – Системний аналіз

Затверджено редакційно-видавничою  
радою університету,  
протокол № 1 від 04.06.14.

Харків  
НТУ «ХПІ»  
2014

Методичні вказівки до лабораторної роботи «Використання показчиків на функцію в програмах мовою С++» з курсу «Програмування» для студентів напрямку 6.040302 – Інформатика і курсу «Програмування та алгоритмічні мови» для студентів напрямку 6.040303 – Системний аналіз / Уклад. М. І. Безменов, О. М. Безменова. – Х. : НТУ «ХПІ», 2014. – 13 с.

Укладачі: М. І. Безменов,  
О. М. Безменова

Рецензент І. П. Гамаюн

Кафедра системного аналізу і управління

## ВСТУП

Крім звичайних покажчиків у C++ визначені так називані покажчики на функцію. Ці покажчики дозволяють реалізувати альтернативний метод виклику функцій і є досить ефективним засобом для побудови меню.

**Метою** даної лабораторної роботи є освоєння методики використання покажчиків на функцію в програмах мовою C++.

### 1. ТЕОРЕТИЧНІ ОСНОВИ

У C++ ім'я функції є константним покажчиком; його значення – це адреса розташування функції у пам'яті.

У програмі можливе визначення змінних, що є покажчиками на функцію. У визначенні такого покажчика повинен бути той же тип, що й тип значення, яке повертає функція, а також ідентична сигнатура.

Визначення покажчика на функцію має такий вигляд:

```
тип_функції (*ім'я_змінної) (специфікація_параметрів) ;
```

Приклади:

```
int (*fPtr1) (double x); // Покажчик на функцію, що повертає
    // значення типу int і має один параметр типу double
void (*fPtr2) (float * a, int n); // Покажчик на void-
    // функцію, перший параметр якої є покажчиком
    // на float, а другий має тип int
char* (*fPtr3) (char * s, int n); // Покажчик на функцію,
    // що повертає покажчик на char і має два параметри –
    // покажчик на char і int
```

У програмі такій змінній-покажчику можна присвоювати ім'я функції з тим самим типом значення, що повертається, і сигнатурою. У подальшому можна викликати функцію по покажчику на неї (ніби покажчик є іменем функції) або за адресою (шляхом розіменування покажчика). Наприклад:

```
int f(double a)
{
    return 2 * a;
}
int main()
{
```

```

fPtr1 = f;
// cout << f(5.5);
cout << fPtr1(5.5);
// Можна виконати розіменування покажчика:
//   cout << (*fPtr1)(5.5);
}

```

Можливим є опис масиву покажчиків на функцію:

```

тип_функції (*ім'я_масиву_покажчиків)
(специфікація_параметрів) [розмір_масиву];

```

Наприклад:

```

char* (*fPtr4) (char *, int) [10];

```

Можна визначити ім'я для типу покажчика на функцію за допомогою **typedef**:

```

typedef тип_функції (*ім'я_типу_покажчика) (специфікація_параметрів);

```

Наприклад:

```

typedef void (*TPtv) (double);
TPtv Ptv, PtvAr[20];

```

Функція може повертати покажчик на функцію. Наприклад, можливі такі прототипи функцій:

```

int (*f1(char)) (double); // Функція з одним параметром
// типу char, яка повертає покажчик на функцію
// з одним параметром типу double і значенням,
// що повертається, типу int
void (*f2(double *, int)) (int); // Функція з двома
// параметрами, перший з яких має тип double *,
// а другий - тип int. Значення, що повертається -
// покажчик на void-функцію з одним параметром типу int
bool (*f3()) (char *, int); // Функція без параметрів,
// яка повертає покажчик на функцію
// типу bool з двома параметрами, перший
// з яких має тип char *, а другий - тип int

```

Використання **typedef** у цьому разі значно спрощує сприйняття коду. Наведені вище описи можуть бути записані так:

```

typedef int (*TPtf1) (double);
TPtf1 f1(char);

```

```

typedef void (*TPtf2) (int);
TPtf2 f2(double *, int);

typedef bool (*TPtf3) (char *, int);
TPtf3 f3(); // Можна інакше: TPtf3 f3(void);

```

## 2. ПРИКЛАД ПРОГРАМИ

Дано дійсні числа  $a, b, c, d_1, d_2$  і натуральне число  $n$  ( $n \geq 2$ ). Числа  $a, b, c$  задають коефіцієнти деякої безперервної функції, визначеної на інтервалі  $[d_1, d_2]$ . Виконати табулювання однієї з двох функцій  $y = a \sin(bx + c)$  і  $y = a(x + b)^2 + c$  на інтервалі  $[d_1, d_2]$  для  $n$  рівновіддалених точок. Визначити функцію табулювання, яка залежить від функції з чотирма параметрами (3 коефіцієнти і значення аргументу). Для вибору функції, табулювання якої повинно бути здійснене, використати функцію, що повертає покажчик на функцію, яка табулюватиметься.

### Розв'язок.

```

#include <iostream >
#include <conio.h>
using namespace std;
// Визначаємо тип TFunc - покажчик на функцію
// з чотирма параметрами типу double,
// яка повертає значення типу double
typedef
double (*TFunc) (double a, double b, double c, double x);
// Нижче наведено прототипи двох функцій
// з чотирма параметрами типу double,
// що повертають значення типу double
double SinABC(double a, double b, double c, double x);
double ParabolaABC(double a, double b, double c, double x);
// Нижче наведено два варіанти прототипу функції з
// параметром, що є покажчиком на функцію
void Tabulation(double d1, double d2, int n,
double a, double b, double c, TFunc f);
/*
void Tabulation(double d1, double d2, int n,
double a, double b, double c,
double(*f) (double, double, double, double));
*/
// Прототип функції, що повертає покажчик на функцію

```

```

TFunc menu();
    // У наступному рядку наведено інший варіант прототипу
// double (*menu())(double, double, double, double);
// Головна функція:
int main()
{
    TFunc item;        // Змінна item визначена за допомогою
// раніше визначеного типу TFunc як покажчик на функцію
// з чотирма параметрами і результатом типу double
// Можна інакше:
// double (*item)(float,float,float,double);
double a, b, c, d1, d2;
int n;
cout << "Enter coefficients of the function (a b c): ";
cin >> a >> b >> c;
cout << "Enter the bounds of the interval"
      " d1 and d2 (d1 < d2): ";
cin >> d1 >> d2;
cout << "Enter number of nodes: n = ";
cin >> n;
item = menu();        // У item записується
// адреса функції або NULL
if (item == NULL)
    cout << "Function number is entered incorrectly";
else // Використання покажчика item
    Tabulation(d1, d2, n, a, b, c, item);
cout << "\nPress any key to exit. ";
_getch ();
}

double SinABC(double a, double b, double c, double x)
{
    return a * sin(b * x + c);
}

double ParabolaABC(double a, double b, double c, double x)
{
    return a *(x + b)*(x + b) + c;
}

void Tabulation(double d1,double d2,int n,
                double a, double b, double c, TFunc f)
{

```

```

double dx;
dx = (d2 - d1) / (n - 1);
cout << '\n';
for (int i = 1; i <= n; i++)
    cout << f(a, b, c, d1 + (i - 1) * dx) << ' ';
}

/*          Заголовок функції menu можна записати так:
double (*menu())(double, double, double, double)
          У програмі використана скорочена форма:
*/
TFunc menu()
{
    // Визначаємо масив покажчиків на функцію й здійснюємо
    // його ініціалізацію
    TFunc fun_items[] = {SinABC, ParabolABC};
/*          Можна інакше:
(*fun_items[])(double, double, double, double) =
          {SinABC, ParabolABC};
*/
int key;
cout << "Selecting a function:\n"
      << "1 - sine, 2 - parabola\n";
cin >> key;
switch (key)
{
    case 1 return fun_items[key - 1]; // функцій або
    default: return NULL; // NULL при помилці вибору
}
}

```

### 3. ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ

За час, відведений для виконання лабораторної роботи (2 академічні години), студент повинен:

1. Розробити алгоритм розв'язання задачі, запропонованої для програмування.
2. Здійснити програмну реалізацію розробленого алгоритму.
3. Здійснити налаштування програми, виправивши синтаксичні та логічні помилки.
4. Підібрати тестові дані для перевірки програми, включаючи виняткові випадки.

5. Оформити звіт до лабораторної роботи.
6. Відповісти на контрольні запитання.

#### 4. ВАРІАНТИ ЗАДАЧ

1. Дано дійсні числа  $a, b, c, d_1, d_2, \varepsilon$  ( $d_1 < d_2, \varepsilon > 0$ ). Обчислити визначений інтеграл від функції  $f(x) = ax^2 + bx + c$  на інтервалі  $[d_1, d_2]$  із точністю  $\varepsilon$ . Як окремі функції реалізувати три методи прямокутників і метод трапецій. Визначити функцію, що повертає покажчик на функцію, яка реалізує конкретний метод наближеного обчислення визначеного інтегралу, і використати її у програмі.

*Визначений інтеграл* деякої функції, заданої на інтервалі  $[a, b]$ , – це площа криволінійної трапеції, обмеженої кривою, віссю абсцис і вертикальними лініями, що проходять через межі інтервалу. Для наближеного обчислення визначеного інтеграла *методом прямокутників* інтервал  $[a, b]$  розбивають на підінтервали, на яких будують прямокутники, висоти яких дорівнюють значенню функції на лівій (правій) межі відповідного підінтервалу або в його середині (відповідно маємо три методи прямокутників). Як значення визначеного інтеграла приймають суму площ таких прямокутників. Зазвичай здійснюють багаторазове обчислення наближеного значення інтегралу, подвоюючи від одного кроку до іншого кількість підінтервалів, допоки модуль різниці між наближеними значеннями інтегралу на двох сусідніх кроках не стане меншим деякого достатньо малого додатного числа  $\varepsilon$ . Число  $\varepsilon$  у такому разі називають *точністю обчислення*.

У *методі трапецій* замість прямокутників використовують трапеції, побудовані на отриманих підінтервалах.

2. Оформити три функції: 1) для обчислення площі трикутника за довжиною основи і висоти; 2) площі прямокутника за довжинами його суміжних сторін; 3) площі сектора круга за значеннями радіуса круга і внутрішнього кута сектора. Дано інформацію про тип фігури та її геометричні розміри. Обчислити її площу. Для вибору функції обчислення площі використати спеціально розроблену функції, що повертає покажчик на функцію. Використовуючи описані функції, обчислити площі фігур, характеристики яких наведені у файлі, й записати результат обчислень в інший файл.

3. Визначити функції для обчислення суми, добутку, частки двох звичайних дробів. Вибір операції над дробами здійснювати за допомогою функції, що повертає покажчик на функцію. Написати програму, що реалізує вказані дії над звичайними дробами.

4. Визначити в програмі дві функції для обчислення косинуса:

$$\cos z = 1 - \frac{z^2}{2!} + \frac{z^4}{4!} \mp \dots \quad \text{та} \quad \cos z = \prod_{k=0}^{\infty} \left\{ 1 - \left[ \frac{2z}{\pi(2k+1)} \right]^2 \right\}, \quad \text{обмежуючись у}$$

першій функції  $n$  доданками, а в другій –  $n$  співмножники, де  $n$  – задане натуральне число. Для вибору методу подання косинуса скористатися функцією, що повертає покажчик на функцію. Порівняти косинус деякого значення, обчислений за допомогою стандартної бібліотечної функції, зі значеннями, отримані за допомогою функцій, визначених у програмі.

5. Дано дійсні числа  $a, b, \varepsilon$  ( $a < b, \varepsilon > 0$ ). Дано також деяке рівняння  $f(x) = 0$ , де  $f(x)$  – монотонна функція на інтервалі  $[a, b]$ . Відомо, що це рівняння на інтервалі  $[a, b]$  має один і тільки один корінь. Знайти цей корінь з точністю  $\varepsilon$ . Для обчислення кореня використати один з двох методів – поділу відрізка навпіл і хорд. Функція, що реалізує меню вибору методу розв’язання рівняння, повинна повертати покажчик на функцію.

*Метод поділу відрізка навпіл* для відшукування кореня рівняння полягає в наступному. Спочатку здійснюється поділ відрізка на дві рівні частини, після чого з двох отриманих відрізків вибирається той, на якому перебуває корінь (значення функції на кінцях цього відрізка мають різні знаки або одне зі значень дорівнює нулю). Якщо довжина відрізка, що містить точний розв’язок, більша заданого досить малого додатного значення  $\varepsilon$ , то процес повторюється з розглядом нового відрізка.

*Метод хорд* (див. рис. 1) на першому своєму кроці передбачає побудову відрізка прямої (хорди), яка з’єднує точки  $(a, f(a))$  й  $(b, f(b))$ , і відшуванні точки перехрещення хорди з віссю абсцис (точка  $c$  на рис 4.1), яка вважається початковим наближенням кореня рівняння. Далі співставляються знаки значень  $f(a)$ ,  $f(b)$  і  $f(c)$  й здійснюється звуження інтервалу  $[a, b]$  шляхом перенесення однієї з її меж  $a$  або  $b$  у точку  $c$ . На наступному кроці будується нова хорда і відшукується нове наближення кореня рівняння. Процес продовжується

до тих пір, поки відстань між двома наближеннями кореня, отриманими на двох сусідніх кроках не стане за модулем меншою за достатньо мале додатне значення  $\varepsilon$ .

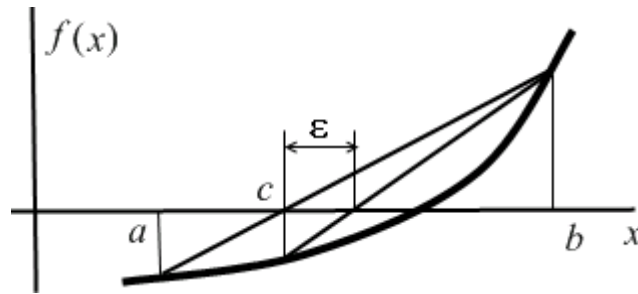


Рис. 1. Метод хорд

6. Дано масив дійсних чисел. Упорядкувати його за незростанням. У вигляді функцій реалізувати два методи впорядкування масиву. Побудувати меню вибору методу сортування, що базується на використанні показників на функції.
7. У матричній алгебрі існує поняття норми  $\|\mathbf{A}\|$  прямокутної матриці  $\mathbf{A} = (a_{ij})_{mn}$ . Запропоновано декілька методів обчислення норми дійсної матриці. Деякі з них такі:

$$\|\mathbf{A}\|_{\text{I}} = \max_{j=1,2,\dots,n} \sum_{i=1}^m |a_{ij}|;$$

$$\|\mathbf{A}\|_{\text{II}} = \max_{i=1,2,\dots,m} \sum_{j=1}^n |a_{ij}|;$$

$$\|\mathbf{A}\|_{\text{I}} = \sum_{i=1}^m \sum_{j=1}^n |a_{ij}|;$$

$$\|\mathbf{A}\|_{\text{2}} = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2} \quad (\text{евклідова норма}).$$

Для кожного з наведених вище варіантів визначення норми описати функцію, що обчислює норму дійсної матриці  $\mathbf{A}$  розміром  $m \times n$ .

Дано натуральні числа  $m$ ,  $n$  і дійсна матриця  $\mathbf{A}$  розміром  $m \times n$ . Знайти норми всіх підматриць цієї матриці, утворених:

- першими  $k$  рядками,  $k = 1, 2, \dots, m$ ;
- першими  $k$  стовпцями,  $k = 1, 2, \dots, n$ .

Вибір методу обчислення норми здійснювати використанням визначеної у програмі функції, що повертає показник на функцію.

8. Дано натуральні числа  $m, n, p, q$  і дві матриці  $\mathbf{A}$  (розміром  $m \times n$ ) і  $\mathbf{B}$  (розміром  $p \times q$ ). Визначити функції підсумовування і множення двох матриць з контролем можливості виконання цих дій. Визначити також функцію, що реалізує меню вибору дій над матрицями з використанням покажчиків на функцію.

*Сумою* двох матриць  $\mathbf{A}$  і  $\mathbf{B}$ , що мають однакові розміри, називається така матриця  $\mathbf{C}$  того ж розміру, елементи якої дорівнюють сумі відповідних елементів матриць-доданків.

*Добутком* двох матриць  $\mathbf{A}$  (розміром  $m \times n$ ) і  $\mathbf{B}$  (розміром  $n \times p$ ) називається така матриця  $\mathbf{C}$  (розміром  $m \times p$ ), елементи якої дорівнюють сумі попарних добутків відповідних елементів рядка першої матриці і стовпця другої матриці:

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}, i = 1, 2, \dots, m, j = 1, 2, \dots, p.$$

9. Визначити функції отримання координат максимального елементу квадратного масиву у виділених на рис. 1 частинах. Вибір функції здійснювати за допомогою меню, яке базується на функції, що повертає покажчик на функцію.

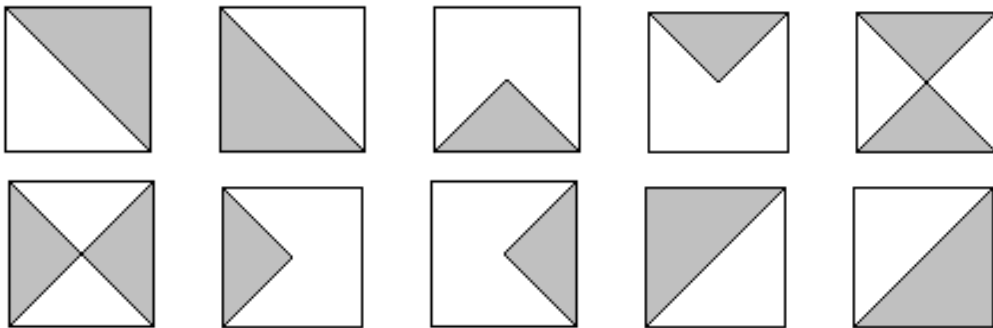


Рис. 1. Варіанти частин масивів

10. Без використання стандартних функцій пошуку в рядках визначити власні функції пошуку  $k$ -го ( $k$  – натуральне число) входження заданого підрядка в рядок із відліком від початку і від кінця рядка. Реалізувати меню вибору напрямку пошуку, що базується на використанні покажчиків на функції.

## 5. КОНТРОЛЬНІ ЗАПИТАННЯ

1. Як визначається покажчик на функцію?

2. Що виступає як значення покажчика на функцію?
3. Чим є ім'я функції у С++?
4. Як описується масив покажчиків на функцію?
5. Як визначити власний тип – покажчик на функцію?
6. Наведіть приклад прототипу функції, один або декілька параметрів якої мають тип «покажчик на функцію».
7. Наведіть приклад прототипу функції, що повертає значення яке має тип «покажчик на функцію».
8. Наведіть приклади прототипів функцій, що повертають покажчик на функцію.

## СПИСОК ЛІТЕРАТУРИ

1. Керниган, Б. Язык программирования Си / Б. Керниган, Д. Ритчи. – М. : Финансы и статистика, 1992. – 272 с.
2. Павловская, Т. А. С/С++. Программирование на языке высокого уровня / Т. А. Павловская. – СПб. : Питер, 2003. – 461 с.
3. Подбельский, В. В. Программирование на языке Си / В. В. Подбельский, С. С. Фомин. – М. : Финансы и статистика, 1999. – 600 с.
4. Страуструп, Б. Язык программирования Си++ : Второе издание / Б. Страуструп. – К. : ДиаСофт, 1993. – Ч. 1. – 264 с. ; Ч. 2. – 296 с.
5. Подбельский, В. В. Язык Си++ / В. В. Подбельский. – М. : Финансы и статистика, 1999. – 560 с.
6. Либерти, Дж. Освой самостоятельно С++ за 21 день : учеб. пособ. / Джесс Либерти. – М. : Вильямс, 2001. – 816 с.
7. Савитч, У. Язык С++. Курс объектно-ориентированного программирования / Уолтер Савитч. – М. : Вильямс, 2001. – 704 с.
8. Шилдт, Г. С++: руководство для начинающих / Герберт Шилдт. – М. : Вильямс, 2005. – 672 с.
9. Шилдт, Г. Самоучитель С++ / Г. Шилдт. – СПб. : ВHV-Петербург, 2003. – 688 с.
10. Шилдт, Г. Полный справочник по С++ / Герберт Шилдт. – М. : Вильямс, 2006. – 800 с.

Навчальне видання  
Методичні вказівки  
до лабораторної роботи  
«Використання покажчиків на функцію в програмах мовою C++»  
з курсу «Програмування» для студентів напряму  
6.040302 – Інформатика і курсу «Програмування  
та алгоритмічні мови» для студентів напряму  
6.040303 – Системний аналіз

Укладачі: БЕЗМЕНОВ Микола Іванович,  
БЕЗМЕНОВА Ольга Миколаївна

Відповідальний за випуск О. С. Куценко  
Роботу до видання рекомендував О. В. Горілий

За авторською редакцією

План 2014 р., поз. 108.

Підп. до друку 07.07.2014 р. Формат 60×84 1/16. Папір офсетний.  
Riso-друк. Гарнітура Таймс. Ум. друк. арк. 0,7. Наклад 50 пр.  
Зам. № 280. Ціна договірна.

---

Видавець і виготовлювач  
Видавничий центр НТУ «ХП»,  
вул. Фрунзе, 21, Харків, 61002.

Свідоцтво суб'єкта видавничої справи ДК № 3657 від 27.12.2009 р.