Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Кафедра **«**Информационные технологии и автоматизированные системы**»**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №5**

Дисциплина: «Основы алгоритмизации и программирования»

Семестр 2

Тема: Наследование. Виртуальные функции. Полиморфизм.

Вариант 6.

Выполнил работу

Студент группы РИС-22-1Б

Поважный В. Е.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

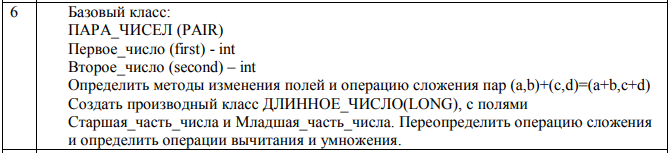
г. Пермь-2023

**Введение**

Для выполнения лабораторной работы требуется сделать отчет программы. Создание программ – отличный способ практики программирования.

**Постановка задачи**

1. Определить пользовательский класс.
2. Определить иерархию классов, в основе которой будет находиться абстрактный класс (см. лабораторную работу №4).
3. Определить класс Вектор, элементами которого будут указатели на объекты иерархии классов.
4. Перегрузить для класса Вектор операцию вывода объектов с помощью потоков.
5. В основной функции продемонстрировать перегруженные операции и полиморфизм Вектора.



**Код программы**

**ооп 5.cpp**

#include <iostream>

#include <cmath>

#include <ctime>

#include "pair.h"

#include "rightAngled.h"

using namespace std;

int main()

{

srand(time(NULL));

Pair pair;

Pair pair1;

std::cout << pair << std::endl;

std::cout << pair1 << std::endl;

pair + pair1;

cout << pair << endl;

LongPair rightangled(3, 4);

rightangled.calculate();

std::cout << rightangled.get() << std::endl;

return 0;

}

**rightAngled.h**

**#include** <cmath>

#include "pair.h"

class LongPair : public Pair

{

public:

LongPair(int f, int s) : Pair(f, s) {}

void calculate()

{

hyp = first + second;

}

int get() const

{

return hyp;

}

private:

int hyp;

};

**pair.h**

**#pragma** once

#include <iostream>

class Pair

{

protected:

int first;

int second;

public:

Pair()

{

first = rand() % 10;

second = rand() % 10;

}

Pair(int f, int s) : first(f), second(s) {}

void set\_first(int f)

{

first = f;

}

void set\_second(int s)

{

second = s;

}

int get\_first() const

{

return first;

}

int get\_second() const

{

return second;

}

friend std::ostream& operator << (std::ostream& s, const Pair& pair)

{

return s << "(" << pair.first << ", " << pair.second << ")";

}

Pair operator + (Pair& pair1)

{

Pair newPair(this->first += pair1.first, this->second += pair1.second);

return newPair;

}

virtual int product() const

{

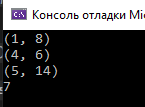
return first \* second;

}

virtual ~Pair() {}

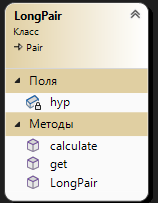
};

**Вывод программы**

****

***Рисунок 1 – вывод программы***

**UML диаграмма**

****

***Рисунок 2 – UML диаграмма***

**Вывод**

Программа выполняет свою задачу.

**Заключение**

Для решения задачи потребовались знания языка программирования, а конкретнее C++. Программа выполняет те условия, что были указаны в постановке задачи и работает без проблем.