**В этот файл пишем:**

1. **Имя того, кто собеседовал**
2. **Своё имя**
3. **Вопросы**
4. **Составляйте свою историю в отдельности, пожалуйста, чтобы понимать, какой собеседующий, какие вопросы задает.**
5. **Пишите, по возможности, вкратце, что ответили на вопрос.**

Собеседовал Андрей Кротов

Собеседовался Виталик Ковров

1. **Алгоритмы**
   1. Сложность
      1. o(n), O(n), омега
   2. Сортировки, какие когда лучше использовать,худший средний случай
2. **Java8** 
   1. Стримы
   2. Параллельные стримы
      1. Недостатки
      2. Когда можно использовать, а когда не надо?
3. **Коллекции**
   1. Что в map передается?
4. **Многопоточность**
   1. Volatile
5. **Git**
   1. Git flow
   2. Rebase
6. **JAVA EE**
   1. Сервлеты
   2. Куда сохраняются данные
      1. В сервлет контекст
7. **Spring**
   1. Spring Core
      1. IoC
   2. MVC
      1. Структура, как запрос приходит?
      2. Аннотация assign
   3. Boot
   4. DATA
      1. JDBC Template
   5. AOP
   6. Транзакции
8. **Системы сборки**
   1. Maven & Gradle отличия

Собеседовался Сергей Байрамов

1. **Проекты**
   1. Учебный проект, сказал что использовали канбан и прогали на сервлетах, писали свой коннекшн пулл и юзали jdbc.
      1. Спросил про канбан и отличия от аджаил, какие столбцы были и где больше всего карточек копилось в канбане
   2. Текущий проект, сказал что аджаил и парное программирование практикуем, пишем тесты
      1. Спросил основные преимущества парного программирования и знаю ли откуда пришло (экстримальное программирование)
      2. Рассказал про аджаил и митинги
      3. Спросил какие есть возможности слежения за чистотой кода (сонар линт, тдд)
      4. Поговорили про тдд, его преимуществах (про итерационность и красно-зеленое программирование(вроде так))
   3. Тестирование, какие бывают
      1. Юнит тесты в чем преимущество и какие требования к ним предъявляют
      2. Mock (аннотации, моки через *mock(...)*)
2. **Базы данных, а точнее jdbc, hibernate, spring data**
   1. Jdbc, как использовали на учебном проекте, что такое транзакция и есть ли они в jdbc
   2. Hibernate
      1. Про orm, и мапинг моделей на базу данных (entity)
      2. Язык написания запросов к базе данных в виде сигнатур методов
      3. Транзакции
   3. Spring data
      1. Что умеет
      2. Transactional
3. **Spring**
   1. Что такое и зачем нужен
   2. Что за слабое и сильное связывание, почему сильное это плохо
   3. Какие преимущества дает слабое связывание
   4. Как его можно добиться?
   5. IoC
      1. В чем выражается, почему называется инверсией управления?
      2. Бины и их описание в программе, иниты и дестрой, а также scope бинов, какие бывают, какой по-умолчанию
      3. А если добавить Spring MVC то какие скоупы добавятся?
      4. Можно ли написать свой скоуп? (Да)
      5. Офтоп: можете почитать про язык эспрешенов в xml конфигурации
      6. А вот если у нас есть транзакшнал над приватным методом, то она сработает?
         1. Нет
         2. А как сделать чтобы да?
         3. BeanDefinitionReader
         4. Про свои бин лоадеры и в целом процесс создания бинов и их жизненный цикл
   6. Spring MVC
      1. Совсем немного про то что это (диспатчеризация)
   7. Внезапные Servlets
      1. Что такое и как они управляются
      2. А если нам нужно аутентифицировать пользователя? Сессии
      3. А глобально? Статик переменная, но тут вообще имелся в виду servlet context
      4. А вот если мы хотим перед сервлетами что-то делать? Filters
      5. А есть ли подобное в спринге?
   8. AOP (был после транзакций на самом деле)
      1. Что же это такое?
      2. Основные понятия аоп
      3. Какие места подсоединения бывают
      4. Примеры использования помимо логирования (логи, транзакции и авторизация)
4. **Системы сборки и гит**
   1. Чем отличается мавен от грейдла
   2. Гит и как он используется
      1. Отличие ребейз от мерджа
      2. В чем преимущество ребейз от мерджа? Один потомок
      3. А вот у нас есть сотня коммитов и на сотом мы выяснили что прога падает, первый коммит точно рабочий, расскажите как быстро найти рабочий? Git bisect (бинарный поиск по коммитам)
5. **Concurrent**
   1. Как можно запустить поток
   2. Синхронизация, как она работает
   3. Volatile
   4. Экзекуторы
      1. Какие есть
      2. В чем отличия
      3. В чем преимущества
      4. Как синхронизировать
      5. Cyclic barrier
   5. Коллекции конкаррент
      1. Какие бывают, отличия
      2. CopyOnWriteArray
   6. Callable и Future
      1. Что такое и зачем
   7. ComplitableFuture
      1. Я не знал, но предположил, что гет там не блокируется и пляшут от этого
6. **Алгоритмы**
   1. Характеристики (сложность, память, устойчивость)
   2. Примеры алгоритмов
   3. Сложность поиска бинарного и в HashMap без коллизий

Собеседовала Александра Шагал

Собеседовался Слава Поважнюк

1. **Spring**
   1. Спросили, что на проекте использует, что сделал, какие фичи реализовать
      1. Что использовали на проекте и что знаете?
   2. Спринг бут, какие вопросы решает.
   3. Спринг кор
      1. IoC
      2. DI
      3. Модули
      4. Вкратце по ним
      5. Жизненный цикл бина
         1. Стадии
         2. Какой паттерн реализует
         3. Почему синглтон
            1. Почему, есть ли еще
   4. Спринг Дата
      1. Почему хорошо, а JDBC это плохо
      2. Чем репозитории отличается от
      3. Транзакции
   5. Спринг Секьюрити
   6. Спринг АОП
      1. Что такое, что делает?
      2. Прокси паттерн
      3. Джойн кат, джойн поинт.
      4. Типичный пример реализации
         1. В логгерах.
   7. Спринг контекст
2. **Многопоточность**
   1. Синохронайзеры
   2. Волатил
   3. Атомик
   4. Как создать тред
   5. Dead-lock
      1. Синхронайз
      2. Как разрулить
   6. Синхронайзеры
      1. Почему нужно использовать
      2. Какие он знает.
      3. CyclicBarrier
         1. Пример
   7. Проблема многопоточности
      1. Race condition
      2. Starvation
      3. Livelock
      4. Dead-lock
   8. Демоны
3. **Коллекции**
   1. Иерархия
   2. Добавил про Map
      1. Не наследуется от iterable
   3. Коллекции наследуются от Iterable
   4. HashMap
   5. Когда добавляется каждый раз или копируется массив?
      1. Нужно рассказать про капасити
   6. Рассказал, что в мапе тоже есть капасити
   7. Как выглядит хэш-функция в исходнике
   8. Как коллизии решаются в мапе?
      1. После 8 элементов мапа превращается в дерево.
   9. List, Queue, Set
      1. Что это ?
      2. Зачем?
   10. Чем сет отличается от листа
   11. Как возможна ситуация, когда в кей на найти в мапе?
       1. Тогда, когда меняется объект.
4. **Алгоритмы**
   1. Вставка в начало и в конец
      1. Сложность
      2. о(n)
   2. Как систем эррей копи отрабатывает?
5. **Java 8**
   1. Стримы
   2. Хорошо ли, если стрим не финализируется?
      1. Это плохо, потому что это работает….
   3. Future
   4. Optional
   5. Функциональные интерфейсы.
   6. Что такое компаратор, консьюмер, предикат.
6. **SQL**
   1. Сделать запрос Many-to-Many.
      1. Сделал кросс запроc
   2. Найти авторов, где суммарный рейтинг больше 10.
   3. Join
7. **HTTP**
   1. Параметры
   2. Вернет ли что - нибудь, если варибл не передал
8. **Тесты**
   1. Чем отличаются Unit от интеграционных тестов?

Собеседовал Роман Лапин

Собеседовался Мальков Максим

1. Чем scram отличается от kanban ?
2. Опишите рабочий цикл над задачей?
   1. Как вы определяете сложность?
   2. В каких единицах можно определять сложность?
   3. Можно ли мапить story points на время?
3. Опишите процесс code review
   1. На что следует обратить внимание при code review?
   2. Какие Bad practice вы будете искать ?
   3. Как вы будете организовывать code review очень большой таски?
4. Опишите git flow
   1. В чем основные бонусы этого подхода ?
   2. Что такое Hot Fix ?
   3. Какие другие подходы к организации git flow вы знаете ?
5. Какие типы вложенных классов вы знаете?
   1. Когда использовать Nested class?
   2. Когда использовать Static Nested class ?
6. Что такое сигнатура метода ?
   1. Можно ли её менять при наследовании?
   2. Даны методы method(Object)  
       method(Integer)  
       method(String)  
      Что произойдет при вызове method(null)?
7. Представим у вас есть main\_thread который создает и запускает thread\_one, в результате своей работы thread\_one кидает checked exception, что при этом произойдет ?
   1. Как получить брошенный thread\_one’ом Exception ?
   2. Поможет ли Future ?
   3. Как сделать неблокирующим способом? (Имелась в виду completablefuture)
8. Какие типы Error вы знаете ?
   1. Как вызвать OutOfMemoryError ?
9. Какие механизмы сборки мусора вы знаете ?
   1. Что происходит при Stop the World?
   2. Чем отличается работа других gc от serializable ?
   3. Как отмечаются объекты до которых уже не добраться ?
10. Какие типы паттернов вы знаете?
    1. Когда использовать Adapter?
    2. Когда использовать Visitor?
11. Liskov substitution principle как его нарушить ?
12. Какие ограничения накладываются на переменные передаваемые в лямбда-выражения?
13. Дана HashMap<byte[],String>() чем она плоха?
    1. Как будет вычисляться hashCode ?
    2. Как её исправить?
    3. Что будет если изменить ключи у объектов в контейнере?
14. Какие ограничения накладываются на элементы помещаемые в TreeMap?
    1. Можно ли положить туда элементы не реализующие Comparable ?
    2. Какова сложность удаления элемента из TreeMap?
15. Где в Spring используется паттерн Chain of Responsibility ?
16. Для чего можно применить AOP?
17. Чем плоха технология MyBatis ?
18. Практическое задание: Найти max элемент linkedList
    1. Усложнение max определять по формуле max = val \* (n -i)  
       Где n - размер связного списка, он неизвестен  
       i - индекс ноды в связном списке
    2. Сделать за один проход  
       Интервьюер сказал, что сам не знает возможно ли это.

Собеседовала Ксения Алексеенко

Собеседовался Василий Толкачев

**1. Spring**

a. Если у интерфейса стоит @Autowire и несколько классов реализуют  
интерфейс что будет

b. Чем RestController отличается от обычного

c. Если класс имплементирует интерфейс и дополнительно содержит какие-то методы, если создать прокси по этому классу, попадут эти методы в интерфейс или нет

d. AOP

2. **Многопоточность**

a. В каком случае плоха CuncurrentHahMap

b. ComplitableFuture

c. Что с эксепшенами в методе run происходит

d. Вообще разницу между потокобезопасными мапами

**3. Коллекции**

a.LinkedHashMap – когда используется

b.Что будет если писать в CuncurrentHashMap во время  
итерации по ней

c. Как поведет себя TreeMap если в качестве ключа null передать

**4. Java 8**

a. Написать immutable класс с одним полем List<Date>

**5. SQL**   
Сделать запрос с join

**6. Тесты**

a. Про моки говорил

b. Про интеграционные и unit тесты

7. **Hibernate** (Транзакции, propagation)

8. **Git** как мы сквошим на проекте, как конфликты разрешать, merge rebase и тд

9. **Сервлеты**, про **DispatcherServlet,** как создать сервлет

10. Кто быстрее PreparedStatement или обычный и за счёт чего

Собеседовала Софья Жук

Собеседовался Продан Иван

1. **Flow в Java**
   1. Вопрос звучал именно так. “Какой flow в Java?”
   2. Видимо, ожидается ответ про компиляцию в байткод - загрузку с помощью класслоадеров - стек и хип - очистку мусора. Но это не точно.
2. **Что такое javac? Где дефолтно хранятся скомпилированные классы? Из чего складывается название скомп-го класса и как выглядит?**
3. **Java Collections**
   1. Какие интерфейсы наследуются от Collection?
   2. Является ли Collection наиболее высоким интерфейсом в иерархии коллекций? (нет, наследуется от Iterable)
   3. Что, если у нас есть TreeSet, в который мы закидываем Integer’ы по возрастанию?
   4. Есть TreeSet. Ты удаляешь из него элемент и видишь, что он не удалился. Твои действия?
4. **Spring. Spring core. Основные модули. Что решает.** 
   1. Что такое bean
   2. Отличие BeanFactory от ApplicationContext
5. **Aop**
   1. Что такое, зачем нужен
   2. Где дефолтно используется в спринге?
6. **Spring MVC. Что привносит, как работает.**
7. **Servlets & JSP** - тоже очень общий разговор с упоминанием контекста, фильтров, листенеров
   1. Life-cycle servlet’а
8. **Отличия м/у Java JDBC, Spring JDBC, Spring Data** (без особых подробностей, просто общее понимание какие проблемы решаются и как в целом выглядит работа с ними).
9. **Какие классы используются при работе с Java JDBC**
10. **SQL vs NoSQL.**
    1. Почему бы не выбросить SQL раз NoSQL так хорош?
    2. ACID
    3. Написать какой-нибудь JOIN
11. **Java 8**
    1. Что в ней появилось нового
    2. Являются ли Optional панацеей от null? (нет, потому что некоторые одаренные люди не вызывают isPresent())
12. **Java.util.concurrency**
    1. Что принесла с собой
    2. Как упрощает работу
    3. Пример когда с последовательным захватом мониторов двух объектов (потенциальный deadlock)
    4. Как улучшить упомянутый выше код
    5. Что такое race condition
    6. Как остановить Thread. Почему stop() - @Depracted
    7. Какие задачи за нас решают инструменты из java.util.concurrency по работе с потоки
    8. Когда стоит использовать ExecutorService’ы
13. **Тестирование**
    1. Какие библиотеки существуют?
    2. Зачем вообще нужно?
    3. Какие объекты стоит/не стоит мокать?
    4. Когда не стоит использовать тестирование?
    5. Unit test/Integration test
14. **Agile vs Waterfall**
    1. Что такое waterfall?
    2. Что такое agile?
    3. Почему agile лучше?
    4. Что происходит во время спринта?
    5. Что такое ретроспектива?
    6. В чем суть гибкости? А что если заказчик посреди спринта очень захочет впилить новую фичу?
15. **Задача - есть строка на вход, определить является ли она палиндром (с обоих концов читается одинаково).**

Собеседовала Анна Осипова

Собеседовалась Мария Боровикова

**1. Как работаете на текущем на проекте (процессы)?**

* 1. Какие митинги включает в себя Scrum
  2. Отличие scrum от kanban
  3. Как измеряется скорость команды в scrum и kanban

**2.** **Какую систему сборки используешь на проекте? Для чего они нужны?**

1. Как работает gradle
2. Какие системы сборки еще знаешь?
3. Отличие gradle от maven.

**3. Какие этапы разработки у вас? Как изменения перетекают в проект (намек на git flow)**

**4. Что знаешь про гит?**

1. Что содержит коммит
2. Что происходит при команде merge
3. Отличия merge от rebase
4. Что делают revert, reset и amend
5. Какие виды reset есть (hard, soft и mixed)

**5. Расскажи за ООП**

1. Полиморфизм, инкапсуляция наследование. Как реализуются в джаве

**6. Java Core**

1. Equals, hashcode (для чего, как связаны)
2. Методы класса Object
3. Модификаторы доступа
4. Какие-нибудь ключевые слова в java, за что отвечают (рассказала про static, final, default)
5. Для чего нужны finally и finalize
6. Как работает gc

**7. Коллекции**

1. Иерархию коллекций нарисовать
2. Есть TreeSet и HashSet, как организованы, чем отличаются
3. Что такое Map, реализации, как внутри устроена
4. (Вопрос странненький) Чем Stream отличается от коллекции?   
   Я сказала, что коллекция - это хранилище некоторых однотипных элементов, а Stream это некоторый поток, который нам поставляет элементы.   
   Интервьюер помахала положительно головой…

**8**. **SQL**

1. Есть таблица EMPLOYEE и SALARY, SALARY содержит ссылку на EMPLOYEE, написать запрос для вывода сотрудника с наибольшей з/п

**9. Spring Framework**

1. Что это вообще и зачем
2. Что такое IoC и как оно реализовано в Spring
3. Ж/ц бина
4. Какие у бина бывают scope и какой по умолчанию
5. (!!!) Если у приложения два контейнера, то бин со скоупом синглтон будет единственным? Ответ: в рамках одного контейнера - да, а в рамках приложения - нет, поэтому для таких случаев есть scope “application”.
6. Что в описании бина есть?
7. Когда выполняется метод destroy и всегда ли он выполняется обязательно?
8. Как бин подтягивается куда-нибудь
9. Какие конфигурации есть
10. Какие аннотации есть для создания бинов

**10. Spring JDBC**

1. Какой тут основной класс, что в себе хранит (про JDBC Template рассказать)
2. Типы Statement’ов
3. Зачем нужен PreparedStatement, если есть просто Statement (чтобы не было sql-инъекций)
4. Что такое SQL-инъекция

**11. Spring AOP**

1. По всем терминам пройтись (Aspect, join point, point cut, advice)
2. Виды advice’ов
3. Как реализуется, то есть как создаются прокси
4. (!!!)Можно ли у private метода поставить аннотацию @Transactional? Как в данном случае создастся прокся? Ответ: прокси делаются либо через dynamic proxy, если класс реализует интерфейс (тогда прокси имплементит этот интерфейс) и через CGLIB, если класс не реализует интерфейс (тогда он делает наследника этого класса). И в том и в другом случае private не попадет в прокси, т.к. В интерфейсе private методы не разрешены, а при наследовании private метод не виден наследнику. Как-то так...

**12. Multithreading**

1. Чем поток отличается от процесса
2. Какие бывают состояния у потока
3. Рассказать про wait/notify
4. Блоки sinchronized
5. Синхронизированные коллекции, чем HashMap отличается от ConcurrentHashMap
6. Рассказать все, че помню про ExecutorService и его наследников, как работает Executors
7. Описать принцип работы ForkJoinPool’а
8. Что такое Future и зачем она нужна

**13. Java 8**

1. Отличие map от flatMap
2. Написать программу:  
   Создать два класса: 1) Автор (поля: имя, возраст), 2)Книга (поля: список с

авторами, заглавие и количество страниц).

В третьем классе (Main) создать две коллекции (Авторов и Книг) и

провернуть с ними несколько действий на стримах:

I) Вернуть список книг, где книга больше 200 страниц

II) Найти самую большую и самую маленькую книгу

III) Вывести книги, у которых только по одному автору

IV) Отсортировать список книг по заглавию

V) Вывести самую большую книгу определенного автора

**14. Тестирование**

1. Как проходит тестирование на проекте
2. Что такое unit тестирование, что такое интеграционное
3. Что такое Mock, рассказать про все это дело

**Собеседовал: Николай Карулин.**

**Собеседовался: Виталий Семков**

В целом интервьюер крайне приятный. Есть полное ощущение, что следующее собеседование с ним пройдёт иначе для другого кандидата, что, собственно, он и подтвердил после беседы. Он видит тему, в которой ты понимаешь и уходит от неё, но дольше общается о темах, в которых ты плаваешь. Попросил продумать архитектуру приложения “магазин игр” и полностью спрашивал по всему стеку. На комментарии про то, что мы что-то не проходили реагирует адекватно.

Agile: отличие от водопада, спринты и все около того.

Gitflow: как бы ты выстроил на своем проекте с обоснованием

ООП: Все концепции с примерами и пояснение модификаторов.

Collections: Рассказал в общем про интерфейсы - кто за что, глубоко по map.

Алгоритмы: хорошо ответил про устойчивость. Вспомнил про Радикс, деревья.

Concurrency: всё по чуть-чуть, syncrhonized, race condition, semaphor, monitor, executor, future, CAS.

Spring: про бин, конфиги, core, autowire, boot не спрашивал, на предложение написать наше приложение на boot, предложил собрать в WAR вместо boot.

Data: написали Entity и Repository, по всей Data пообщались, (Евгений Борисов “Да, Та!” должно помочь).

AOP: в общих чертах

MVC и Spring MVC: подзавалился я на контроллерах основательно, надо уметь писать.

Servlets: не очень глубоко, но понимать надо. Контейнер, жизненный цикл.

Паттерны и SOLID: в общих чертах пообщались.

**Собеседовал: Mariia Zueva.**

**Собеседовался: Evgenii Musin.**

В целом интервью прошло в довольно нормальной обстановке. Концепция строилась на работе с кодом (открывается идея, пишется код, спрашиваются вопросы по нему). Есть практические задания из простых, типа КодингБат.

1 JAVA CORE

Собеседование было завязано на этом, потрачено полтора часа.

1.1 Наследование, инкапсуляция, полиморфизм.

1.2 Что можно менять при переопределении метода

1.3 Принцип подстановки Барбары ~~Стрейзн~~ Лисков.

1.4 Методы класса object

1.5 Зачем нужен интерфейс клонабл, если все равно везде есть метод клон

1.6 Equals & Hashcode Зачем их переопределять, почему это важно для коллекций

1.7 Мутабельность\иммутабельность и коллекции

1.8 Автобоксинг, анбоксинг, восходящие и нисходящие преобразования

1.9 Называть экзотические интерфейсы в коллекциях

1.10 Методы TreeSet, отличающиеся от сета

1.11 Как внутри устроен HashMap

1.12 Алгоритмические сложности добавления, удаления элементов и поиска в коллекции

1.13 Какие сортировки в стандартной библиотеке вы знаете

1.14 Стабильные\нестабильные сортировки

2 Spring и компания

2.1 Воркфлоу веб приложения развернутого на сервере со спринг (вылетает реквест от клиента и что дальше)

\что может вернуть handlermapper

\@ResponceBody - что такое и зачем

\что можно передать как параметр в контроллер (кроме реквест и респонс)

\какие есть аннотации для инъекции бина

\способы конфигурации спринга

\транзакшонал - как работает (паттерн)

\аоп - что такое как работает

2.2 Аннотации хибернейта (в том числе мэни ту мэни и все такое)

2.3 Спрингбут - что такое, зачем нужен

2.4 Тестирование (юнит, моки)

3 java8 (optional - что такое, зачем нужен, как создать, терминальные, нетерминальные операции, фьючи)

Собеседовала Мария Зуева

Собеседовался Роман Колганов

1. Core
   1. Таблица переопределения методов
   2. Модификаторы доступа
   3. GC и типы
   4. Classloaders
   5. Как можно finally прекратить?
      1. Уронить JVM
      2. System.exit
   6. Ранее и позднее связывание
      1. Дженерики - поздний полиморфизм
   7. Super и extends
      1. PECS
      2. Почему мы не можем добавить, когда делаем extends?
         1. Скастовать нельзя - уточнить
   8. Зачем нужны checked Exception.
      1. Какой критерий, что программа поведет себя некорректно.
      2. Если есть внешняя зависимость, которую ты не можешь контролировать, то ты
      3. К примеру, есть файл, который ты не можешь перехватить
2. Collection
   1. Иерархии
   2. Нестандартные коллекции
      1. Что ты знаешь, кроме list, set, map
   3. Hashset, treeset - устройство
   4. Какие сортировки где используются?
   5. Treeset
      1. Comparable
      2. Comparator
   6. Сложность добавления, удаления в коллекциях.
3. Многопоточное программирование
   1. Вайт нотивай
   2. Синхронайз
   3. Дано 1000 файлов, надо обработать - комплетабл фьюча, так как он не блокирующая, ты ей выстраиваешь чейн.
   4. Циклбайрьер
   5. Форк джойн пул что под капотом
   6. Валатайл
   7. Хэппес бефор
4. Спринг
   1. IoC - зачем нужен
      1. DI
   2. Способы создать контекст 4 способа
      1. Конфиг xml
      2. Аннотация
      3. Java
      4. Груви
      5. Котлин
   3. Этапы жизни бина
   4. Что храниться в контейнере?
      1. Бин дефенитион ридер
      2. Бин пост процессов
      3. Вызовется ли дестрой метод у прототайп?
   5. Scope бинов
      1. Аппликейшн еще есть
   6. Чем плохая field injection бинов?
   7. АОП весь рассказал
   8. Транзакция по умолчанию
      1. Require
   9. Транзакция над приват методом.
   10. Dynamic proxy и cglib
   11. Data
       1. Как создается entity
       2. У тебя есть лист и нет entity для листа, как будет проинициализирована? Spring data entity list
       3. Борисов spring data
       4. Чем плоха fetch lazy, когда закрывается контекст, то она не факт, что отработает, если мы закрыли контест
   12. MVC
       1. У тебя контроллер отдает view и model, что делается дальше?
          1. Как формируется view? Класс html чего-то там.
       2. Контроллер все хавает.
       3. Какие способы передать параметры параметры к контроллер
          1. Requestbody
       4. Какие еще параметры есть у URL кроме реквест маппинг?
          1. Хидер
          2. Урл
          3. Контент тайп
          4. Метод
5. Git
   1. Отличия мердж и ребейса?
   2. Как решаются мердж конфликты при мердж и ребейз?
6. Тестирование
   1. Интеграционное и юнит
   2. Какие мы пишем тесты?
   3. Мокито - общие концепции
7. Java8
   1. Optional
      1. Куда не рекомендуемся
         1. в поля не надо
         2. Консьюмер
         3. В параметр метода не надо ставить
   2. Стримы
8. Алгоритмы
   1. Назвать что-то, кроме красно-черного дерева
      1. АВЛ.
   2. Сложность
9. Задача
   1. Даны пути к файлам, нужно сделать лист путей в виде string. Сделать мапу string к long, где стринг тот же путь, а long длина файла.

**Собеседовал: Ксения Антоненко, Ирина Васильева**

**Собеседовался: Рамиль Галимов**

1. Рассказать про проект. Какие задачи выполнял

2. Как процесс работы организован, какие особенности у Канбана, в чем отличие Канбан от Скрама, как производительность команды оценивается в скраме.

3. Git. Как на проекте используется git (какие ветки используем), как это делается в git-flow. Как отменить коммит, который запушили/ не запушили.

4. Какой энвайрмент на проекте, куда дженкинс деплоит, откуда берет билд.

5. Какой билд тул используем на проекте. В чем отличия Maven от Gradle

6. Тестирование. Написать тест, который тестирует метод из класса А. В классе А есть зависимость от класса В, тестируемый метод из класса А вызывает метод из класса В. Реализовать с помощью моков. Сделал без аннотаций, потом попросили тоже самое, но с аннотациями.

7. Коллекции. Как устроена хэш-мапа, мутабельный ключ, как переопределять equals и hashcode. Как хэш функция вычисляется, можно ли просто указать что хэш-функция выдает всегда 42 и что будет. Различия hashmap, syncrhonizedMap и concurrentHashMap. Как решаются коллизии в мапе, какое изменение мапы появилось в java 8. Рассказать про loadFactor и initialCapacity, как происходит расширение мапы. Как сохранить в мапу оптимальным образом 1 млн. элементов, если заранее известно количество элементов.

8. Написать иммутабельный класс у которого два интовых поля и одно поле типа Data. Рассказать про конструктор копирования. Как правильно организовать геттер для даты. Рассказать как правильно организовать глубокое клонирование.

9. Java8. Написать стрим который фильтрует коллекцию по двум заданным условиям одновременно и возвращает новую коллекцию.

10. Многопоточность. Проблему visibility и race condition как решить. Рассказть про volatile, synchronized, atomic, CAS-операции(как работают). В каких случаях применяется блокирующая и неблокирующая синхронизация. Что такое lock, где используется. Что произойдет если по HashMap во время итеррации добавлять/удалять элементы, то же самое для syncrhonizedMap, как решить эту проблему.

11. Spring. IoC и DI. Как реализуется проксирование бинов. Какие классы можно сделать бинами. Как работает Dynamic Proxy и CGLib. Основные аннотации для бинов в чем отличие. Что делает @RestController. Скоуп бинов. Как заинжектить прототайп в синглтон.

12. Аннотация @Transactional как работает, куда можно ставить. Как будет реализовано прокси если над методом а() класса А (имплементит Comparable), поставить @Transactional.

13. Spring AOP. Рассказать что такое, основные термины, виды адвайсов. Для каких ситуаций используется Arround адвайс.

13. HTTP. Отличие методов. Что можно в гет и пост передавать.

14. SQL. Написать join для двух таблиц с заданным условием.

15. Hibernate. Что такое, для чего используется. Как организовать bidirectional one-to-many связь между двумя энтити, какие аннотации использовать. Для чего нужна аннотация @Id, виды GeneratedType, виды fetching. В какой ситуации лучше использовать JDBC, а не Hibernate. Недостатки JDBC.

16. Паттерны. Какие применял. Рассказать про фабрику, декоратор, адаптер и прокси. В чем отличие декоратора от прокси.

17. Сортировки. Какие виды знаю, что такое устойчивость сортировки, виды устойчивых сортировок. Возможная проблема у квик сорта. Как оптимально отсортировать 1 млн строк (radix sort).

Собеседовали Юлия Комышова и Ирина Васильева

Собеседовался Сергей Сарваров

1. **Рассказать про проект** 
   1. О чем проект?
   2. Что делал на проекте?
2. **Методология программирования** 
   1. По какой методологии работаете на проекте?
   2. Какие роли на проекте? Есть ли scrum master?
   3. Sprint’ы на проекте, сколько длятся?
   4. Спросили, что происходит в конце спринта?
      1. Презентации и Ретроспектива
      2. Препланинг и планинг на следующий спринт?
   5. Отличия Waterfall, Scrum и Kanban
3. **Система контроля версий**
   1. Как устроена на проекте проверка кода?
      1. Рассказал про то, что делаем мердж реквест, лайки собираем.
   2. Какую систему контроля версий используете?
   3. Рассказать про git flow на проекте
   4. Сквошите ли коммиты? В чем заключается сквош?
   5. Чем отличается merge от rebase?
   6. Зачем нужен git reset?
   7. В какую ветку мерджатся изменения?
   8. Push, fetch, pull
4. **Системы сборки**
   1. Какую используете на проекте? С какими работал?
   2. Чем отличается maven от gradle?
      1. Один на XML, другой на groovy
   3. Рассказать про scope
      1. Compile
      2. Provided
      3. Runtime
      4. И т.п.
   4. Зачем нужны плагины?
   5. Есть ли зависимости, которые загружаются по умолчанию? Или непосредственно все надо прописывать в конфиге?
      1. Ответил, что есть то, что подгружается по умолчанию.
   6. Рассказать про формирование JAR. Когда он формируется, что для этого надо сделать?
      1. package
   7. Репозитории Gradle
5. **Тестирование**
   1. Какую систему тестирование на проекте применяете?
   2. Junit
      1. Сходу рассказал про assertEquals, assertTrue, assertFalse, assertThrow
      2. Спросили потом про аннотации @Before, @BeforeClass
   3. TDD - чем хорош и что такое?
   4. BDD - зачем и для чего?
   5. Что такое mock’и?
   6. Что такое spy?
6. **ООП**
   1. Какие парадигмы программирования знаешь?
      1. ООП
      2. Декларативное
      3. И подобные
   2. Рассказать про принципы ООП
   3. Как в Java выражается тот или иной принцип ООП?
7. **JavaCore**
   1. Поговорили про модификаторы доступа в контексте инкапсуляции
   2. Интерфейсы
      1. Какие модификаторы доступа у полей интерфейса по умолчанию?
      2. А у методов?
      3. Отличия между интерфейсом и абстрактным классом.
      4. Какие модификаторы доступа можно ставить дефолтным методам?
      5. Множественное наследование у интерфейсов
      6. Ромбовидное наследование
   3. Чем отличаются мутабельность и иммутабельность?
   4. Класс Object
      1. Перечислить все методы
      2. Какой метод за что отвечает?
      3. Рассказал про соглашения equals и hashCode
      4. Для чего они применяются?
   5. Как организована память в Java?
      1. Рассказал про stack и heap
      2. Рассказал про permgen, oldgen, younggen (eden, survivior),
      3. Что такое GC? Как он подчищает память?
      4. Какие GC бывают?
   6. Параметризованные классы
      1. Wildcard - T extends Number
   7. JDBC
      1. Что такое DAO?
      2. Отличия statement от preparedStatrement.
         1. Как мы получаем результат? - через resultset
      3. Может ли мы менять БД?
         1. Да, заменяя драйвер
      4. Какой класс управляет драйвером?
8. **Многопоточное программирование**
   1. Зачем нужно, какие проблемы решает?
   2. Чем отличается процесс от потока?
   3. Как можно создать поток?
      1. Рассказал про runnable и thread
      2. Сказал, что надо юзать start() вместо run() - не запустит поток
   4. У нас есть два потока, пытаются что-то делать с одной переменной. Что произойдет?
      1. Получится неконсистентное значение
      2. Нужно использовать volatile. Рассказал, что это.
      3. Рассказал про кеши L1, L2, L3
   5. А что будет, если два потока работают с одним методом?
      1. Рассказал про синхронизацию.
      2. Про мониторы, что это такое? Какие применяются, что в них пишется ссылка на поток
   6. Чем синхронизация методов отличается от синхронизации на блоках? В чем суть?
      1. Sync mehod() почти эквивалентент method { Sunc (this) { .. } }
   7. Как синхронизировать статический метод?
      1. Sync (nameOfClass.class() { .. }
   8. Рассказать про executors
      1. Какие бывают?
         1. Рассказал про всякие singleThreadPool, FixedThreadPool и пр.
      2. Можно ли в executor запихнуть несколько тасок?
         1. Можно. В тот же cachedThreadPool
9. **Spring**
   1. Spring Core
      1. Что такое IoC
      2. Как реализован IoC в спринг? - через DI
      3. Жизненный цикл бина - Борисов
         1. XmlBeanDefinitionReader, BeanDefinition и вот это вот всё ;)
   2. Spring AOP
      1. Рассказать, что это такое
      2. Joinpoint, Pointcut, Advice, aspect
      3. Какие аннотации применяются, если аннотировать, а не xml конфиг писать?
      4. Proxy-объект, упомянул про паттерн proxy
   3. Spring MVC
      1. Как работает? На каком паттерне строится?
      2. Handler mapping, controller's, View resolver, view
      3. Рассказал сам про аннотации @PathVariable, @RequestParam
   4. Rest
      1. Зачем нужно и что такое?
      2. Требования rest.
   5. SOAP
      1. Какой формат сразу вспоминаем, когда мы говорим о SOAP? - xml
10. **Веб-сервисы**
    1. Что такое HTTP?
    2. Какие методы HTTP знаешь?
       1. Сказал про get post
       2. Подсказали про остальные, напомнив про CRUD
    3. По какому порту работает HTTP? - я про это говорил, не помню, спрашивали или нет
    4. Какие ответы могут приходить по HTTP
       1. Все вот эти 1xx, 2xx, 3xx, 4xx, 5xx
    5. О чем говорит 403 ошибка?
       1. Доступ запрещен
11. **SQL**
    1. У нас есть две таблицы, в одной id страны и ее имя, в другой id страны и количество запусков спутников. Вывести страну, у которой не было запусков