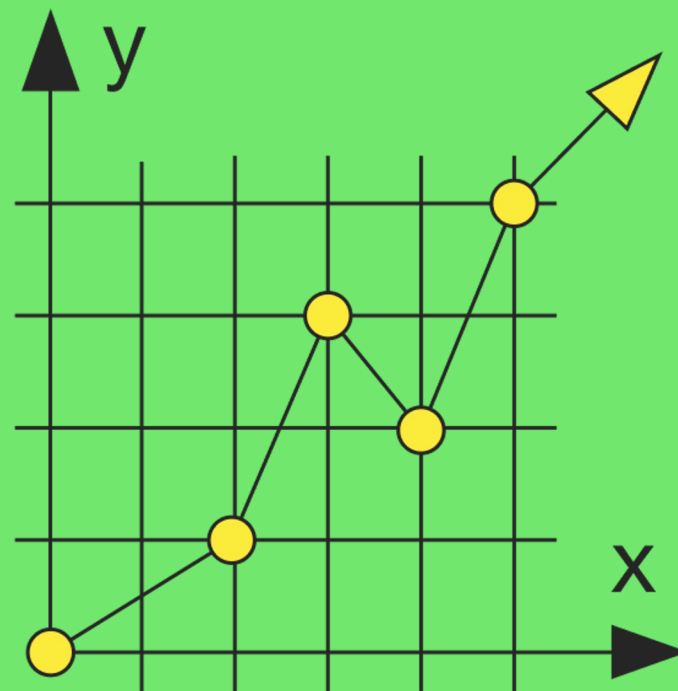


От простого к сложному

Семинар 5





Содержание урока



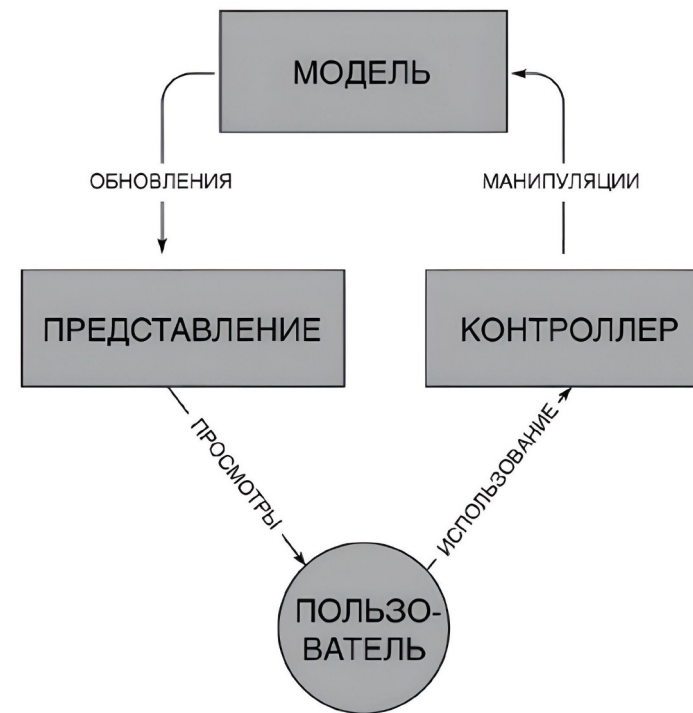
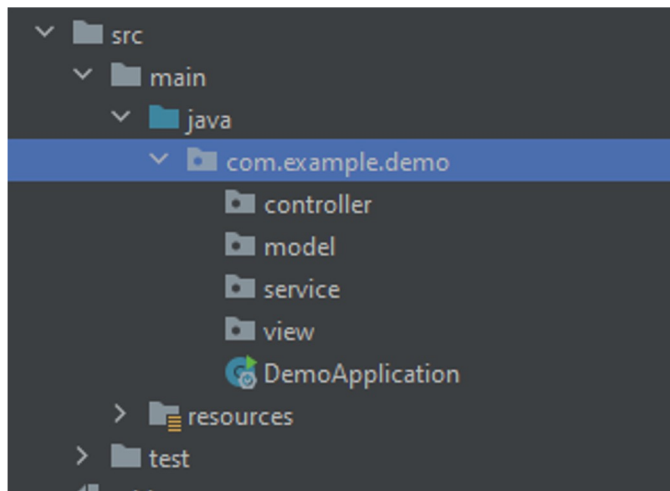
Что будет на уроке сегодня

- 📌 Повторим теорию, разберем, что такое MVC
- 📌 Получим практические навыки в проектировании приложения
- 📌 Научимся проектировать MVC приложения с необходимым уровнем абстракции



Разберем теоретические моменты

Поговорим о структуре MVC





Перейдем к практике



Задача 1 - Data/model

- 📌 Создать package – model. Работу продолжаем в нем
- 📌 Реализовать абстрактный класс User и его наследники Student и Teacher. Родитель имеет в себе общие данные (пример: фио, год рождения и тд), а наследники собственные параметры (какие, по вашему выбору – Пример: studentId для Student, teacherId для Teacher)



Поставьте видео на паузу
и выполните задание



Задача 2 - Service

- 📌 Создать package – service. Работу продолжаем в нем
- 📌 Создать класс DataService с методами в по управлению сущностями User (create, read) и агрегирующий всех пользователей заведенных в системе
- 📌 StudentId должны быть созданы по алгоритму – максимальный id в коллекции + 1. Аналогично и для TeacherId. Здесь решение через перебор элементов и instanceOf



Поставьте видео на паузу
и выполните задание



Задача 3 - View/Dto

- 📌 Создать package – view. Работу продолжаем в нем
- 📌 Создать класс StudentView, содержащий в себе метод вывода в консоль данных студента поданных на вход



Поставьте видео на паузу
и выполните задание



Задача 4 - Controller

- 📌 Создать package – controller. Работу продолжаем в нем
- 📌 Создать класс Controller агрегирующий в себе необходимые классы в виде полей, а инициализируем прямо в поле.
- 📌 Создаем метод созданиеСтудента – реализующий логику, путем вызова соответствующих методов интерфейсов:
 - 📌 Создания Студента в сервисе
 - 📌 Возвращение всех студентов в сервисе
 - 📌 Вызов view и передача списка найденных студентов



Поставьте видео на паузу
и выполните задание



Домашнее задание



Домашнее задание

- 📌 Создать класс УчебнаяГруппа содержащая в себе поля Преподаватель и список Студентов
- 📌 Создать класс УчебнаяГруппаСервис, в котором реализована функция(входные параметры - (Teacher, List<Strudent>)) формирования из Студентов и Преподавателя УчебнойГруппы и возвращения его
- 📌 Создать метод в Контроллере, в котором формируется учебная группа, путем вызова метода из сервиса
- 📌 Все вышеуказанное создать согласно принципам ООП пройденным на семинаре



Подведем итоги



Спасибо за работу!