

**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 12**

*Название работы:* Шаблоны, исключения

**Выполнил студент группы № М3113**

**Крамской Вадим Вадимович**

**Подпись:**



**Санкт-Петербург**

**2023**

## Условие:

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.

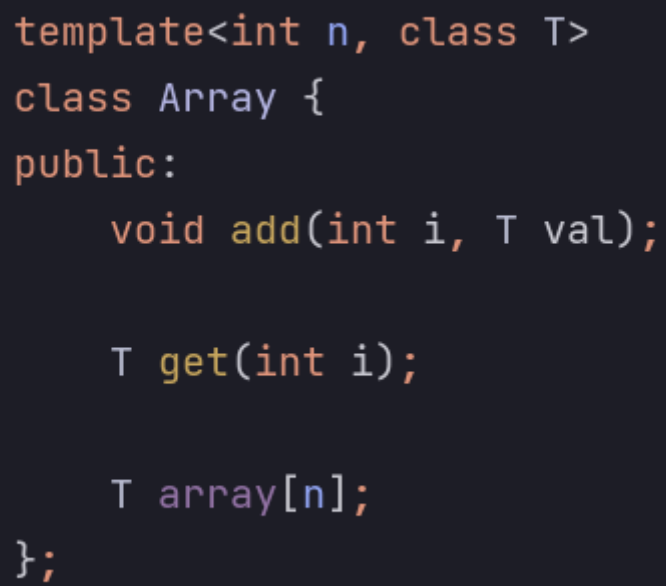
Функции			
	Описание	Параметры шаблона	
A	Поиск максимального элемента в массиве.	Тип элементов массива	
B	Поиск минимального из двух элементов.	Тип элементов.	
C	Меняет значения двух переменных одного типа местами (swap)	Тип переменных.	
D	Вычисляет N-ю (целую) степень числа	Степень (показатель) – целое число. Тип переменной.	
Классы			
	Описание	Параметры шаблона	Исключения
E	Стек из макс. N элементов типа T	int N, class T	Переполнение стека. Изъятие элемента из пустого стека.
F	Массив из N элементов типа T	int N, class T	Обращение по несуществующему индексу.
G	Очередь из макс. N элементов типа T	int N, class T	Переполнение очереди. Изъятие элемента из пустой очереди.
H	Матрица NxM элементов типа T	int N, int M, class T	Обращение по несуществующему индексу.

## Решение

### 1) Функция возведения в степень

```
template<typename T>
T power(int degree, T our) {
    return pow(our, degree);
}
```

### 2) Класс массива



```
template<int n, class T>
class Array {
public:
    void add(int i, T val);

    T get(int i);

    T array[n];
};
```

3)Добавление и удаление из массива

```

template<int n, class T>
void Array<n, T>::add(int i, T val) {
    if (i < 0 || i >= n)
        throw UnavailableIdxException();
    this->array[i] = val;
}

template<int n, class T>
T Array<n, T>::get(int i) {
    if (i < 0 || i >= n)
        throw UnavailableIdxException();
    return this->array[i];
}

```

#### 4) Класс-родитель исключения

```

class BaseException {
protected:
    std::string text;
public:
    virtual std::string what() = 0;
};

```

#### 5) Класс-наследник исключения

```
class UnavailableIdxException final : public BaseException {
public:
    UnavailableIdxException() {
        text = "This index unavailable, please try again \n";
    }

    UnavailableIdxException(std::string error) {
        text = std::move(error);
    }

    std::string what() override {
        return text;
    }
};
```