Министерство высшего образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (ПНИПУ)

Электротехнический факультет

Дисциплина: основы алгоритмизации и программирования, 2 семестр

ОТЧЁТ

«Лабораторная работа ООП № 5. Наследование.

Виртуальные функции. Полиморфизм»

Выполнил

Студент группы РИС-22-2Б

Хохряков А.С.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Пермь 2023

# Содержание отчета

1. Постановка задачи.
2. Анализ задачи.
3. UML-Диаграмма
4. Ответы на вопросы
5. Текст программы

Постановка задачи

1. Определить абстрактный класс.

2. Определить иерархию классов, в основе которой будет находиться абстрактный класс.

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.

Вариант 3:

Базовый класс:

ПАРА ЧИСЕЛ (PAIR)

Первое число (first) - int

Второе число (second) - int

Определить методы изменения полей и вычисления произведения чисел.

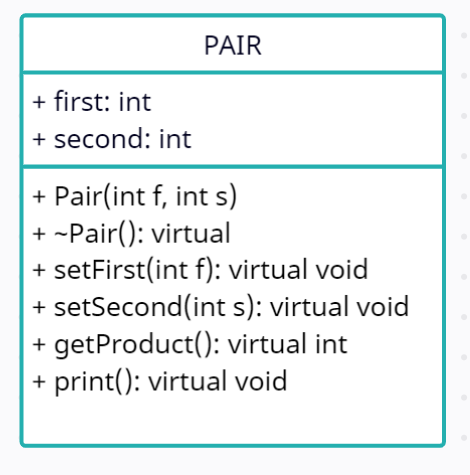
Создать производный класс ПРЯМОУГОЛЬНЫЙ ТРЕУГОЛЬНИК

((RIGHT ANGLED), с полями-катетами. Определить метод вычисления гипотенузы.

Анализ задачи

1. Определяется базовый класс Pair с защищенными переменными first и second, конструктором, деструктором, абстрактными методами setFirst, setSecond, getProduct и методом print, который выводит на экран значения first, second и getProduct.
2. Определяется класс RightAngledTriangle, наследующийся от Pair. Он имеет конструктор, деструктор, метод getHypotenuse, который вычисляет и возвращает длину гипотенузы, и переопределяет метод print, добавляя вывод значения гипотенузы.
3. Определяется класс Vector с приватным вектором elements. Он имеет метод add, который добавляет вектор элементов типа shared\_ptr<Pair>, и метод print, который выводит на экран значения элементов вектора, вызывая их методы print.
4. В функции main() создается объект класса Vector v.
5. Получаются значения f и s.
6. Создается shared\_ptr на объект класса RightAngledTriangle с передачей значений f и s в качестве параметров конструктора.
7. Вызывается метод add объекта v, передавая ему созданный shared\_ptr.
8. Вызывается метод print объекта v, выводящий значения элементов вектора на экран.

UML – Диаграмма



Ответы на вопросы

1. Чисто виртуальный метод - это метод, который объявлен в базовом классе без определения и используется для реализации полиморфизма. Он отличается от виртуального метода тем, что чисто виртуальный метод не имеет реализации по умолчанию и должен быть переопределен в производных классах, в то время как виртуальный метод может иметь реализацию по умолчанию.
2. Абстрактный класс - это класс, который содержит чисто виртуальные методы, которые должны быть реализованы в производных классах. Объекты абстрактного класса не могут быть созданы напрямую.
3. Абстрактные классы предназначены для создания иерархий классов, где базовый класс определяет общие свойства и методы, а производные классы могут добавлять специфические свойства и методы.
4. Полиморфные функции - это функции, которые могут принимать объекты разных классов и вызывать соответствующий метод для каждого класса.
5. Полиморфизм - это принцип объектно-ориентированного программирования, который позволяет объектам одного класса использовать методы другого класса, который он наследует или реализует. Принцип подстановки - это принцип, который гарантирует, что объекты производных классов могут использоваться везде, где используются объекты базового класса.
6. Примеры иерархий с использованием абстрактных классов:

* Животное (абстрактный класс) -> Млекопитающее -> Кошка, Собака
* Транспортное средство (абстрактный класс) -> Автомобиль -> Легковой автомобиль, Грузовик
* Фигура (абстрактный класс) -> Круг, Квадрат, Прямоугольник

1. Примеры полиморфных функций:

* Функция, которая принимает объекты разных классов и вызывает соответствующий метод для каждого класса.
* Функция, которая принимает указатель на базовый класс и вызывает виртуальный метод.

1. Механизм позднего связывания используется в тех случаях, когда необходимо вызвать метод производного класса через указатель на базовый класс, который может указывать на объект производного класса. Позднее связывание обеспечивает гибкость и возможность использования полиморфизма.

Текст программы

#include <iostream>

#include <vector>

#include <memory>

#include <cmath>

using namespace std;

class Pair {

protected:

int first;

int second;

public:

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

virtual ~Pair() {}

virtual void setFirst(int f) { first = f; }

virtual void setSecond(int s) { second = s; }

virtual int getProduct() { return first \* second; }

virtual void print() { cout << "(" << first << ", " << second << ", " << getProduct() << ")"; }

};

class RightAngledTriangle : public Pair {

public:

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

~RightAngledTriangle() {}

double getHypotenuse() { return sqrt(first \* first + second \* second); }

void print() { Pair::print(); cout << ", hypotenuse: " << getHypotenuse(); }

};

class Vector {

private:

vector<shared\_ptr<Pair>> elements;

public:

void add(shared\_ptr<Pair> p) { elements.push\_back(p); }

void print() {

for (auto& p : elements) {

p->print();

cout << " ";

}

cout << endl;

}

};

int main() {

Vector v;

int f, s;

cout << "Enter the first and second cathetus of a right-angled triangle:" << endl;

cin >> f >> s;

shared\_ptr<RightAngledTriangle> p1 = make\_shared<RightAngledTriangle>(f, s);

v.add(p1);

v.print();

return 0;

}