

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ  
УНИВЕРСИТЕТ им. В. Г. ШУХОВА» (БГТУ им. В.Г. Шухова)

Кафедра программного обеспечения  
вычислительной техники и автоматизированных  
систем

**Лабораторная работа №4**

по дисциплине: ООП

тема: «Классы»

Выполнил: студент группы ВТ-231

Масленников Д. А.

Проверили:  
Буханов Д. Г.

Белгород 2025

**Цель работы:** приобретение практических навыков создания класса на языке C++.

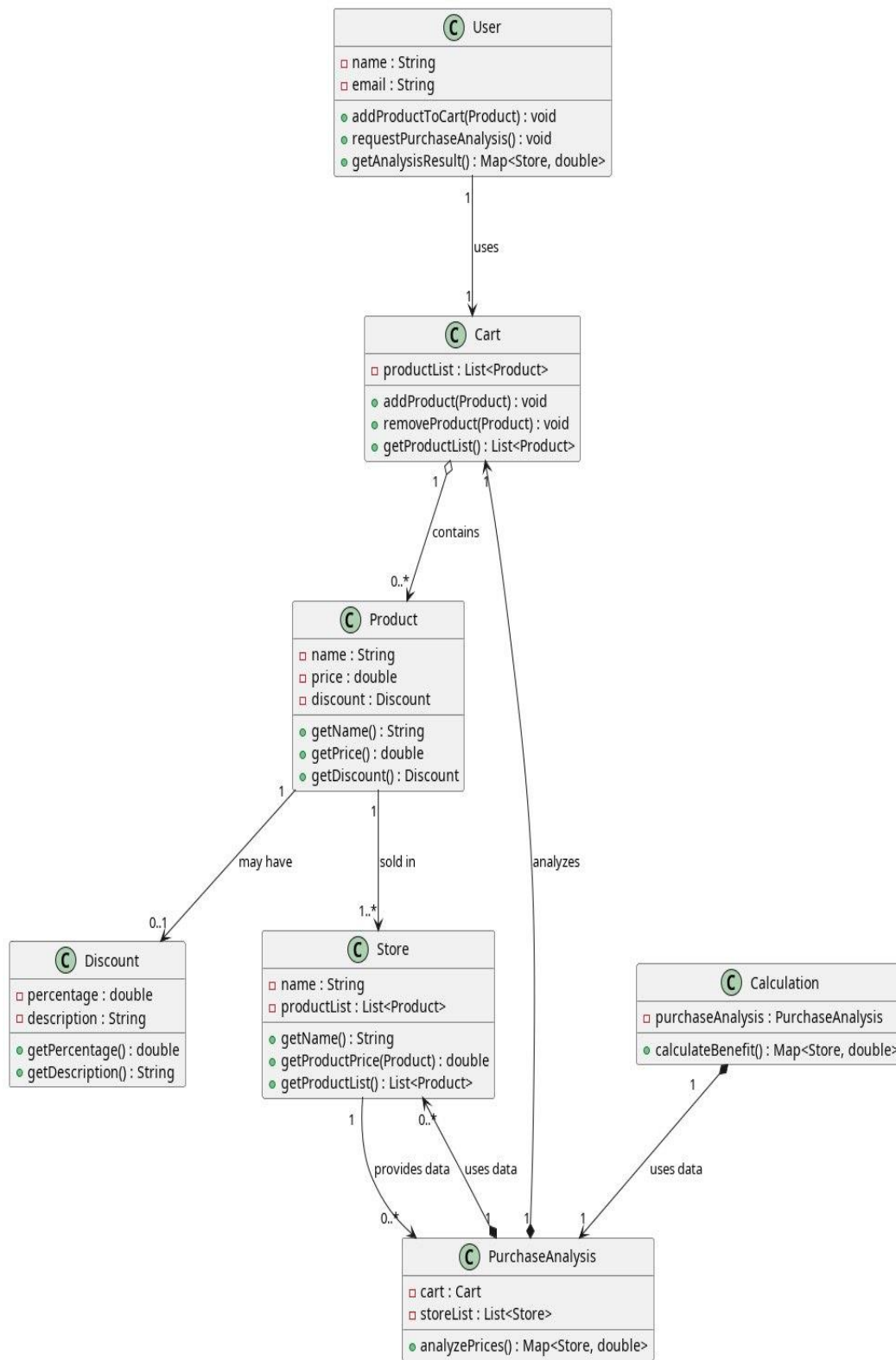
**Задание:** выполнить построение диаграммы объектов (не менее 7-ми объектов) в соответствии с заданием варианта (табл. 1). По заданному описанию класса в соответствии с вариантом создать класс на языке C++ и переопределить указанные операции.

#### Вариант 4

4	Выполнить построение объектной модели следующей предметной области: “система расчета выгодной покупки”.	<p>Создать класс Money, разработав следующие элементы класса:</p> <p>а. Поля:</p> <p>int first; //номинал купюры</p> <p>int second; //количество купюр</p> <p>б. Конструктор, позволяющий создать экземпляр класса с заданными значениям полей.</p> <p>в. Методы, позволяющие:</p> <p>вывести номинал и количество купюр;</p> <p>определить, хватит ли денежных средств на покупку товара на сумму рублей.</p> <p>определить, сколько шт товара стоимости рублей можно купить на имеющиеся денежные средства.</p> <p>г. Перегрузить (переопределить):</p> <p>сложение(унарное, постфиксное);</p> <p>сложение, совмещённое с присваиванием;</p>
---	---	--

## Задание 1:

Выполнить построение объектной модели следующей предметной области: «система расчета выгодной покупки»



## Задание 2:

Создать класс Money, разработав следующие элементы класса:

а. Поля:

`int first; //номинал купюры`

`int second; //количество купюр`

б. Конструктор, позволяющий создать экземпляр класса с заданными значениям полей.

в. Методы, позволяющие:

вывести номинал и количество купюр;

определить, хватит ли денежных средств на покупку товара на сумму рублей.

определить, сколько шт товара стоимости рублей можно купить на имеющиеся денежные средства.

г. Перегрузить (переопределить):

сложение(унарное, постфиксное);

**сложение, совмещённое с присваиванием;**

Main.cpp

```
#include "header.hpp"
```

```
int main() {
```

```
    Money wallet(100, 5);
```

```
    Money extra(500, 1);
```

```
    Money less(50, 5);
```

```
    wallet.print();
```

```
    extra.print();
```

```
    less.print();
```

```
    int productPrice = 450;
```

```
    if (wallet.canAfford(productPrice)) std::cout << "можете купить " << wallet.howManyItems(productPrice) << "
товаров" << std::endl;
```

```
    else {
```

```
        std::cout << "не хватит" << std::endl;
```

```
    }
```

```
    Money x = wallet++;
```

```
    wallet.print();
```

```
    x.print();
```

```

wallet += extra;
wallet.print();

wallet += less;
wallet.print();

return 0;
}

```

## interface.cpp

```

#include "header.hpp"

Money::Money() : nominal(0), count(0) {}

Money::Money(int f, int s) : nominal(f), count(s) {}

void Money::print() const {
    std::cout << "номинал - " << nominal << " количество - " << count << std::endl;
}

bool Money::canAfford(int price) const {
    return (nominal * count) >= price;
}

int Money::howManyItems(int price) const {
    return (nominal * count) / price;
}

Money operator+= (Money& a1, const Money& a2) {
    if (a1.nominal == a2.nominal) a1.count += a2.count;
    else {
        int exchangeRate;
        if (a1.nominal > a2.nominal) {
            exchangeRate = a1.nominal / a2.nominal;
            a1.count += a2.count / exchangeRate;
        }
        else {
            exchangeRate = a2.nominal / a1.nominal;
            a1.count += a2.count * exchangeRate;
        }
    }
    return a1;
}

const Money operator++(Money& a1, int) {
    Money oldCount = a1;

```

```
    a1.count++;  
    return oldCount;  
}
```

header.hpp

```
#ifndef HEADER_HPP  
  
#define HEADER_HPP  
  
#include <iostream>  
  
class Money {  
private:  
    int nominal;  
    int count;  
  
public:  
    Money();  
    Money(int f, int s);  
    void print() const;  
    bool canAfford(int price) const;  
    int howManyItems(int price) const;  
    friend Money operator+= (Money& a1, const Money& a2);  
    friend const Money operator++(Money& a1, int);  
};  
  
#endif
```

**Вывод:** приобрели практические навыки создания операторов класса на языке C++. так же научились строить диаграмму классов