Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

Дисциплина: «Основы алгоритмизации и программирования»

Тема: Классы и объекты. Инкапсуляция.

Семестр 2

Выполнил работу

Студент группы РИС-22-1Б

Токарев Павел Аркадьевич

Проверил

Доцент кафедры ИТАС

Полякова Ольга Андреевна

Г. Пермь-2023

**Постановка задачи**

1.Реализовать определение нового класса.

2.Полу first-положительное число, отражающее номинал купюры, а second-их количество.

2.Продемонстрировать различные способы создания объектов в главной функции.

3.Реализовать метод инициализации init, ручной ввод через метод read, вывод на экран через метод show и метод summa, вычисляющий произведение номинала и количества купюр.

4.Реализовать внешнюю функцию make , принимающую в качестве аргументов поля класса и возвращающую экземпляр класса.

**Анализ**

1.Определим класс money и его методы в заголовочном файле Money.h

2.Метод init реализуем присваиванием полям класса значений аргументов.

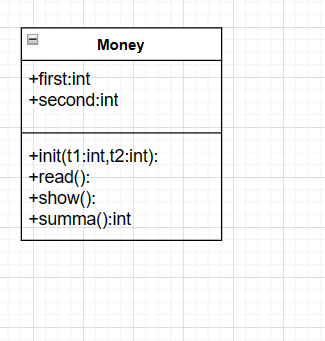
3.Методы show и read реализуем через потоковый вводи и вывод.

4.Метод summa возвращает значение, равное first\*second

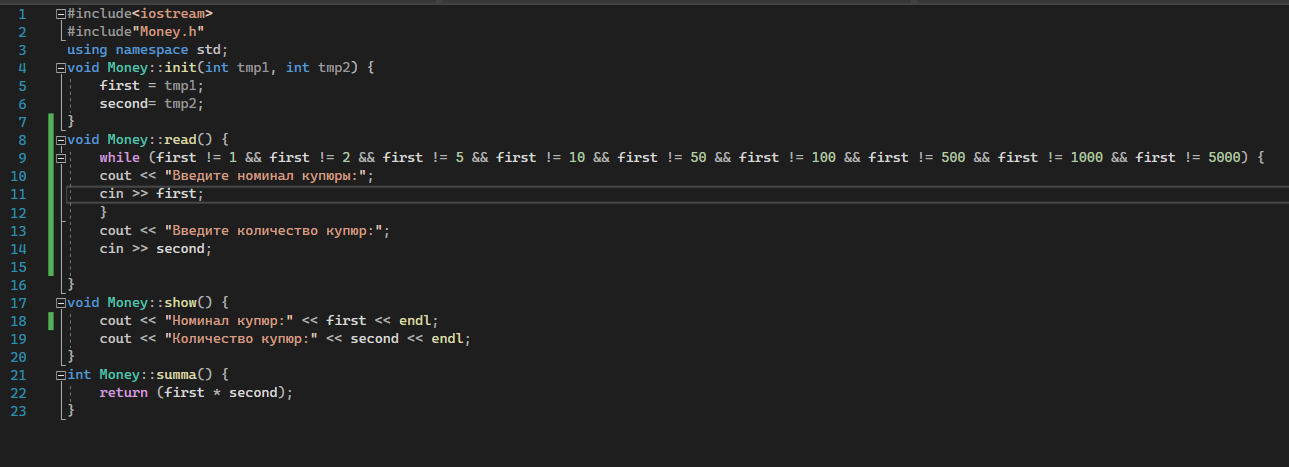
5.Внешняя функция make возвращает объект, поля которого заполнены с помощью метода init.

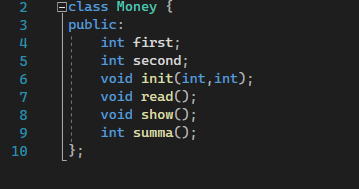
6.В функции main покажем различные способы создания объектов и массивов объектов.

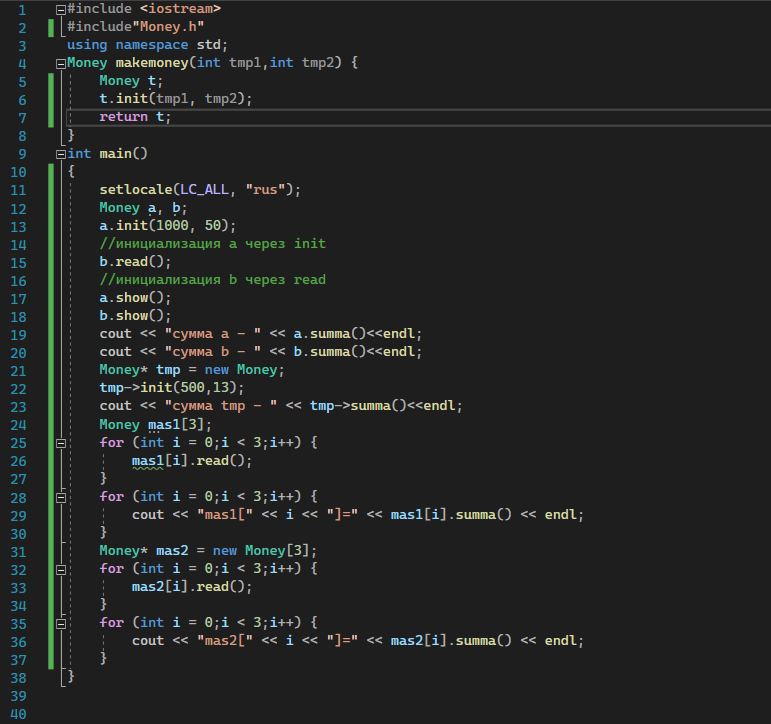
**Диаграмма UML**

****

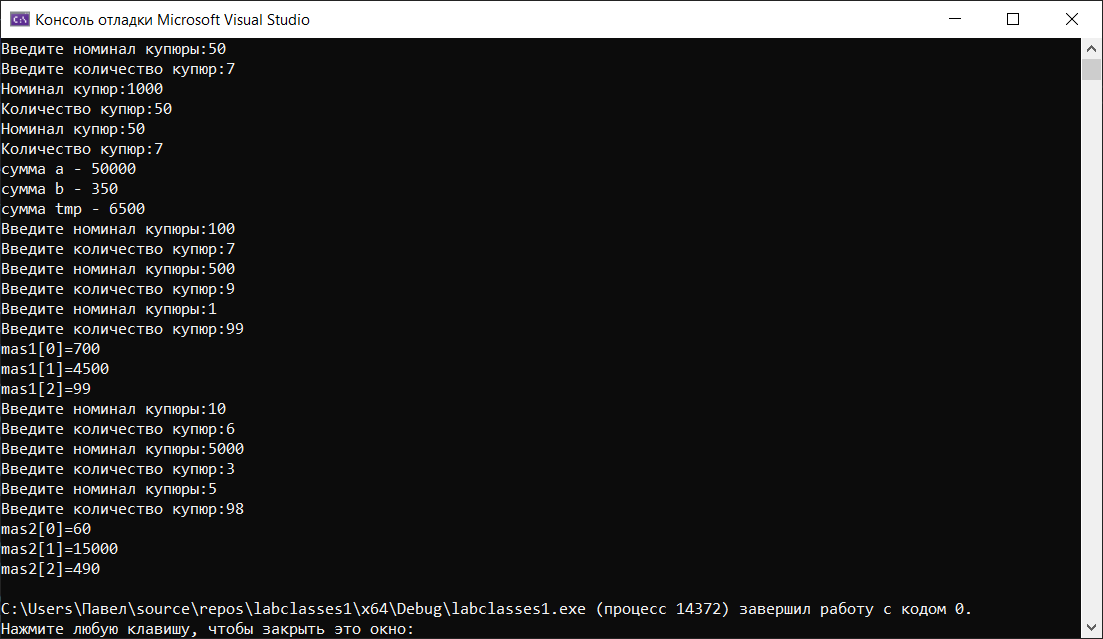
**Код программы**

****

****

****

**Результаты работы программы**

****

**Анализ результатов**

Как видно из результата работы программы, без ошибок выполняется создание объектов класса всеми способами. Программа корректно вычисляет денежную сумму.

**Контрольные вопросы**

1.Класс-абстрактный тип данных, определяемый пользователем. Он позволяет смоделировать реальный объект с данными и функциями, позволяющими с ним взаимодействовать.

2.Объект класса-экземпляр, созданный по его подобию и обладающий его атрибутами.

3.Поля класса-переменные внутри класса, образующие его структуру.

4.Методы-функции внутри класса, определяющие операции над объектом.

5.Спецификаторы доступа используются для управления видимостью компонентов класса.

6.При использовании спецификатор public компоненты экземпляра класса видны в любой части программы и к ним можно обратиться.

7.При спецификаторе доступа private компоненты класса не будут доступны извне. Поля с private доступом могут изменяться public методами.

8.По умолчанию при объявлении класса через слово class используется спецификатор доступа private.

9.По умолчанию при объявлении класса через слово struct используется спецификатор доступа public.

10.При описании интерфейса класса должен использоваться спецификатор public, так как интерфейс определяет открытые методы и свойства, доступные для использования в других частях программы.

11.С помощью модификаторов. Модификатор-метод для задания атрибута класса.

12.С помощью селекторов. Селектор-метод, возвращающий значение атрибута.

13.К полю name можно обратиться с помощью оператор доступа к членам класса ->.

14. Можно обратиться к полю name объекта s через оператора доступа к члену класса (.) следующим образом: s.name.

15.Невозможно напрямую обратиться к полям класса, однако можно изменить или получить значение поля с помощью селекторов и модификаторов.

s->setname(“name”);

name=s.getname();

16.Невозможно обратиться к полям класса так как по умолчанию используется спецификатор доступа private.

17.Можно обратиться к полю name как с помощью оператора доступа (.) , так и с помощью селекторов и модификаторов, поскольку используется спецификатор доступа public.