

**Министр науки и высшего образования Российской
Федерации**

**Федеральное государственное автономное
образовательное учреждение высшего образования**

**«Национальный исследовательский университет
ИТМО»**

**Факультет информационных технологий и
программирования**

Лабораторная работа №5

Шаблоны, исключения

Выполнил студент группы № М3119

Самигуллин Руслан Рустамович

Подпись:

Проверил:

Повышев Владислав Вячеславович

Санкт-Петербург
2024

Написать шаблонную функцию или класс согласно варианту.
Описать класс-исключение или иерархию классов-исключений.
Генерировать исключения в соответствующих исключительных ситуациях.
Если у вас есть другие предложения по исключительным ситуациям — используйте их.

```
#include <iostream>
#include <stdexcept> // Для std::out_of_range
using namespace std;

// Шаблоная функция для поиска минимального из двух элементов
template<typename T>
T findMin(const T& a, const T& b) {
    if (a <= b) {
        return a;
    } else {
        return b;
    }
}

// Шаблонный класс для матрицы N*M элементов типа T
template<int N, int M, class T>
class Matrix {
private:
    T data[N][M]; // Двумерный массив элементов типа T
    int num_rows;
    int num_cols;

public:
    // Конструктор по умолчанию
    Matrix(int rows, int cols) : num_rows(rows), num_cols(cols) {
        if (num_cols < 0 || num_rows < 0) {
            throw std::invalid_argument("Невозможно создать матрицу с отрицательным количеством строк или столбцов");
        }
    }

    // Метод для доступа к элементу матрицы
    T& at(int i, int j) {
        if (i < 0 || i >= N || j < 0 || j >= M) {
            throw std::out_of_range("Индекс выходит за пределы допустимого диапазона");
        }
        return data[i][j];
    }
};

int main() {
    int a, b;
    cin >> a >> b;
    int min_int = findMin(a, b); // Выводится результат: 3
    std::cout << "Минимальное целое число: " << min_int << std::endl;

    double c, d;
    cin >> c >> d;
    double min_double = findMin(c, d); // Выводится результат: 2.71
    std::cout << "Минимальное дробное число: " << min_double << std::endl;

    int rows, cols;

    std::cout << "Введите количество строк матрицы: ";
    std::cin >> rows;

    std::cout << "Введите количество столбцов матрицы: ";
    std::cin >> cols;

    Matrix<100, 100, int> matrix(rows, cols);

    // Заполнение матрицы значениями, введенными пользователем
    for (int i = 1; i <= rows; ++i) {
        for (int j = 1; j <= cols; ++j) {
            std::cout << "Введите элемент [" << i << "][" << j << "]: ";
            std::cin >> matrix.at(i-1, j-1);
        }
    }

    // Печать матрицы на экран
    std::cout << "Матрица " << rows << "x" << cols << ":\n" << std::endl;
    for (int i = 0; i < rows; ++i) {
        for (int j = 0; j < cols; ++j) {
            std::cout << matrix.at(i, j) << " ";
        }
        std::cout << std::endl;
    }

    try {
        int x, y;
        std::cout << "Введите индексы элемента для доступа (i j): ";
        std::cin >> x >> y;

        // Проверка допустимости индексов и доступ к элементу матрицы
        int value = matrix.at(x-1, y-1);
        std::cout << "Значение по индексу (" << x << ", " << y << "): " << value << std::endl;
    }
    catch (const std::invalid_argument& ex) {
        std::cerr << "Ошибка: " << ex.what() << std::endl;
    }
    catch (const std::out_of_range& ex) {
        std::cerr << "Ошибка: " << ex.what() << std::endl;
    }

    return 0;
}
```

Результат:
12 13
Минимальное целое число: 12
0.5 0.6
Минимальное дробное число: 0.5
Введите количество строк матрицы: 2
Введите количество столбцов матрицы: 3
Введите элемент [1][1]: 1
Введите элемент [1][2]: 2
Введите элемент [1][3]: 3
Введите элемент [2][1]: 4
Введите элемент [2][2]: 5
Введите элемент [2][3]: 6
Матрица 2x3:
1 2 3
4 5 6
Введите индексы элемента для доступа (i j): 1 2
Значение по индексу (1, 2): 2

В программе реализованы шаблонные функции и класс для работы с минимальными значениями и матрицами.

1. Шаблонная функция - findMin:

- Эта функция принимает два аргумента любого сравнимого типа данных и возвращает минимальное из них.

- Функция использует шаблонный параметр T, что позволяет ей работать с различными типами данных.

- Примеры использования функции:

`int min_int = findMin(a, b);` - находит минимальное целое число между a и b.

`double min_double = findMin(c, d);` - находит минимальное дробное число между c и d.

2. Шаблонный класс - Matrix:

- Класс представляет матрицу размером NxM, где N и M - заданные при создании объекта числа строк и столбцов соответственно.

- Конструктор класса проверяет, что количество строк и столбцов неотрицательно.

- Класс предоставляет метод `at` для доступа к элементу матрицы по индексам.

- Метод проверяет допустимость индексов и выбрасывает исключение в случае выхода за границы матрицы.

- Пример создания объекта класса: `Matrix<100, 100, int> matrix(rows, cols);` - создает матрицу с заданным числом строк и столбцов типа `int`.

3. Ввод и вывод данных:

- Пользователь вводит значения для нахождения минимального числа, размеров матрицы и ее элементов.

- После ввода данных выводится результат нахождения минимального числа и матрица с введенными значениями.

4. Обработка исключений:

- Исключения `std::invalid_argument` и `std::out_of_range` выбрасываются при недопустимых аргументах конструктора и при попытке доступа к элементу матрицы с недопустимыми индексами соответственно.

- В случае возникновения исключений выводится соответствующее сообщение об ошибке.

Вопросы:

1. Что такое шаблон? Что бывает шаблонным?

Ответ: это обобщенный блок кода, который работает с разными типами данных или значениями. Шаблонным бывает функция, которая работает с обобщенными типами данных или значениями и класс, который хранит и обрабатывает данные.

2. Как компилируется код с шаблонами?

Ответ: Код с шаблонами компилируется в два этапа, во время компиляции и во время инстанциации. Во время компиляции проверяется синтаксис и типы данных. Компилятор проверяет, нет ли ошибок в коде и генерирует объектный код. Во время инстанциации компилятор создает конкретные версии шаблонов для каждого типа данных, с которыми они используются. Это происходит только тогда, когда шаблонная функция или класс используется в коде.

3. Какая специализация шаблонов в вашей реализации?

Ответ: в моем коде отсутствуют специализации реализации шаблонов. Однако, шаблонная функция `findMin` и шаблонный класс `Matrix` представляют собой, как обобщенные шаблоны.

4. Что такое исключение в C++? Как работает механизм исключений?

Ответ: это механизм обработки ошибок, который позволяет программисту обрабатывать исключения, которые могут появиться во время выполнения программы. Когда происходит такая ситуация, программа создает объект исключения, который передается по стеку вызовов в поисках блока кода, способного его обработать. Когда соответствующий блок обработки найден, управление передается туда, и код обработки выполняется. Если блок обработки не найден, программа завершает работу, а стек разматывается, освобождая ресурсы.

5. В каких ситуациях в вашей программе бросаются исключения и почему?

Продемонстрируйте класс-исключение или иерархию классов-исключений, которые необходимо было реализовать по условию.

Исключение `std::invalid_argument` бросается при создании объекта класса `Matrix`, если количество строк или столбцов задается отрицательным числом.

Исключение `std::out_of_range` бросается при попытке доступа к элементу матрицы с индексами, выходящими за допустимые границы.