

Федеральное государственное автономное  
образовательное учреждение высшего  
образования  
«Национальный исследовательский университет  
ИТМО»

Факультет Информационных технологий и программирования

Программирование на C++

Лабораторная работа №5, Шаблоны, исключения.

Вариант 5

Выполнил: Гаджиев Саид М3115

Санкт-Петербург

2023 г.

## Шаблоны, исключения.

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.

### Вариант 5

Решение:

```
#include <iostream>
#include <stdexcept>

template<typename T>
T find_max(T *arr, int size) {
    if (size <= 0) {
        throw std::invalid_argument("Array size must be positive.");
    }
    T max_val = arr[0];
    for (int i = 1; i < size; i++) {
        if (arr[i] > max_val) {
            max_val = arr[i];
        }
    }
    return max_val;
}

template <int N, class T>
class Array {
private:
    T arr[N];

public:
    T &operator[](int index) {
        if (index < 0 || index >= N) {
            throw std::out_of_range("Array index out of range: " +
std::to_string(index));
        }
        return arr[index];
    }
};

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    double double_arr[] = {1.1, 2.2, 3.3, 4.4, 5.5};
    try {
        std::cout << "Max value of int array: " << find_max(arr, 5) <<
std::endl;
        std::cout << "Max value of double array: " << find_max(double_arr, 5)
<< std::endl;
        // Uncomment this to test invalid argument exception
        // std::cout << "Max value of empty array: " << find_max(int_arr, 0)
<< std::endl;
    }
    catch (const std::invalid_argument& e) {
        std::cerr << "Exception: " << e.what() << std::endl;
    }

    Array<5, int> int_arr;
    try {
        int_arr[0] = 1;
        int_arr[1] = 2;
```

```

std::cout << "Element at index 0: " << int_arr[0] << std::endl;
std::cout << "Element at index 1: " << int_arr[1] << std::endl;
// Uncomment this to test out of range exception
// std::cout << "Element at index -1: " << int_arr[-1] << std::endl;
}
catch (const std::out_of_range& e) {
    std::cerr << "Exception: " << e.what() << std::endl;
}
return 0;
}

```

**find\_max** - это функция-шаблон, которая принимает указатель на массив любого типа и возвращает максимальное значение в массиве. Если размер массива отрицательный или равен нулю, функция генерирует исключение `std::invalid_argument`.

**Array** - это класс-шаблон, который представляет статический массив заданного размера *N* элементов любого типа *T*. Класс предоставляет оператор доступа к элементам `operator[]`, который возвращает ссылку на элемент массива по индексу. Если индекс находится за пределами допустимого диапазона, оператор генерирует исключение `std::out_of_range`.

В функции `main` создаются два массива: `arr` и `double_arr`, содержащие целочисленные и вещественные числа соответственно. Затем вызывается функция `find_max` для каждого массива, и результат выводится на экран. Также есть закомментированный вызов функции `find_max` с нулевым размером массива для тестирования генерации исключения.

Затем создается объект класса `Array` с размером 5 элементов и типом `int`. Оператор `[]` используется для доступа к элементам массива, а также есть закомментированный вызов оператора `[]` с недопустимым индексом для тестирования генерации исключения.

В случае возникновения исключительных ситуаций (например, неверный размер массива или выход индекса за пределы массива) программа генерирует исключение, которое обрабатывается в блоке `catch`. Если исключение генерируется, оно выводится на экран, а программа завершается с кодом возврата 0.