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 «Объекно-ориентированное программирование»

Выполнил студент Чекулаев В. Ю.

Факультет, группа ФКФН, ПО(аб)-81

Проверил Федосеев А. А.

Хабаровск – 2020г.

1. **Постановка задачи**

Согласно варианта 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;

1. **Содержание заголовочного файла класса-шаблона**

#ifndef RIGHTTRIANGLE\_H

#define RIGHTTRIANGLE\_H

#include<cstddef>

#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*

1. **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

1. **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<cstddef>

#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 << " ";

}

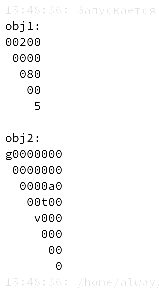
}

std::cout << "\n";

}

}

Вывод программы:

**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:

