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**

Дисциплина: «Информатика»

"Классы и объекты. Наследование. Виртуальные функции. Полиморфизм."

Семестр 2

Выполнил работу

Студент группы РИС-22-1Б

Бадртдинов Т.З

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Г. Пермь-2023

# Постановка задачи

1. Определить абстрактный класс.

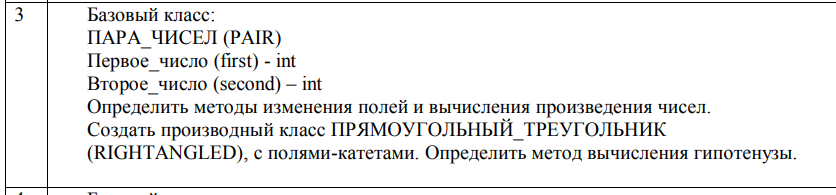
2. Определить иерархию классов, в основе которой будет находиться абстрактный класс (см. лабораторную работу №4).

3. Определить класс Вектор, элементами которого будут указатели на объекты иерархии классов.

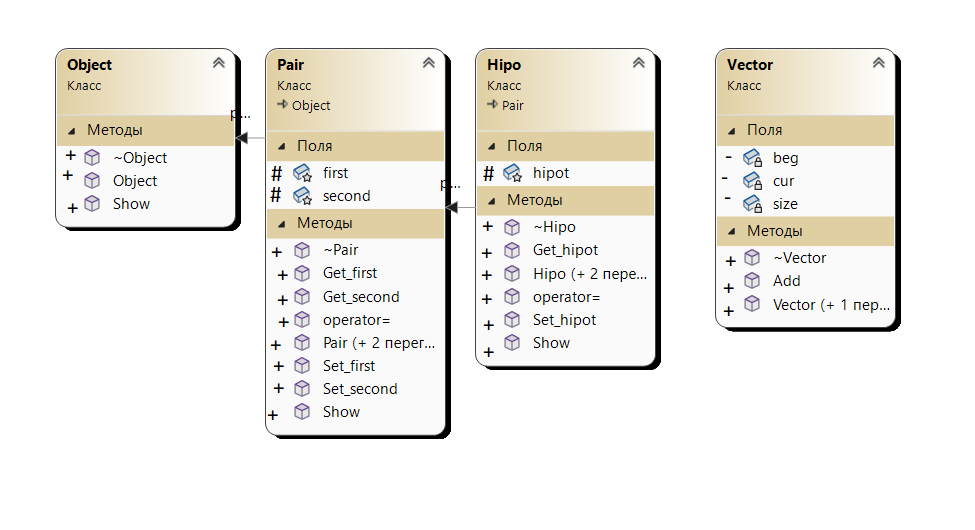
4. Перегрузить для класса Вектор операцию вывода объектов с помощью потоков.

5. В основной функции продемонстрировать перегруженные операции и полиморфизм Векторa.

Вариант 3



Описание класса

**

# Код программы(object.h)

#pragma once

class Object

{

public: Object(void){}

public: ~Object(void){}

virtual void Show()=0;//чисто виртуальная функция

};

# Код программы(vector.cpp)

#include "Vector.h"

//конструктор без параметров

Vector::Vector(void)

{

beg=0;

size=0;

cur=0;

}

//деструктор

Vector::~Vector(void)

{

if(beg!=0)delete [] beg;

beg=0;

} //конструктор с параметрами

Vector::Vector(int n) {

beg=new Object\*[n];

cur=0;

size=n;

}

//добавление объекта, на который указывает указатель p в вектор

void Vector::Add(Object \*p)

{

if(cur<size)

{

beg[cur]=p; cur++;

}

}

//операция вывода

ostream& operator<<(ostream&out,const Vector&v)

{

if(v.size==0) out<<"Empty"<<endl;

Object \*\*p=v.beg;//указатель на указатель типа Object

for(int i=0;i<v.cur;i++)

{

(\*p)->Show();//вызов метода Show() (позднее связывание)

p++;//передвигаем указатель на следующий объект

}

return out;

}

# Код программы(hipotenuse.cpp)

#include "Hipotenuse.h"

#include <math.h>

//конструктор без параметров

Hipo::Hipo(void):Pair()

{

hipot=0;

}

//дестрктор

Hipo::~Hipo(void) { }

//конструктор с параметрами

Hipo::Hipo(int C, int P, int G):Pair(C,P)

{

hipot= sqrt(C \* C+ P \* P);

}

//конструктор копирования

Hipo::Hipo(const Hipo &L)

{

first=L.first;

second=L.second;

}

//модификатор

void Hipo::Set\_hipot(int G)

{

hipot=G;

}

//оперция присваивания

Hipo& Hipo::operator=(const Hipo& l)

{

if (&l == this)return \*this;

second = l.second;

first = l.first;

hipot = l.hipot;

return \*this;

}

//операция ввода

istream& operator>>(istream&in,Hipo&l)

{

cout<<"\nfirst:";in>>l.first;

cout<<"\nsecond:";in>>l.second;

l.hipot = sqrt(l.first \* l.first + l.second \* l.second);

return in;

}

//операция вывода

ostream& operator<<(ostream& out, const Hipo& l)

{

out << "\nfirst : " << l.first;

out << "\nsecond : " << l.second;

out << "\nhipot : " << l.hipot;

out << "\n";

return out;

}

# Код программы(pair.cpp)

#include "Pair.h"

//конструктор без параметров

Pair::Pair(void)

{

; first=0; second=0;

} //деструктор

Pair::~Pair(void) { }

//конструктор с параметрами

Pair::Pair(int C,int P)

{

first=C; second=P;

} //конструктор копирования

Pair::Pair(const Pair& t)

{

first = t.first;

second = t.second;

}

//модификаторы

void Pair::Set\_first(int C)

{

first=C;

}

void Pair::Set\_second(int P)

{

second=P;

}

//перегрузка операции присваивания

Pair& Pair::operator=(const Pair&c)

{

if(&c==this)return \*this;

second=c.second;

first=c.first;

return \*this;

}

//глобальная функция для ввода

istream& operator>>(istream&in,Pair&c)

{

cout<<"\nfirst:";in>>c.first;

cout<<"\nsecond:";in>>c.second;

return in;

}

//глобальная функция для вывода

ostream& operator<<(ostream&out,const Pair&c)

{

out<<"\nfirst: "<<c.first;

out<<"\nsecond : "<<c.second;

out<<"\n";

return out;

}

# Код программы(main)

#include "Object.h"

#include "Pair.h"

#include "Hipotenuse.h"

#include "Vector.h"

#include <string>

#include <iostream>

using namespace std;

void main() {

Vector v(5);//вектор из 5 элементов

Pair a;//объект класса Pair

cin>>a;

Hipo b;// объект класса Hipo

cin>>b;

Object\*p=&a;//ставим указатель на объект класса Pair

v.Add(p);//добавляем объект в вектор

p=&b;//ставим указатель на объект класса

v.Add(p); //добавляем объект в вектор

cout<<v;//вывод вектора

}

# Работы программы

