



# C# / .NET

(chatbot: разработка доменной логики)

Артём Трофимушкин

# Взаимосвязи между компонентами

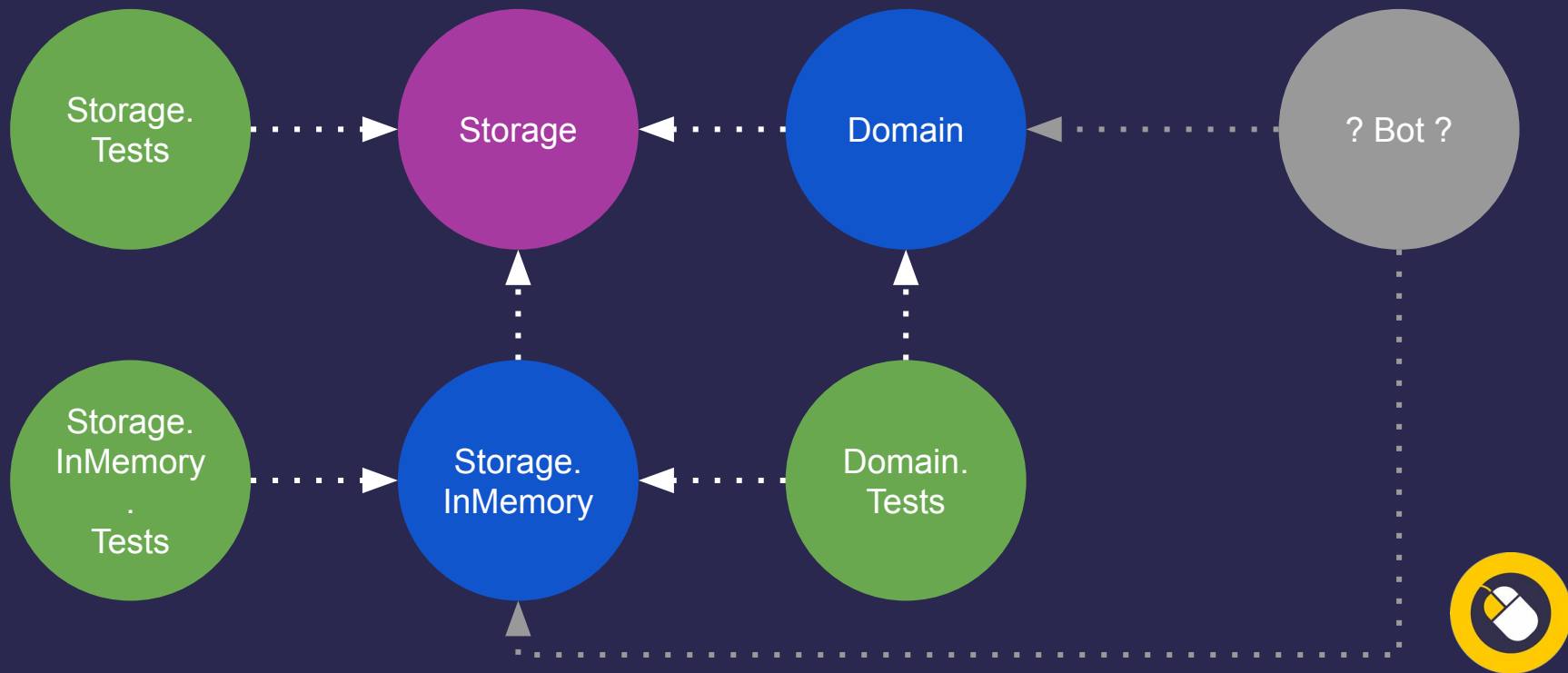
---

Для начала выделим основные сборки, отвечающей за логику работы собственно ремайндера:

- **Reminder.Storage**
  - Библиотека с описанием интерфейсов и классов, которые будут использоваться конкретными реализациями хранилища данных
  - Ей в пару будет создана сборка `Reminder.Storage.Core.Tests` для тестов логики классов, если она будет.
- **Reminder.Storage.InMemory**
  - Библиотека с реализацией хранилища данных в памяти. Реализует все интерфейсы `Reminder.Storage.Core`.
  - Ей в пару будет создана сборка `Reminder.Storage.InMemory.Tests` для тестов логики хранилища.
- **Reminder.Domain**
  - Основная библиотека логики.



# Взаимосвязи между компонентами



# Библиотека `Reminder.Domain`

---

- `Storage.Domain.ReminderService` класс
  - Класс определяет основной доступ к логике работы с напоминаниями.
  - Класс должен принимать в конструктор класс, реализующий интерфейс `IReminderStorage`.
  - Здесь должна быть возможность добавить новое напоминание, посмотреть одно или несколько уже добавленных напоминаний.
  - Здесь должна происходить основная логика по периодической проверке “не пришло ли время вывода какого-либо из напоминаний”.
- `Storage.Domain.Model.XxxReminderModel` классы
  - Классы моделей `ReminderItem` уровня домена — немного разные — каждый для своих нужд:
    - добавление напоминаний,
    - просмотр напоминаний,
    - срабатывание напоминаний.



# Совместная работа в классе

---

Реализация доменной логики ремайндера в библиотеке классов  
Reminder.Domain

Создание библиотеки модульных тестов Reminder.Domain.Tests (NUnit Test Project .NET Core).



# Домашняя работа

---

Покрыть модульными тестами классы сборки `Reminder.Domain`, совместно написанной на уроке.



# Спасибо за внимание.

