Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Электротехнический факультет  
Кафедра «Информационные технологии и автоматизированные системы» направление подготовки: 09.03.04 – «Программная инженерия»

# Лабораторная работа " Перегрузка операций."

Выполнил студент гр. РИС-24-3б

Жиряков Леонид Антонович

Проверил:

Доцент кафедры ИТАС   
Ольга Андреевна Полякова

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (оценка) (подпись)

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

(дата)

г. Пермь, 2024

Вариант 8.

Постановка задачи:

Общая:

1. Определить пользовательский класс.

2. Определить в классе следующие конструкторы: без параметров, с параметрами, копирования.

3. Определить в классе деструктор.

4. Определить в классе компоненты-функции для просмотра и установки полей данных (селекторы и модификаторы).

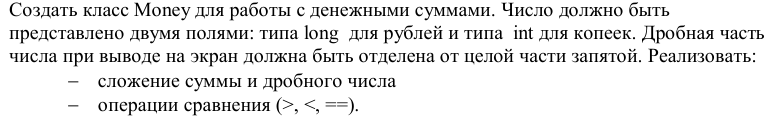
5. Перегрузить операцию присваивания.

6. Перегрузить операции ввода и вывода объектов с помощью потоков.

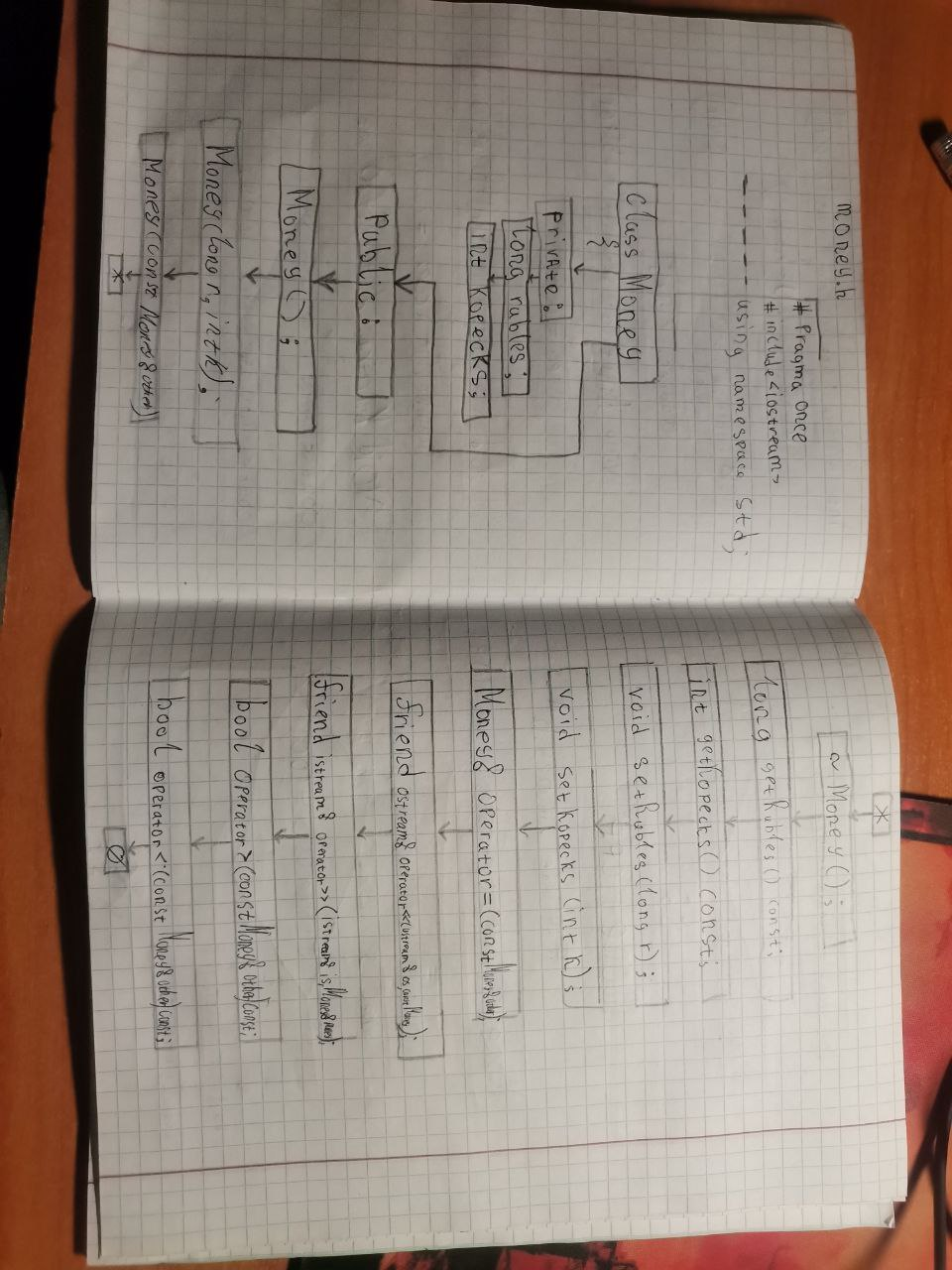
7. Перегрузить операции, указанные в варианте.

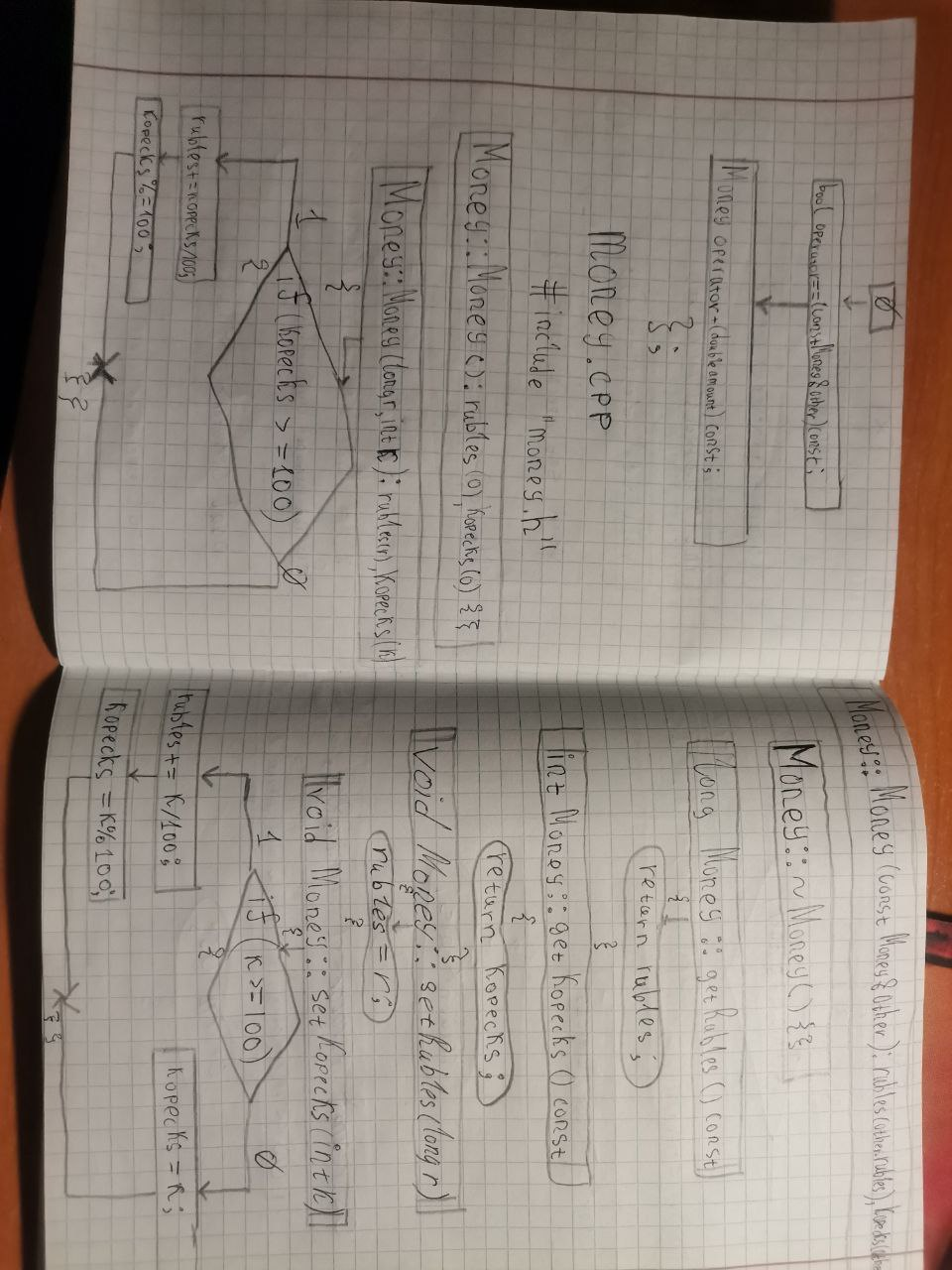
8. Написать программу, в которой продемонстрировать создание объектов и работу всех перегруженных операций.

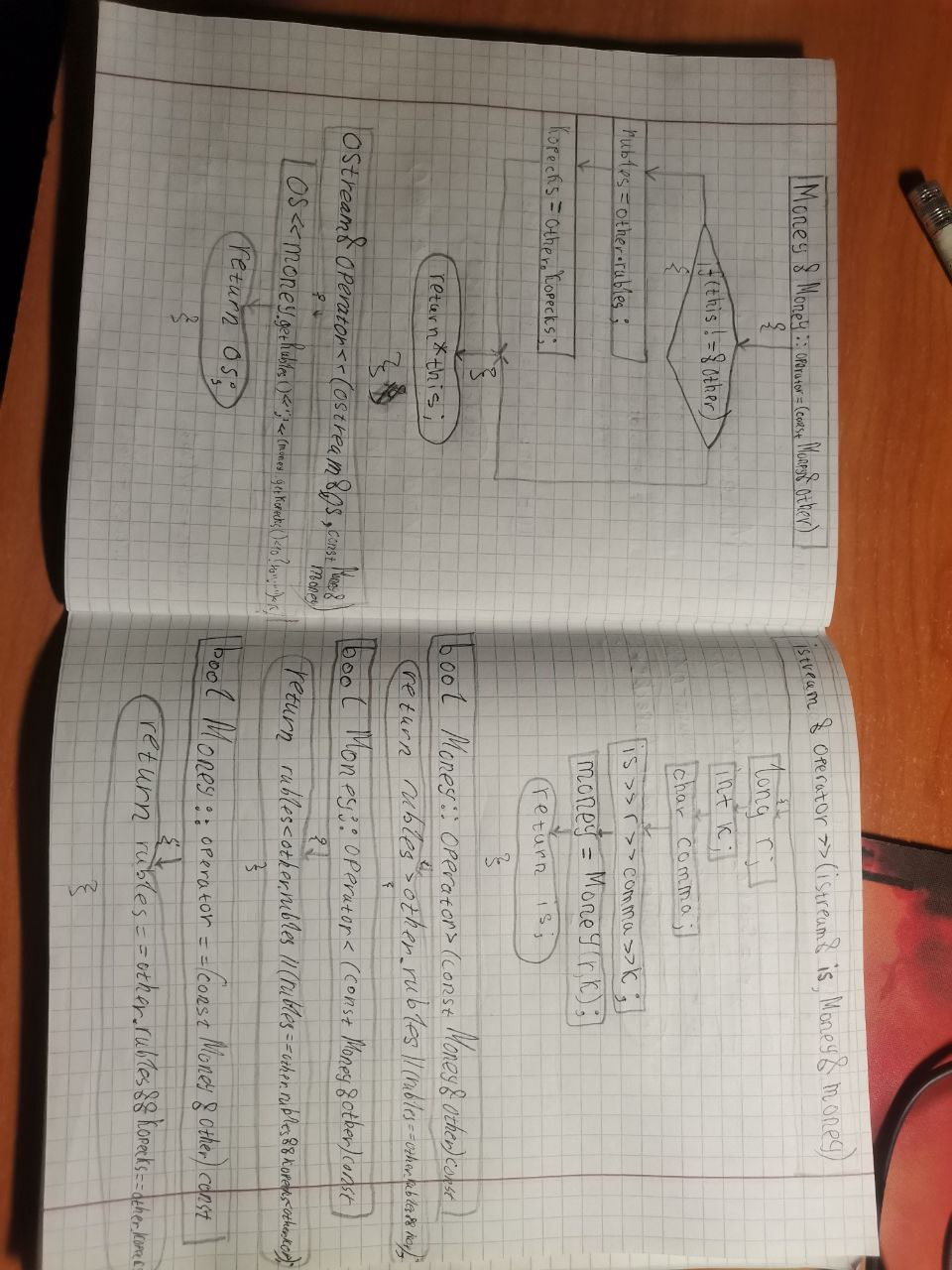
Персональная:

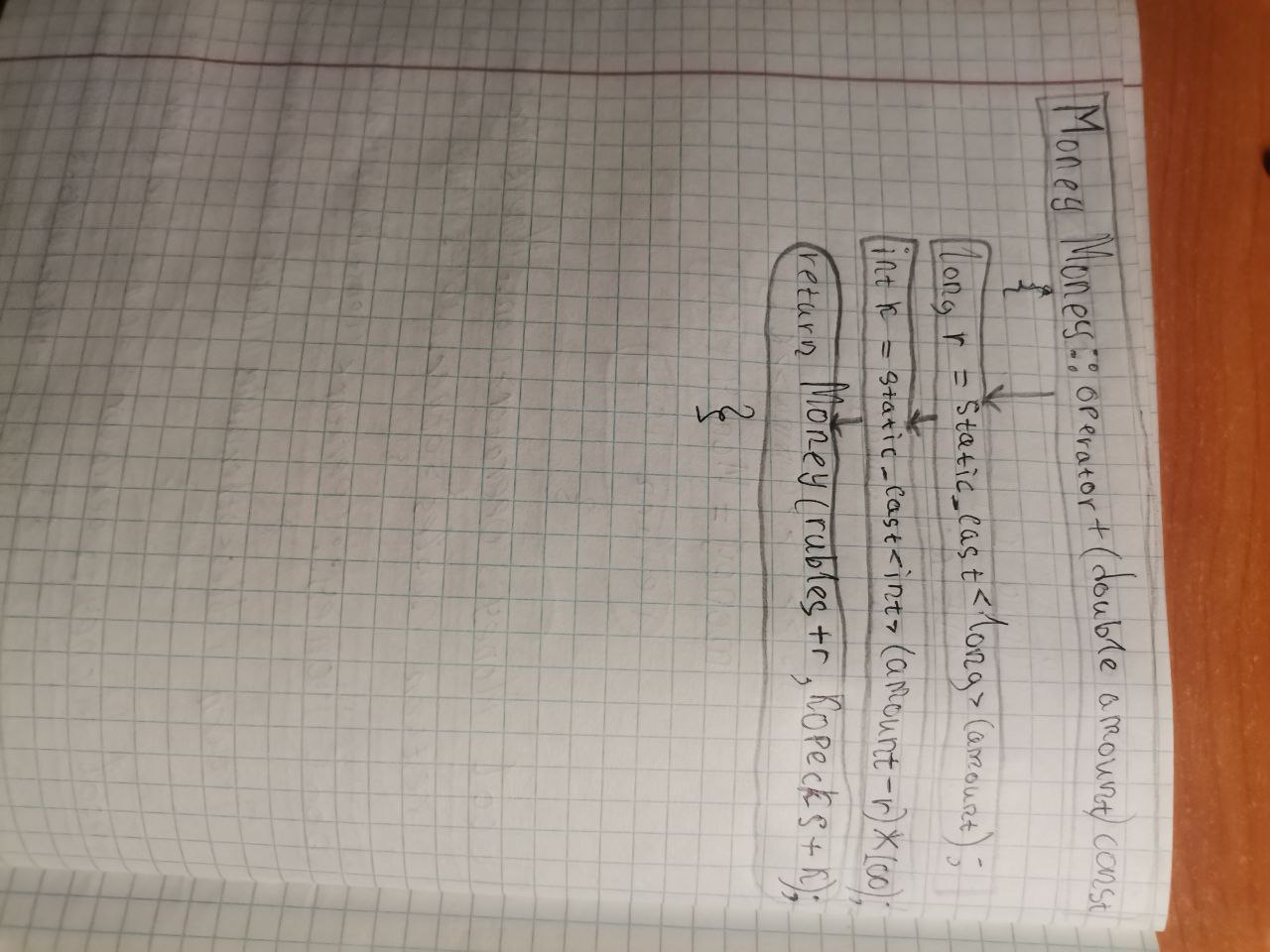


Блок-схема:

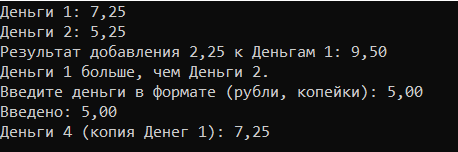








Результат работы:



Контрольные вопросы:

1. Для чего используются дружественные функции и классы?

2. Сформулировать правила описания и особенности дружественных функций.

3. Каким образом можно перегрузить унарные операции?

4. Сколько операндов должна иметь унарная функция-операция, определяемая внутри класса?

5. Сколько операндов должна иметь унарная функция-операция, определяемая вне класса?

6. Сколько операндов должна иметь бинарная функция-операция, определяемая внутри класса?

7. Сколько операндов должна иметь бинарная функция-операция, определяемая

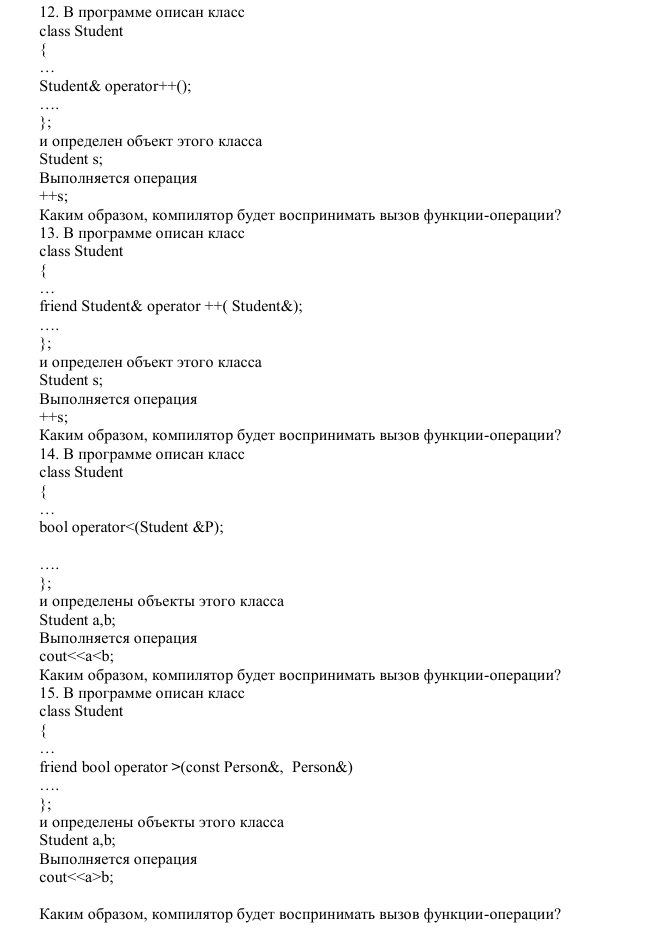
вне класса?

8. Чем отличается перегрузка префиксных и постфиксных унарных операций?

9. Каким образом можно перегрузить операцию присваивания?

10. Что должна возвращать операция присваивания?

11. Каким образом можно перегрузить операции ввода-вывода?



Ответы:

1. **Для чего используются дружественные функции и классы?**  
   Дружественные функции и классы используются для доступа извне к скрытым полям класса, что позволяет расширить интерфейс класса и выполнять действия, не являющиеся свойствами класса, но нуждающиеся в доступе к его скрытым полям.
2. **Сформулировать правила описания и особенности дружественных функций.**
   * Дружественная функция объявляется внутри класса с ключевым словом friend.
   * Она должна принимать объект или ссылку на объект класса в качестве параметра.
   * Дружественная функция может быть обычной функцией или методом другого класса.
   * На нее не распространяются спецификаторы доступа, и место размещения объявления в классе не имеет значения.
   * Одна функция может быть дружественной для нескольких классов.
3. **Каким образом можно перегрузить унарные операции?**  
   Унарные операции можно перегрузить как компонентную функцию класса (нестатический метод без параметров) или как внешнюю (глобальную) функцию с одним параметром типа класса.
4. **Сколько операндов должна иметь унарная функция-операция, определяемая внутри класса?**  
   Унарная функция-операция, определяемая внутри класса, должна иметь один операнд — это вызвавший объект.
5. **Сколько операндов должна иметь унарная функция-операция, определяемая вне класса?**  
   Унарная функция-операция, определяемая вне класса, должна иметь один параметр типа класса.
6. **Сколько операндов должна иметь бинарная функция-операция, определяемая внутри класса?**  
   Бинарная функция-операция, определяемая внутри класса, должна иметь два операнда, где вызвавший объект считается первым операндом.
7. **Сколько операндов должна иметь бинарная функция-операция, определяемая вне класса?**  
   Бинарная функция-операция, определяемая вне класса, должна иметь два параметра типа класса.
8. **Чем отличается перегрузка префиксных и постфиксных унарных операций?**  
   Префиксные унарные операции определяются без параметров, а постфиксные унарные операции требуют первого параметра типа int, который используется только для различения между префиксной и постфиксной формой.
9. **Каким образом можно перегрузить операцию присваивания?**  
   Операция присваивания перегружается путем определения собственной операции-функции, которая возвращает ссылку на объект, для которого она вызвана, и принимает в качестве параметра единственный аргумент — ссылку на присваиваемый объект.
10. **Что должна возвращать операция присваивания?**  
    Операция присваивания должна возвращать ссылку на объект, для которого она была вызвана.

11.**Каким образом можно перегрузить операции ввода-вывода?**Операции ввода-вывода в C++ можно перегрузить, реализуя их как внешние дружественные функции. Это связано с тем, что левыми операндами этих операций являются потоки ввода-вывода (например, std::cin и std::cout), а не объекты пользовательских классов.

### Основные шаги для перегрузки операций ввода-вывода:

1. **Объявление дружественных функций:**
   * Внутри класса объявляются функции operator>> и operator<< как дружественные с помощью ключевого слова friend. Это позволяет им иметь доступ к закрытым полям класса.
2. **Определение функций:**
   * Эти функции принимают поток (std::istream для ввода и std::ostream для вывода) и объект класса в качестве параметров.

12.**Каком образом компилятор будет воспринимать вызов функции-операции?**

Компилятор будет воспринимать вызов функции-операции как вызов нестатического метода operator++, который возвращает ссылку на объект Student.

13.**Каком образом компилятор будет воспринимать вызов функции-операции?**

Компилятор будет воспринимать вызов функции-операции как вызов внешней функции operator++, объявленной как дружественная для класса Student.

14.**Каком образом компилятор будет воспринимать вызов функции-операции?**

Компилятор будет воспринимать вызов функции-операции как вызов метода operator <, который должен быть определен для сравнения объектов типа Student.

15.**Каком образом компилятор будет воспринимать вызов функции-операции?**

Компилятор будет воспринимать вызов функции-операции как вызов внешней дружественной функции operator <<, которая принимает потоки и объект Student.

Ссылка:

**https://github.com/LeonidZhir/-**