# 11 СТВОРЕННЯ ЗАГОЛОВНИХ ФАЙЛІВ

**Мета**: Здобути практичні навички роботи зі створення та використання заголовного файлу.

**11.1 Завдання 1**

Створити програму де необхідно реалізувати функцію для виводу двумірного масиву на екран. Сигнатуру функції помістити в заголовочний файл "printArray.h". А вихідний код функції помітити в файл "printArray.cpp". Після чого визвати із головной функції програми (int main).

**11.1.1 Хід роботи**

**11.1.2 Програмування**

Введення тексту програми:

Лістинг 11.1 — Вихідний код файлу printArray.h

#ifndef PRINTARRAY\_H

#define PRINTARRRAY\_H

void printMatrix(int\*\* A, int a, int b);

#endif

Лістинг 11.2 — Вихідний код файлу printArray.cpp

#include <iostream>

#include "printArray.h"

using namespace std;

void printMatrix(int\*\* A, int a, int b)

{

for (int i = 0; i < a; i++)

{

for (int j = 0; j < a; j++)

cout << A[i][j] << '\t';

cout << endl;

}

}

Лістинг 11.3 — Вихідний код файлу Source.cpp

#include "Fraction.h"

#include "printArray.h"

#include <iostream>

using namespace std;

int i, j, a, b;

int main()

{

cout << "Input b=";

cin >> b;

cout << "Input a=";

cin >> a;

int\*\* A = new int\* [a];

for (int i = 0; i < a; i++)

A[i] = new int[b];

for (int i = 0; i < a; i++)

for (int j = 0; j < b; j++)

A[i][j] = rand() % 100;

printMatrix(A, a, b);

for (int i = 0; i < a; i++)

delete A[i];

delete[]A;

system("pause");

return 0;

}

***11.1.3 Обчислення, обробка і аналіз результатів***

У ході виконання данного завдання отримано наступні результати:

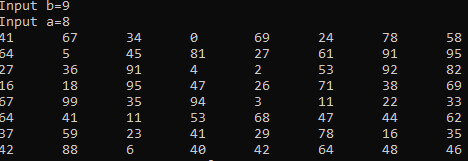


Рис.11.1 — Результат обчислень

**11.2 Завдання 2**

Написати програмний код, який виконує наступні кроки: - створити структуру Point (точка) з полями X та Y (координати) до окремого файлу Point. - розробити функцію для розрахунку відстані між двома точками з іменем getDistance(point, point). - заголовок функції розташувати в заголовочному файлі, код програми в іншому файлі.

**11.2.1 Хід роботи**

**11.2.2 Програмування**

Введення тексту програми:

Лістинг 11.4 — Вихідний код файлу point.h

#ifndef POINT\_H

#define POINT\_H

struct point

{

int a;

int b;

};

double getDistance(point, point);

#endif

Лістинг 11.5 — Вихідний код файлу point.cpp

#include "point.h"

#include <cmath>

using namespace std;

double getDistance(point aPoint, point bPoint)

{

return sqrt((aPoint.a - bPoint.b) \* (aPoint.a - bPoint.a) + (aPoint.b - bPoint.b) \* (aPoint.b - bPoint.b));

}

Лістинг 11.6 — Вихідний код файлу Source.cpp

#include "point.h"

#include <iostream>

using namespace std;

int main()

{

point a, b;

cout << "a.a=";

cin >> a.a;

cout << "a.b=";

cin >> a.b;

cout << "b.a=";

cin >> b.a;

cout << "b.b=";

cin >> b.b;

cout << getDistance(a, b) << endl;

system("pause");

return 0;

}

***11.2.3 Обчислення, обробка і аналіз результатів***

У ході виконання данного завдання отримано наступні результати:



Рис.11.2 — Результат обчислень

**11.3 Завдання 3**

Створити структуру Circle – коло в середні структури є два поля: point (структура Point) - координата на осі координат - radius (типа int або double) - довжина радіусу; - розробити функцію, яка приймає два параметра типу Circle та перевіряє чи перетинаються кола.

**11.3.1 Хід роботи**

**11.3.2 Програмування**

Введення тексту програми:

Лістинг 11.7 — Вихідний код файлу circle.h

#ifndef CIRCLE\_H

#define CIRCLE\_H

struct circle

{

struct Point {

int a;

int b;

} point;

int r;

};

bool isIntersectionCircle(circle circle1, circle circle2);

int getDistance(circle circle1, circle circle2);

circle makeCircle();

#endif

Лістинг 11.8 — Вихідний код файлу circle.cpp

#include<iostream>

#include"Circle.h"

using namespace std;

int getDistance(circle cycle1, circle cycle2)

{

return sqrt(pow(cycle2.point.a - cycle1.point.a, 2) + pow(cycle2.point.b - cycle1.point.b, 2));

}

circle makeCircle()

{

circle result;

cout << "x = ";

cin >> result.point.a;

cout << "y = ";

cin >> result.point.b;

cout << "r = ";

cin >> result.r;

return result;

}

bool isIntersectionCircle(circle cycle1, circle cycle2)

{

int dist = getDistance(cycle1, cycle2);

int sumR = cycle1.r + cycle2.r;

return dist < sumR;

}

Лістинг 11.9 — Вихідний код файлу Source.cpp

#include<iostream>

#include"Circle.h"

using namespace std;

int main()

{

circle cycle1 = makeCircle();

circle cycle2 = makeCircle();

bool isIntersect = isIntersectionCircle(cycle1, cycle2);

cout << "Cycle " << (isIntersect ? "are" : "aren't") << " intersect";

}

***11.3.3 Обчислення, обробка і аналіз результатів***

У ході виконання данного завдання отримано наступні результати:

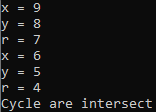


Рис.11.3 — Результат обчислень

**11.4 Програми та обладнання**

При опрацюванні данної лабараторної роботи я використав такі програмні продукти:

* Visual Studio 2022;
* Microsoft Word

**11.5 Висновки**

Попрактикувався у роботі з використанням заголовного файлу