Opracowanie pytań rekrutacyjnych

Zawartość

[1. Pytania w firmach 2](#_Toc63631252)

[1.1. Accenture 2](#_Toc63631253)

[1.1.1. Klasa Object i jej metody 2](#_Toc63631254)

[1.1.1. SQL, left join, zmiana nazwy w tabeli, zmiana wartosci w tabeli, 2](#_Toc63631255)

[1.1.2. ManyToMany, OneToMany, OneToOne 2](#_Toc63631256)

[1.1.3. FoteginKey 2](#_Toc63631257)

[1.1.4. Angielski techniczny-tlumaczenie tekstu 2](#_Toc63631258)

[1.1.5. Angielski normalny- standardowe pytania:zainteresowania, jaki film, o prace, o podróże 2](#_Toc63631259)

[1.1.6. Co to Klasa, Obiekt 2](#_Toc63631260)

[1.1.7. Wyjatki 2](#_Toc63631261)

[1.1.8. Garbage collector 2](#_Toc63631262)

[1.1.9. Modyfikatory dostepu-jakie, po co i dlaczego, jaki bedzie domyslny 2](#_Toc63631263)

[1.1.10. Klasa abstrakcyjna 2](#_Toc63631264)

[1.1.11. Interfejs 2](#_Toc63631265)

[1.1.12. Jak odwrocic zdanie "ale ma kota" 2](#_Toc63631266)

[1.1.13. Po co jest GitHub? 2](#_Toc63631267)

[1.2. I-BS 2](#_Toc63631268)

[2. Pytania z bootcampow 6](#_Toc63631269)

[2.1. Kodilla 6](#_Toc63631270)

[2.2. Coderslab 10](#_Toc63631271)

[3. Podpowiedzi i inne 10](#_Toc63631272)

[3.1. Wskazówki mentora 10](#_Toc63631273)

# Pytania w firmach

## Accenture

### Klasa Object i jej metody

<https://programistajava.pl/klasa-object-poczatek-hierarchii-dziedziczenia/>

### SQL, left join, zmiana nazwy w tabeli, zmiana wartosci w tabeli,  ManyToMany, OneToMany, OneToOne, FoteginKey

### Angielski techniczny-tlumaczenie tekstu

### Angielski normalny- standardowe pytania:zainteresowania, jaki film, o prace, o podróże

### Co to Klasa, Obiekt

### Wyjatki

### Garbage collector

### Modyfikatory dostepu-jakie, po co i dlaczego, jaki bedzie domyslny

### Klasa abstrakcyjna

### Interfejs

### Jak odwrocic zdanie "ale ma kota"

### Po co jest GitHub?

## I-BS

W i-bs rozmowa była w takiej małej salce "konferencyjnej" z jedną osobą lokalną i drugą zdalnie siedziby na telekomie + tv + kamera.

Rozmowa szybko przeszła do części technicznej. Maglowano po kolei ponkty z umiejętności CV.

Pytania z javy, sringa, resta, sql, git, dość szybko wychodziły poza zakres bootcampu.

Przykładowo po przejsciu do springa, po stwierdzeniu, że używałem najnowszej wersji spring boota pierwsze pytanie dotyczyło co powiem o WebFluxach.

- omówić interfejs i klasę abstrakcyjną, różnice

- modyfikatory dostępu

- znaczenie static dla pól, metody klasy

- jak zdefiniować stałą

- dlaczego stała ma być statić

- różnica między klasą a obiektem

- wymienić znane kolekcje

- różnica międzi stosem a fifo

- gdzieś podczas rozmowy padło pytanie dla klas odwzorujących kwadrat i prostokąt, która może dziedziczyć po której (wg. nich prostokąt dziedziczy po kwadracie)

- metody streamów

- wyjątki, checked i unchecked, obsługa;

-znana wersja springa

- znane moduły springa

- zasięgi beanów

- wstrzykiwanie beanów

- tworzenie beanów

- było pytanie o obsługę błędów, nie jestem pewien czy w kontekście springa czy resta; teraz zgaduję, że chodziło o @ControllerAdvice

- co to jest websocket - ogólnie koncept działania

- statusy http, co oznaczają dwusetki, czterysetki pięćsetki

- co zrobić by kontroler zwracał odpowiedzi w json, a co by w xml

rodzaje komponentów (service, controller, repository i ogólny component)

- @Controler a @Restcontroller

- get / post / delete / put

SQL i DB:

- różnice między joinami

- jak zsumować kolumnę

- jak zliczyć

- podać przykład relacji N:M

- jak taką relację zrobić po stronie db

- napisać kwerendęm która zwróci drugi niepowtarzający się wynik

- na jakie zdarzenie zareaguje trigger; kiedy, przed czy po

Themeleaf - w jaki sposób przekazywałem parametry do widoku

Hibernate - jaki jest znany sposób tworzenia zapytań

Gradle - jak zrobić wara

GIT- jakimi komendami przekopiowałbym jeden nowy plik między repozytoriamiW i-bs rozmowa była w takiej małej salce "konferencyjnej" z jedną osobą lokalną i drugą zdalnie siedziby na telekomie + tv + kamera.

Rozmowa szybko przeszła do części technicznej. Maglowano po kolei ponkty z umiejętności CV.

Pytania z javy, sringa, resta, sql, git, dość szybko wychodziły poza zakres bootcampu.

Przykładowo po przejsciu do springa, po stwierdzeniu, że używałem najnowszej wersji spring boota pierwsze pytanie dotyczyło co powiem o WebFluxach.

- omówić interfejs i klasę abstrakcyjną, różnice

- modyfikatory dostępu

- znaczenie static dla pól, metody klasy

- jak zdefiniować stałą

- dlaczego stała ma być statić

- różnica między klasą a obiektem

- wymienić znane kolekcje

- różnica międzi stosem a fifo

- gdzieś podczas rozmowy padło pytanie dla klas odwzorujących kwadrat i prostokąt, która może dziedziczyć po której (wg. nich prostokąt dziedziczy po kwadracie)

- metody streamów

- wyjątki, checked i unchecked, obsługa;

-znana wersja springa

- znane moduły springa

- zasięgi beanów

- wstrzykiwanie beanów

- tworzenie beanów

- było pytanie o obsługę błędów, nie jestem pewien czy w kontekście springa czy resta; teraz zgaduję, że chodziło o @ControllerAdvice

- co to jest websocket - ogólnie koncept działania

- statusy http, co oznaczają dwusetki, czterysetki pięćsetki

- co zrobić by kontroler zwracał odpowiedzi w json, a co by w xml

rodzaje komponentów (service, controller, repository i ogólny component)

- @Controler a @Restcontroller

- get / post / delete / put

SQL i DB:

- różnice między joinami

- jak zsumować kolumnę

- jak zliczyć

- podać przykład relacji N:M

- jak taką relację zrobić po stronie db

- napisać kwerendęm która zwróci drugi niepowtarzający się wynik

- na jakie zdarzenie zareaguje trigger; kiedy, przed czy po

Themeleaf - w jaki sposób przekazywałem parametry do widoku

Hibernate - jaki jest znany sposób tworzenia zapytań

Gradle - jak zrobić wara

GIT- jakimi komendami przekopiowałbym jeden nowy plik między repozytoriami

# Pytania z bootcampow

## Kodilla

1. Różnica pomiędzy klasą a obiektem

2. Czym jest programowanie obiektowe, a strukturalne

3. Czym jest dziedziczenie?

4. Jaka jest różnica pomiędzy klasą abstrakcyjną a interfejsem?

5. Omów budowę interfejsu (metody domyślny, statyczne, metody prywatne – java 9)

6. Jakie znasz rodzaje kolekcji (wszystkie)

7. Różnica pomiędzy ArrayList a LinkedList

8. Do czego służą Mapy?

9. Po co metoda hashCode() i kiedy jest wykorzystywana?

10. Omów HashSet – jaki to rodzaj kolekcji, w jaki sposób porównuje obiekty przy dodawaniu?

11. Czym jest Typ Entry?

12. Czy mapa może zawierać 2 takie same obiekty pod różnymi kluczami ?

13. Czy mapa może zawierać 2 takie same klucze z różnymi obiektami?

14. Jeżeli pod danym kluczem będzie już obiekt, to co się stanie jeżeli dodamy inny obiekt pod ten sam klucz?

15. Czym jest interfejs funkcjonalny. Podaj przykład użycia i zastosowania.

16. Dzięki czemu w Javie 8 dostępne są wyrażenia lambda?

17. Omów klasę Object

18. Jaki jest kontrakt pomiędzy metodą hashCode a equlas?

19. Czy możemy użyć metody hashCode do szybkiego porównywania obiektów?

20. Jak uruchomić aplikację w Javie z linii poleceń?

21. Omów strukturę pliku pom.xml

22. Jak możemy dodać zależności do projektu (maven a gradle) ?

23. Czym jest GIT i jak działa?

24. Jakie znasz komendy GIT’a ?

25. Omów; git pull, git statsh, fetch, merge, rebase

26. Omów TDD. Napisz kalkulator przeliczający liczby dziesiętne na liczby szestastkowe, ósemkowe przy użyciu TDD.

27. Czym jest testowanie oprogramowania?

28. Jak powinna nazywać się klasa testująca, jak powinny nazywać się metody tej klasy, jaka powinna być struktura ciała metody w teście.

29. Różnica pomiędzy @Mock a @Spy a Stub

30. Omów adnotacje @Test, @Ignore, @Before, @AfterClass

31. Co oznacza słowo final w javie i gdzie możemy go użyć?

32. Czy klasę String można rozszerzyć ?

33. Czym jest Stream?

34. Jaka jest różnica w użyciu pomiędzy stream a for-each. Kiedy stream lepiej użyć, a kiedy pętli?

35. Omów .filter(), .map(), reduce() w kontekście streamów

36. Jaka jest różnica pomiędzy map a flatMap() ?

37. Czym jest Optional i jak go używać ?

38. Do czego służy try, catch, finally ?

39. Jaki jest podział wyjątków w Javie, dla każdego typu wyjątku podaj przykład

40. Kiedy utworzysz wyjątek typu checked a kiedy unchecked? Dlaczego?

41. Czy można napisać blok try-finally ?

42. Czy w bloku try-catch, może znaleźć się kolejny blok try-catch, a w nim kolejny? Dlaczego ?

43. Trow i Trows – omów

44. Czym jest SOLID? – wytłumacz na przykładzie

45. Czym jest DRY – wytłumacz na przykładzie, dokonaj refaktroingu kodu

46. Zastosowanie static?

47. Jakie są zakresy zmiennych / metod dostępne w javie?

48. Mając do wykonania prosty system do obsługi biblioteki, jak zorganizujesz pakiety wiedząc, że system ten będzie komunikował się bazą danych i aplikacją frontendową przy użyciu REST.

49. Jakie znasz wzorce projektowe? Omów 3 wybrane, przedstaw 1 z nich w formie kodu (oprócz Singletona)

50. Czym jest Srping?

51. Jaki jest zakres życia beanów w Springu?

52. Czym jest kontener Springa i do czego służy?

53. W jaki sposób możemy wstrzykiwać zależności w Springu?

54. Czym jest IoC a czym jest DI ?

55. Do czego służy adnotacja @Configuration a do czego @Bean?

56. Czym jest ORM ?

57. Różnica pomiędzy Hibernate a Spring Data a JPA

58. Czym jest Servlet?

59. Czym jest transakcyjność, do czego służy adnotacja @Transactional i jakie może przyjmować wartości

60. Czym jest Srping Boot ?

61. Omów adnotacje Adnotacje @Qualifier oraz @Conditional i podaj przykład zastosowania ich.

62. W jaki sposób skonfigurować połączenie z bazą danych w aplikacji SpringBoot

63. Jaka jest różnica pomiędzy adnotacjami @Entity a @Table?

64. Wyjaśnij pojęcie CRUD i ACID

65. Jak wywołać Rollback?

66. Przedstaw w postaci diagramu ERD system obsługujący wypożyczenie książek w bibliotece.

67. Omów relacje 1:1, 1:wiele, wiele:wiele, wiele:1

68. Napisz instrukcje SQL tworzącą następujące tabele w bazie danych:

69. Czym jest klucz główny, a czym jest klucz obcy?

70. W jaki sposób zapewnisz unikalność wartości w danej kolumnie w bazie danych?

71. Jak jest różnica pomiędzy NamedQuery a NamedNativeQuery i w jaki sposób należy je definiować. Przedstaw na przykładzie tabel utworzonych w pytaniu 68.

72. Omów instrukcje JOIN (wszystkie) i podaj przykłady.

73. Czym jest REST ?

74. Jakie znasz metody protokołu HTTP ?

75. Jak jest różnica pomiędzy HTTP 2.0 a 1.1 ?

76. Jaka powinna być poprawna struktura adresów endpointów w API ?

77. Czym jest obiekt DTO?

78. W jaki sposób można utworzyć w Srpingu REST Controller – omów wszystkie możliwości

79. Co oznacza adnotacja @Override i kiedy jest ona wymagana?

80. Co to jest RestTemplate? Omów jego najważniejsze metody i podaj przykłady użycia (getForObject, postForObject itd.)

81. Jaką znasz alternatywę dla RestTemplate?

82. Kiedy powinniśmy logować informacje?

83. Jakie znasz poziomy logowania, jak je zmienić (sl4j)?

84. Czym jest Spring Actuator i jakie daje możliwości?

85. Czym są kursory (SQL)? Omów FETCH?

86. Omów procedury i funkcje (SQL). Napisz procedurę

87. Czym są TRIGERRY i jak ich używać?

88. Kiedy powinniśmy tworzyć wiodki? W jaki sposób użyć widoku w aplikacji z JPA/Spring Data?

89. Do czego służą indeksy? Kiedy i na czym tworzyć indeksy, a kiedy ich nie tworzyć i dlaczego.

90. Czym jest XPath (Absolute, Position, Relative)

91. Omów w jaki sposób piszemy testy w Selenium? Napisz test rejestrujący konto na gmail.

92. Do czego służy Thymeleaf ?

93. Jak tworzyć aplikacje wielojęzyczne z użyciem Springa?

## Coderslab

# Podpowiedzi i inne

## Wskazówki mentora