МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Отчет к лабораторной работе № 3

Тема: **Объектно-ориентированный подход. Классы и объекты**

Дисциплина: **«Информационные технологии и программирование»**

**Выполнил:**

Студент 1 курса группы ПИЖ-б-о-23-1(2) направление подготовки: 09.03.04 «Программная инженерия»

Сабодаш Алексей Игоревич

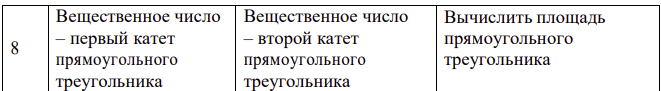
**Проверила:**

Новикова Е. Н.

Ставрополь 2024

**Вариант 8**

**Цель работы**: дать первые представления о программировании в объектно-ориентированном стиле. научится создавать классы объектов

**Задание 1:**  


#include <iostream>

#include <sstream>

#include <cmath>

class RightTriangle {

private:

double side1;

double side2;

public:

RightTriangle(double s1, double s2) : side1(s1), side2(s2) {}

std::string getInfo() {

std::stringstream ss;

ss << "Катет1: " << side1 << ", Катет2: " << side2;

return ss.str();

}

double computeArea() {

return 0.5 \* side1 \* side2;

}

};

int main() {

setlocale(LC\_ALL, "Russian");

double side1, side2;

std::cout << "Катет1: ";

std::cin >> side1;

std::cout << "Катет2: ";

std::cin >> side2;

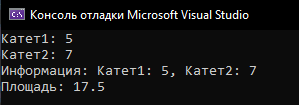
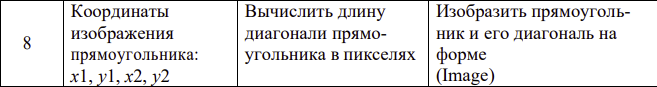
RightTriangle triangle(side1, side2);

std::cout << "Информация: " << triangle.getInfo() << std::endl;

std::cout << "Площадь: " << triangle.computeArea() << std::endl;

return 0;

}

**  
  
Задание 2:**  


#include <iostream>

#include <cmath>

#include <string>

class Rectangle {

private:

int x1, y1, x2, y2;

public:

// Конструктор по умолчанию

Rectangle() : x1(0), y1(0), x2(0), y2(0) {}

// Конструктор перезагрузки с параметрами

Rectangle(int x1, int y1, int x2, int y2) : x1(x1), y1(y1), x2(x2), y2(y2) {}

// Деструктор

~Rectangle() {

std::cout << "Объект уничтожен\n";

}

// Функции обработки данных

double computeDiagonalLength() {

return std::sqrt(std::pow(x2 - x1, 2) + std::pow(y2 - y1, 2));

}

std::string getInfo() {

return "Координаты: (" + std::to\_string(x1) + ", " + std::to\_string(y1) + "), (" +

std::to\_string(x2) + ", " + std::to\_string(y2) + ")";

}

// Функция для отображения прямоугольника и его диагонали на консоли

void drawRectangle() {

int width = std::abs(x2 - x1);

int height = std::abs(y2 - y1);

// Отображение прямоугольника

for (int i = 0; i < height; ++i) {

for (int j = 0; j < width; ++j) {

if (i == 0 || i == height - 1 || j == 0 || j == width - 1)

std::cout << "\*";

else

std::cout << " ";

}

std::cout << std::endl;

}

// Отображение диагонали

double diagonal = computeDiagonalLength();

int diagonalLength = static\_cast<int>(diagonal);

if (diagonalLength > 0) {

if (width > height) {

for (int i = 0; i < height; ++i) {

std::cout << std::string(i, ' ') << "\*\n";

}

}

else {

for (int i = 0; i < width; ++i) {

std::cout << std::string(i, ' ') << "\*\n";

}

}

}

}

};

int main() {

setlocale(LC\_ALL, "Russian");

// Создание объектов

Rectangle rect1(2, 2, 8, 5);

// Вывод информации о прямоугольнике

std::cout << "Информация: " << rect1.getInfo() << std::endl;

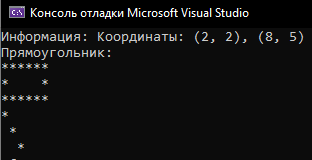
// Отображение прямоугольника и его диагонали на консоли

std::cout << "Прямоугольник:\n";

rect1.drawRectangle();

return 0;

}

  
 **https://github.com/strit595/Progrm**