

## Темы урока

<b>Chatbot: Пишем интерфейсы и вспомогательные классы</b>	<b>1</b>
Reminder.Receiver.Core	1
Reminder.Sender.Core	1
Reminder.Parsing	1
<b>Chatbot: Регистрация собственного бота в Telegram</b>	<b>1</b>
<b>Chatbot: Пишем логику получения и отправки сообщений</b>	<b>2</b>
Reminder.Receiver.Telegram	2
Reminder.Sender.Telegram	2
<b>Chatbot: Модифицируем доменную логику</b>	<b>2</b>
Reminder.Domain	2
<b>Chatbot: Пишем консольное приложение</b>	<b>2</b>
Reminder.App	2
<b>Домашнее задание</b>	<b>2</b>

## Chatbot: Пишем интерфейсы и вспомогательные классы

Показывая слайд со схемой обратить внимание на связи.

Важно, что домен **не оперирует** конкретными реализациями интерфейсов хранилища, приёмника и рассыльщика напоминаний. Он работает на уровне интерфейсов и только в самом приложении Reminder.App происходит внедрение зависимостей (dependency injection) между конкретными имплементациями интерфейсов.

### Reminder.Receiver.Core

- Вместе пишем код по слайдам и коду конечного приложения.

### Reminder.Sender.Core

- Вместе пишем код по слайдам и коду конечного приложения.

### Reminder.Parsing

- Вместе пишем код по слайдам и коду конечного приложения.

## Chatbot: Регистрация собственного бота в Telegram

- Создаём бота,
- Получаем токен.

## Chatbot: Пишем логику получения и отправки сообщений

### Reminder.Receiver.Telegram

- Вместе пишем код по слайдам и коду конечного приложения.

### Reminder.Sender.Telegram

- Вместе пишем код по слайдам и коду конечного приложения.

## Chatbot: Модифицируем доменную логику

### Reminder.Domain

- Вместе пишем код по коду конечного приложения.

## Chatbot: Пишем консольное приложение

### Reminder.App

- Вместе пишем код по коду конечного приложения.

## Домашнее задание

- Покрыть код юнит-тестами.
- Расширить формат допустимых сообщений для установки новых таймеров