МИНИСТЕРСТВО ТРАНСПОРТА И КОММУНИКАЦИЙ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«Белорусский государственный университет транспорта»

Кафедра «Информационно-управляющие системы и технологии»

Отчет  
по лабораторным работам

по дисциплине «Объектно-ориентированное программирование»

Выполнил Проверил

студент группы ГИ-21 ст. преп. каф. «ИУСиТ»

Зайцев И. С. Голдобина Т. А.

Гомель, 2020

Оглавление

[Лабораторная работа № 7 Тема: «Шаблоны классов» 3](#_Toc58581265)

[1.1 Задания 3](#_Toc58581266)

[1.2 Контрольные вопросы 7](#_Toc58581267)

Лабораторная работа № 7  
Тема: «Шаблоны классов»

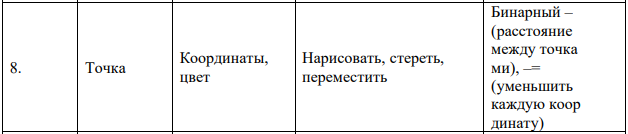
### **Цель**

Научиться использовать шаблоны функций для создания группы однотипных функций, а также шаблоны классов для создания группы связанных типов классов.

## Задания

### Задание 1.1

### Условие



Программный код

#include <iostream>

#include <cmath>

using namespace std;

template <typename T>

class Point

{private:

T coordinate, coordinate2;

string color;

public:

Point(T a=0, T b=0, string c = " ") : coordinate(a), coordinate2(b), color(c){}

void new\_coordinate(T x, T y)

{

this->coordinate = x;

this->coordinate2 = y;

}

void draw()

{

cout << coordinate << "\t" <<coordinate2 << "\t" << color<< endl;

}

void erase1()

{

this->coordinate = 0;

this->coordinate2 = 0;

this->color = " ";

}

void move1(Point &a)

{

cout << a.coordinate <<"\t" << a.coordinate2;

}

void vivod();

float operator - (Point &a)

{

float x = (sqrt((pow(this->coordinate - a.coordinate, 2) + pow(this->coordinate2 - a.coordinate2,2))));

return x;

}

Point operator -= (int x)

{

return Point(this->coordinate - x, this->coordinate2 - x);

}

Point operator += (int x)

{

return Point(this->coordinate + x, this->coordinate2 + x);

}

};

template <typename T>

T enter\_x()

{ T x;

cout << "Enter a coordinate(x): ";

cin >> x;

return x;

}

template <typename T>

T enter\_y()

{ T y;

cout << "Enter a coordinate(y): ";

cin >> y;

cout << "-----------------------------\n";

return y;

}

int main()

{ setlocale(LC\_ALL, "");

int x, y, z, key;

x = enter\_x<int>();

y = enter\_y<int>();

Point <int>first(x, y, "RED");

x = enter\_x<int>();

y = enter\_y<int>();

Point <int>second(x, y, "GREEN");

cout << "0. Exit\n1. Erase coordinates\n2. Move coordinates\n3. Distance between coordinates\n4. Draw:(\n5. Change coordinates\n";

do{

cout << "\nChoose a number: ";

cin >> key;

switch(key)

{

case 1: first.erase1(); second.erase1(); cout << "Coordinates erased!!!"<<endl; break;

case 2: {cout << "How much will we move the coordinates?: ";

cin >> z;

if(z>0)

{

first = (first +=z);

first.move1(first);

cout << endl;

second = (second +=z);

second.move1(second);

}

else

{

first = (first -=(-1\*z));

first.move1(first);

cout << endl;

second = (second -=(-1\*z));

second.move1(second);

}

break;

}

case 3: {float theed = second - first; cout << "Distance between coordinates:\n" << theed; } break;

case 4: first.draw(); second.draw(); cout <<"Drew:)"; break;

case 5: {x = enter\_x<int>(); y = enter\_y<int>(); first.new\_coordinate(x, y);

x = enter\_x<int>(); y = enter\_y<int>(); second.new\_coordinate(x, y);

}break;

}

}while(key!=0);

return 0;

}

### Результат

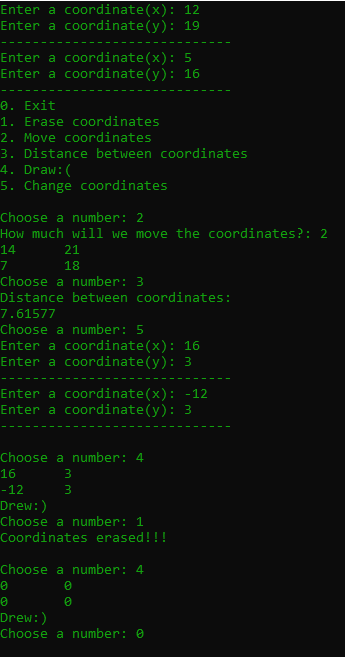


Рисунок 1 – Результат задания 1

## 1.2 Контрольные вопросы

1. Для чего используются шаблоны классов? Что у них общего с шаблонами функций?

Шаблон семейства классов определяет способ построения отдельных классов подобно тому, как класс определяет правила построения и формат отдельных объектов. Этот класс можно рассматривать как некоторое описание множества классов, отличающихся только типами их данных. В С++ используется ключевое слово template для обеспечения параметрического полиморфизма. Шаблоны определения класса и шаблоны определения функции позволяют многократно использовать код, корректно по отношению к различным типам.

1. Как описываются шаблоны классов?

Формат шаблона класса имеет вид:

template <список параметров>

class объявление класса

1. Как создать объект на основе класса, порожденного шаблоном?

Так же, как и объект обычного класса, дополнительно указав параметризацию, например:

ИмяКласса <Параметризация> имяОбъекта (параметры конструктора);

1. Каких типов могут быть фактические параметры шаблонов классов?

Любых.

1. Можно ли описывать в списке параметров шаблона параметры, не определяющие тип?

Можно. Для этого в угловых скобках у нужного параметра вместо слова class пишется нужный тип данных.

### Вывод

В результате лабораторной работы научились работать с шаблонами классов.