

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп'ютерні інформаційні технології»

Лабораторна робота №9

з дисципліни «Об'єктно-орієнтоване програмування»

на тему: «Шаблони класів»

Виконав:
студент гр.ПЗ1911
Сафонов Д. Є.
Прийняла:
Демидович І.М.

Дніпро, 2021

Тема. Шаблони класів

Завдання. Написати програму на мові C++, яка містить шаблонну функцію, шаблонний клас з параметрами-типами та параметрами-константами. Передбачити спеціалізацію шаблону.

Індивідуальні завдання. Клас та функцію обрати за власними вподобаннями для довільної предметної області.

Текст програми.

“main.cpp”

```
#include "matrix.h"
#include "print.h"
#include <iostream>

int main() {
    const size_t R = 4, C = 3;
    Matrix<int, R, C> m;
    for (size_t i = 0; i < R; i++)
        for (size_t j = 0; j < C; j++)
            m.at(i, j) = i * j;

    print(m);
    print_matrix(m);
    std::cout << std::endl;
}
```

“matrix.h”

```
#ifndef __MATRIX_H__
#define __MATRIX_H__

#include <cstdint>

template<class T, size_t ROWS, size_t COLS>
class Matrix {
    T __matrix[ROWS][COLS];
public:
    T get(size_t x, size_t y) const {return __matrix[x][y];}
    T& at(size_t x, size_t y) {return __matrix[x][y];}
};

/* SEPARATE FUNCTION BECAUSE PARTIAL TEMPLATE SPECIFICATION ISNT ALLOWED
FOR FUNCTIONS */
template<class T, size_t ROWS, size_t COLS>
void print_matrix(const Matrix<T, ROWS, COLS>& m) {
    std::cout << "[ ";
    for (size_t i = 0; i < ROWS - 1; i++) {
        _print_row(m, i);
        std::cout << " ], ";
    }
    _print_row(m, ROWS — 1);
    std::cout << " ] ]";
}

template<class T, size_t ROWS, size_t COLS>
void _print_row(const Matrix<T, ROWS, COLS>& m, size_t i) {
    std::cout << "[ ";
    for (size_t j = 0; j < COLS - 1; j++) {
        print(m.get(i, j));
        std::cout << ", ";
    }
    print(m.get(i, COLS — 1));
}
#endif
```

“print.h”

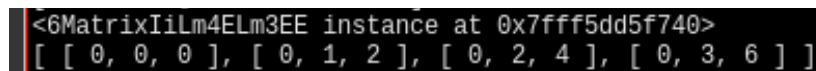
```
#ifndef __PRINT_H__
#define __PRINT_H__

#include <iostream>

template<class T>
void print(const T& obj) {
    std::cout << "<" << typeid(obj).name() << " instance at " << std::hex << &obj << ">" <<
    std::endl;
}

template<>
void print<int>(const int& obj) {
    std::cout << obj;
}
#endif
```

Приклад роботи програми.

A screenshot of a terminal window showing the output of a C++ program. The output consists of two lines. The first line is the address of the object: "<6MatrixIiLm4ELm3EE instance at 0x7fff5dd5f740>". The second line is a 4x3 matrix of integers: "[[0, 0, 0], [0, 1, 2], [0, 2, 4], [0, 3, 6]]".

```
<6MatrixIiLm4ELm3EE instance at 0x7fff5dd5f740>
[ [ 0, 0, 0 ], [ 0, 1, 2 ], [ 0, 2, 4 ], [ 0, 3, 6 ] ]
```

Рисунок 1

Висновки.

На мою думку шаблони — дуже важлива частина не тільки мови C++, а інших мов, у яких вони(або їх аналоги) доступні для використання. Завдяки ним можна дуже сильно скорочувати код програми, і це майже нічого не коштує(тому що компіляція варіанту шаблону виконується тільки коли він потрібен).