

ФГАОУВО «Южно-Уральский государственный университет (НИУ)»

Институт естественных и точных наук

Кафедра «Прикладная математика и программирование»

### ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

по дисциплине «Объектно-ориентированное программирование»

Автор работы

студент группы ЕТ-211

\_\_\_\_\_ А.А.Владимиров

\_\_\_\_\_ 2022 г.

Работа зачтена с оценкой

\_\_\_\_\_

\_\_\_\_\_ А.К.Демидов

\_\_\_\_\_ 2022 г.

Челябинск, 2022

## 1 Постановка задачи

I. Определить класс-шаблон с использованием динамического распределения памяти согласно варианту и необходимые конструкторы и операции, включая конструктор копий, операция присваивания и если указано операцию индексации. При выходе за границу, переполнении и т.п. вызвать исключительную ситуацию (определить собственные классы) для информирования программы, вызвавшей метод.

2. класс массив (одномерный массив элементов заданного типа), в аргументах конструктора задаются номер первого и последнего элемента в массиве как в языке Pascal.  
получение i-го элемента с помощью операции индексации

При определении друзей класса-шаблона использовать [следующий пример](#)

II. Реализовать main с тестами  
(создание объектов и выполнение действий с ними, в т.ч. действие, приводящее к возникновению исключительной ситуации, которую необходимо перехватить)

## 2 Описание интерфейса классов

```
struct arrayerror { // базовый класс для ошибок
    virtual ~arrayerror() {} // деструктор
    virtual const char *what() const=0; // сообщение для
печати
};

struct arrayoutofrange: arrayerror {
    const char *what() const {return "Индекс вне
диапазона";} // сообщение для печати
};

template <typename T>
class Array {
    T *a; // указатель на данные в стеке
    int first, // текущее количество
        end;
public:
    Array(int first, int end): first(first), end(end),
a(new T[1+end-first]) {} // конструктор
    Array(const Array<T> &); // конструктор копий
    ~Array() throw() {delete []a;} // деструктор
    T &operator[](int); // поиск по ключу
```

```

    T operator[](int) const; // поиск по ключу
};

```

### 3 Описание тестов для проверки классов

```

int main()
{
    Array<int> obj(2, 10);
    cout<<"Тест 1. Добавление\n";
    try {
        for (int i=2;; i++) {
            obj[i] = i;
        }
    }
    catch (arrayerror &e) {
        cout<<e.what();
    }

    cout<<"\nТест 2. Получение по индексу\n";
    try {
        for (int i=2;; i++) {
            cout << (obj[i]) << ' ';
        }
    }
    catch (arrayerror &e) {
        cout<<"\n"<<e.what();
    }
    return 0;
}

```

#### Полученные результаты

Тест 1. Добавление

Индекс вне диапазона

Тест 2. Получение по индексу

2 3 4 5 6 7 8 9 10

Индекс вне диапазона

### 4 Листинг реализации класса

```

template <typename T>
Array<T>::Array(const Array <T> &array): a(new
T[1+array.end-array.first]), first(array.first),
end(array.end) {
    for (int i = first; i <= end; i++)
        a[i]=array.a[i];
}

template <typename T>

```

```

T &Array<T>::operator[](int index)
{
    if(index < first or index > end) throw
arrayoutofrange();
    return a[index-first];
}

template <typename T>
T Array<T>::operator[](int index) const
{
    if(index < first or index > end) throw
arrayoutofrange();
    return a[index-first];
}

```