

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут
імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 4 з дисципліни
«Основи програмування 2. Модульне програмування»

Успадкування та поліморфізм

Варіант 9

Виконав студент ІП-14 Демченко Філіпп Ігорович
(шифр, прізвище, ім'я, по батькові)

Перевірів Вітковська Ірина Іванівна
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота 4

Успадкування та поліморфізм

Мета – вивчити механізми створення та використання класів і об'єктів.

Завдання.

9. Створити клас TMatrix, який представляє матрицю і містить методи для обчислення детермінанта та суми елементів матриці. На основі цього класу створити класи-нащадки, які представляють квадратні матриці 2-го та 3-го порядку. За допомогою цих класів обчислити вираз

$$S = \left(\sum_{i=1}^3 \sum_{j=1}^3 a_{ij} \right) + |A| + |B|,$$

де $A = \|a_{ij}\|_1^3$ – матриця 3-го порядку, а $B = \|b_{ij}\|_1^2$ – матриця 2-го порядку.

Код програми

1) C++

Tmatrix.hpp

```
• TMatrix.hpp CPP/TMatrix.hpp/TMatrix/PrintMatrix()

#ifndef TMATRIX_H
#define TMATRIX_H

class TMatrix {
protected:
    unsigned int dimension;
    int **matrix;

public:
    TMatrix(unsigned int i_dimension);
    TMatrix();
    virtual ~TMatrix();

    int Sum() const;
    virtual int Det() const = 0;

    void EnterMatrix();
    void PrintMatrix();
};

#endif
```

Реалізація методів класу Tmatrix. Tmatrix.cpp

```
TMatrix.cpp CPP/TMatrix.cpp/~TMatrix()

#include "TMatrix.hpp"
#include <cstdio>

TMatrix::TMatrix(unsigned int i_dimension)
{
    dimension = i_dimension;
    matrix = new int *[dimension];

    for (unsigned int i = 0; i < dimension; ++i){
        matrix[i] = new int [dimension];
    }
}

TMatrix::TMatrix()
{
    dimension = 0;
    matrix = 0;
}

TMatrix::~TMatrix()
{
    for (unsigned int i = 0; i < dimension; ++i){
        if (matrix[i])
            delete [] matrix[i];
    }

    if (matrix)
        delete [] matrix;
}
```

```

void IMatrix::EnterMatrix()
{
    printf("Enter %d-dimension matrix row by row:\n", dimension);

    for (unsigned int i = 0; i < dimension; ++i){
        putchar('\t');
        for (unsigned int j = 0; j < dimension; ++j){
            scanf("%d", &matrix[i][j]);
        }
    }
    putchar('\n');
}

void TMatrix::PrintMatrix()
{
    for (unsigned int i = 0; i < dimension; ++i){
        putchar('\t');
        for (unsigned int j = 0; j < dimension; ++j){
            printf("%-5d ", matrix[i][j]);
        }
        putchar('\n');
    }
    putchar('\n');
}

int TMatrix::Sum() const
{
    int result = 0;

    for (unsigned int i = 0; i < dimension; ++i){
        for (unsigned int j = 0; j < dimension; ++j){
            result += matrix[i][j];
        }
    }
    return result;
}

```

Похідні класи Matrix2 та Matrix3

```
Matrix2.hpp CPP/Matrix2.hpp Matrix2
#ifndef MATRIX2_H
#define MATRIX2_H

#include "TMatrix.hpp"

class Matrix2 : public TMatrix {
public:
    Matrix2() : TMatrix(2) {}
    virtual ~Matrix2() {}

    virtual int Det() const {
        return matrix[0][0] * matrix[1][1] - matrix[0][1] * matrix[1][0];
    }
};

#endif
```

```
Matrix3.hpp CPP/Matrix3.hpp Matrix3 Det() const

class Matrix3 : public TMatrix {
public:
    Matrix3() : TMatrix(3) {}
    virtual ~Matrix3() {}

    virtual int Det() const {
        return + matrix[0][0] * matrix[1][1] * matrix[2][2]
            + matrix[0][1] * matrix[1][2] * matrix[2][0]
            + matrix[1][0] * matrix[2][1] * matrix[0][2]
            - matrix[0][2] * matrix[1][1] * matrix[2][0]
            - matrix[0][1] * matrix[1][0] * matrix[2][2]
            - matrix[1][2] * matrix[2][1] * matrix[0][0];
    }
};

#endif
```

Головна програма. MainProgram.cpp

```
MainProgram.cpp CPP/MainProgram.cpp main(int, char **)

#include "Matrix2.hpp"
#include "Matrix3.hpp"
#include <cstdio>

int main(int argc, char **argv)
{
    Matrix3 A;
    Matrix2 B;

    A.EnterMatrix();
    B.EnterMatrix();

    printf("sum(A) = %d\n", A.Sum());
    printf("det(A) = %d\n", A.Det());
    printf("det(B) = %d\n", B.Det());

    printf("sum(A) + det(A) + det(B) = %d\n", A.Sum() + A.Det() + B.Det());

    return 0;
}
```

Результат виконання

```
filipp@machine: ~/programming/KPI/labs_op/II_semestr/lab_05/CPP
filipp@machine:~/programming/KPI/labs_op/II_semestr/lab_05$ cd CPP
filipp@machine:~/programming/KPI/labs_op/II_semestr/lab_05/CPP$ ./a.out
Enter 3-dimension matrix row by row:
  1 -3 4
  2 3 5
  5 -3 4

Enter 2-dimension matrix row by row:
  1 2
  5 6

sum(A) = 18
det(A) = -108
det(B) = -4
sum(A) + det(A) + det(B) = -94
filipp@machine:~/programming/KPI/labs_op/II_semestr/lab_05/CPP$
```

2) Python

Класс Tmatrix

```
TMatrix.py PY/TMatrix.py
class TMatrix():
    def __init__(self, dimension):
        self.dimension = dimension
        self.matrix = []

    def EnterMatrix(self):
        print(f"Enter {self.dimension} matrix row by row:")
        for i in range(self.dimension):
            self.matrix.append(list(map(int, input().split())))

    def PrintMatrix(self):
        for i in range(self.dimension):
            for j in range(self.dimension):
                print(self.matrix[i][j], end = " ")
            print()

    def Sum(self):
        result = 0
        for i in range(self.dimension):
            for j in range(self.dimension):
                result += self.matrix[i][j]
        return result

    def Det():
        pass
```

Matrix2.py

```
Matrix2.py PY/Matrix2.py Matrix2/ __init__
from TMatrix import TMatrix

class Matrix2(TMatrix):
    def __init__(self):
        super().__init__(2)

    def Det(self):
        return self.matrix[0][0] * self.matrix[1][1] - self.matrix[0][1] * self.matrix[1][0]
```

Matrix3.py

```
Matrix3.py PY/Matrix3.py Matrix3/ Det
from TMatrix import TMatrix

class Matrix3(TMatrix):
    def __init__(self):
        super().__init__(3)

    def Det(self):
        return self.matrix[0][0] * self.matrix[1][1] * self.matrix[2][2] + self.matrix[0][1] * self.matrix[1][2] * self.mat
```


Головна програма. MainProgram.py

```
from Matrix2 import Matrix2
from Matrix3 import Matrix3


A = Matrix3()
B = Matrix2()

A.EnterMatrix()
B.EnterMatrix()

print(f"sum(A) = {A.Sum()}")
print(f"det(A) = {A.Det()}")
print(f"det(B) = {B.Det()}")

print(f"sum(A) + det(A) + det(B) = {A.Sum() + A.Det() + B.Det()}")
```

Результат виконання



```
filipp@machine: ~/programming/KPI/labs_op/II_semestr/lab_05/PY
filipp@machine:~/programming/KPI/labs_op/II_semestr/lab_05/PY$ python3 MainProgram.py
Enter 3 matrix row by row:
1 4 7
-5 6 4
3 4 5
Enter 2 matrix row by row:
3 4
5 6
sum(A) = 29
det(A) = -104
det(B) = -2
sum(A) + det(A) + det(B) = -77
filipp@machine:~/programming/KPI/labs_op/II_semestr/lab_05/PY$
```

Висновок.

В цій лабораторній роботі було вивчено основні механізми наслідування класів та засвоєно поняття поліморфізму.