**Бильдина Екатерина, гр.3775, Архитектура ПО (01)**

**ЗАДАНИЕ\_01:**

**Введение в понятие архитектуры, проектирование ПО и жизненный цикл программного продукта. UML-диаграммы**

Задание: Возьмите работу с ООП, Добавтье туда все возможное что мы смотрели на семинаре, и сделайте по ней UML диограмму

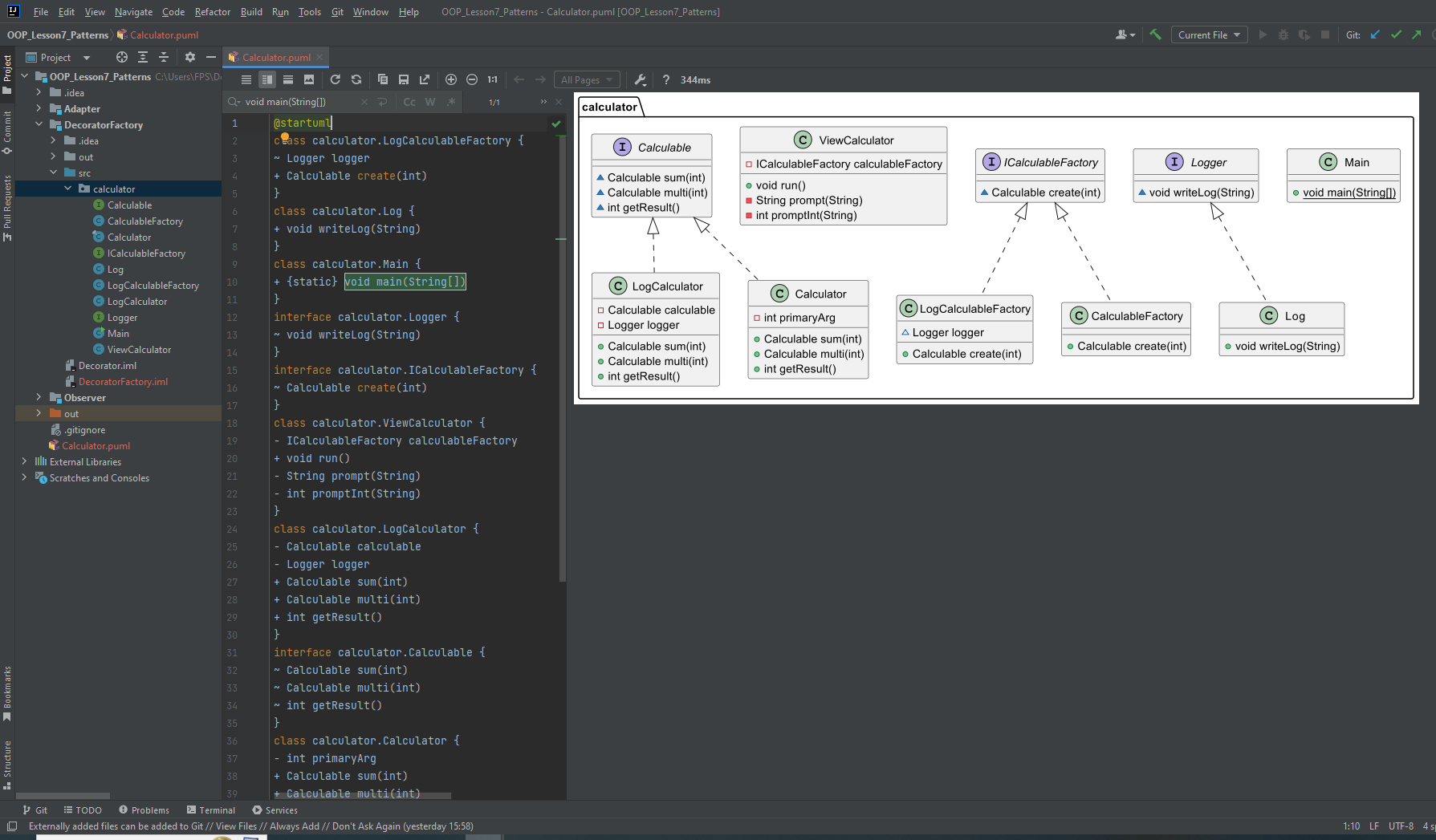
**ФОРМАТ СДАЧИ ДЗ:**

\*Коллеги, обратите внимание! Если вы работаете на основе примера с семинара, прикрепите, пожалуйста, файл в формате .puml. Это позволит мне лучше оценить вашу работу и, возможно, добавить дополнительные примеры для визуализации. Также не забывайте читать комментарии после домашнего задания. В них содержатся примеры ручного добавления связей и отношений. И если возможно, отправляйте ссылку на ваш GitHub.  
\*

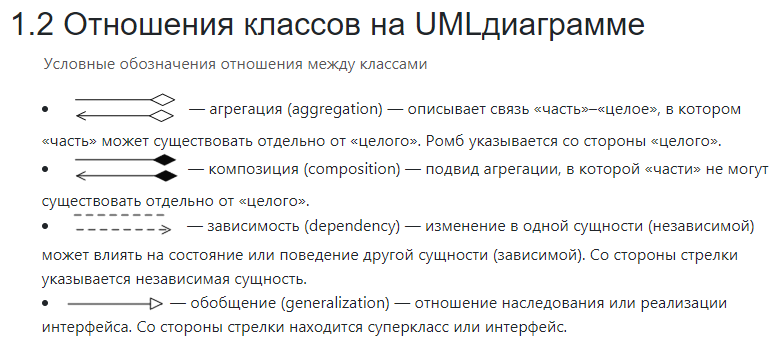
**РЕШЕНИЕ:**

UML — это язык графического описания для объектного моделирования в области разработки ПО, моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

Проще говоря - генерация UML-диаграмм из Java-кода — это простой и эффективный способ понять структуру и взаимодействие классов в программе.

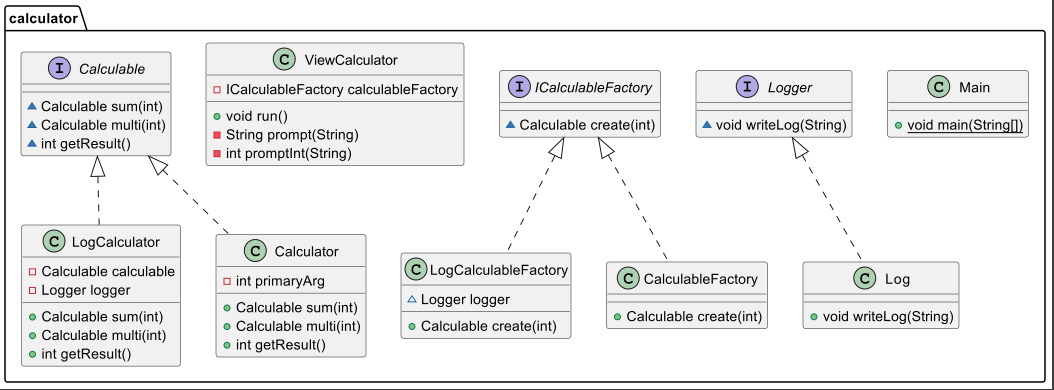


Отношения классов на UML-диаграмме:

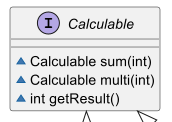


Поработаем с нашим проектом:

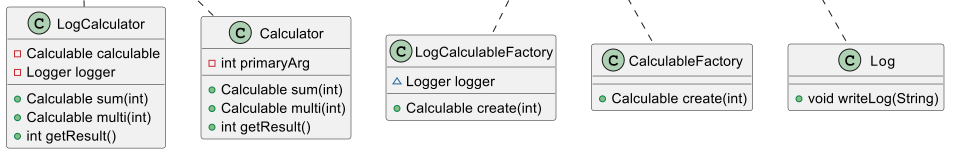
Есть проект calculator включающий себя классы и нтерфейсы.



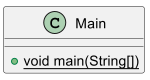
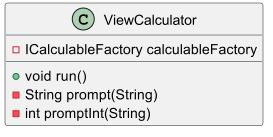
Интерфейсы:



Классы, которые зависят от интерфейсов.

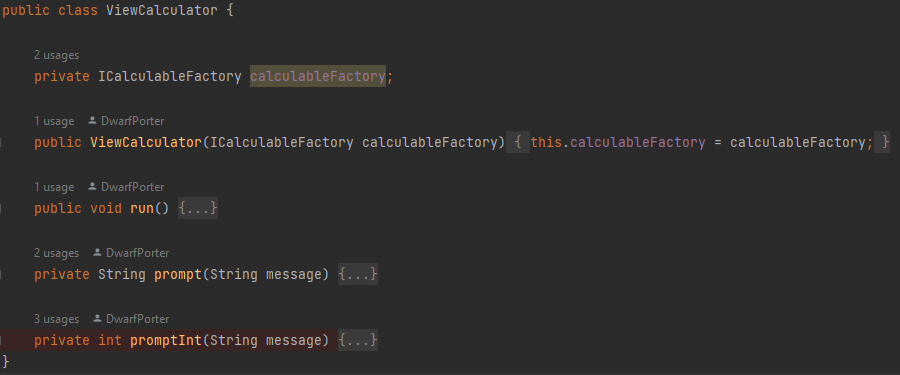
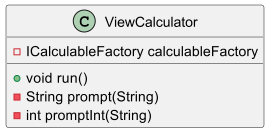


И “самостоятельные” классы



Так же мы видим визуализацию типов Методов и переменных. А именно:

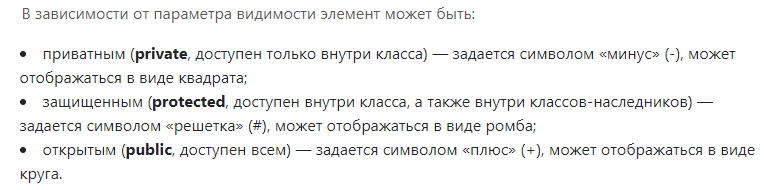
**Класс / Атрибуты / Операции - набор объектов со схожими свойствами**



Рассмотрим public class ViewCalculator.

В нем создан приватный объект класса ICalculableFactory с именем calculableFactory.

И методы публичный run и два приватных promt, promtInt.



Класс – это описание набора объектов с одинаковыми атрибутами, операциями, связями и семантикой.  
  
Графически класс изображается в виде прямоугольника, разделенного на 3 блока горизонтальными линиями:

имя класса

атрибуты (свойства) класса

операции (методы) класса.

Для атрибутов и операций может быть указан один из трех типов видимости:

— — private (частный)

# — protected (защищенный)

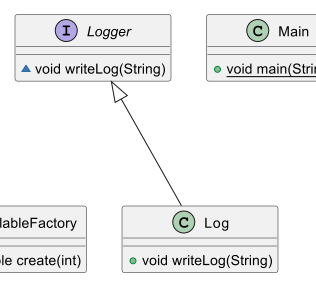
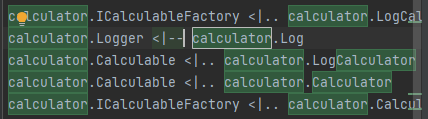
+ — public (общий)

Видимость для полей и методов указывается в виде левого символа в строке с именем соответствующего элемента.  
  
Каждый класс должен обладать именем, отличающим его от других классов. Имя – это текстовая строка. Имя класса может состоять из любого числа букв, цифр и знаков препинания (за исключением двоеточия и точки) и может записываться в несколько строк.  
На практике обычно используются краткие имена классов, взятые из словаря моделируемой системы. Каждое слово в имени класса традиционно пишут с заглавной буквы (верблюжья конвенция), например Sensor (Датчик) или TemperatureSensor (ДатчикТемпературы).  
  
Для абстрактного класса имя класса записывается курсивом.

Абстрактные методы класса обозначаются курсивным шрифтом.  
Статические методы класса обозначаются подчеркиванием.

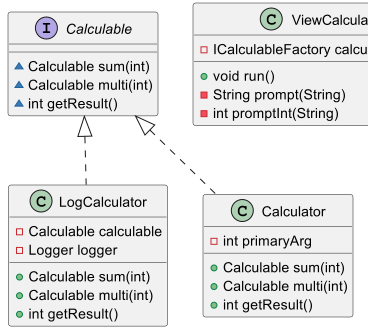
НАСЛЕДОВАНИЕ <|--

**C:Log наследует I:Logger**



РЕАЛИЗАЦИЯ ИНТЕРФЕЙСА <|..

**Классы Calculator и LogCalculator реализуют Интерфейс Calculable**



calculator.Calculable <|.. calculator.LogCalculator

calculator.Calculable <|.. calculator.Calculator

Основные элеметы в UML диаграмме:

