**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНТЕЛЕКТУАЛЬНИХ**

**ТЕХНОЛОГІЙ І ЗВ’ЯЗКУ**

**Звіт**

**з дисципліни Проектний Практикум**

**Практична робота №7**

**на тему: «Проектування і опрацювання програми з віртуальними функціями»**

Виконав: студент 3 курсу, групи ІПЗ-3.04 спеціальності

121 Інженерія програмного забезпечення

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Бухта М.М.

Перевірив\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Багачук Д.Г.

**Одеса  2023**

**МЕТА РОБОТИ**

Засвоєння поняття віртуальної функції; набуття практичних навичок їх оголошення та використання.

**ЗАВДАННЯ**

**Опис завдання:**

1. Розробити і реалізувати програмно клас “Число”.
2. Розробити ієрархію класів: число (базовий), матриця (похідний).
3. Розробити та реалізувати програмно віртуальну функцію, яка обчислює факторіал числа, заданого в базовому класі, і використовується похідним класом, що містить масив цілих чисел, факторіали яких слід підрахувати та вивести у таблицю.

**Код програми:**

**main.cpp**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Laboratory work #6; \*

\* Student Bukhta Mykyta; \*

\* Grade: 3; \*

\* Group Software Engineering 3.04; \* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

#include "Matrix.hpp"

#include <random>

#include <iostream>

using namespace lab\_7;

int32\_t random(int32\_t min, int32\_t max) {

static std::random\_device dev;

static std::mt19937 rng(dev());

std::uniform\_int\_distribution<std::mt19937::result\_type> dist6(min, max); // distribution in range [1, 6]

return dist6(rng);

}

int main(int argc, char \*\*argv) {

Matrix matrix{10};

for (auto &arr : matrix.m\_matrix) {

for (auto &elem : arr) {

elem = random(1, 10);

}

}

matrix.printMatrix();

std::cout << "\n\n-----------------------\n\n" << std::endl;

matrix.printMatrixFactorial();

return 0;

}

**Number.hpp**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Laboratory work #6; \*

\* Student Bukhta Mykyta; \*

\* Grade: 3; \*

\* Group Software Engineering 3.04; \* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

#ifndef BUKHTAMYKYTA\_LAB7\_NUMBER\_HPP

#define BUKHTAMYKYTA\_LAB7\_NUMBER\_HPP

#include <inttypes.h>

namespace lab\_7 {

class Number {

public:

Number(uint64\_t number);

virtual ~Number(void) = default;

virtual uint64\_t factorial(int64\_t val) const noexcept;

protected:

int64\_t m\_number;

};

} // !lab\_7;

#endif // !BUKHTAMYKYTA\_LAB7\_NUMBER\_HPP;

**Number.cpp**

#include "Number.hpp"

namespace lab\_7 {

Number::Number(uint64\_t number) : m\_number(number) {

}

uint64\_t Number::factorial(int64\_t val) const noexcept {

if (val <= 0) {

return 0;

}

int64\_t ret{1};

for (int64\_t i{2}; i < val; ++i) {

ret \*= i;

}

return ret;

}

};

**Matrix.hpp**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Laboratory work #6; \*

\* Student Bukhta Mykyta; \*

\* Grade: 3; \*

\* Group Software Engineering 3.04; \* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

#ifndef BUKHTAMYKYTA\_LAB7\_MATRIX\_HPP

#define BUKHTAMYKYTA\_LAB7\_MATRIX\_HPP

#include "Number.hpp"

#include <array>

#include <vector>

#define \_\_MATRIX\_SIZE\_BY\_SPEC\_\_ 100

namespace lab\_7 {

class Matrix : Number {

public:

Matrix(int64\_t number);

void printMatrixFactorial(void) const noexcept;

void printMatrix(void) const noexcept;

public:

std::vector<std::array<int64\_t, \_\_MATRIX\_SIZE\_BY\_SPEC\_\_>> m\_matrix;

};

} // !lab\_7;

#endif // !BUKHTAMYKYTA\_LAB7\_MATRIX\_HPP;

**Matrix.cpp**

#include "Matrix.hpp"

#include <iostream>

namespace lab\_7 {

Matrix::Matrix(int64\_t number) : Number(number) {

m\_matrix.resize(number);

}

void Matrix::printMatrixFactorial(void) const noexcept {

for (const auto &arr : m\_matrix) {

std::cout << "| ";

for (const auto &elem : arr) {

std::cout << this->factorial(elem) << ", ";

}

std::cout << "|" << std::endl;

}

}

void Matrix::printMatrix(void) const noexcept {

for (const auto &arr : m\_matrix) {

std::cout << "| ";

for (const auto &elem : arr) {

std::cout << elem << ", ";

}

std::cout << "|" << std::endl;

}

}

} // !lab\_7;

**Результат виконання:**

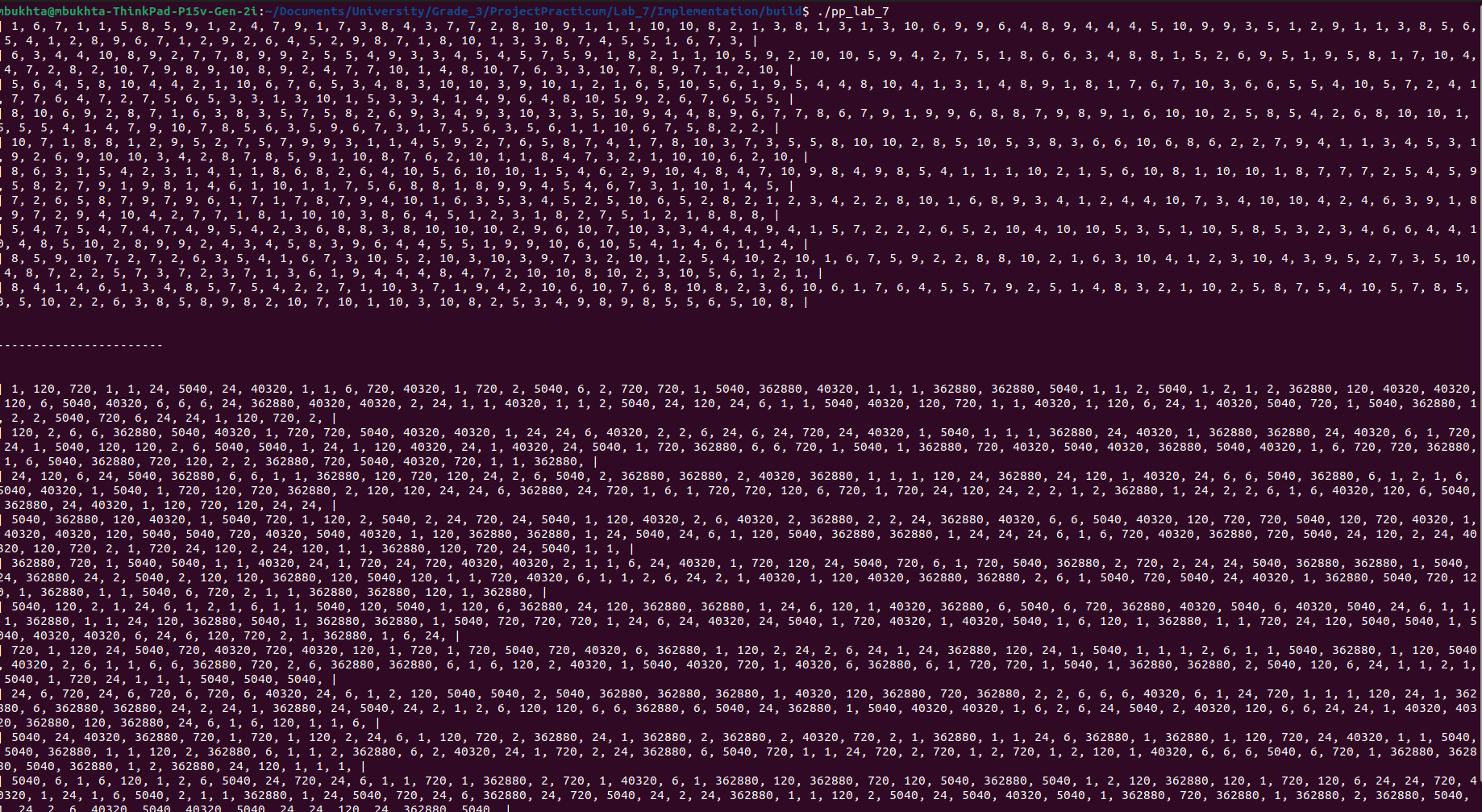


Рисунок 1.1 – результат виконання програми