

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

С. Ю. Плєснецов

МЕТОДИЧНІ ВКАЗІВКИ

до виконання розрахункового завдання
з навчальної дисципліни «Прикладне програмування»
для студентів спеціальностей 175, G6
«Інформаційно-вимірювальні технології»

Затверджено
редакційно-видавничою
радою НТУ «ХПІ»,
протокол № 1 від 13.02.2025 р.

Харків
НТУ «ХПІ»
2025

Методичні вказівки до виконання розрахункового завдання з навчальної дисципліни «Прикладне програмування» для студентів спеціальностей 175, G6 «Інформаційно-вимірювальні технології» / уклад. С. Ю. Плєснецов. – Харків: НТУ «ХПІ», 2025. – 20 с.

Укладач С. Ю. Плєснецов

Рецензент С. Г. Львов

Кафедра інформаційно-вимірювальних технологій

ЗМІСТ

Вступ.....	4
Варіанти завдання	6
Вимоги до оформлення звіту з виконання завдання	10
Коротка довідкова інформація про структурні елементи мови C++	11
Приклад оформлення виконаного розрахункового завдання.....	16
Перелік рекомендованих джерел інформації для виконання завдань мовою C++	19

ВСТУП

Практичне завдання за дисципліною «Прикладне програмування» є засобом засвоєння, закріплення та перевірки знань, вмінь та навичок, придбаних при вивченні предмету.

Сучасне програмне забезпечення часто базується на принципі об'єктності, або об'єктно-орієнтованому програмуванні. Об'єктно-орієнтоване програмування (ООП) є одним з найбільш поширених методів прикладного програмування, який дає можливість структурування оптимізації та підвищення надійності виконання програмного коду.

ООП має набір принципів та особливостей, що відокремлюють його від класичного структурного або функціонального програмування, в першу чергу через широке, майже ексклюзивне використання об'єктів різних класів як носіїв блоків алгоритму, а також через наявність значного масиву особливостей, пов'язаних з наробленими за останні десятиліття шаблонами проектування об'єктно-орієнтованого програмного забезпечення.

Об'єктний підхід поширюється не тільки на безпосередньо розробку структури програми, але й на додаткові інструменти, які можуть бути важливими чи необхідними при реалізації програмних засобів, що розробляється для рішення тієї чи іншої задачі.

ООП визначається як парадигма програмування, заснована на концепції «об'єктів», які можуть містити дані та код: дані у вигляді полів (*атрибути* або *властивості*), і код у формі процедур та функцій (*методи*).

Особливістю об'єктів є те, що власні процедури об'єкта можуть отримувати доступ до полів даних і змінювати їх. В ООП комп'ютерні програми створюються шляхом створення їх з об'єктів, які взаємодіють один з одним. ООП підтримується багатьма мовами програмування.

В більшості випадків об'єкт є реалізацією абстрактного класу. Інакше кажучи, об'єкти є екземплярами класів, які також визначають їх типи, поля, методи тощо.

Багато з найбільш широко використовуваних мов програмування (такі як C++, Java, Python тощо) є мультипарадигмальними, і вони більшою чи меншою мірою підтримують об'єктно-орієнтоване програмування, як правило, у поєднанні з імперативним, процедурним програмуванням.

В даних методичних вказівках наведено завдання для виконання розрахункової роботи в межах дисципліни «Прикладне програмування». Вони спрямовані на закріплення знань та навичок, що необхідні для вирішення базових задач при програмуванні додатків з комплексним наповненням (додатки, що поєднують аналітичні – *розрахунок, обробка даних*, тощо – та мультимедійні – *візуалізація* – функції, а також операції збереження та відновлення даних), а також вимоги до структурування та оформлення звітів про виконання цих завдань.

ВАРІАНТИ ЗАВДАНЬ

Завдання за дисципліною «Прикладне програмування» включає в себе розробку програмного засобу та анування результатів цієї розробки. Важливим аспектом задачі є використання принципів ООП в реалізації програмного рішення, і якість їх використання є ключовим елементом в оцінюванні виконаного завдання викладачем.

Мова програмування, якою виконується завдання, або програмне середовище, у якому воно виконується, не є принциповими, за умови, якщо цією мовою чи у цьому середовищі можливе повне виконання задач, сформульованих у завданні. Рекомендованими мовами програмування для виконання завдань є: C++ (MS Visual Studio, Eclipse, тощо), C#, Delphi (Delphi, Lazarus).

Завдання наведено у двадцяти варіантах. Варіант обирається відповідно до номера у журналі академічної групи. За бажанням студент може виконати завдання варіанту, що не співпадає з його номером, за умовою, що цей варіант не виконується іншим студентом у групі та про це повідомлено викладача. Варіанти завдань наведено в табл. 1.

При виконанні завдання необхідним елементом програмного рішення є використання об'єктного підходу.

Вирішене завдання має містити:

1. Принаймні два об'єкти різних класів, розроблених виконавцем;
2. Взаємодію об'єктів;
3. Імплементацию шаблону ООП;
4. Реалізацію графічного інтерфейсу користувача та/або візуалізацію даних.

Таблиця 1 – Варіанти практичного завдання
за курсом «Прикладне програмування»

Номер варіанту	Завдання
1	2
1.	<p><i>Побудова математичної моделі</i></p> <p>Дано функцію $y(x) = a \cdot x^2 + b \cdot x + c$. Побудувати графік цієї функції на заданому користувачем інтервалі. Вивести перелік значень функції в межах цього інтервалу. Коефіцієнти функції задаються користувачем. Аргументом та функцією є значення x та y.</p>
2.	<p><i>База даних</i></p> <p>Реалізувати програмний засіб, здатний зберігати інформацію про співробітників у вигляді умовно необмеженої кількості записів – фамілія, ім'я, стать, вік, професія, посада. Зберегти ці дані на диску, відновити ці дані з диску. Вивести перелік співробітників, що відповідає певному значенню запису (наприклад, всіх співробітників з певною посадою). Бажано реалізувати додаток у вигляді додатку з графічним інтерфейсом. Формат збереження даних довільний.</p>
3.	<p><i>Функціональний аналіз</i></p> <p>Дано функцію $y(\varphi) = a \cdot \sin(b \cdot \varphi)$. Коефіцієнти функції задаються користувачем. Аргументом та функцією є значення φ та y. Знайти екстремуми цієї функції на заданому користувачем інтервалі, вивести їх, та візуалізувати.</p>
4.	<p><i>Побудова математичної моделі</i></p> <p>Дано функцію $y(x) = a \cdot x^3 + b \cdot x^2 + c \cdot x + d$. Побудувати графік цієї функції на заданому користувачем інтервалі. Вивести перелік значень функції в межах цього інтервалу. Коефіцієнти функції задаються користувачем. Аргументом та функцією є значення x та y.</p>
5.	<p><i>Аналіз даних</i></p> <p>Створити набір даних у вигляді таблиці розміром $w \times h$ використовуючи генератор випадкових чисел, зі значеннями у діапазоні, що задається користувачем. Вивести отриману таблицю. Відсортувати всі рядки таблиці за зростанням. Вивести отриману таблицю. Знайти найменше та найбільше значення в таблиці, вивести їх та їх координати. Бажано реалізувати додаток у вигляді додатку з графічним інтерфейсом.</p>
6.	<p><i>Функціональний аналіз</i></p> <p>Дано функцію $y(\varphi) = a \cdot (\sin(b \cdot \varphi) + \cos(c \cdot \varphi))$. Коефіцієнти функції задаються користувачем. Аргументом та функцією є значення φ та y. Знайти екстремуми цієї функції на заданому користувачем інтервалі, вивести їх, та візуалізувати.</p>
7.	<p><i>Побудова математичної моделі</i></p> <p>Дано функцію $y(x) = \frac{a}{b \cdot x^c}$. Побудувати графік цієї функції на заданому користувачем інтервалі. Вивести перелік значень функції в межах цього інтервалу. Коефіцієнти функції задаються користувачем. Аргументом та функцією є значення x та y.</p>

1	2
8.	<p><i>Вовк, коза та капуста</i> Дано класичну задачу про вовка, козу та капусту. На одному березі річки знаходяться рибалка з човном, якому необхідно перевезти на інший берег вовка, козу та капусту. В човен можуть поміститись лише рибалка та будь-який з інших трьох об'єктів. При цьому, якщо на березі без рибалки залишаються одночасно вовк та коза або коза та капуста, задача вважається не вирішеною. Переміщення об'єктів між човном та берегом, а також човна між берегами виконує користувач. Бажано реалізувати додаток у вигляді додатку з графічним інтерфейсом.</p>
9.	<p><i>Функціональний аналіз</i> Дано функцію $y(\varphi) = a \cdot \cos(b \cdot \varphi)$. Коефіцієнти функції задаються користувачем. Аргументом та функцією є значення φ та y. Знайти екстремуми цієї функції на заданому користувачем інтервалі, вивести їх, та візуалізувати. Значення a, b задає користувач.</p>
10.	<p><i>Побудова математичної моделі</i> Дано функцію $y(x) = a \cdot b^{c \cdot x}$. Побудувати графік цієї функції на заданому користувачем інтервалі. Вивести перелік значень функції в межах цього інтервалу. Коефіцієнти функції задаються користувачем. Аргументом та функцією є значення x та y.</p>
11.	<p><i>Програмна інтерполяція</i> Дано функцію $y(x) = a \cdot x^b + c$. Обчислити значення функції з кроком 1 на заданому користувачем інтервалі, довжина якого має перевищувати 10. Знайти за допомогою засобів інтерполяції значення функції у заданій користувачем точці, та порівняти це значення з аналітично обчисленим. Вивести результат графічно та текстом. a, b, c задає користувач.</p>
12.	<p><i>База даних</i> Реалізувати програмний засіб, здатний зберігати інформацію про товари на складі у вигляді умовно необмеженої кількості записів – назва, маса, об'єм, кількість одиниць, вартість, виробник. Зберегти ці дані на диску, відновити ці дані з диску. Вивести перелік товарів, що відповідає певному значенню запису (наприклад, всі товари з певним виробником). Бажано реалізувати додаток у вигляді додатку з графічним інтерфейсом. Формат збереження даних довільний.</p>
13.	<p><i>Програмна інтерполяція</i> Дано функцію $y(x) = a \cdot x^2 + b \cdot x + c$. Обчислити значення функції з кроком 1 на заданому користувачем інтервалі, довжина якого має перевищувати 10. Знайти за допомогою засобів інтерполяції значення функції у заданій користувачем точці, та порівняти це значення з аналітично обчисленим. Вивести результат графічно та текстом. a, b, c задає користувач.</p>
14.	<p><i>Рухомі елементи</i> Візуалізувати переміщення об'єкту за наведеним рівнянням руху: $y(t) = A \sin(f \cdot t)$, $x(t) = t$, де значення A та f задає користувач, а t є позначенням часу у секундах. Візуалізувати рух об'єкту за заданим рівнянням таким чином, щоб початкова точка його руху відповідала лівій частині вікна чи екрану, кінцева – правій частині вікна чи екрану, та об'єкт не покидав межі вікна чи екрану за увесь час переміщення. Початкова точка руху – 0 с, кінцева – 10 с.</p>

1	2
15.	<p><i>Системи рівнянь</i> Дано систему рівнянь виду:</p> $\begin{cases} a \cdot x + b \cdot y = c \\ d \cdot x + e \cdot y = f \end{cases}$ <p>де a, b, c, d, e, f – коефіцієнти, що задає користувач, x, y – невідомі. Знайти рішення цієї системи рівнянь та візуалізувати його.</p>
16.	<p><i>Матриця</i> Користувач задає квадратну матрицю A довільного розміру ($w \times w$). Виконати з даною матрицею наступні операції та вивести на екран їх результат: транспонування, обчислення визначника. При виконанні обчислення визначника рекомендується використовувати метод Гауса.</p>
17.	<p><i>Рухомі елементи</i> Візуалізувати переміщення об'єкту за наведеним рівнянням руху: $y(t) = A \sin(f \cdot t) + b \cdot t$, $x(t) = t$ де значення A, b та f задає користувач, а $t \in$ позначенням часу у секундах. Візуалізувати рух об'єкту за заданим рівнянням таким чином, щоб початкова точка його руху відповідала лівій частині вікна чи екрану, кінцева – правій частині вікна чи екрану, та об'єкт не покидав межі вікна чи екрану за увесь час переміщення. Початкова точка руху – 0 с, кінцева – 10 с.</p>
18.	<p><i>Квадратне рівняння</i> Знайти корені квадратного рівняння виду $a \cdot x^2 + b \cdot x + c = 0$ та візуалізувати рішення цього рівняння. a, b, c задає користувач, x – невідома.</p>
19.	<p><i>Програмна інтерполяція</i> Дано функцію $y(x) = a \cdot x^3 + b$. Обчислити значення функції з кроком 1 на заданому користувачем інтервалі, довжина якого має перевищувати 10. Знайти за допомогою засобів інтерполяції значення функції у заданій користувачем точці, та порівняти це значення з аналітично обчисленим. Вивести результат графічно та текстом. a, b задає користувач.</p>
20.	<p><i>Керований куб</i> Візуалізувати тривимірний куб, який може обертатись навколо двох вісей натисканнями клавіш «вверх» та «вниз», «вліво» та «вправо». Крок чи інтенсивність обертання – довільні чи регульовані. Вершину куба, що знаходиться найближче до «камери» позначити графічно, обчислити та вивести відстань до точки розміщення «камери».</p>

ВИМОГИ ДО ОФОРМЛЕННЯ ЗВІТУ З ВИКОНАННЯ ЗАВДАННЯ

Виконане завдання оформлюється у вигляді звіту. Звіт складається з:

- 1) титульного аркуша, де вказано прізвище виконавця, назву курсу, номер варіанту;
- 2) аркуша, де наведено номер варіанту та викладено завдання;
- 3) опису рішень, прийнятих для побудови алгоритму;
- 4) лістингу вихідного коду розробленого програмного рішення;
- 5) скріншотів програмного засобу у режимі виконання;
- 6) висновків;
- 7) за необхідності – посилань на використані джерела.

Текст завдання виконується шрифтом *Times New Roman*, розмір 14 пт, міжрядковий інтервал 1,5, без додаткового інтервалу між абзацами.

Заголовки підрозділів розміщуються посередині, виділяються жирним та набираються прописними літерами.

Лістинги наводяться шрифтом Courier New, розмір 10 або 12 пт, інтервал 1 .. 1,25. Рекомендується використовувати засоби форматування коду, наприклад сервіс <https://hilite.me/>

У лістингів, скріншотів та таблиць має бути номер та назва, на кожний з цих елементів має бути посилання в тексті. За наявності посилань на джерела інформації – їх номер має бути вказаним у квадратних дужках в місці посилання.

Перелік використаних джерел рекомендується оформлювати згідно вимог ДСТУ 8302:2015 Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання [1].

КОРОТКА ДОВІДКОВА ІНФОРМАЦІЯ ПРО СТРУКТУРНІ ЕЛЕМЕНТИ МОВИ C++

Наведемо короткий перелік структурних елементів мови C++. Цей перелік є далеким від вичерпного. Для повних даних по синтаксису мови C++ рекомендуємо звертатись до джерел [2, 3]. Докладні довідкові відомості доступні за посиланнями [4–6].

1. Включення файлу заголовку

```
#include <назва>
```

```
#include "назва файлу.h"
```

2.1. Об'явлення змінних

```
тип_змінної змінна1=початкове значення, змінна2=початкове значення,  
змінна3;
```

2.2. Операції зі змінними

```
v1 = 10; //присвоєння значення
```

```
v3 = v1 + v2; // складання
```

```
v3 = v1 - v2; // віднімання
```

```
v3 = v1 * v2; // множення
```

```
v3 = v1 / v2; // ділення
```

```
v1++; // збільшення значення на 1
```

2.3. Логічні операції

Ці операції повертають бульове (логічне) значення, тобто true або false

```
(a == b) //дорівнює
```

```
(a != b) //не дорівнює
```

```
(a < b) // менше
```

```
(a > b) //більше
```

```
(a <= b) //менше чи дорівнює
```

```
(a >= b) //більше чи дорівнює
```



```

class назва_класу : public клас_попередник
{
private:
    тип_поля: приватне_поле_1;
    тип_поля: приватне_поле_2, приватне_поле_3;
public:
    тип_поля: публічне_поле_1;
    тип_поля: публічне_поле_2, публічне_поле_3;

    назва_класу()
    {
        //дії в конструкторі класу
    }
    тип назва_методу()
    {
        //дії в межах методу класу
        return значення відповідного типу;
    }
}

```

```

} назва_об'єкту1, назва_об'єкту2 .....;

```

8. Умовне розгалуження

```

if (логічний_вираз)
    { // код, якщо (логічний_вираз == true) }
else
    { // код, якщо (логічний_вираз == false) }

```

9. Цикл з лічильником

```

for (int лічильник = початкове_значення; лічильник <= кінцеве_значення;
лічильник++)
{
    // операції в межах циклу
    // будуть повторюватись, поки ((лічильник <= кінцеве_значення) == true)
}

```

10. Цикл з передумовою

while (логічний_вираз)

```
{
    // код, що буде повторюватись, поки (логічний_вираз == true)
}
```

Приклад простого коду з основними синтаксичними елементами

```
#include <iostream>
#include <vector>

using namespace std;

int n;

class solver
{
private:
    vector<float> values = {};
public:
    solver()
    {
        values.clear();
    };
    void getvalues(int att)
    {
        if (att > 1)
        {
            values.clear();
            for (int i = 0; i < att; i++)
            {
                float k = 0;
                cout << "Enter a value: \n";
                cin >> k;
                values.push_back(k);
            }
        }
    }
    int getvaluessize()
    {
        return values.size();
    }
    float findaverage()
    {
        float sum=0; int c=0;
        for (int i = 0; i < getvaluessize(); i++)
        {
            sum += values[i];
            c++;
        }
        return sum / c;
    }
} solver1;

int main()
```

```
{
    std::cout << "Input vector size:\n";
    cin >> n;
    if (n > 0)
    {
        solver1.getvalues(n);
        cout << "Average value:" << solver1.findaverage() << endl;
    }
    else
    {
        cout << "Not enough values\n";
    }
    system("pause");
}
```

**ПРИКЛАД ОФОРМЛЕННЯ ВИКОНАНОГО
РОЗРАХУНКОВОГО ЗАВДАННЯ**

Міністерство освіти і науки України
Національний технічний університет
«Харківський політехнічний інститут»
Кафедра інформаційно-вимірювальних систем

Розрахункове завдання
за дисципліною
«Прикладне програмування»
Варіант ____

Виконав студент гр. _____

Перевірив _____

Харків 20__

1. ВИХІДНІ ДАНІ ЗАВДАННЯ

Виконати аналіз функції $y(x) = \frac{ax^3+bx}{c}$ для заданих користувачем a, b, c на інтервалі $[x_1; x_2]$. Вивести значення екстремумів, візуалізувати функцію.

2. ХІД ВИКОНАННЯ ЗАВДАННЯ

Розроблено програмний засіб для виконання поставленої задачі. Використано мову програмування C++, компілятор Visual Studio 2017. Програму реалізовано для Windows API [1]. Вихідний код наведено у лістингу 1, скріншот додатку у режимі виконання – на рис. 1.

Використано наступні програмні бібліотеки:

В алгоритмі реалізовано об'єкти наступних класів:...

Лістинг 1 – Вихідний код розробленого додатку

```
#pragma once

#include "targetver.h"

#define WIN32_LEAN_AND_MEAN
#include <windows.h>

#include "stdafx.h"
#include <stdlib.h>
#include <malloc.h>
#include <memory.h>
#include <tchar.h>
#include "resource.h"
#include <string>

using namespace std;

#define MAX_LOADSTRING 100

#define EDIT_A 1024
#define EDIT_B 1025
#define EDIT_C 1026
#define EDIT_X1 1027
#define EDIT_X2 1028
#define STATIC_MIN 1029
#define STATIC_MAX 1030
#define BUTTON_EXEC 1031
#define PANEL_OUT 1032

.....
```

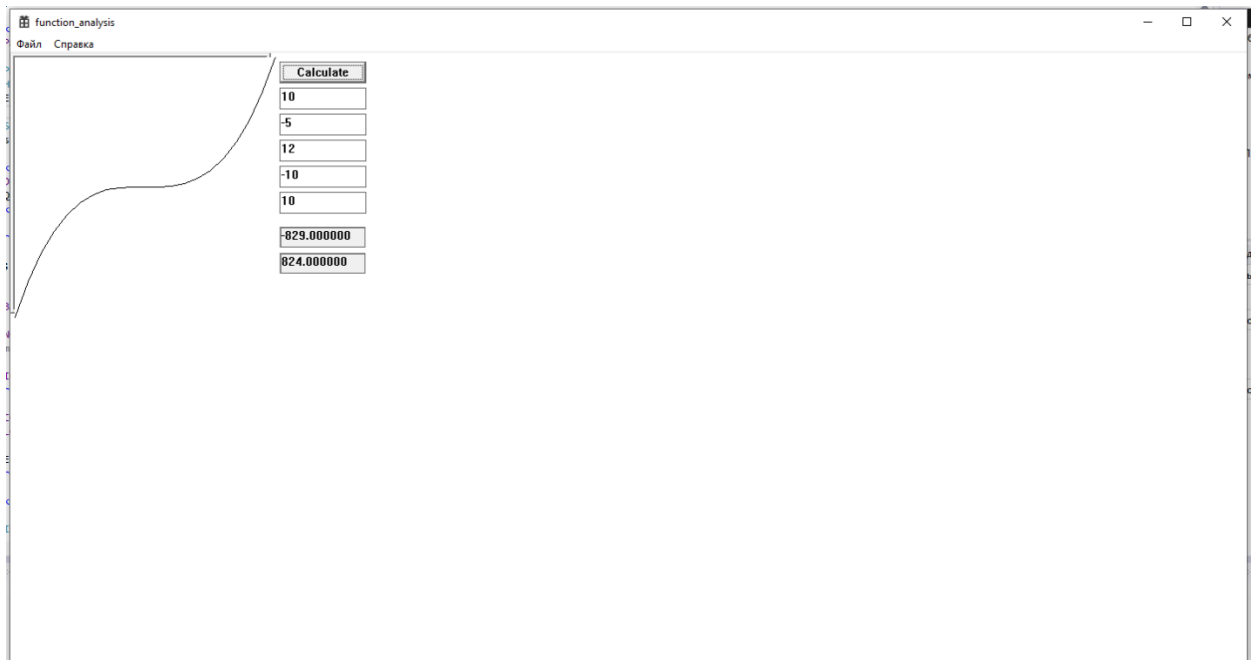


Рисунок 1 – Скріншот додатку у режимі виконання

ВИСНОВКИ

1. Розроблено програмний засіб за використанням API, мови програмування, IDE
2. Реалізовано наступні елементи об'єктного підходу: ...

ПЕРЕЛІК ДЖЕРЕЛ ІНФОРМАЦІЇ

1. Microsoft Developer Network [Електронний ресурс] / Microsoft // Режим доступу: <https://docs.microsoft.com/> – Назва з екрана. – Дата публікації : 10.01.2022. – Дата перегляду : 10.01.2022.

**ПЕРЕЛІК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ ІНФОРМАЦІЇ
ДЛЯ ВИКОНАННЯ ЗАВДАНЬ МОВОЮ C++**

1. ДСТУ 8302:2015 Інформація та документація. Бібліографічне посилання.
Загальні положення та правила складання
2. <https://isocpp.org/std/the-standard>
3. <http://www.cplusplus.com/>
4. <https://www.tutorialspoint.com/cplusplus/index.htm>
5. https://www.w3schools.com/cpp/cpp_oop.asp
6. <https://www.educative.io/blog/object-oriented-programming>

Навчальне видання

МЕТОДИЧНІ ВКАЗІВКИ

до виконання розрахункового завдання
з навчальної дисципліни «Прикладне програмування»
для студентів спеціальностей 175, G6
«Інформаційно-вимірювальні технології»

Укладач ПЛЕСНЕЦОВ Сергій Юрійович

Відповідальний за випуск

проф. Григоренко І.В.

Роботу до видання рекомендувала

доц. Дроздова Т.В.

В авторській редакції

План 2025 р., поз. 153

Видавничий центр НТУ «ХП».

Свідоцтво про державну реєстрацію ДК № 5478 від 21.08.2017 р.

61002, Харків, вул. Кирпичова, 2.

Електронне видання