

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тихоокеанский государственный университет»

Кафедра «Программное обеспечение вычислительной техники и  
автоматизированных систем»

## Реализация шаблонов классов

Лабораторная работа №3

по дисциплине «Объектно-ориентированное программирование»

Выполнил студент

Чекулаев В. Ю.

Факультет, группа

ФКФН, ПО(аб)-81

Проверил

Федосеев А. А.

Хабаровск – 2020г.

## 1. Постановка задачи

Согласно варианту реализовать класс-шаблон для реализации заданной структуры хранения. Для обработки ошибочных ситуаций использовать механизм исключений.

Написать тестовое приложение, демонстрирующее работу класса для данных простого и сложного типа (например запись), корректную обработку ошибок.

Вариант 6 : правый верхний треугольник квадратной матрицы. Размеры указываются в конструкторе. Операции:

- чтение элемента по индексу;
- запись элемента по индексу;

## 2. Содержание заголовочного файла класса-шаблона

```
#ifndef RIGHTTRIANGLE_H
#define RIGHTTRIANGLE_H

#include<cstdint>
#include<iostream>

template<class T>
class RightTriangle{
public:
    RightTriangle(int n = 2);
    ~RightTriangle(){ if(m_triangle){ delete[] m_triangle; } }

    T get_at(int i, int j) const;
    void set_at(int i, int j,const T& elem);

private:
    int m_n;
    T* m_triangle;
};

template<class T>
RightTriangle<T>::RightTriangle(int n) : m_n(n), m_triangle(nullptr){
    try{
        if(n <= 1) throw(1);

        u_int amount(0);
        while(n){
```

```

        amount += n;
        --n;
    }

    m_triangle = new T[amount]();
}
catch(int){
    std::cout << "RightTriangle<T>::RightTriangle(int n)::Uncorrect
size" << "\n";
    exit(1);
}
catch(std::bad_alloc){
    std::cout << "RightTriangle<T>::RightTriangle(int n)::Failed to
allocate memory" << "\n";
    exit(1);
}
}

template<class T>
T RightTriangle<T>::get_at(int i, int j) const{
    --i; --j;

    try{
        if(i > j || (i >= m_n && j >= m_n) || i < 0 || j < 0) throw(1);

        size_t n(m_n), pol(0);
        --n;

        while(i){
            pol += n;
            --n; --i;
        }
        pol += j;

        return m_triangle[pol];
    }
    catch(int){
        std::cout << "RightTriangle<T>::get_at(int i, int j)::Index out of
range\n";
        exit(1);
    }
}

template<class T>
void RightTriangle<T>::set_at(int i, int j, const T& elem){
    --i; --j;

    try{
        if(i > j || (i >= m_n && j >= m_n) || i < 0 || j < 0) throw(1);

        size_t n(m_n), pol(0);
        --n;

        while(i){
            pol += n;

```

```

        --n; --i;
    }
    pol += j;

    m_triangle[pol] = elem;
}
catch(int){
    std::cout << "RightTriangle<T>::set_at(int i, int j,const T&
elem)::Index out of range\n";
    exit(1);
}
}

#endif // RIGHTTRIANGLE_H

```

### 3. Список исключительных ситуаций при работе класса-шаблона:

- Неверное задание размеров матрицы в конструкторе
- Невозможность выделения достаточного количества памяти для хранения матрицы
- Неверное задание индексов при чтении матрицы
- Неверное задание индексов при записи значения в матрицу

#### 4. Текст одного из методов класса-шаблона, реализующего поставленную задачу

```
template<class T>
void RightTriangle<T>::set_at(size_t i, size_t j, const T& elem){
    --i; --j;

    try{
        if(i > j || (i >= m_n && j >= m_n) || i < 0 || j < 0) throw(1);

        size_t n(m_n), pol(0);
        --n;

        while(i){
            pol += n;
            --n; --i;
        }
        pol += j;

        m_triangle[pol] = elem;
    }
    catch(int){
        std::cout << "RightTriangle<T>::set_at(size_t i, size_t j, const T&
elem)::Index out of range\n";
        exit(1);
    }
}
```

#### 5. Пример работы класса-шаблона для двух разных типов

Содержание файла main.cpp:

```
#include<iostream>

#include<cstdint>

#include"righttriangle.h"

class TestClass{
public:
    TestClass(char c = '0') : m_c(c){}
    ~TestClass(){}

    char get() const{
        return m_c;
    }

private:
    char m_c;
};
```

```

int main(){
    size_t n = 5;
    RightTriangle<double> obj1(n);
    obj1.set_at(3, 4, 8);
    obj1.set_at(1, 3, 2);
    obj1.set_at(5, 5, 5);

    std::cout << "obj1:" << "\n";
    for(size_t i = 1; i <= n; ++i){
        for(size_t j = 1; j <= n; ++j){
            if(i <= j){
                std::cout << obj1.get_at(i, j);
            } else{
                std::cout << " ";
            }
        }
        std::cout << "\n";
    }

    n = 8;
    RightTriangle<TestClass> obj2(n);
    obj2.set_at(3, 7, TestClass('a'));
    obj2.set_at(1, 1, TestClass('g'));
    obj2.set_at(4, 6, TestClass('t'));
    obj2.set_at(5, 5, TestClass('v'));

    std::cout << "\nobj2:" << "\n";
    for(size_t i = 1; i <= n; ++i){
        for(size_t j = 1; j <= n; ++j){
            if(i <= j){
                std::cout << obj2.get_at(i, j).get();
            } else{
                std::cout << " ";
            }
        }
        std::cout << "\n";
    }
}

```

Вывод программы:

```
15:48:36: Запускается
obj1:
00200
0000
080
00
5

obj2:
g00000000
00000000
0000a0
00t00
v000
000
00
0
15:48:36: /home/alway,
```

## 6. Пример работы класса-шаблона по обработке исключительных ситуаций

Изменение в файле main.cpp:

Строка 41: «obj2.set\_at(4, 6, TestClass('t'))» изменено на «obj2.set\_at(4, -6, TestClass('t'))»

Вывод программы:

```
16:35:25: Запускается /home/alway/Документы/содт/3 курс/ООП/3/суд-3-Desktop
obj1:
00200
0000
080
00
5
RightTriangle<T>::set_at(int i, int j,const T& elem)::Index out of range
16:35:25: /home/alway/Документы/содт/3 курс/ООП/3/суд-3-Desktop_04_5_15_1
```