Что такое Java Enterprise Edition?

Большинство программистов, начинающих программировать на Java, очень скоро задаются этим вопросом. Найти ответ на этот вопрос можно, например, через Google: <http://ru.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition> Альтернативный ответ содержится в списке тем для собеседования по Java. Это техническая шкала, по которой вас будут оценивать для работы Java-программистом.

* Методологии разработки программного обеспечения. Основные подходы (Waterfall, Agile, [RUP](http://ru.wikipedia.org/wiki/Rational_Unified_Process)), жизненные циклы. Основные отличия.
* UML. Основные виды диаграм. Принципы их построения.
* Java IDE
  + Eclipse
  + IDEA
  + NetBeans
* Системы контроля версий. CVS, SVN, Mercurial. Зачем нужны, как работать.
* Системы сборки (Ant, Maven).
* [Javadoc](http://docs.oracle.com/javase/7/docs/technotes/tools/windows/javadoc.html) [Дескрипторы Javadoc.](http://ru.wikipedia.org/wiki/Javadoc)
* Отладка кода. Debug. Логирование. Система логирования Log4j. Стандартные аппендеры. Layout
* Типы данных. Примитивы, обертки. Класс Object. Основные методы.
* JVM. Устройство памяти (стек, куча). Garbage Collector.
* Интерфейсы, абстрактные классы. Принципиальные отличия. Конструкторы. Блоки инициализации. Внутренние классы. Анонимные классы.
* Модификаторы доступа, static, final.
* Collections, устройство коллекций, примеры использования для решения задач
  + Интерфейсы Collection, Map, List, Set
  + Основные реализации. Их отличия
  + Итераторы
* Шаблоны в Java (generics).
* Сериализация объектов. Где применяется и зачем.
* Java multi-threading:
  + JMM: взаимодействие потоков
  + Блокировки
  + Средства в Java: java.util.concurrent, atomic types
  + Обработка исключений:
  + Ловля исключения. try-catch-finally
  + Виды исключений (Throwable, Error, Exception). Стандартные исключения Java (RuntimeException etc.)
  + [Антипаттерны обработки ошибок](https://dev64.wordpress.com/2012/10/07/exception-handling-antipatterns/)
* Реляционные базы данных
  + Реляционная модель (отношения, кортежи, атрибуты)
  + Нормализация и нормальные формы
  + 12 правил Кодда
  + Ограничения целостности
  + Индексы
  + Хранимые процедуры и функции
  + Триггеры
* SQL
  + SELECT
  + JOIN: разновидности, особенности
  + INSERT, UPDATE, DELETE, CREATE, MERGE, ALTER
* Advanced

GoF Design Patterns (уметь нарисовать с помощью UML, объяснять основные отличия похожих)

* + - Adapter / Wrapper
    - Decorator
    - Proxy
    - Abstract factory. Factory method
    - Chain of responsibility
    - Iterator
    - Observer
    - Strategy
    - Template method
  + Другие шаблоны: MVC, DAO, [Dependency Injection](https://dev64.wordpress.com/2013/05/14/dependency-inversion/), Lazy initialization, Pool. Антипаттерны.
  + Сильная и слабая связанности кода. Когда и зачем нужна слабая связанность.
  + [Unit testing + junit](https://dev64.wordpress.com/2013/04/18/unit-testing/). TDD.
  + Continuous Integration (Jenkins, TeamCity). Что такое, зачем нужно.
  + PL/SQL
    - Синтаксис SQL, специфичный для Oracle
    - Программные блоки в PL/SQL
    - Особенности написания package в Oracle
    - Хранение данных (сегменты, блоки)
    - Планы выполнения
  + [JDBC. Иерархия классов](https://dev64.wordpress.com/2012/04/18/jdbc-specification-review/), пути получения Connection, способ выполнения запроса в базу. Обработка результатов запроса.
  + Web-services:
    - REST
    - SOAP
  + [XML](https://dev64.wordpress.com/xml/)
    - XSLT
    - [XPath](https://dev64.wordpress.com/2012/04/18/xpath-tutorial/)
    - [SAX](https://dev64.wordpress.com/2012/04/28/learning-sax/)/[DOM](https://dev64.wordpress.com/java/#DOM)
  + JavaScript, JSON.
  + Application Servers. Общее представление, устройство. Разновидности. Weblogic AS.
  + Понятие ORM, frameworks:
    - Hibernate, JDO, JPA
    - EJB
  + Enterprise Java Beans:
    - Разновидности
    - Жизненные циклы бинов, пул бинов
    - BMP / CMP
    - Дескрипторы
    - Интерфейсы бинов (home & remote)
  + JNDI.
  + Транзакции
    - Основные понятия, когда и зачем применяется
    - Транзакции в DB
    - JTA (user transactions, interface synchronization)
    - ACID
  + [Servlets](http://www.java2ee.ru/servlets/index.html), [JSP](http://www.java2ee.ru/jsp/):
    - Понятие, предназначение
    - Дескриптор web.xml, основные сущности
    - Жизненный цикл сервлета
    - Request, Response, HttpSession etc.
    - Фильтры