# Теоретическая часть

* Понятие об объекте Promise – реализация асинхронного кода в JavaScript
* Создание Promise(), методы resolve() и reject()
* Метод then() и обработка успешного результата операции Promise
* Метод catch() и обработка ошибок при работе Promise
* Методы Promise.all() и Promise.race() для работы с коллекцией промисов
* Реализация AJAX на базе промисов
* Введение в Fetch API – асинхронная реализация AJAX средствами ядра JavaScript
* Асинхронные валидаторы в Angular – реализация на базе промисов
* Асинхронные сервисы в Angualar – реализация на базе промисов

# Практическая часть

Разработайте приложение **Angular** с использованием изученных возможностей Angular. Применяйте формы, валидаторы (синхронные и асинхронные), данные храните в локальном хранилище с использованием имитации таблицы базы данных **InMemoryWebApiModule**. Сервис доступа к «базе данных» выполните асинхронным с использованием промисов.

Разбивка на компоненты и организация их взаимодействия – на Ваше усмотрение. В один из компонентов выводите текст этого задания.

**Задача 1.** Дана коллекция – описание студентов спортивного клуба «Кодокан». Данные хранить в имитации таблицы базы данных **InMemoryWebApiModule**. Объект класса Студент имеет следующие поля: идентификатор, фамилия и инициал, дата рождения, номер телефона, адрес электронной почты, спортивный разряд (по возрастанию: 10й кю, 9й кю, …, 2й кю, 1й кю, 1й дан, 2й дан).

Реализуйте REST-запросы к коллекции (добавление и редактирование сведений о студенте выполняйте в форме. Валидацию выполняйте стандартными валидаторами Angular).

С помощью запросов с методом GET выполняйте следующие обработки:

* Вывести всех студентов всех групп
* Вывести список тренеров
* Вывести студентов, упорядочив по группам, внутри групп – по убыванию разрядов, внутри разряда – по алфавиту
* Вывести студентов выбранного тренера. Ввод параметра запроса в форме, примените валидацию стандартным валидатором Angular: поле обязательно для заполнения
* Вывести студентов с заданной датой рождения. Ввод параметра запроса в форме, примените валидацию стандартным валидатором Angular: поле обязательно для заполнения
* Вывести студентов с заданным диапазоном разрядов. Ввод параметров запроса в форме, примените стандартную валидацию для проверки заполнения и валидацию асинхронными валидатором для проверки правильности диапазона
* Вывести студентов с заданным диапазоном возраста. Ввод параметров запроса в форме, примените стандартную валидацию для проверки заполнения и валидацию асинхронными валидатором для проверки правильности диапазона
* Вывести самых молодых студентов
* Вывести самых старых студентов

# Дополнительно

Запись занятия можно скачать [**по этой ссылке**](https://cloud.mail.ru/public/K62Q/EQBJHXwwj), материалы занятия в прикрепленном архиве.