Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Пермский национальный исследовательский политехнический университет» (ПНИПУ)

Электротехнический факультет

Кафедра информационных технологий и автоматизированных систем

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

**Лабораторная работа по классам № 5**

**по дисциплине**

**«Основы алгоритмизации и программирования»**

**семестр** 2

Выполнил студент гр. ИВТ-21-1б

Балдин Алексей Игоревич

Проверил:

Яруллин Денис Владимирович

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

(оценка) (подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

г. Пермь - 2022

**Цель**

Создание иерархии классов с использованием простого наследования и абстрактного класса.

Изучение полиморфизма и виртуальных методов.

**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 Вектора

**Исходные данные**

Вариант 5:

Базовый класс:

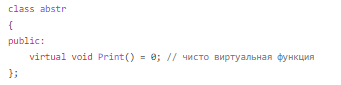
ПАРА\_ЧИСЕЛ (PAIR)

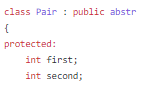
Первое число (first) - int Второе число (second) – int

Определить методы изменения полей и операцию сложения пар (a, b)+(c,d)=(a+b,c+d) Создать производный класс ДЕНЕЖНАЯ\_СУММА(MONEY), с полями Рубли и Копейки. Переопределить операцию сложения и определить операции вычитания и деления денежных сумм.

Сначала создаём тип данных «class» под именем Pair. Поля классов first и second c типом данных (int), находятся в private. В модификаторе доступа – public, создаем конструкторы по умолчанию, c параметрами, деструктор, также перегружаю оператор (+), который как раз-таки складывает поля first и second, еще методы (геттеры и сеттеры) – т.е. установка значений в private поля. Тут же в public прописываем дружественность операторов (>>) и (<<).

Создаю абстрактный класс:

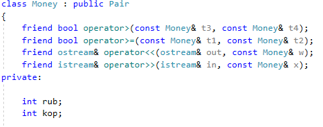
  
Здесь, создаю класс Pair – в нем наши поля под модификатором доступа – protected, это значит эти поля могут наследоваться в другие классы, также Pair наследует абстрактный класс:



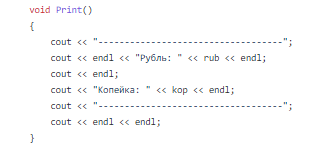
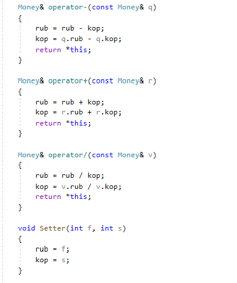
Далее в коде создаются конструкторы, деструкторы. Тут создаю дружественные прототипы перегруженных операторов ввода и вывода:



Далее, создаю еще один класс, который будет наследовать класс Pair, а это значит можно будет пользоваться полями класса Pair в классе Money. Также создаю прототипы дружественных операторов ввода и вывода, и прототипы перегруженных операторов «больше» и «больше равно». Также присутствуют поля класса в private секции: rub, kop.



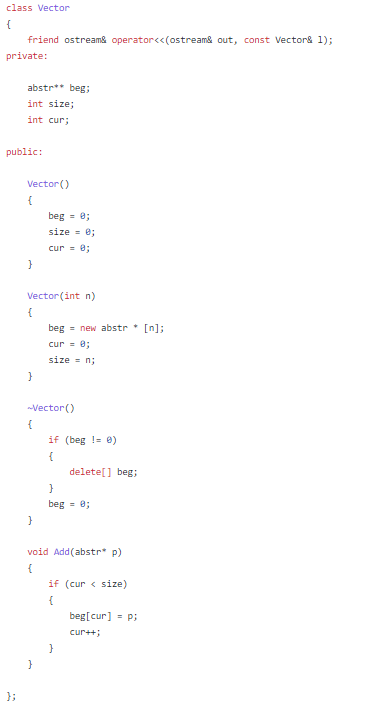
Перегружаю оператор (-), оператор (+), деления, и также присутствует метод (Setter) – для установки значения полей private. Есть метод Print:



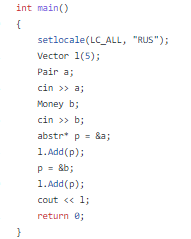
Перегрузка операторов (>=), (>), <<, >> - под условие задачи(В глобальном поле кода):



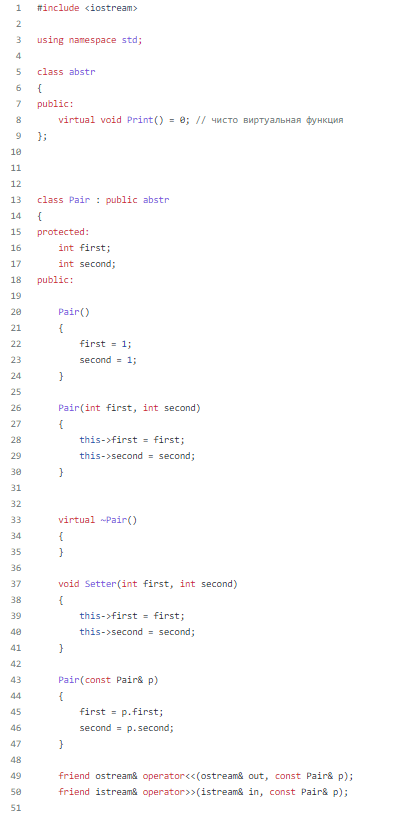
Создаю еще класс: Vector, у него определены поля private, дружественный оператор вывода. Также есть метод Add.

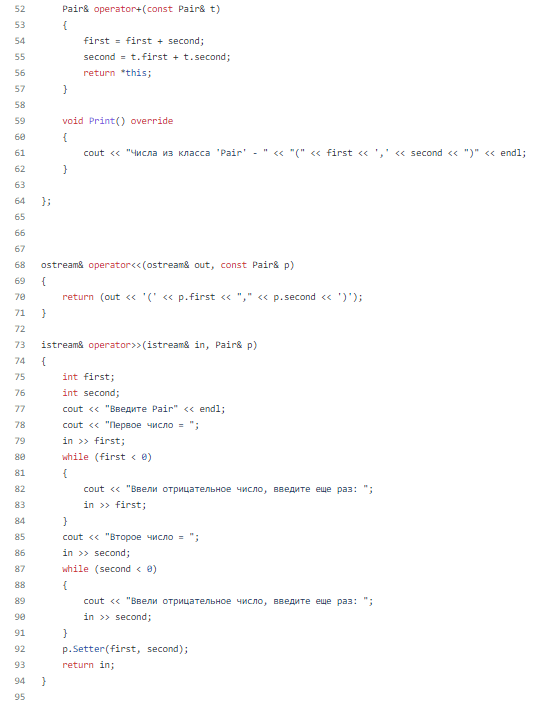


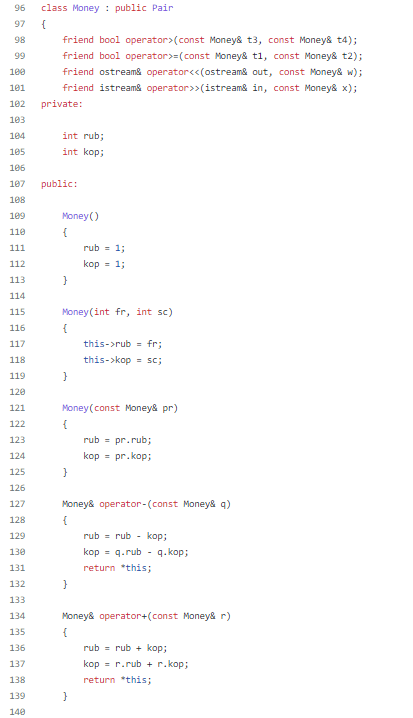
В main() создаю поля типа Pair, Vector, Money, вызываю функцию Add через объект класса и вывожу по условию.



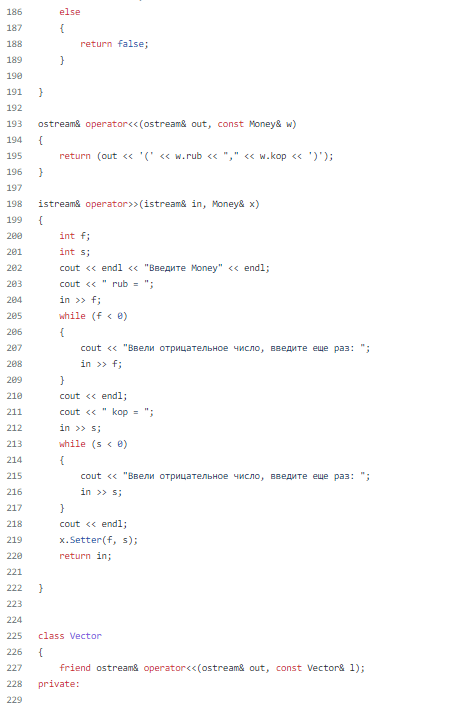
Код программы:

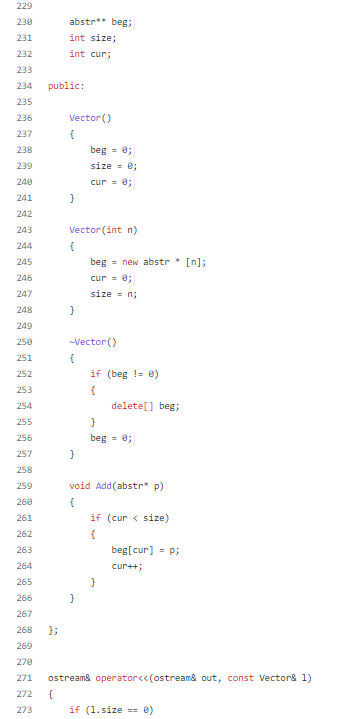


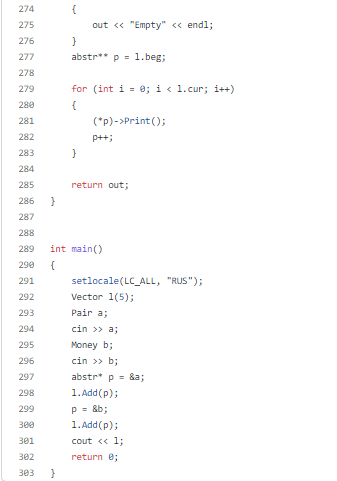












Результат программы:

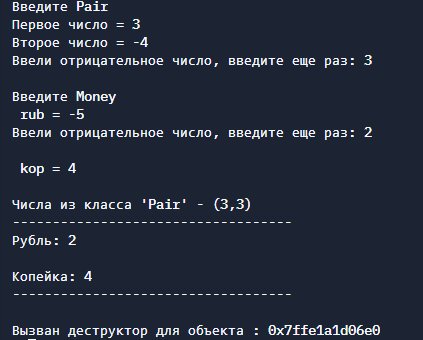


Диаграмма к задаче:

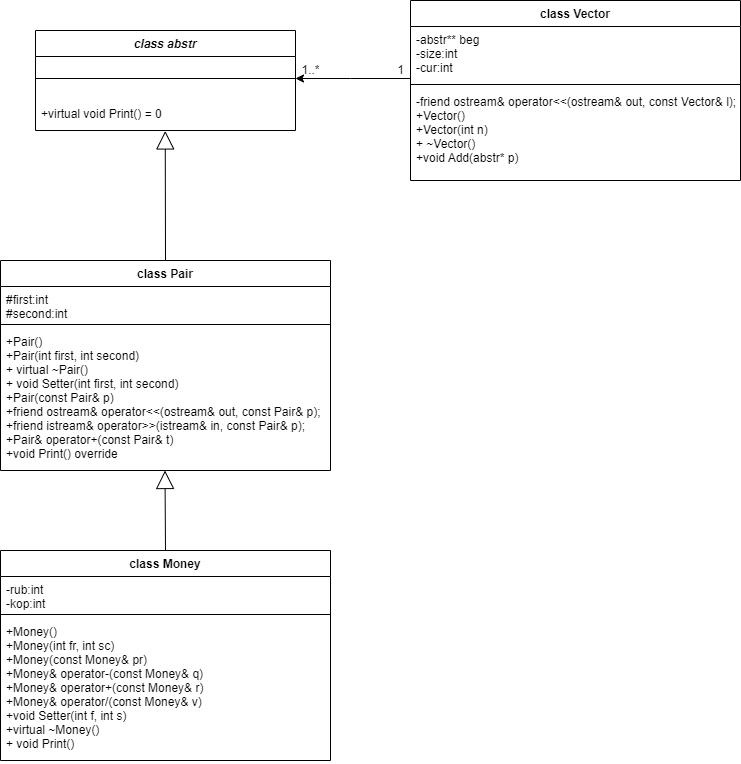


Рисунок 1

**Контрольные вопросы:**

* 1. Какой метод называется чисто виртуальным? Чем он отличается от виртуального метода?
  2. Какой класс называется абстрактным?
  3. Для чего предназначены абстрактные классы?
  4. Что такое полиморфные функции?
  5. Чем полиморфизм отличается от принципа подстановки?
  6. Привести примеры иерархий с использованием абстрактных классов.
  7. Привести примеры полиморфных функций.
  8. В каких случаях используется механизм позднего связывания?

**Ответы на вопросы:**

1. «Виртуальный» означает, что метод может быть переопределен в подклассах, но имеет непосредственную реализацию в базовом классе. «Чистый виртуальный» означает, что это виртуальный метод без прямой реализации. Такой метод должен быть переопределен хотя бы один раз в иерархии наследования - если у класса есть какие-либо нереализованные виртуальные методы, объекты этого класса не могут быть созданы, и компиляция завершится неудачно.
2. Абстрактным называется класс, имеющий чисто виртуальные функции. Чисто виртуальная функция – это функция, которая не определена в базовом классе, и обязательно должна быть перегружена в классах наследниках.
3. Абстрактные классы — это такие классы, которые не могут создавать объекты и нужны лишь для того, чтобы дополнить работу основных классов и избавить нас от повторяющегося кода.
4. Полиморфной функцией является такая функция, которая может вызываться с аргументами различного типа, а фактический выполняемый код зависит от типа аргументов.
5. Одинаковая спецификация – это реализация принципа подстановки, т.е. принцип подстановки характеризует спецификацию классов, а полиморфизм – их поведение.
6. Class Tara{virtual void print()=0;} <= class Meshok:Tara <=class Box:Meshok <= class Bochka:Box, class Gruz
7. Данный вопрос и предыдущий – продемонстрированы в решении лаб.р.
8. Позднее связывание является одним из самых мощных добавлений языка С++ к возможностям языка С. Платой за такое увеличение мощи программы служит некоторое уменьшение ее скорости исполнения. Поэтому использование позднего связывания оправдано только тогда, когда оно улучшает структурированность и управляемость программы