

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
Пермский национальный исследовательский политехнический университет  
Электротехнический факультет  
Кафедра информационных технологий и автоматизированных систем

**ОТЧЕТ**  
**о работе по информатике**  
Семестр: 2

На тему: «Классы и объекты. Перегруженные операции в классах.»

Выполнил студент ИВТ-22-26:  
Дияров Тимур Артурович

Проверил доцент кафедры ИТАС:  
Полякова Ольга Андреевна

Пермь 2023

## Постановка задачи

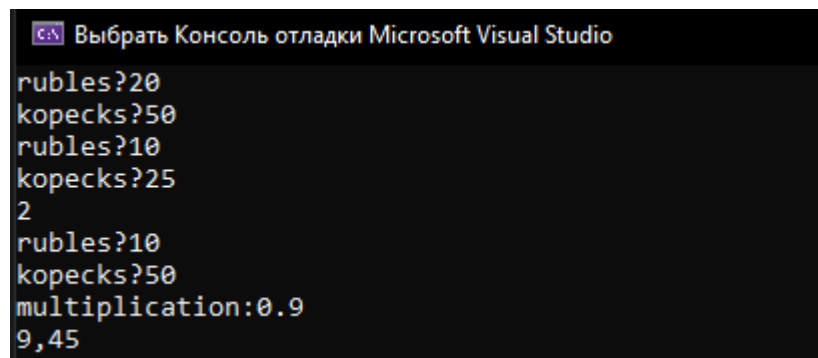
Создать класс Money для работы с денежными суммами. Число должно быть представлено двумя полями: типа long для рублей и типа int для копеек. Дробная часть числа при выводе на экран должна быть отделена от целой части запятой. Реализовать: деление сумм, умножение суммы на дробное число.

## Описание класса

Класс “money” с двумя полями “rubles” (типа long) и “kopecks” (типа int). В структуре имеются четыре метода:

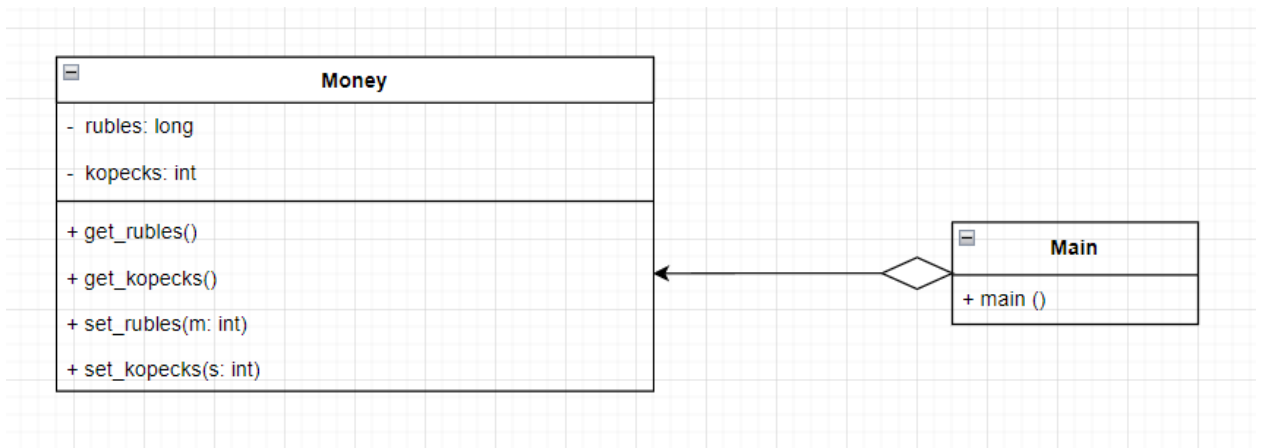
- get\_rubles(): инициализация переменной для рублей
- get\_kopecks(): инициализация переменной для копеек
- set\_rubles(int m) и set\_kopecks(int s): преобразование вышеупомянутых переменных в одно число

## Скриншот работы программы



```
Выбрать Консоль отладки Microsoft Visual Studio
rubles?20
kopecks?50
rubles?10
kopecks?25
2
rubles?10
kopecks?50
multiplication:0.9
9,45
```

## UML-диаграмма



## Контрольные вопросы

1. Дружественные функции и классы используются для доступа к скрытым полям класса и переопределения операций.
2. Для описания дружественной функции нужно использовать слово friend перед её объявлением. Эта функция не является членом класса, но имеет доступ к его частным и защищенным элементам.
3. Унарные операции можно перегрузить объявлением двух разных функций с одним именем, но разными подписями:

```
Complex operator!() const {
    return Complex(real * (-1), imaginary * (-1));
}

Complex operator*(Complex other) {
    return Complex(real * other.real + imaginary * other.imaginary,
        imaginary * other.real - real * other.imaginary);
}
```

4. Функция-операция должна иметь хотя бы одну операнду, которая должна быть объектом класса, для которого она определена.
5. То же, что и в номере 4.
6. То же, что и в номере 4, но операнд должно быть два.
7. То же, что и в номере 6.
8. Перегрузка префиксных и постфиксных унарных операций отличаются тем, что в префиксных унарных операциях операнды передаются перед именем функции, а в постфиксных унарных операциях - после.
9. В C++ можно перегрузить операцию присваивания, используя конструкцию `operator =`.
10. Операция присваивания должна возвращать ссылку на то, что ей присваивается.

11. Операция присваивания должна возвращать ссылку на то, что ей присваивается.
12. Вызов операции будет восприниматься как вызов операнды, которая интерпретируется как объект, к которому применяется оператор.
13. То же, что и в номере 12.
14. То же, что и в номере 12, но операнда вызывается на два объекта класса.
15. То же, что и в номере 14.