

**Вопрос 1:** Как вы решаете какой бин инжектировать, если у вас несколько подходящих бинов.

A : Заинжектится близлежащий по дереву пакетов

B : Заинжектится тот, который Singleton

C : Заинжектится тот, который Prototype

D : С помощью @Primary, @Qualifier. И @Bean если выбор надо сделать в рантайме.

**Вопрос 2:** Какими свойствами не обладают транзакции?

A : Атомарность.

B : Изолированность.

C : Устойчивость.

D : Вариативность.

**Вопрос 3:** База данных библиотеки содержит следующие таблицы:

Необходимо найти всех авторов, имя которых начинается с "А".

A : SELECT a.id, a.name FROM author WHERE a.name is "A%";

B : SELECT a.id, a.name FROM author WHERE a.name sounds like "A%";

C : SELECT a.id, a.name FROM author WHERE a.name from "A%";

D : SELECT a.id, a.name FROM author WHERE a.name similar to "A%";



**Вопрос 4:** Для чего необходима денормализация в реляционных базах данных?

A : Откат к использованию предыдущей версии БД.

B : Удаление редко используемых данных.

C : Удаление любых данных.

D : Повышения производительности.

**Вопрос 5:** Почему иногда мы используем `@ResponseBody`, а иногда `ResponseEntity`?

A : `@RE` для более кастомного ответа

B : `@RB` для более кастомного ответа

C : `@RE` и `@RB` полностью одинаковы

D : `@RE` и `@RB` используются в концептуально разных случаях

**Вопрос 6:** Что будет выведено на консоль:

A : 16

B : 156

C : Будет выброшено исключение

D : Ошибка компиляции

```
public static void main(String[] args) {  
  
    try {  
        System.out.print("1");  
        int a = 2 / 0;  
        System.out.print("2");  
    } catch (IllegalArgumentException e) {  
        System.out.print("3");  
    } catch (NullPointerException e) {  
        System.out.print("4");  
    } catch (Exception e) {  
        System.out.print("5");  
    } finally {  
        System.out.print("6");  
    }  
  
}
```

**Вопрос 7:** База данных библиотеки содержит следующие таблицы:

Необходимо выбрать всех авторов, у которых есть книжка с названием «Букварь»

A : `SELECT * FROM author a WHERE exists(SELECT * FROM book b JOIN author_to_book atb ON b.id = atb.book_id WHERE atb.author_id = a.id AND b.title = 'Букварь');`

B : `SELECT a.id, a.name FROM author a JOIN book b ON b.id = atb.book.id WHERE book.title= 'Букварь';`

C : `SELECT a.id, a.name FROM author a JOIN author_to_book atb ON b.id = book.id WHERE book.title= 'Букварь';`

D : `SELECT a.id, a.name FROM author a WHERE exists(SELECT 1 FROM book b JOIN author_to_book atb ON b.id = atb.book_id WHERE atb.author_id = a.id AND b.title = 'Букварь');`



**Вопрос 8:** Как получить имя текущего исполняемого теста в JUnit 4?

A : @невозможно

B : TestName-Rule

C : Вызвать Tests.getCurrentTestName()

D : С помощью @TestName

**Вопрос 9:** Какая из строчек кода подойдет, что бы создать новый поток, который выведет в консоль "Thread is in Running state":

A : `new Test().run();`

B : `new Thread(new Test()).start();`

C : `new Test(new Thread()).run();`

D : Другое

```
public class Test implements Runnable {
    public void run() {
        System.out.println("Thread is in Running state");
    }

    public static void main(String args[]) {
        /* Как создать новый поток? */
    }
}
```

5:13 PM

**Вопрос 10:** Как проверить метод, вызывается два раза с помощью mockito verify()?

A : невозможно

B : `verify(mockObject, atLeast(2)).someMethod("was called at least twice");`

C : два раза сделать invoke

D : Повесить аннотацию @Twice

**Вопрос 11:** Где должны располагаться статические (css, js, html) ресурсы в Spring MVC приложении?

A : Это настраиваемо

B : В папке lib

C : В папке resources

D : В папке web

**Вопрос 12:** Каким будет результат выполнения кода:

A : Вывод в консоль строк, в следующем порядке: AB CD AB...

B : Вывод в консоль строк A B C D, но порядок не определен

C : Вывод в консоль строк, в следующем порядке: ABCD ABCD

D : Другое

```
public class Test extends Thread {
    public void run() {
        for (int i = 0; i < 3; i++) {
            System.out.println("A");
            System.out.println("B");
        }
    }
}

class ThreadDemo extends Thread {
    public void run() {
        for (int i = 0; i < 3; i++) {
            System.out.println("C");
            System.out.println("D");
        }
    }
}

public static void main(String args[]) {
    Test t1 = new Test();
    ThreadDemo t2 = new ThreadDemo();
    t1.start();
    t2.start();
}
```

**Вопрос 13:** Как запустить тестовые методы в определенном порядке в JUnit4?

A : @FixMethodOrder(.....)

B : @Ascending

C : Тесты будут выполнены в хаотичном порядке

D : @RunWith

**Вопрос 14:** Каким будет результат выполнения кода:

A : Ничего

B : Программа выведет в консоль пары x и y, которые не всегда будут равны (x = 1; y = 2;)

C : Программа выведет в консоль пары x и y, которые не всегда будут равны друг другу

D : Другое

```
public class Test implements Runnable {
    private int x = 0;
    private int y = 0;

    public static void main(String[] args) {
        Test obj = new Test();
        (new Thread(obj)).start();
        (new Thread(obj)).start();
    }

    public synchronized void run() {
        for (int i = 0; i < 10; i++) {
            x++;
            y++;
            System.out.println("x = " + x + "; y = " + y);
        }
    }
}
```

**Вопрос 15:** Как проверить частную функцию или класс с частными методами, полями или внутренними классами?

A : Нету возможности

B : Reflection API

C : Классы-тесты имеют доступ к private элементам класса

D : С помощью механизма сериализации/десериализации

**Вопрос 16:** Какой паттерн проектирования применяется в Hibernate для ленивой загрузки?

A : Декоратор

B : Прокси

C : Фабрика

D : Фасад

**Вопрос 17:** Аннотация, которой необходимо пометить класс, который не используется как самостоятельный Entity-класс, а только как часть других Entity.

A : @Embedded

B : @Embeddable

C : @Entity

D : @Transient

**Вопрос 18:** В каком случае при работе с коллекциями будет выброшено `java.util.ConcurrentModificationException`?

A : Когда мы пытаемся модифицировать коллекцию из нескольких потоков одновременно

B : Когда мы пытаемся модифицировать коллекцию во время итерирования по ней

C : При модификации пустой коллекции

D : Ничего из перечисленного

**Вопрос 19:** Что будет выведено в консоль:

A : 1.. 2..

B : 1.. 2.. 3..

C : Ничего

D : Другое

```
public class Test extends Thread {  
    public static void main(String[] args) {  
        Test t = new Test();  
        t.run();  
    }  
  
    public void run() {  
        for (int i = 1; i < 3; ++i) {  
            System.out.print(i + "..");  
        }  
    }  
}
```

**Вопрос 20:** Чем отличается JDBC Statement от PreparedStatement?

A : Statement используется для выполнения хранимых процедур, PreparedStatement используется для выполнения прекомпилированных запросов с параметрами.

B : Statement используется для выполнения простых SQL-запросов к БД, PreparedStatement используется для выполнения прекомпилированных запросов с параметрами.

C : Statement используется для выполнения SQL-запросов к БД, PreparedStatement используется для выполнения хранимых процедур.

D : PreparedStatement используется для выполнения простых SQL-запросов к БД, Statement используется для выполнения прекомпилированных запросов с параметрами.

**Вопрос 21:** Что вернет метод `session.load()`, если в БД нет подходящей записи?

A : `Optional.empty()`

B : Будет выброшено исключение

C : `null`

D : Будет создана новая запись с переданным `Id` и вернется ссылка на нее

**Вопрос 22:** Есть переменная объявленная как `ArrayList<? extends Number> numbers`, объект какого типа нельзя присвоить такой переменной?

A : `ArrayList<Number>`

B : `ArrayList<Object>`

C : `ArrayList<Double>`

D : Ничего из перечисленного

**Вопрос 23:** Что предпочитаете использовать для конфигурации Spring - xml или аннотирование?

A : XML конфигурация невозможна

B : С точки зрения результата – без разницы

C : Аннотирование. Так как этот способ в рантайме быстрее

D : XML. Так как этот способ в рантайме быстрее

**Вопрос 24:** Что такое план запроса?

A : Документ, описывающий порядок запросов в БД.

B : Схема взаимодействия между различными объектами базы данных.

C : Порядок располнения SQL запросов в Java приложении.

D : Последовательность операций, необходимая для получения результата SQL-запроса.

**Вопрос 25:** В чем разница между IOC (Inversion of Control) и Application Context?

A : IOC реализация AC

B : AC реализация IOC

C : это одно и тоже

D : Это концептуально разные вещи

**Вопрос 26:** Каким будет результат выполнения кода:

- A : Вывод строк в консоль в следующем порядке Multithreading Example Java Thread
- B : Не возможно однозначно определить какие строки будут выведены и в каком порядке
- C : Вывод строк в консоль в следующем порядке Java Thread ThreadExample JavaMultithreading
- D : Другое

```
public class Test extends Thread {  
    final StringBuffer sb1 = new StringBuffer();  
    final StringBuffer sb2 = new StringBuffer();  
  
    public static void main(String args[]) {  
        final Test h = new Test();  
  
        new Thread() {  
            public void run() {  
                synchronized (this) {  
                    h.sb1.append("Java");  
                    h.sb2.append("Thread");  
                    System.out.println(h.sb1);  
                    System.out.println(h.sb2);  
                }  
            }  
        }.start();  
  
        new Thread() {  
            public void run() {  
                synchronized (this) {  
                    h.sb1.append("Multithreading");  
                    h.sb2.append("Example");  
                    System.out.println(h.sb2);  
                    System.out.println(h.sb1);  
                }  
            }  
        }.start();  
    }  
}
```

**Вопрос 27:** Какой метод позволяет получить Connection из Session

- A : session.getConnection()
- B : session.connection()
- C : session.initConnection()
- D : ни один из перечисленных

**Вопрос 28:** Что из перечисленного не является состоянием Entity?

- A : Detached
- B : Attached
- C : Managed
- D : New



**Вопрос 29:** Какие различия между Mock и Spy

A : S если не был подвержен stubbing-у, то вызовет настоящую реализацию

B : Концептуально разные вещи

C : Нет отличий

D : Spy не существует

**Вопрос 30:** Что такое JDBC?

A : Библиотека для работы с потоками ввода\вывода

B : JDBC используются для организации взаимодействия между приложением и базой данных.

C : Технология организации доступа к файловой системе

D : это набор Java Api, организованный в виде службы каталогов, который позволяет Java-клиентам открывать и просматривать данные и объекты по их именам.

---

Результат собеседования.

Java core.

Правильных ответов: 4 Из : 5

Многопточность.

Правильных ответов: 3 Из : 5

Spring.

Правильных ответов: 3 Из : 5

JPA / Hibernate.

Правильных ответов: 4 Из : 5

Тестирование.

Правильных ответов: 4 Из : 5

SQL.

Правильных ответов: 3 Из : 5

Вы набрали : 84 баллов

Собеседование : пройдено