

Шпаргалка

Топ-50 вопросов

по Spring для собеседований
на Java Junior

О нас

FAANG School – IT онлайн-школа с первоклассными образовательными продуктами, которая выпускает разработчиков с гибким мышлением и мощным стеком.

Наша миссия

Помогаем реализовать амбиции, стать профессиональными разработчиками и получить работу мечты в лучших IT-компаниях в кратчайшие сроки!

Что такое Java Буткемп?

Java Буткемп — это интенсивное обучение в формате командной работы над мощным проектом, где ты получишь максимально приближенный к реальному опыт для успешного трудоустройства с высокой зарплатой в кратчайшие сроки.

Для этого мы адаптировали практики разработки BigTech-компаний, внедрили передовые инструменты индустрии и организовали работу по гибкой методологии Scrum, чтобы ты выделялся среди других кандидатов на рынке.

**Стремишься стать востребованным разработчиком и получить работу мечты в кратчайшие сроки?
Скорее записывайся на Java Буткемп!**

[ЗАПИСАТЬСЯ НА ЭКСКУРСИЮ](#)

Spring — это мощный фреймворк, позволяющий эффективно создавать масштабируемые и легко сопровождаемые приложения на Java.

Здесь собраны **50 ключевых вопросов**, которые помогут систематизировать знания и подготовиться к техническим собеседованиям на Java Junior.

Изучая их, ты получишь общее представление о ключевых модулях и возможностях фреймворка, а также узнашь тонкости его конфигурации, веб-разработки и управления транзакциями.

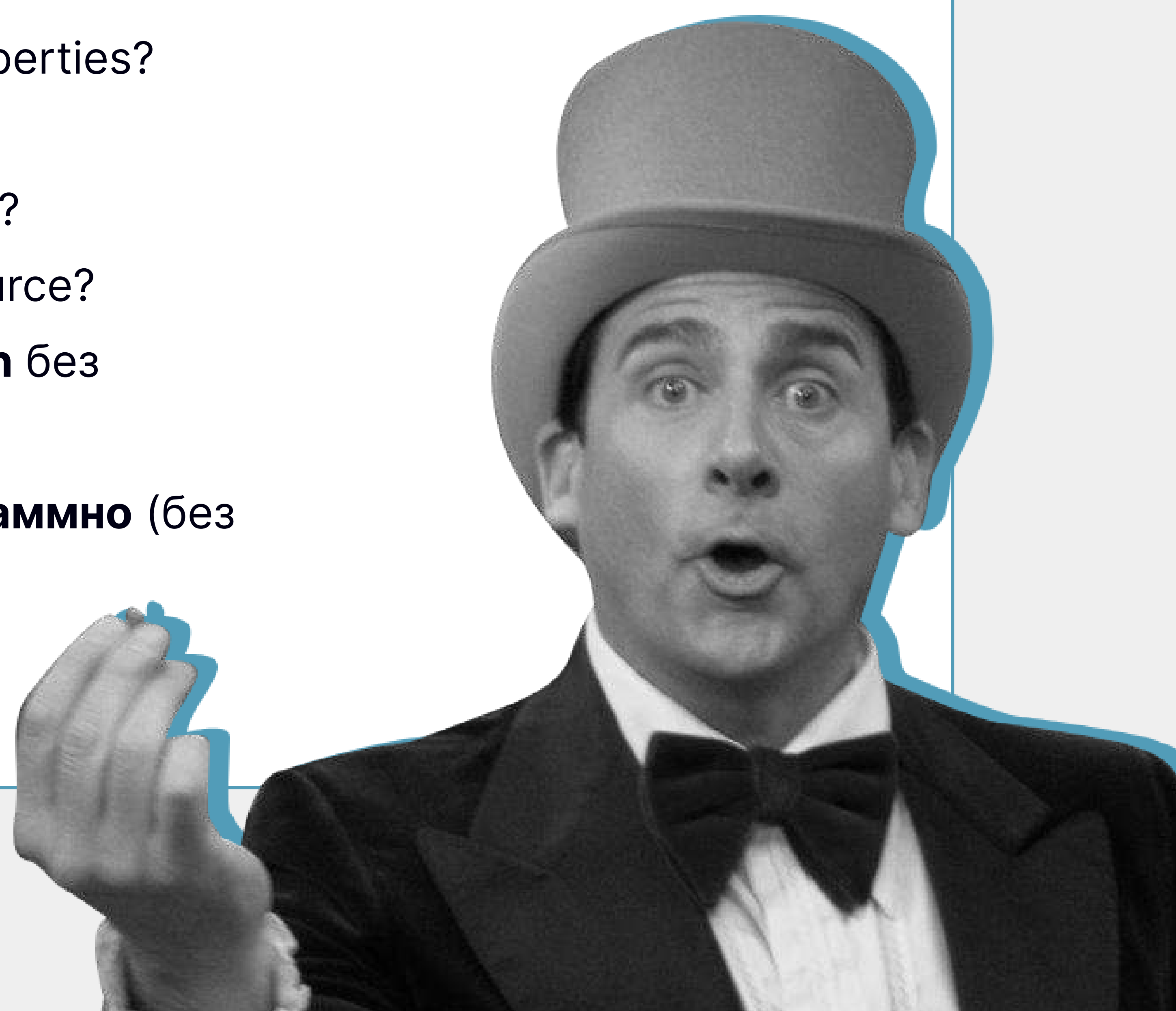
Основы Spring Framework

1. Что такое Spring Framework и зачем он нужен?
2. В чем основная идея **Inversion of Control (IoC)** и как Spring её реализует?
3. Что такое **Dependency Injection (DI)** в Spring?
4. Чем отличается **Field Injection**, **Constructor Injection** и **Setter Injection**?
5. Как создать и управлять Spring Bean-ами?
6. Что такое **Bean Scope** и какие их виды существуют?
7. Как работает аннотация `@Component` и чем она отличается от `@Service` и `@Repository`?
8. Что делает аннотация `@Autowired` и какие есть альтернативы её использованию?
9. Что такое `@Qualifier` и когда его стоит использовать?
10. Как работает `@Primary` и зачем он нужен?

Конфигурация Spring

1. Чем отличается аннотация `@Bean` от `@Component`?
2. Как загрузить конфигурацию в Spring через `application.properties` или `application.yml`?
3. Как использовать `@ConfigurationProperties` для управления настройками приложения?
4. В чем разница между `@Value` и `@ConfigurationProperties`?
5. Как работает **Spring Profile** и зачем он нужен?
6. Как настроить и использовать **Spring Environment**?
7. Чем отличается `@PropertySource` от `@ImportResource`?
8. Как можно создать **пользовательский Spring Bean** без использования `@Component`?
9. Как в Spring можно зарегистрировать Bean **программно** (без аннотаций)?
10. Как включить и использовать `@Lazy` в Spring?

spring



Spring Web (REST, MVC, Контроллеры)

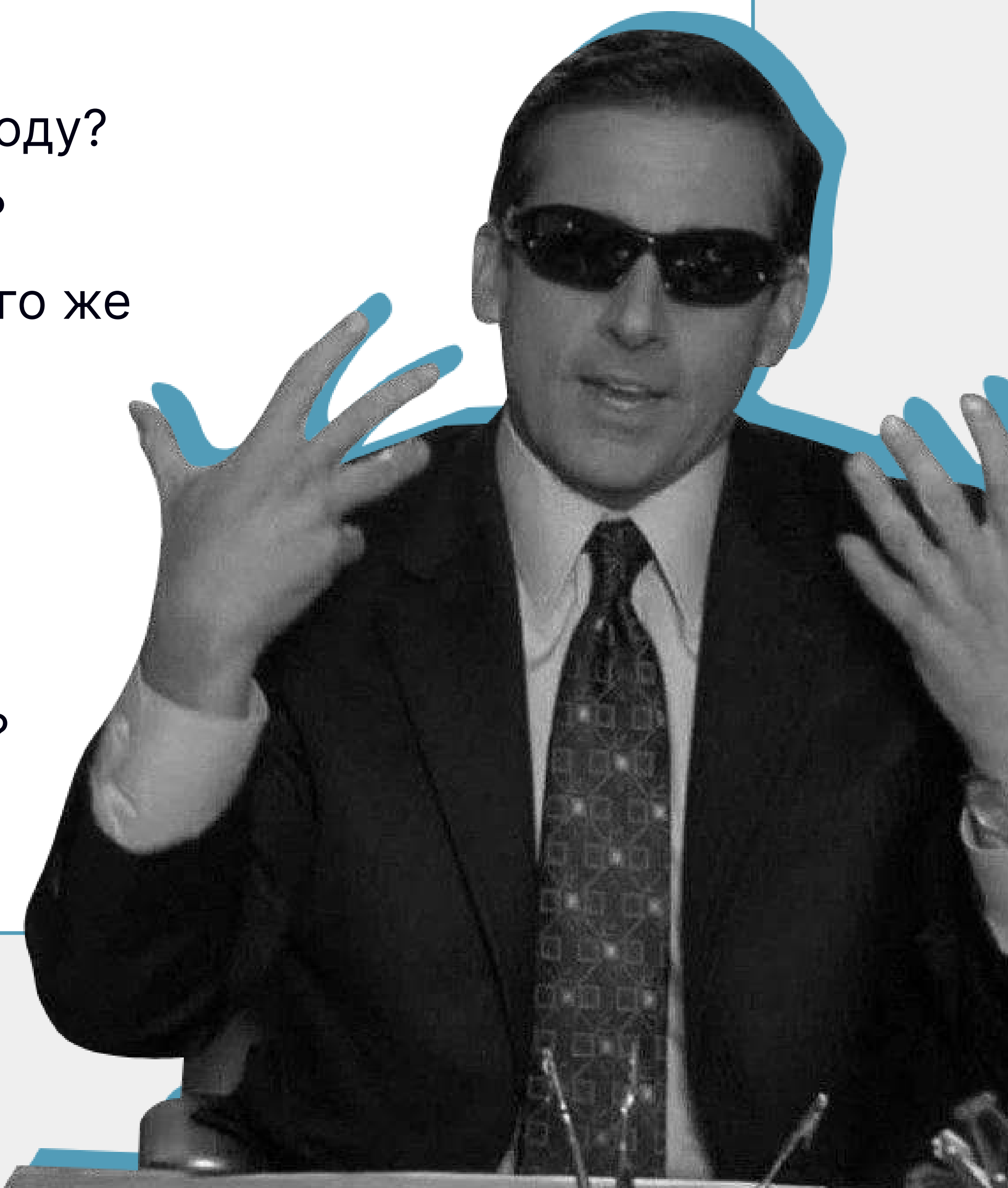
1. Чем отличается @RestController от @Controller?
2. Как работает @RequestMapping и его сокращенные аннотации (@GetMapping, @PostMapping, и т. д.)?
3. В чем разница между @RequestParam и @PathVariable?
4. Как передавать данные в @RequestBody?
5. Как вернуть код ответа HTTP 201 (Created) в Spring?
6. Как работает аннотация @ResponseStatus?
7. Как передавать заголовки HTTP через @RequestHeader?
8. Как обработать параметры запроса через @RequestParam и @RequestBody в одном методе?
9. Как настроить обработку ошибок в REST API через @ExceptionHandler?
10. Как работает ResponseEntity и зачем его использовать?

Spring Boot и его особенности

1. В чем разница между Spring и Spring Boot?
2. Что делает аннотация @SpringBootApplication?
3. Что такое Spring Boot Starter и зачем он нужен?
4. Как изменить порт, на котором работает Spring Boot приложение?
5. Как добавить кастомный баннер в Spring Boot?
6. Как работает CommandLineRunner и ApplicationRunner?
7. Как запустить Spring Boot приложение в виде **JAR** и **WAR**?
8. Как отключить баннер Spring Boot при запуске?
9. Какие существуют способы **логирования** в Spring Boot?
10. Как в Spring Boot сделать поддержку **международализации (i18n)**?

Spring Transactions и управление данными

1. Что такое аннотация @Transactional и зачем она нужна?
2. Чем отличается propagation от isolation в @Transactional?
3. Как в Spring отключить автоматическое управление транзакциями?
4. Чем отличается rollbackFor от noRollbackFor в @Transactional?
5. Что произойдет, если @Transactional применяется к private методу?
6. Можно ли использовать @Transactional внутри @Async метода?
7. Влияет ли @Transactional на вызовы метода внутри одного и того же бина?
8. Что такое **Read-Only Transactions** в Spring?
9. Как работают аннотации @EnableTransactionManagement и @Transactional вместе?
10. Как выполнить транзакцию вручную через TransactionTemplate?





Надеемся, что эта шпаргалка
будет полезна для тебя
и поможет лучше
подготовиться к
собеседованиям!

А если ты хочешь узнать
больше о том, как именно тебе
начать карьеру в IT в 2025,
то переходи по ссылке
и получай **БЕСПЛАТНУЮ**
карьерную консультацию
от команды FAANG School!

[ЗАПИСАТЬСЯ](#)

Подписывайся на нас
в соц.сетях:

 [YouTube](#)

 [VK](#)

