Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №18.3**

Дисциплина: «Информатика»

Тема: Структуры данных

Вариант 7

Выполнила:

Студент группы ИВТ-20-2б

Солдатов Алексей Павлович

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цель задачи**

Создание консольного приложения, состоящего из нескольких файлов в системе программирования Visual Studio.

Использование перегруженных операций в классах

Для достижения поставленной цели необходимо решить следующие задачи:

* Проанализировать теоретические сведения о ООП
* Проанализировать понятия классов, инструменты для их создания
* Разобрать поля и методы классов
* Понять, как пользоваться перегруженными операциями в классах
* Реализовать алгоритм поставленной задачи
* Создать отдельные файлы: .cpp для описания методов класса, .cpp для описания главной функции, .h для описания класса

Постановка задачи

Базовый класс:

ПАРА\_ЧИСЕЛ (PAIR)

Первое\_число (first) - int

Второе\_число (second) – int

Определить методы изменения полей и вычисления произведения чисел. Создать производный класс ПРЯМОУГОЛЬНЫЙ\_ТРЕУГОЛЬНИК (RIGHTANGLED), с полями-катетами. Определить метод вычисления гипотенузы.

Анализ задачи

1. Определить какие операции должны быть выполнены по заданию:

* Создание класса

class Money

* Создание Конструктора

Money::Money(long int rubs, int kops)

* Создание метода вывода информации класса
* Ввод данных

1. Для решения задачи используются переменные:

Несколько переменных целочисленного значения для ввода чисел

int rub = -1;

int kop = -1;

Поле типа double для хранения числа

double sum;

Переменные целочисленного типа для работы конструктора

long int rubs, int kops

1. Ввод данных осуществляется посредством функции

cin >> rub;

Вывод данных осуществляется посредством функции

cout << "Введи количество рублей: ";

1. Создание метода Print, который печатает данные конкретного контейнера

void Money::Print(double s)

{

cout << "Сумма = " << s << endl;

}

1. Создание контейнера

Money::Money(long int rubs, int kops)

{

double sum1 = kops;

sum1 = sum1 / 100;

sum1 = sum1 + rubs;

sum = sum1;

}

Код

Main.cpp:

#include "Rightangled.h"

#include "pair.h"

#include <iostream>

using namespace std;

int CorrectFirstNum()

{

int d;

cout << "Введите первое число: ";

cin >> d;

while (d < 1)

{

cout << "Введите число больше 0: ";

cin >> d;

}

return d;

}

int CorrectSecondNum()

{

int d;

cout << "Введите второе число: ";

cin >> d;

while (d < 1)

{

cout << "Введите число больше 0: ";

cin >> d;

}

return d;

}

int main()

{

setlocale(LC\_ALL, "rus");

Pair A(1, 1);

A.SetFirst(777);

A.SetSecond(666);

A.Show();

//entering Rightangled

int first, second;

first = CorrectFirstNum();

second = CorrectSecondNum();

//print class Rightangled

Rightangled D(32, 13);

Rightangled C(first, second);

C.PrintInf();

cout << "=============================" << endl;

C.Hyp(first, second);

}

pair.cpp:

#include "rightangled.h"

#include "Pair.h"

#include <iostream>

using namespace std;

Pair::Pair() { first = 0; second = 0; }

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

this->first = f;

this->second = s;

}

Pair::Pair(const Pair& t)

{

first = t.first;

second = t.second;

}

void Pair::SetFirst(int f)

{

this->first = f;

}

void Pair::SetSecond(int s)

{

this->second = s;

}

int Pair::GetFirst()

{

return first;

}

int Pair::GetSecond()

{

return second;

}

//methods

void Pair::Show()

{

cout << "First: " << first << endl;

cout << "Second: " << second << endl;

}

pair.h:

#pragma once

#include <iostream>

using namespace std;

class Pair

{

int first;

int second;

public:

//constructors

Pair();

Pair(int d, int m);

Pair(const Pair& t);

//destructor

~Pair() {};

//Getter and Setters

void SetFirst(int f);

void SetSecond(int s);

int GetFirst();

int GetSecond();

void Show();

void IncreaseFirst();

void IncreaseSecond();

Pair& operator=(Pair& t);

friend istream& operator >> (istream& cin, Pair& t);

friend ostream& operator << (ostream& cout, Pair& t);

};

rightangled.h:

#pragma once

#include "Pair.h"

#include <iostream>

#include <math.h>

using namespace std;

class Rightangled: public Pair

{

int FirstKat, SecondKat;

public:

//constructors

Rightangled() : Pair(), FirstKat(0), SecondKat(0) {};

Rightangled(int d, int m) : Pair(d, m) {}//,FirstKat(d), SecondKat(m) {};

Rightangled(Pair t, int d, int m) : Pair(t), FirstKat(d), SecondKat(m) {}

void PrintInf()

{

cout << "\nПервый катет равен:\t" << FirstKat << endl;

cout << "Второй катет равен:\t" << SecondKat << endl;

}

void Hyp(int FirstKat, int SecondKat)

{

cout << "Гипотенуза равна: " << sqrt(pow(FirstKat, 2) + pow(SecondKat, 2)) << endl << endl;

}

~Rightangled() {};

void SetFirstKat(int d)

{

if (d < 1)

{

cout << "error! FirstKat out of range!" << endl;

}

else this->FirstKat = FirstKat;

};

void SetSecondKat(int m)

{

if (m < 1)

{

cout << "error! SecondKat out of range!" << endl;

}

this->SecondKat = m;

};

int GetFirstKat() { return FirstKat; }

int GetSecondKat() { return SecondKat; }

Rightangled& operator=(Pair& t)

{

this->SetFirst(t.GetFirst());

this->SetSecond(t.GetSecond());

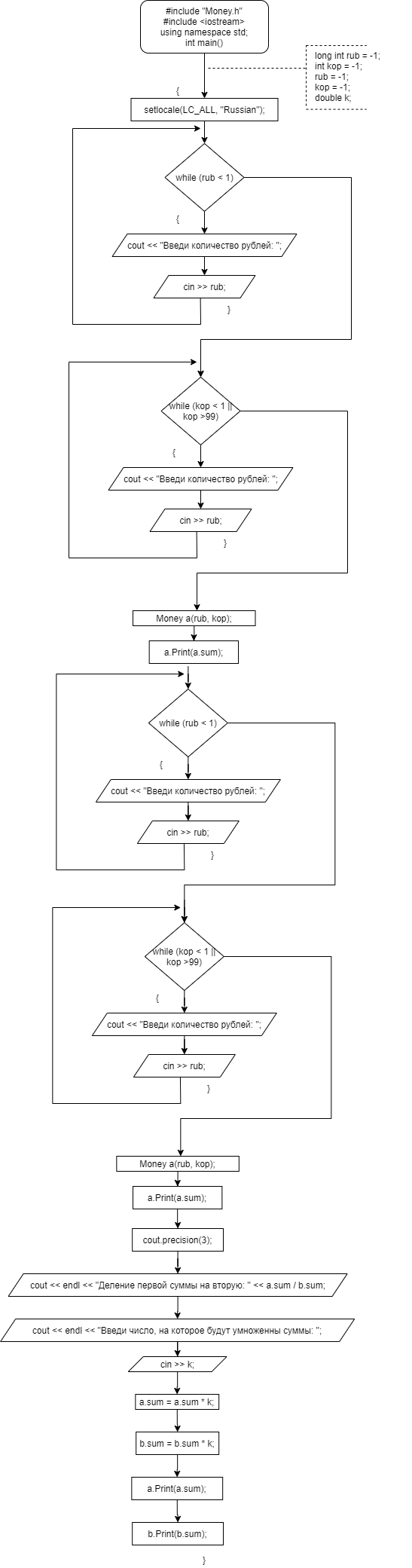
return \*this;

}

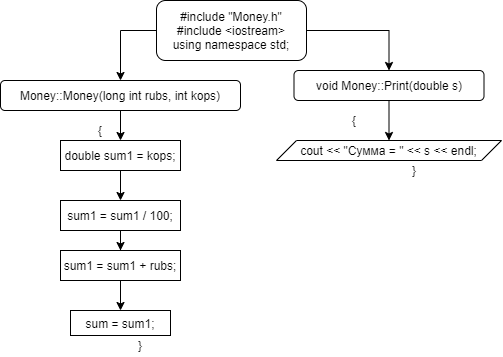
};

Блок-схема

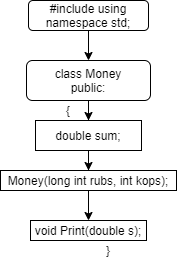
Main.cpp:



pair.cpp:



pair.h:



Работа кода

