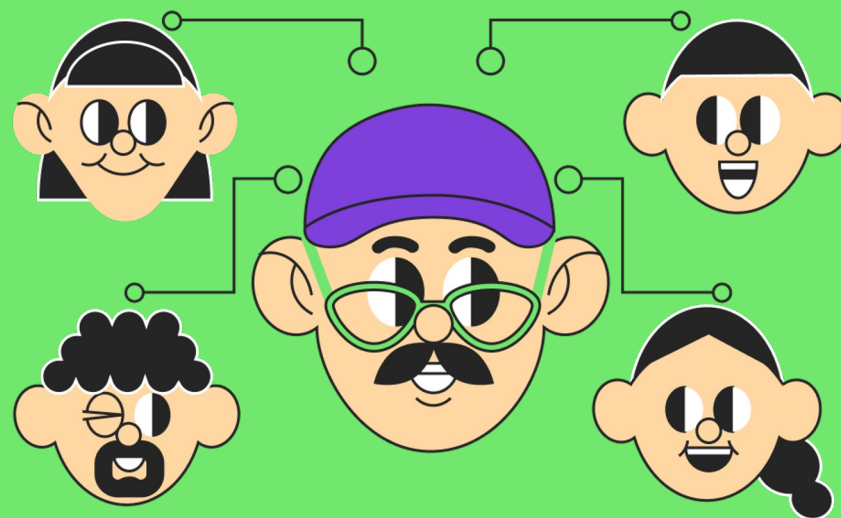


# Абстракции и интерфейсы

Семинар 2





## Как эффективно заниматься в асинхронном формате?

- 📌 Не забывайте смотреть лекцию перед тем, как перейти к семинару;
- 📌 Во время выполнения заданий ставьте видео на паузу и пробуйте выполнить его самостоятельно;
- 📌 После самостоятельного выполнения задания, обязательно посмотрите как преподаватель выполняет его и анализируйте свои действия;
- 📌 Обязательно выполняйте домашнее задание после семинара;
- 📌 Регулярно освежайте материал в памяти с помощью конспектов к лекциям, видеолекций и других удобных вам материалов.



Ставьте видео на паузу и выполняйте задания




WELCOME



# Содержание урока



## Что будет на уроке сегодня

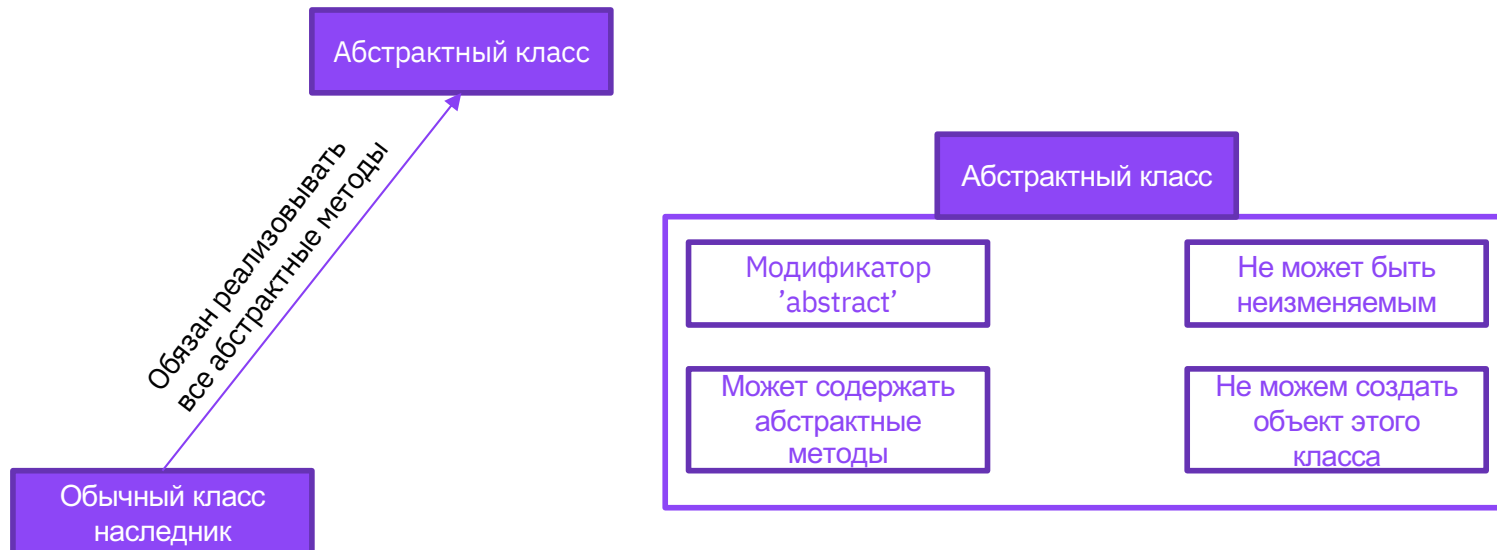
-  Повторим теорию базовых определений ООП - абстракция, наследование
-  Получим практические навыки в построении абстрактных сущностей и интерфейсов
-  Научимся правильно использовать абстрактные классы и интерфейсы



Повторим основные теоретические  
моменты с лекции

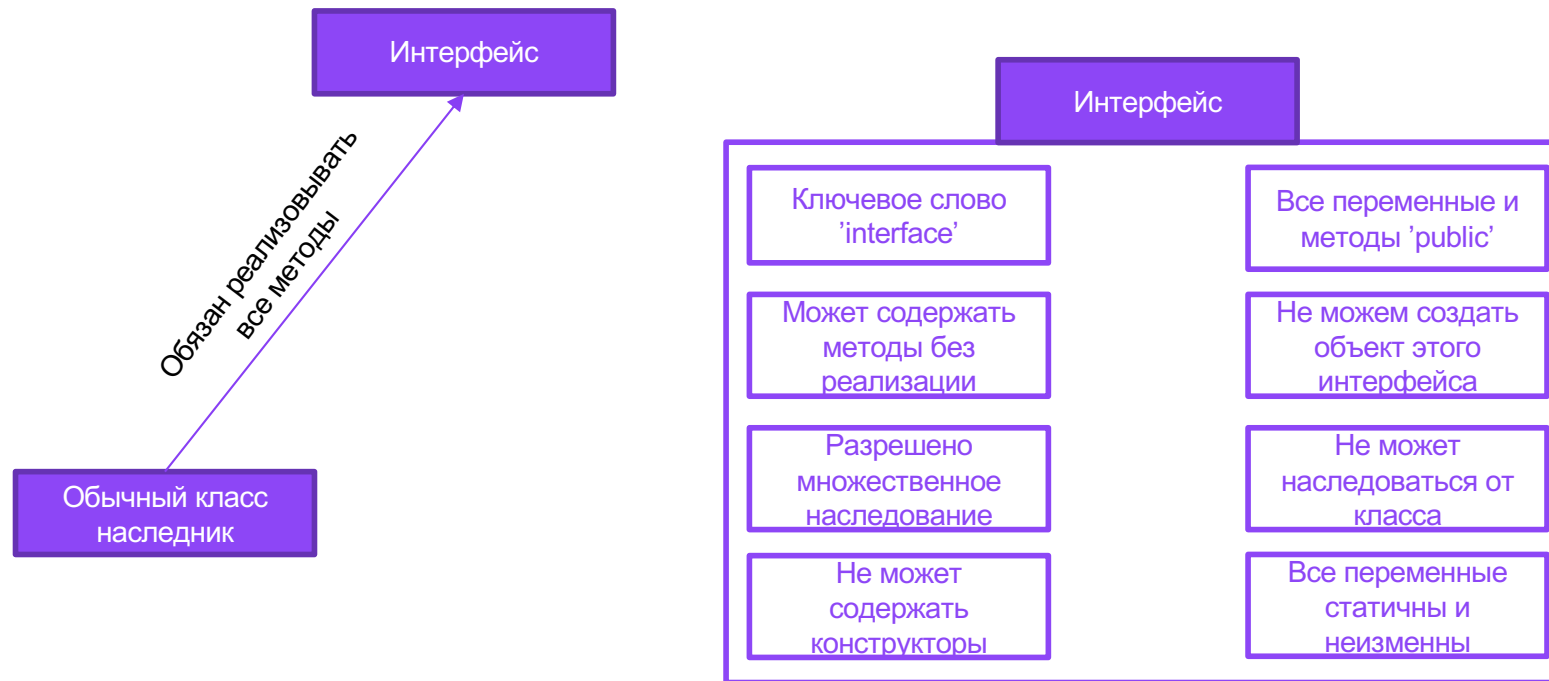


## Что такое абстрактный класс?



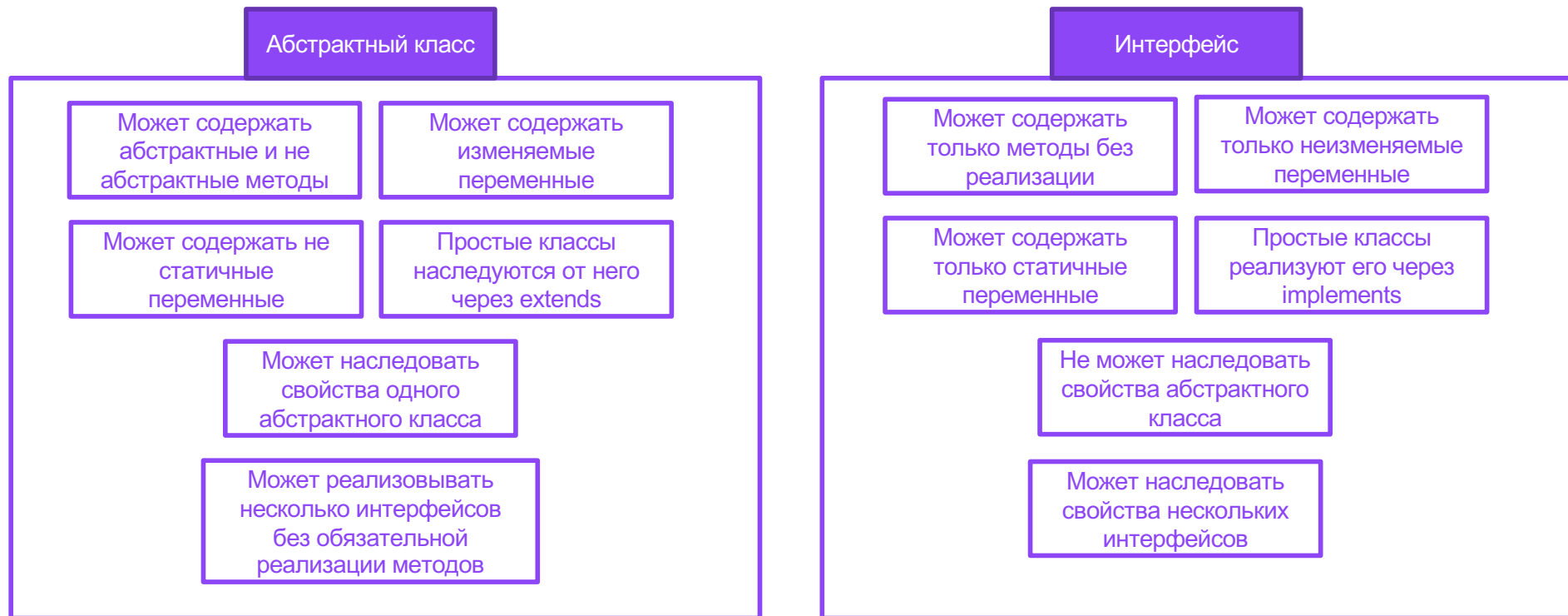


## Что такое интерфейс?





## Чем отличается абстрактный класс от интерфейса?







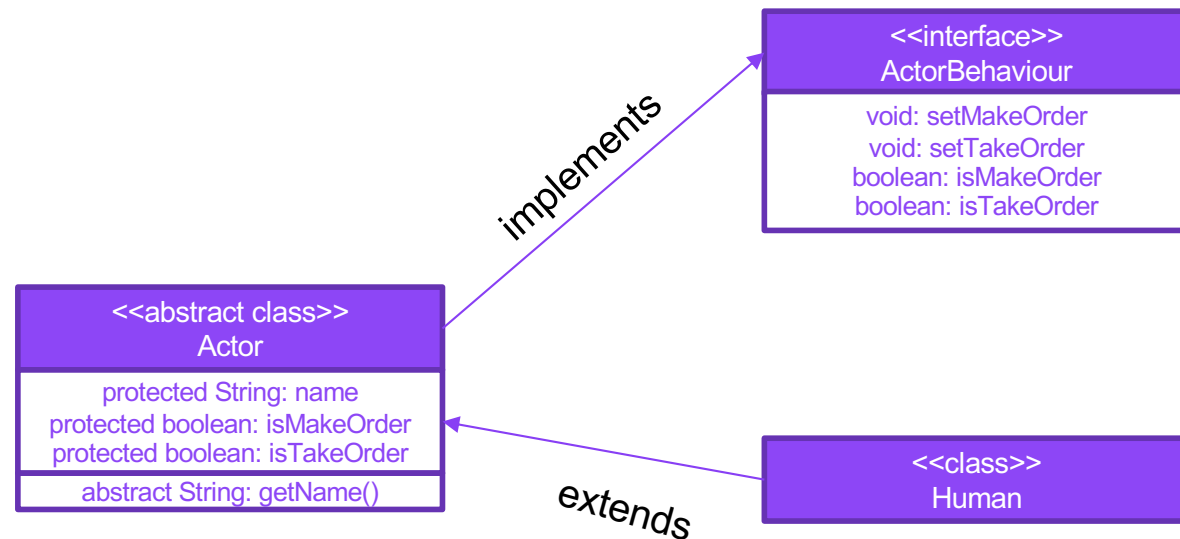
Перейдем к практике



## Задание 1

### Необходимо реализовать:

1. Интерфейс ActorBehaviour, который будет содержать описание возможных действий актора в очереди/магазине
2. Абстрактный класс Actor, который хранит в себе параметры актора, включая состояние готовности сделать заказ и факт получения заказа.  
Дополнение: для большего понимания, можно сделать методы-геттеры для имени и прочих “персональных данных” abstract
3. Класс Human, который должен наследоваться от Actor и реализовывать ActorBehaviour

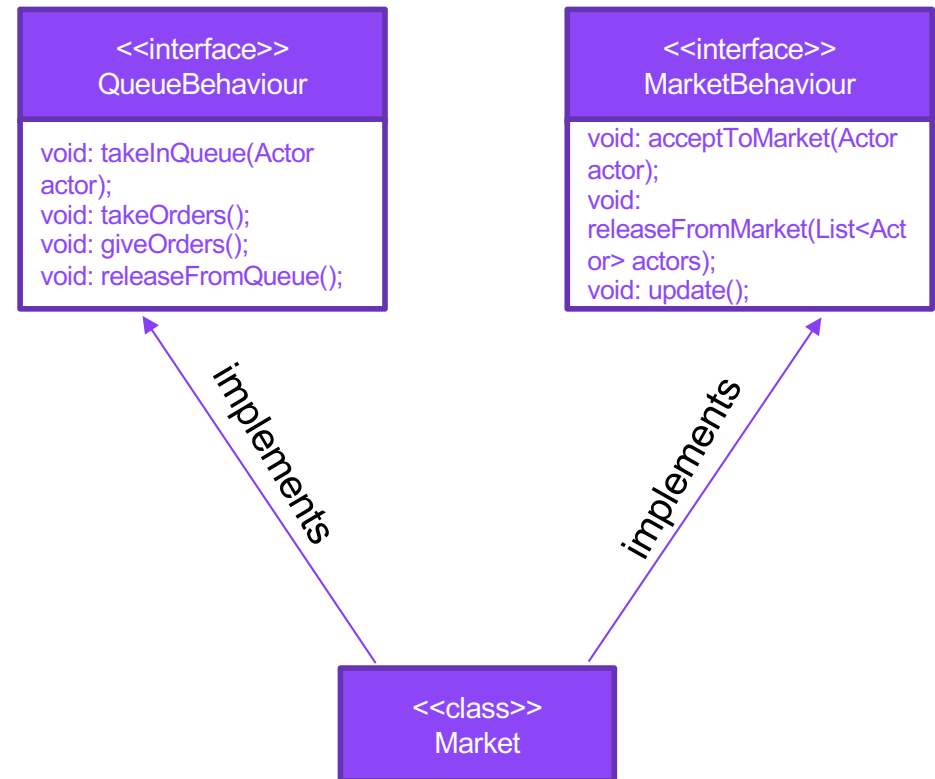




## Задание 2

**Необходимо реализовать строение классов, без конкретной реализации:**

1. Интерфейс QueueBehaviour, который описывает логику очереди – помещение в/освобождение из очереди, принятие/отдача заказа
2. Интерфейс MarketBehaviour, который описывает логику магазина – приход/уход покупателей, обновление состояния магазина
3. Класс Market, который реализовывает два вышеуказанных интерфейса и хранит в списке список людей в очереди в различных статусах

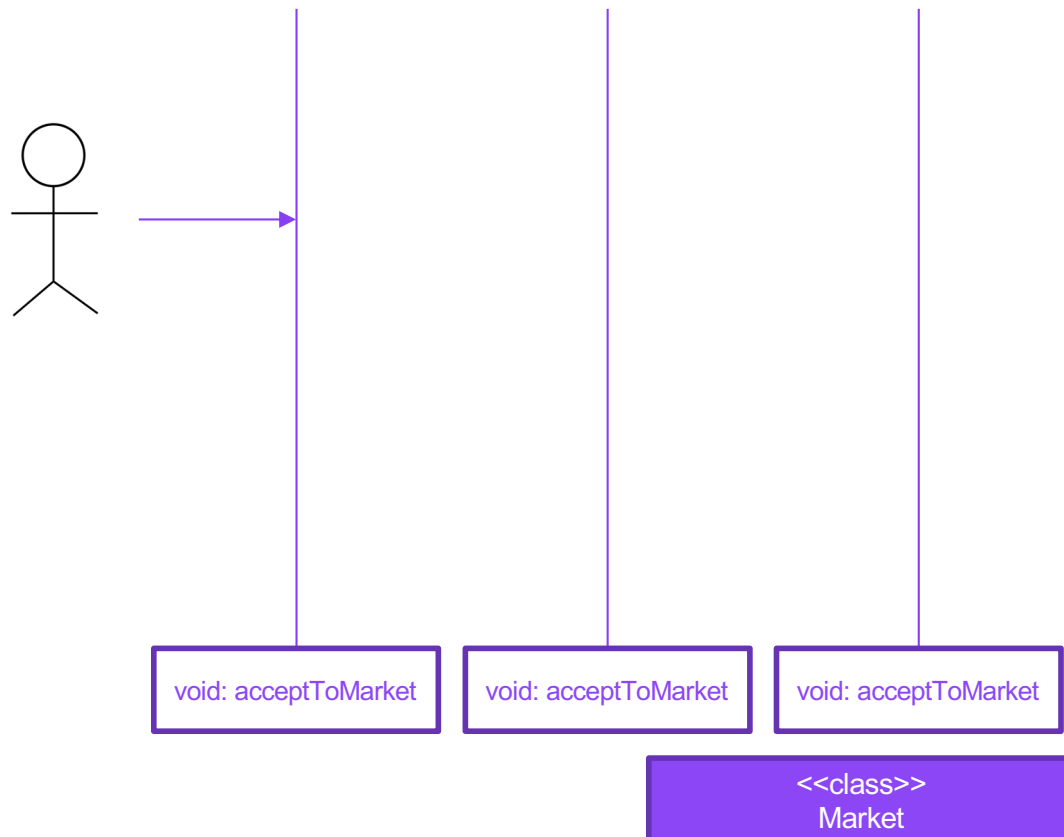




# Домашнее задание



## Домашнее задание



Реализовать класс Market и все методы, которые он обязан реализовывать.

Методы из интерфейса QueueBehaviour, имитируют работу очереди, MarketBehaviour – помещает и удаляет человека из очереди, метод update – обновляет состояние магазина (принимает и отдает заказы)



Подведем итоги



Спасибо за работу!