**Министерство образования и науки Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**

**МЕХАНИКИ И ОПТИКИ”**

**ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**ОТЧЕТ**

**по Лабораторной работе № 10**

**«Решение задач с использованием шаблонов классов»**

Специальность 09.02.03 «Программирование в компьютерных системах»

Дисциплина «Прикладное программирование»

|  |  |
| --- | --- |
| Преподаватель:  Сорокин Д.Ю.  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2019г.  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Выполнил:  студент группы Y2335  Котлицкий С.А. |

Санкт-Петербург

2018/2019

**Цель работы:** продемонстрировать работу с механизмом шаблонов в объектно-ориентированном программировании

**Практическое задание:** Реализовать шаблон класса из класса восьмой лабораторной работы. Объект описать минимально возможным количеством полей, остальные характеристики вычислять динамически с помощью методов. Класс должен содержать минимум 2 конструктора: конструктор по умолчанию и параметризованный. Перегрузить операторы записи и чтения потока, для обеспечения потокового ввода-вывода. Перегрузить минимум 2 стандартных оператора(унарный и бинарный).

Написать программу, выполняющую:

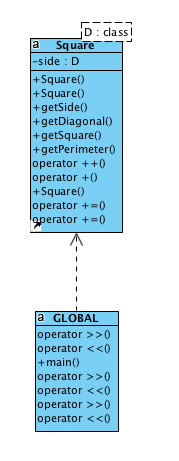
1. создание 2-х экземпляров шаблона класса, с инстанцированием для разных типов данных, с динамическим выделением памяти, первый с параметрами по умолчанию, второй с параметрами заданными пользователем;
2. вывод всех характеристик заданного экземпляра класса на экран;
3. демонстрацию работы перегруженных операторов;

освобождение выделенной памяти

**Индивидуальное задание:**

1. Квадрат.

Диаграмма классов программы представлена ниже (Рисунок 1):



*Рисунок 1 Диаграмма классов программы*

Код программы:

#include <iostream>

#include <math.h>

using namespace std;

template <class D>

class Square {

D side;

public:

Square(){

side = 1;

}

Square(D side){

this -> side = side;

}

D getDiagonal(){

return sqrt(2 \* side \* side);

}

D getSquare(){

return side \* side;

}

D getPerimeter(){

return side \* 4;

}

Square<D>&operator++(int){

side++;

return \*this;

}

Square<D>&operator+=(Square<D>&C) {

this -> side += C.side;

return \*this;

}

template <class D1> friend istream&operator>>(istream&i, Square<D1>&C);

template <class D1> friend ostream&operator<<(ostream&o, Square<D1>&C);

};

template <class D>

istream&operator>>(istream&i, Square<D>&C) {

cout << "Введите длину стороны" << endl;

i >> C.side;

return i;

}

template <class D>

ostream&operator<<(ostream&o, Square<D>&C) {

o << "side = " << C.side << endl;

return o;

}

int main(int argc, const char \* argv[]) {

Square<float> \*A = new Square<float>(5), \*B = new Square<float>();

cin>>\*B;

\*A+=\*B;

cout << "A ";

cout<<\*A;

cout << "B ";

cout<<\*B;

cout << "Perimeter A " << A -> getPerimeter() << "\nPerimeter B " << B -> getPerimeter() << endl;

cout << "Diagonal A " << A -> getDiagonal() << "\nDiagonal B " << B -> getDiagonal() << endl;

cout << "Square A " << A -> getSquare() << "\nSquare B " << B -> getSquare() << endl;

delete A;

delete B;

return 0;

}

Протокол программы:

Введите длину стороны

2

A side = 7

B side = 2

Perimeter A 28

Perimeter B 8

Diagonal A 9.8995

Diagonal B 2.82843

Square A 49

Square B 4

Program ended with exit code: 0