

**Министерство 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

Вариант 21.

Шаблоны, исключения.

**Выполнил студент группы № М3111
Соловьев Михаил Александрович.**

Санкт-Петербург
2023

В данном задании необходимо написать шаблонную функцию или класс согласно варианту. Описать класс-исключение или иерархию классов-исключений. Генерировать исключения в соответствующих исключительных ситуациях.

Код:

```
exception.h
#ifndef COURSE_C__EXCEPTION_H
#define COURSE_C__EXCEPTION_H

#include <iostream>
#include <exception>

class Exception : public std::exception {
public:
    Exception(const char* errr) : error(errr) {}
    const char* what() const noexcept {
        return error;
    }
private:
    const char* error;
};

template<typename T, int n>
class Array {
private:
    T arr[n];
public:
    Array() = default;

    Array(int cnt) {
        if (cnt > n) {
            throw Exception("ERROR!");
        }
        for (int i = 0; i < cnt; i++) {
            std::cin >> arr[i];
        }
    }

    T& operator[](int i) {
        if (i < 0 || i >= n) {
```

```

        throw Exception("Index out of range!");
    }

    return arr[i];
}
};

```

```

#endif //COURSE_C___EXCEPTION_H

```

main.cpp

```

#include "exception.h"
#include "template.h"
#include <iostream>

```

```

int main() {
    int a[5] = {1, 2, 3, 4};
    std::cout << findMax<int>(a, 5);
    //try {
    //    std::cout << findMax<int>(a, 0);
    //} catch (Exception& e) {
    //    std::cerr << "Error: " << e.what() << '\n';
    //}
}

```

```

    Array<int, 5> arr(5);
    try {
        std::cout << arr[8];
    } catch (Exception& e) {
        std::cerr << "Error: " << e.what() << '\n';
    }
    return 0;
}

```

template.h

```

#ifndef COURSE_C___TEMPLATE_H
#define COURSE_C___TEMPLATE_H

```

```

#include "exception.h"

```

```

template<typename T>
T findMax(T *arr, int n) {
    if (n <= 0) {

```

```
        throw Exception("Array size must be greater than 0");
    }

    T maxElement = arr[0];
    for (int i = 1; i < n; i++) {
        if(arr[i] > maxElement) {
            maxElement = arr[i];
        }
    }

    return maxElement;
}

#endif //COURSE_C___TEMPLATE_H
```

Вывод:

Я написал шаблонную функцию и научился работать с исключениями