**Практическая работа №1**

**«Проектирование классов»**

**Выполнил**

Пантелеев Данила

Группа 328-1

Вариант 3

Скрипт программы:

#include <iostream>

using namespace std;

class Triangle {

private:

int a, b, angle;

public:

Triangle(int x, int y, int ang) {

a = x;

b = y;

angle = ang;

}

int getSideA() {

return a;

}

int getSideB() {

return b;

}

int getSideAngle() {

return angle;

}

int decrease(int proc) {

return angle -= proc;

}

int increase(int proc) {

return angle += proc;

}

int kindOfAngle() {

if (angle < 90) {

return 1;

}

else if (angle == 90) {

return 2;

}

else if (angle > 90 && angle < 180) {

return 3;

}

}

int kindOfTriang() {

if (a == b and angle == 60) {

return 1;

}

else if (a == b) {

return 2;

}

else if (a != b) {

return 3;

}

}

};

int main() {

setlocale(LC\_ALL, "RUS");

int sideA, sideB, sideAngle, proc;

s1:cout << "Введите первую сторону = ";

cin >> sideA;

if (!cin) {

cout << "Неверное значение" << endl;

cin.clear();

cin.ignore(1000, '\n');

goto s1;

}

s2:cout << "Введите вторую сторону = ";

cin >> sideB;

if (!cin) {

cout << "Неверное значение" << endl;

cin.clear();

cin.ignore(1000, '\n');

goto s2;

}

s3:cout << "Введите угол = ";

cin >> sideAngle;

cout << endl;

if (!cin) {

cout << "Неверное значение" << endl;

cin.clear();

cin.ignore(1000, '\n');

goto s3;

}

Triangle tr1(sideA, sideB, sideAngle);

cout << "Сторона 1 - " << tr1.getSideA() << endl;

cout << "Сторона 2 - " << tr1.getSideB() << endl;

cout << "Угол - " << tr1.getSideAngle() << endl << endl;

cout << "Вид треугольника:" << endl;

if (tr1.kindOfTriang() == 1) {

cout << "Равносторонний" << endl << endl;

}

else if (tr1.kindOfTriang() == 2) {

cout << "Равнобедренный" << endl << endl;

}

else if (tr1.kindOfTriang() == 3) {

cout << "Разносторонний" << endl << endl;

}

cout << "Значение угла:" << endl;

if (tr1.kindOfAngle() == 1) {

cout << "Острый" << endl << endl;

}

else if (tr1.kindOfAngle() == 2) {

cout << "Прямой" << endl << endl;

}

else if (tr1.kindOfAngle() == 3) {

cout << "Тупой" << endl << endl;

}

ch:cout << "Выберите операцию увеличения(1) либо уменьшения(2) угла" << endl;

int choice;

cin >> choice;

if (choice == 1) {

cout << "Введите проценты = ";

cin >> proc;

tr1.increase(proc);

cout << endl;

cout << "Сторона 1 - " << tr1.getSideA() << endl;

cout << "Сторона 2 - " << tr1.getSideB() << endl;

cout << "Угол - " << tr1.getSideAngle() << endl << endl;

cout << "Вид треугольника:" << endl;

if (tr1.kindOfTriang() == 1) {

cout << "Равносторонний" << endl << endl;

}

else if (tr1.kindOfTriang() == 2) {

cout << "Равнобедренный" << endl << endl;

}

else if (tr1.kindOfTriang() == 3) {

cout << "Разносторонний" << endl << endl;

}

cout << "Значение угла:" << endl;

if (tr1.kindOfAngle() == 1) {

cout << "Острый" << endl << endl;

}

else if (tr1.kindOfAngle() == 2) {

cout << "Прямой" << endl << endl;

}

else if (tr1.kindOfAngle() == 3) {

cout << "Тупой" << endl << endl;

}

}

else if (choice == 2) {

cout << "Введите проценты = ";

cin >> proc;

tr1.decrease(proc);

cout << endl;

cout << "Сторона 1 - " << tr1.getSideA() << endl;

cout << "Сторона 2 - " << tr1.getSideB() << endl;

cout << "Угол - " << tr1.getSideAngle() << endl << endl;

cout << "Вид треугольника:" << endl;

if (tr1.kindOfTriang() == 1) {

cout << "Равносторонний" << endl << endl;

}

else if (tr1.kindOfTriang() == 2) {

cout << "Равнобедренный" << endl << endl;

}

else if (tr1.kindOfTriang() == 3) {

cout << "Разносторонний" << endl << endl;

}

cout << "Значение угла:" << endl;

if (tr1.kindOfAngle() == 1) {

cout << "Острый" << endl << endl;

}

else if (tr1.kindOfAngle() == 2) {

cout << "Прямой" << endl << endl;

}

else if (tr1.kindOfAngle() == 3) {

cout << "Тупой" << endl << endl;

}

}

else if (choice != 1 || choice != 2) {

cout << "Чтобы выбрать операцию увеличение или уменьшения угла выберите цифры 1 или 2" << endl << endl;

cin.clear();

goto ch;

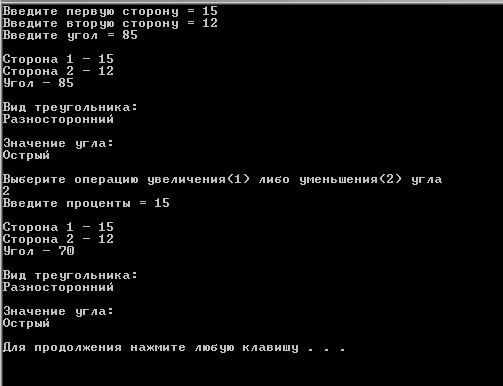
}

system("pause");

return 0;

}

Выполнение программы:



Контрольные вопросы:

1. Инкапсуляция, наследование, полиморфизм.
2. Класс – шаблон для создания объектов.
3. **class <имя\_класса>**

**{**

**<закрытые\_элементы\_класса>**

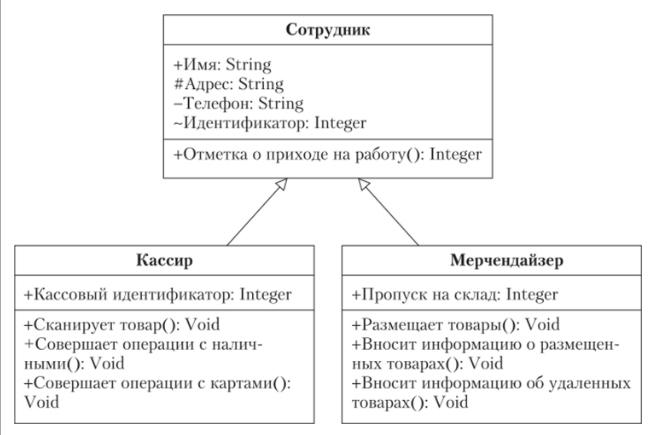
**protected:**

**<защищённые\_элементы\_класса>**

**public:**

**<открытые\_элементы\_класса> }**

**<список\_объектов> ;**

1. Диаграмма классов (англ. class diagram) — структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними.
2. 
3. Именованное свойство класса, описывающее диапазон значений, которые может принимать экземпляр атрибута.