МИНИСТЕРСТВО ТРАНСПОРТА РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА (МИИТ)»

Кафедра АСУ

**Отчет**

По лабораторной работе

по дисциплине   
«Языки программирования высокого уровня»

Тема: «НАСЛЕДОВАНИЕ И ВИРТУАЛЬНЫЕ МЕТОДЫ»

Задание № 5

Вариант № 4

Выполнил:   
 ст. гр. УВА-111:

Букин А. А.

Проверил:

Савушкин С.А.

Москва 2020 г.

**Содержание:**

[**Постановка задачи 3**](#_heading=h.30j0zll)

[**Общие требования 3**](#_heading=h.1fob9te)

[**Задание 3**](#_heading=h.3znysh7)

[**Авторское понимание и формализация задачи**](#_heading=h.2et92p0) **3**

[**Текст программы**](#_heading=h.tyjcwt) **4**

[**Результаты тестирования**](#_heading=h.3dy6vkm) **7**

[**Вывод**](#_heading=h.1t3h5sf) **8**

**Постановка задачи**

**Общие требования**

Во всех заданиях реализовать вывод на экран, методы получения значений полей и методы установки значений полей, а также необходимые конструкторы (если это не указано в задании явно). Конструкторы и методы обязательно должны проверять параметры на допустимость; в случае неправильных данных — выводить сообщение об ошибке и заканчивать работу. Преобразование в строку реализовать в виде функции преобразования toString().

Для демонстрации работы с объектами нового типа во всех заданиях требуется написать главную функцию. В программе должны присутствовать различные способы создания объектов и массивов объектов. Программа должна демонстрировать использование всех функций и методов.

Продемонстрировать принцип подстановки.

**Задание**

Создать класс Pair (пара чисел); определить методы изменения полей и вычисления произведения чисел. Определить производный класс Rectangle (прямоугольник) с полями-сторонами. Определить методы вычисления периметра и площади прямоугольника.

**Авторское понимание и формализация задачи**

Мы создаем класс Pair и класс Rectangle, для работы с ними. Вводим числа в пару и стороны в Rectangle. Так же реализуем вычисления периметра и площади прямоугольника. И получившийся результат выводим на экран.

**Текст программы**

**Файл – lab5.cpp**

**#include <string>**

**#include <iostream>**

**#include <sstream>**

**using namespace std;**

**class Pair {**

**protected:**

**double x; // число 1**

**double y; // число 2**

**public:**

**Pair(double x, double y) : x(x), y(y) {}; // конструктор инициализации**

**Pair(const Pair &p) : x(p.x), y(p.y) {}; // конструктор копирования**

**Pair() : x(0.0), y(0.0) {}; // конструктор без аргументов**

**string toString();**

**double mult() { return (x \* y); };**

**void setX();**

**void setY();**

**double getX();**

**double getY();**

**};**

**string Pair::toString() {**

**ostringstream s;**

**s << "Пара: { " << x << ", " << y << " }";**

**return s.str();**

**}**

**class Rectangle : Pair {**

**public:**

**Rectangle(double nx, double ny) { // конструктор инициализации**

**if (nx < 0.0 || ny < 0.0) {**

**cout << "Ошибочные значения, завершение программы..." << endl;**

**exit(0);**

**}**

**x = nx;**

**y = ny;**

**};**

**Rectangle(const Rectangle &p) { x = p.x; y = p.y; }; // конструктор копирования**

**Rectangle(); // конструктор без аргументов**

**string toString(); //подстановка функции (override)**

**double square() { return mult(); }; // использование функции из предка**

**double perimeter() { return ((x + y) \* 2); };**

**};**

**Rectangle::Rectangle() {**

**double nx;**

**double ny;**

**cout << "Введите первую сторону: ";**

**cin >> nx;**

**cout << "Введите вторую сторону: ";**

**cin >> ny;**

**if (nx <= 0.0 || ny <= 0.0) {**

**cout << "Ошибочные значения, завершение программы..." << endl;**

**exit(0);**

**}**

**x = nx;**

**y = ny;**

**}**

**string Rectangle::toString() {**

**ostringstream s;**

**s << "Прямоугольник: { " << x << ", " << y << " } c периметром "**

**<< perimeter() << " и площадью " << square();**

**return s.str();**

**}**

**int main()**

**{**

**cout << "┍━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┑" << endl;**

**cout << "| Демонстрация работы перегрузок |" << endl;**

**cout << "┕━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┙" << endl << endl;**

**Pair p1 = Pair();**

**cout << "По умолчанию : " << p1.toString() << endl;**

**Pair p2 = Pair(1, 3);**

**cout << "Произведение пары " << p2.toString() << " равно " << p2.mult() << endl;**

**Pair p3 = Pair(p2);**

**cout << endl << "Копирование : " << p2.toString() << " -> " << p3.toString() << endl;**

**Rectangle r = Rectangle(); // введение данных с терминала**

**cout << endl << r.toString() << endl << endl; // использование принципа подстановки**

**cout << "┍━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┑" << endl;**

**cout << "| Демонстрация работы с классами |" << endl;**

**cout << "┕━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┙" << endl << endl;**

**Pair \*pair\_heap = new Pair(2, 5);**

**cout << "pair\_heap: " << pair\_heap->toString() << endl << endl;**

**int rect\_num = 5;**

**Rectangle \*array[rect\_num];**

**for (int i = 1; i <= rect\_num; i++) {**

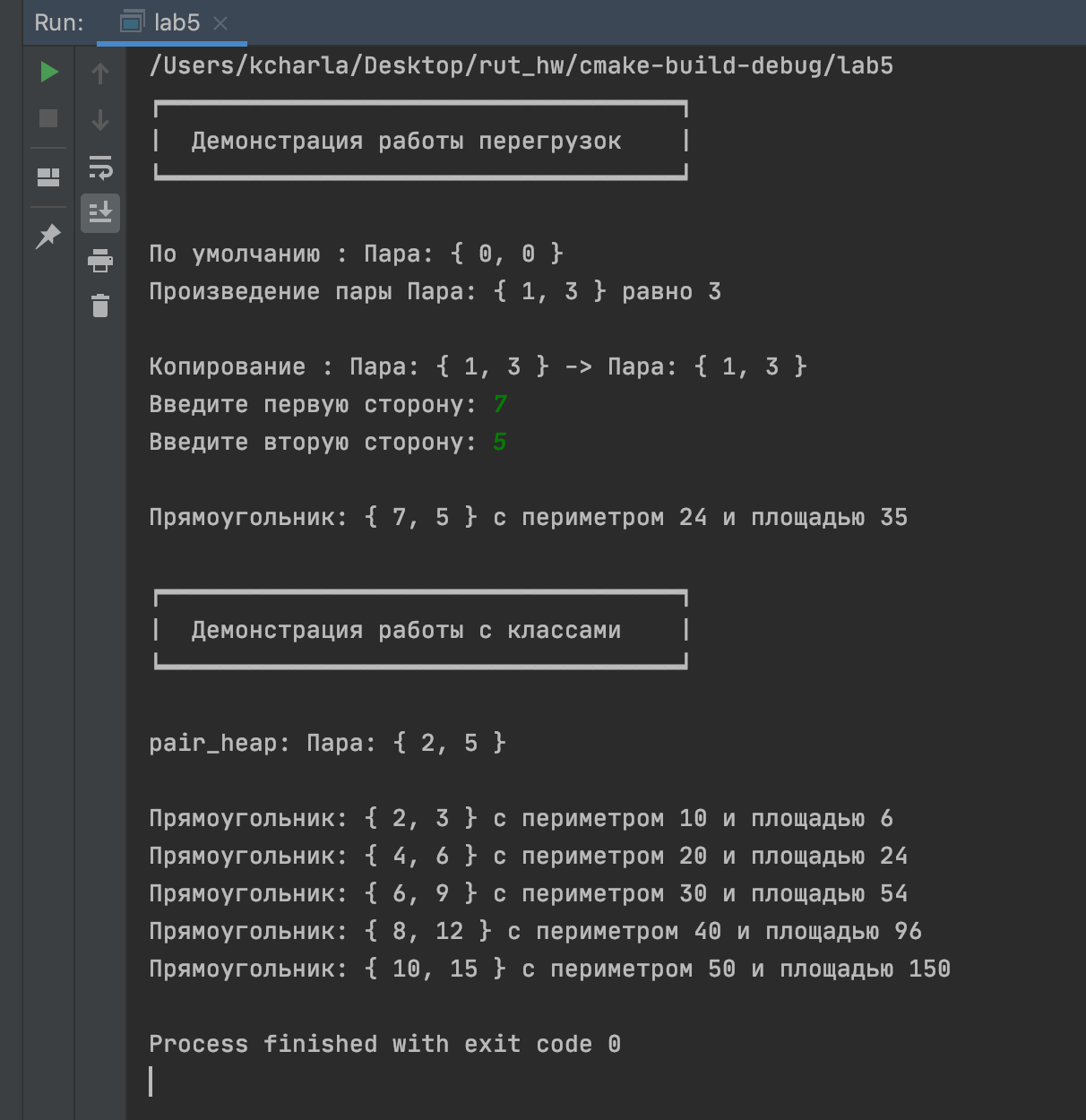
**array[i] = new Rectangle(i \* 2, i \* 3);**

**cout << array[i]->toString() << endl;**

**}**

**}**

**Результаты тестирования**

****

**Вывод**

Программа выдает ожидаемые результаты и работает корректно. Поскольку полученные результаты совпали с ожидаемыми, программу можно считать правильной. Я научился наследовать классы объектов.