j-labs software specialists
ul. Montwiłła-Mireckiego 25
30-426 Kraków
Tel +48 12 268-16-90
Email: j-labs@j-labs.pl
www.j-labs.pl

Zagadnienia do nauki - JAVA

Zasady pracy z dokumentem

- Celem tego dokumentu jest upewnienie się, że posiada się wiedzę z zakresu, który obejmuje dokument.
- Należy realizować główne podpunkty zaczynając od pierwszego – kolejność jest kluczowa.
- W drugiej części dokumentu znajdują się pytania kontrolne. Są one pomocne przy powtórce wiedzy, ale nie pokrywają w 100% wiedzy z danej dziedziny.
- Jeśli posiadasz wiedzę z danego punktu idź dalej.
- Po zrealizowaniu jednego punktu głównego należy przejść do następnego (kolejność tematów do powtórki jest od najważniejszego do najmniej ważnego).
- Jeśli jakąś technologie masz w CV (czyli masz komercyjne doświadczenie) upewnij się, że umiesz odpowiadać na pytania z zakresu tej technologii. Nawet, jeśli ta technologia nie jest zawarta w niniejszym dokumencie.
- SQL, Java i inne podstawy często są warunkiem koniecznym, choć niewystarczającym do pozytywnej oceny technicznej.

HINT: Wiele tematów jest wyjaśnionych w Wikipedii w sposób prosty i całościowy – warto zwrócić uwagę na to miejsce.

j-labs software specialists
ul. Montwiłła-Mireckiego 25
30-426 Kraków
Tel +48 12 268-16-90
Email: j-labs@j-labs.pl
www.j-labs.pl

Tematy do powtórki

OCJP/OCJA

Nauka Javy poprzez przygotowanie do egzaminu OCJP/OCJA (dawniej SCJP). Ten element jest podstawowym elementem (choć nie jedynym) wymaganym przez klientów. Wysoki poziom znajomości Javy jest też niezbędny do pisania dobrego kodu.

- Testowanie własnej wiedzy – na początek należy sprawdzić swoją wiedzę. Przykładowe egzaminy można znaleźć tu:
  - http://www.oraclestudy.com/java-and-middleware-certification
  - https://www.whizlabs.com/ocpjp-scjp/
  - http://enthuware.com/

  W przypadku uzyskiwania wyniku na poziomie zdawalności egzaminu, czyli 65%-75% i więcej należy zaprzestać nauki. W przypadku niższych wyników należy doszlifować najsłabsze elementy w pierwszej kolejności.
- Wątki! Przeczytać i nauczyć się całości (nie zależnie od wyników testowania):
  - http://download.oracle.com/javase/tutorial/essential/concurrency/
- Wyjątki → Praktycznie podczas każdej rozmowy padają pytania z tego obszaru.
  - http://docs.oracle.com/javase/tutorial/essential/exceptions/index.html

Algorytmika

Należy przerobić wszystkie poniższe zagadnienia ze zrozumieniem.

- Kluczowym elementem jest umiejętność rozwiązywania zadań algorytmicznych na kartce papieru – warto to poćwiczyć, gdyż programowanie bez użycia IDE może być (szczególnie z początku) problemem. Należy zwracać uwagę na to, ile pamięci się alokuje i unikać rekurencji, podczas kodowania rozwiązania.
- Pojęcie złożoności obliczeniowej. Należy być przygotowanym na dyskusje o złożoności obliczeniowej i pamięciowej różnych algorytmów.
- Złożoności obliczeniowe:

- dostępu do elementu (mając jego wartość) w tablicy,
- dostępu do elementu (mając jego wartość) w tablicy posortowanej,
- dostępu do elementu (mając jego wartość) w tablicy hashowanej,
- złożoność dostępu do elementu w drzewie (np. binary tree),
- złożoność obliczeniowa quicksorta,
- złożoność obliczeniowa sortowania bąbelkowego, przez wstawianie i przez wybieranie.

Przydatna tabelka dotycząca złożoności obliczeniowej powyższych operacji: http://bigocheatsheet.com/.

- Problem 5 filozofów i jakie są możliwe rozwiązania.
- Problem producenta i konsumenta oraz jego implementacja.
- Znajomość algorytmów: silnia, fibonacci.

JPA (lub Hibernate)

Z uwagi na to, że istnieje silny związek miedzy JPA i Hibernate nie ma większego znaczenia czy z poniższych zagadnień posiadana wiedza będzie dotyczyć Hibernate, czy JPA.

- Mapowanie klas (encji) na tabele w bazie danych.
- Budowanie relacji miedzy obiektami (OneToOne, ManyToOne itd. wraz z parametrami relacji).
- fetchType – parametr relacji miedzy klasami modelu.
- Definiowanie PK (PrimaryKey).
- Mapowanie pól na kolumny w tabeli (w tym parametry mapowania miedzy innymi: null, not null, unique, length itd.).
- Dziedziczenie encji (strategie, wady i zalety każdej z nich).
- Cascade, Delete Orphan.
- Podstawowe adnotacje – należy znać na pamięć podstawowe adnotacje związane z JPA/Hibernate dotyczące powyższych tematów.
- Cache L1 i L2.
- Zapoznanie się z:

http://docs.jboss.org/hibernate/core/3.3/reference/en/html/queryhql.html

- Load vs Get.
- Optimistic locking.
- SessionFactory.
- Criteria API.

EJB

Jeśli masz doświadczenie zawodowe w EJB znajomość poniższych zagadnień jest kluczowa.

- Rodzaje bean'ów (MDB też jest bean'em, a Entities co do zasady nie są Beanami).
- Interfejs Local vs. Remote – o co chodzi.
- Transakcje (rodzaje oraz jak się robi rollback transakcji (*)).
- Session context (zwany także EJB context) – co to jest i do czego można użyć.
- Podstawowe adnotacje służące do: wstrzykiwania zależności, wstrzykiwania session contextu, definiowania beanow, definiowania transakcyjności, definiowania interfejsów.
- JMS: Topic vs. Queue.

Spring

Jeśli masz doświadczenie zawodowe w Spring znajomość poniższych zagadnień jest kluczowa:

- Transakcje (rodzaje oraz jak się robi rollback transakcji (*)).
- Sposoby propagacji transakcji.
- Podstawowe adnotacje w szczególności: wstrzykiwania zależności i innych elementów (@Value), definiowania beanów, definiowania transakcyjności.
- Scopy bean'ów Spring'owych.
- Ogólne pojęcie o Spring MVC.
- Proxy w Springu.
- ApplicationContext.

j-labs software specialists
ul. Montwiłła-Mireckiego 25
30-426 Kraków
Tel +48 12 268-16-90
Email: j-labs@j-labs.pl
www.j-labs.pl

Wzorce projektowe

- Znajomość podstawowych wzorców projektowych takich jak:
  - singleton (dlaczego to czasem antypattern)
  - fabryka
  - mvc
  - dekorator
  - obserwator
  - dao
  - proxy
  - dto
  - facade pattern (b. ważny wzorzec)
  - adapter
- Znajomość wzorców z GoF (*Gang of Four)*

  http://geekswithblogs.net/subodhnpushpak/archive/2009/09/18/the-23-gang-of-four-design-patterns-.-revisited.aspx

SQL i bazy danych

- Znajomość podstaw takich jak: SELECT, INSERT, DELETE, UPDATE.
- Znajomość składni i sposobu działania JOIN, INNER JOIN, OUTER JOIN, HAVING, GROUP BY, ORDER BY.
- Znajomość funkcji agregujących: COUNT, MAX, MIN, SUM.
- Znajomość PK i Unique index.
- Znajomość składni zapytań zagnieżdżonych.
- Zasada dziania indeksów w bazie danych wraz z ich negatywnymi aspektami.
- Znajomość pojęcia ACID.
- Ogólna znajomość poziomów Izolacji.
- (Gdy w projekcie jest MS SQL Server) READ_COMMITED vs READ_COMMITED_SNAPSHOT w MS SQL Server.

j-labs software specialists
ul. Montwiłła-Mireckiego 25
30-426 Kraków
Tel +48 12 268-16-90
Email: j-labs@j-labs.pl
www.j-labs.pl

- Na czym polega indeksowanie i po co się go używa.

Web Security

Wiedza z bezpieczeństwa aplikacji WEBowych jest ważna, ale można ją powtórzyć pod koniec nauki:

- SQL injection (co to jest, jak to zrobić, jak się zabezpieczyć).
- XSS (co to jest, jak zrobić taki atak, jak się zabezpieczyć) i filtrowanie/kodowanie znaków specjalnych:

  http://www.youtube.com/watch?v=6UIJPjiUwfM (w pigułce).
- pojęcia dotyczące „szyfrowania" haseł: sól (salt) i sekret (secret).
- CSRF.
- Jak działa HTTPS.
- HMAC, SHA.

AJAX, JavaScript, CSS, HTML, React, Redux itd.

Ten punkt warto potraktować wyjątkowo poważnie w sytuacji, gdy projekt kładzie na to nacisk.

- Dziedziczenie w JavaScript.
- Możliwe wartości dla właściwości overflow w CSS i do czego służy.
- Narzędzia do testowania i walidacji kodu w JavaScript.
- JavaScript === vs ==.
- Związek JSON z JavaScript?
- W kontekście JavaScript: JSON vs XML.
- JavaScript: this.
- CSS: important.
- AJAX: ajaxComplete, ajaxSend, ajaxError.
- Protokół http: POST/GET, sesja, kategorie błędów.
- Propagate w JavaScripcie.
- Różnice miedzy state, a props (React).

j-labs software specialists
ul. Montwiłła-Mireckiego 25
30-426 Kraków
Tel +48 12 268-16-90
Email: j-labs@j-labs.pl
www.j-labs.pl

- Czym jest komponent Reactowy?
- Co to jest reducer?
- Jak działają akcje w Reduxie?
- Co to jest Closure?

j-labs software specialists
ul. Montwiłła-Mireckiego 25
30-426 Kraków
Tel +48 12 268-16-90
Email: j-labs@j-labs.pl
www.j-labs.pl

Pytania kontrolne

OCJP/OCJA – Java 8

- Pytania z egzaminu (linki powyżej). Należy osiągnąć wynik z testowego egzaminu na poziomie 65% lub więcej.
- Jak stworzyć i wystartować wątek?
- Jak zastopować wątek? (Wskazówka: stop jest deprecated).
- Jakiego typu może być zmienna w switch(x){}? (Wskazówka: rożni się to miedzy wersjami Javy).
- Jeśli hashcode() dla dwóch obiektów jest taki sam, to czy equals() może/musi być true?
- Co powoduje wait, notify, notifyAll?
- Co to jest synchronized?
- Co to jest volatile?
- Mamy kod:

  ```
  public static void foo(X instance) {
   instance.setA(1);
   instance.setB(2);
   instance = new X();
   instance.setA(3);
   instance.setB(4);
  }
  ```

  Jakie wartości będą w zmiennej X.a i X.b?
- Jakiej listy użyjesz, gdy przechowujesz 2000 obiektów i masz zamiar wstawiać między wybrane elementy nowe elementy?
- Po co się stosuje prywatny konstruktor dla klasy wewnętrznej?
- Co się stanie jak rozszerzę klasę String?

j-labs software specialists
ul. Montwiłła-Mireckiego 25
30-426 Kraków
Tel +48 12 268-16-90
Email: j-labs@j-labs.pl
www.j-labs.pl

- Jaka jest różnica między String, StringBuffer i StringBuilder?
- Jakie są sposoby synchronizacji?
- Omów klasyfikacje błędów (exceptions) w Javie (w tym RemoteException)?
- Jak parsować XML w Javie?
- Co to jest autoboxing? Czym różni się char od Char?
- Czy poniższy kod się skompiluje? Dlaczego?

List<String> l = ...

List<Object> la = l;

- Czym się różni Error od Exception?
- Czym się różni Exception od RuntimeException?
- Czy można ziterować wszystkie możliwe wartości Enuma?
- Jakie znasz klasy do obsługi dat?
- Jak stworzyć własna adnotacje? (nie jest wymagana kompleksowa wiedza, a ogólne pojęcie)
- Wyjaśnij działanie HashMap.
- Czym są checked exceptions w Javie?
- Czy konstructor może być final?
- Czy jeśli nadpisujemy equals, to trzeba też nadpisać hashcode?
- Czym się różni LinkedList od ArrayList?
- Co nowego jest w Java 8?
- Opisz mechanizm ThreadLocal.
- Jak działa Garbage Collector?
- Co to jest interfejs Callable i do czego służy?
- Jak działa parser XML DOM?
- Jak działa parser XML SAXParser?
- Jak powinna wyglądać klasa Immutable?
- Jak działa lambda w Java 8?
- Czy kod w Finally wykona się po returnie w Try?

j-labs software specialists
ul. Montwiłła-Mireckiego 25
30-426 Kraków
Tel +48 12 268-16-90
Email: j-labs@j-labs.pl
www.j-labs.pl

- Inner vs nested class.
- Po co stosuje się prywatny konstruktor?
- Wymień metody klasy Object.
- XOR w Javie.
- Na czym polega Dynamic Binding?
- String pools w Javie.
- Rodzaje Garbage Collectorow oraz do czego można użyć każdego z nich?
- Do czego można wykorzystać copy constructor?
- Co to jest Iterator? Jak działa?
- Czym rożni się fail-fast od fail-safe?
- Jak działają parallel streams w Java 8?
- Co zmieniło się w API stream'ów w Javie 8?
- Jak działa map i filter w Javie 8? Napisz przykład użycia.
- Jak działa Optional?
- Metody default w interfejsach – jak działają?
- Jak zmieniło się Date API w Java 8?
- Który z nowych feature'ów w Java 9 podoba Ci się najbardziej i dlaczego?

Algorytmika

- O czym nam mówi złożoność obliczeniowa?
- Jaka jest złożoność obliczeniowa:
    - dostępu do elementu (mając jego wartość) w tablicy
    - dostępu do elementu (mając jego wartość) w tablicy posortowanej
    - dostępu do elementu (mając jego wartość) w tablicy hashowanej
    - złożoność dostępu do elementu w drzewie (np. binary tree)
    - złożoność obliczeniowa quicksorta
    - złożoność obliczeniowa sortowania bąbelkowego, przez wstawianie i przez wybieranie

j-labs software specialists
ul. Montwiłła-Mireckiego 25
30-426 Kraków
Tel +48 12 268-16-90
Email: j-labs@j-labs.pl
www.j-labs.pl

- Na czym polega problem 5 filozofów i jakie są możliwe rozwiązania?
- Na czym polega problem producenta i konsumenta oraz jego implementacja?
- Napisz algorytm liczący silnie (bez rekurencji).
- Napisz algorytm liczący kolejne elementy ciągu Fibonacciego (bez rekurencji).
- Napisz algorytm odwracania jednokierunkowej listy odsyłaczowej (bez alokowania nowej listy).
- Co to jest „stabilność algorytmu sortowania"?
- W jakim typie należy trzymać ceny (np. ceny produktów w sklepie internetowym)? Dlaczego double nie jest dobrym typem do przechowywania informacji o kwotach pieniężnych i co można zamiast niego użyć?
- Przykładowe algorytmy do poćwiczenia na kartce:
    - Silnia(n)
    - Fibonacci(n)
    - Odwracanie jednokierunkowej listy odsyłaczowej.
    - Znajdowanie najbliższego wspólnego rodzica w drzewie mając podane węzły drzewa.
    - Sprawdź, czy liczba jest potęgą innej liczby np. 3.
    - Sprawdź, czy liczba jest potęgą 2 (3 możliwe podejścia: bruteforce, logarytm, przesunięcia bitowe).
    - Odwróć string (NIE alokując nowego stringa).
    - Algorytm justowania stringa.
    - Napisz funkcje, która odpowiada na pytanie czy pierwszy znak w podanym Stringu jest z dużej czy małej litery.
    - Oblicz pole koła.
    - Napisać funkcję, która dla zadanego na wejściu (dowolnego) String'a zamienia pierwszy znak na jego duży odpowiednik (jeśli jest literą).
    - Zaimplementuj hashcode dla String (kluczowe: znajomość zasad hashcode w Java oraz wykorzystanie 4 bajtów).

j-labs software specialists
ul. Montwiłła-Mireckiego 25
30-426 Kraków
Tel +48 12 268-16-90
Email: j-labs@j-labs.pl
www.j-labs.pl

JPA (lub Hibernate)

- Opisz dziedziczenie klas modelu w JPA (rodzaje, jak to się przekłada na bazę, jak rekordy są łączone, co znajduje się w nieużywanych kolumnach, jakich adnotacji trzeba użyć)
- Co powoduje FetchType.LAZY?
- Co powoduje FetchType.EAGER?
- Jak skonfigurować Hibernate'a by rozwiązać problem, gdy dwóch użytkowników próbuje zmodyfikować tą samą daną jednocześnie.
- Jakie są rodzaje cache w Hibernate? Czym się różni cache L1 i L2 w Hibernate/JPA?
- Co to jest Query Cache?
- Jak zabezpieczyć się przed powstawaniem sierotek podczas operacji merge w Hibernate (parametr adnotacji @OneToMany – deleteOrphan)?
- Jak zmapować klasę na tabele w bazie danych?
- Jak się ma JPA do Hibernate?
- Jak pozbyć się problemu wielu selektów przy relacji 1-n?

- Mamy taką sytuacje w JPA/Hibernate:

```
entity = entityManager.read(Entity.class,id);
entity.setProperty("x");
result = entityManager.createQuery("SELECT e FROM Entity e").getResultList();
// ta kwerenda wyciąga wszystko, łącznie z encją 'id'

for (Entity e : result) {
entity.getProperty(); <-- jaką wartość ma to pole, jeśli jest to
encja dla której wołaliśmy 'setProperty("x")'
}
```

- Jak działa manualny flush?
- Czy można używać obiektów entity (obiekty modelu JPA) w warstwie prezentacji? Jakie są możliwe problemy/przeszkody/ryzyka? (dotyczy m.in. lazy loading)?

j-labs software specialists
ul. Montwiłła-Mireckiego 25
30-426 Kraków
Tel +48 12 268-16-90
Email: j-labs@j-labs.pl
www.j-labs.pl

EJB 3

Jeśli masz doświadczenie zawodowe w EJB znajomość poniższych zagadnień jest kluczowa.

- Jak cofnąć transakcje bez użycia Exception'a?
- Jak wstrzyknąć EJB context?
- Wymień rodzaje beanów.
- Opisz rodzaje transakcji EJB.
- Co się stanie jeśli w transakcji zawołamy metodę oznaczoną: „@TransactionAttribute(TransactionAttributeType.MANDATORY)"?
- Co się stanie, jeśli w transakcji zawołamy metodę oznaczoną: „@TransactionAttribute(TransactionAttributeType.REQUIRES_NEW)"?
- Jaką adnotacja oznaczyć klasę, aby stworzyć beana sesyjnego stanowego oraz sesyjnego bez stanowego?
- Jak stworzyć message driven beana?
- Jak w beanie wstrzyknąć innego beana (jaka adnotacja)?
- Jak sprawdzić nazwę zalogowanego użytkownika z poziomu beana?
- Jak sprawdzić z poziomu beana sesyjnego czy użytkownik posiada dana role?
- Co to jest JTA?
- Co to jest JMS?
- Co to jest JAAS?
- Do czego możemy użyć JNDI?
- Czym różni się interfejs Local od Remote.
- Jakie zalety posiada interfejs Local?
- Jakie zalety posiada interfejs Remote?
- Jaka jest różnica miedzy Topic a Queue?

j-labs software specialists
ul. Montwiłła-Mireckiego 25
30-426 Kraków
Tel +48 12 268-16-90
Email: j-labs@j-labs.pl
www.j-labs.pl

Spring

Jeśli masz doświadczenie zawodowe w Spring znajomość poniższych zagadnień jest kluczowa.

- Jak w springu zrobić, aby bean/metoda była transakcyjna?
- Co to jest Spring security?
- Co to jest Spring MVC?
- Omów wzorzec Inversion of Control.
- Jak w Spring zdefiniować Bean?
- Do czego służą poniższe adnotacje: @Autowired, @Resource, @Service, @Controller, @RequestMapping, @Component, @Configuratio, @Repository?
- Jakie są scopy dla beanow w Springu?
- Do czego służy @Value?
- Jak działa Proxy w Springu? Sposoby tworzenia Proxy.
- Jakie są popularne implementacje ApplicationContext?
- Interfejs ApplicationContextAware.
- Wymień możliwe tryby/sposoby Autowiringu.
- Pojęcia związane z AOP.

Wzorce projektowe

- Na czym polega wzorzec: facade pattern?
- Na czym polega dependency injection?
- Na czym polega IoC?
- Na czym polega wzorzec action command?
- Na czym polega wzorzec Factory method?
- Na czym polega wzorzec Template method?
- Zaimplementuj singleton z bezpiecznym dostępem wielowątkowym (pamiętać o synchronizacji / double checking pattern / ew. SingletonHolder z bezpiecznym dostępem bez konieczności synchronizacji).

j-labs software specialists
ul. Montwiłła-Mireckiego 25
30-426 Kraków
Tel +48 12 268-16-90
Email: j-labs@j-labs.pl
www.j-labs.pl

- Ile może być instancji singletona w aplikacji zdeployowanej na 5 komputerach gdzie na każdym komputerze uruchomione są 2 serwery aplikacyjne?
- Omów AOP na przykładzie AspectJ?
- Omów MVC.
- Omów MVP.
- Wybierz trzy wzorce i omów. Wzorce z:

http://java.sun.com/blueprints/corej2eepatterns/Patterns/index.html

- Wzorzec Memento.
- Przykłady zastosowania wzorców projektowych w standardowej bibliotece Javy.

SQL

- Napisz SQL wypisujący powtórzone wartości w bazie (znajomość Group by w połączeniu z count i where, having, union). Wskazówka: należy użyć having lub innych SQLowych konstrukcji zamiast specyficznych dla konkretnej bazy poleceń.
- Jakie są negatywne aspekty zakładania indexu na kolumnę/kolumny?
- Jak działa (ogólna idea) index w bazie danych?
- Co daje założenie indexu?
- Jakie wady ma index?
- Jaka jest różnica między INNER JOIN a OUTER JOIN?
- Czym się różni widok od tabeli?
- Jaka jest różnica między PK a Unique index?
- Co to jest Foreign Key?

j-labs software specialists
ul. Montwiłła-Mireckiego 25
30-426 Kraków
Tel +48 12 268-16-90
Email: j-labs@j-labs.pl
www.j-labs.pl

SQL – Specyficzne dla baz danych

Jeśli z daną bazą masz doświadczenie powinieneś umieć odpowiedzieć na poniższe pytania.

- MS SQL Server: Jaka jest różnica między: READ_COMMITED vs READ_COMMITED_SNAPSHOT w MS SQL Server.
- Oracle: Co to jest DUAL?
- Oracle: Po co używa się konstrukcji: select * from DUAL?
- Oracle: Co zawierają widoki: all_*, user_*

WebSecurity

- Czy możliwy jest HQL injection? (Podpowiedź: TAK – warto przeanalizować temat, dlaczego jest możliwy)
- Jak się zabezpieczyć przez SQL injection?
- Jak się zabezpieczyć przed XSS (ogólnie oraz w technologii, w jakiej pracowałeś)?
- Co to jest CSRF i jak się przed tym zabezpieczyć?
- Co to jest „sól" (salt) (dot. przechowywania haseł w bazie danych)?
- Co to jest hashowanie i kiedy się je stosuje?
- HMAC, SHA.
- Co to jest CA?

AJAX, JavaScript, CSS, HTML itd.

Ten punkt warto potraktować wyjątkowo poważnie w sytuacji, gdy projekt kładzie na to nacisk.

- Czy w Javascript można overload-ować metody?
- JavaScript: this – opisz o co chodzi.
- Jaka jest różnica między a==b a a===b w JavaScript?
- Opisz jak działa dziedziczenie w JavaScript.
- Jakie są możliwe wartości dla właściwości overflow w CSS i do czego to służy?
- Jakie znasz narzędzia do testowania i walidacji kodu w JavaScript?
- Jak ma się JSON do JavaScript?

j-labs software specialists
ul. Montwiłła-Mireckiego 25
30-426 Kraków
Tel +48 12 268-16-90
Email: j-labs@j-labs.pl
www.j-labs.pl

- Dlaczego lepiej z JavaScriptu łączyć się do serwera przy użyciu JSON a nie XML?
- Co oznacza w CSS important?
- AJAX: co oznaczają ajaxComplete, ajaxSend, ajaxError?
- AJAX: co to jest ajax push?
- Jak ma się JSON do JavaScript?
- Jaka jest różnica między http POST i http GET?
- Jak działa sesja http z punktu widzenia protokołu http – co, jak i kiedy jest przesyłane?
- Jakie typy danych są dostępne w JavaScript?
- Do czego służy i jak działa operator typeof?
- Jak dodawać i usuwać propertiesy w runtimie w JavaScripcie? (wskazówka: słowo kluczowe prorotype).
- Co to jest closure w JavaScripcie?

MISC

- Co to są wymagania niefunkcjonalne?
- Jak ustawić maksymalna wartość pamięci dla JVM w Tomcat/JBoss?
- Omów model OSI/ISO.
- Zadanie z dokumentami (dokument + historyczne + powiązane) - zaprojektuj strukturę bazy danych, w której:
    - można przechowywać dokumenty wraz z ich updatami (wersje historyczne) jak i powiązaniami do innych;
    - dodatkowe pytania do tego pytania:
        - co trzeba zrobić, aby znaleźć dokumenty zależne w zaprojektowanej strukturze?
        - co trzeba zrobić, aby znaleźć dokumenty historyczne w zaprojektowanej strukturze?
        - co trzeba zrobić, aby znaleźć rodzica?
- Co to jest JAX-WS?

j-labs software specialists
ul. Montwiłła-Mireckiego 25
30-426 Kraków
Tel +48 12 268-16-90
Email: j-labs@j-labs.pl
www.j-labs.pl

- Co to jest JAXB?
- Co to jest JAX-RS?
- Jaka jest różnica między: optimistic locking a pesimistic locking?
- Czym się rożnią aggregation i composition?
- Czym się różni composition od inheritance?
- Omów podstawowe elementy Scrum.
- Omów DDD.
- Omów TDD.
- Co to jest polimorfizm?
- Co to jest dziedziczenie? Czym różni się kompozycji? Kiedy stosować jedno, kiedy drugie?
- Co to jest enkapsulacja?
- Co można się dowiedzieć z pliku WSDL?
- Jakie znasz skróty klawiszowe IDE (np. Eclipse) w którym pracujesz? Wymień przynajmniej 6.
- Czym się różni test jednostkowy od integracyjnego?
- Czym charakteryzuje się dobry test jednostkowy?
- Co to jest biblioteka do mock'owania?
- Co to jest RMI?
- XML Schema – opisz co to jest i do czego służy?
- Klasyfikacja błędów w protokole http?
- Autoryzacja i autentykacja (uwierzytelnianie) - czym są?
- Omów zasady tworzenia poprawnego kodu.
- Jak się ma polimorfizm do dziedziczenia?
- Narysuj diagram sekwencji (UML).
- Narysuj diagram kompozycji (UML).
- Co to jest JMS?
- Mamy dwa wiadra 7l i 4l. Jak odmierzyć 5l?
- Jest godzina 3:15. Jaki kąt tworzą ze sobą wskazówki zegara?

j-labs software specialists
ul. Montwiłła-Mireckiego 25
30-426 Kraków
Tel +48 12 268-16-90
Email: j-labs@j-labs.pl
www.j-labs.pl

- Dane jest osiem identycznych z wyglądu kulek. Jedna z nich jest cięższa. Znajdź ją przy pomocy wagi szalkowej w jak najmniejszej liczbie ważeń.
- Napisz testy jednostkowe do funkcji sgn(x) (wartość bezwzględna).
- Omów zasadę SOLID.
- Jak mock'ować statyczna metodę? Jakiej biblioteki do mock'owania użyć.
- Do czego służy PMD?
- Do czego służy checkstyle?
- Czym jest programowanie funkcyjne?
- Jak kontrolujesz jakość kodu?
- Jakich używałeś profilerów w Javie?

j-labs software specialists
ul. Montwiłła-Mireckiego 25
30-426 Kraków
Tel +48 12 268-16-90
Email: j-labs@j-labs.pl
www.j-labs.pl

Dodatkowa wiedza

TX w EJB i Spring

- EJB. W EJB domyślnie jedynie exceptiony dziedziczące po RuntimeException rollbackują transakcje. Wyrzucenie exceptiona dziedziczącego po Exception nie rollbackuje transakcji.

  Czyli, żeby zrollbackować transakcje trzeba:
  - rzucić coś co dziedziczy po RuntimeException,
    albo
  - własny exception zaanotować: @ApplicationException(rollback = true),
    albo
  - na session context wołamy setRollbackOnly()
    @Resource
    private EJBContext sctx; //te dwie linijki wstrzykują session context
    i potem
    sctx.setRollbackOnly()

    http://www.jguru.com/faq/view.jsp?EID=723614

Uwaga: Czy rzucany przez kontener java.rmi.RemoteException powoduje efekt podobny do zachowania opisanego powyżej dla RuntimeException?

- Spring. W Springu jedynie exceptiony dziedziczące po RuntimeException (runtime, unchecked exceptions) domyślnie rollbackuja transakcje. Wyrzucenie exceptiona dziedziczącego po Exception nie rollbackuje transakcji. Jest to taka sama reguła jak w EJB: unchecked exception cofa, checked nie.

  Czyli, aby zrollbackować transakcje należy:

  - ✓ rzucić coś co dziedziczy po RuntimeException,
    albo
  - ✓ oznaczyć metodę @Transactional(rollbackFor=MineFineNewException.class),
    albo
  - ✓ programowo:
    TransactionAspectSupport.currentTransactionStatus().setRollbackOnly();

http://static.springsource.org/spring/docs/current/spring-framework-reference/htmlsingle/spring-framework-reference.html#transaction-declarative-rolling-back

- ✓ Proponujemy także doczytać o @Transactional http://static.springsource.org/spring/docs/current/spring-framework-reference/htmlsingle/spring-framework-reference.html#transaction-declarative-annotations.

- Wzorzec projektowy
  Stara dobra praktyka mówi, aby nie mieć problemów z półcommitami i innymi dziwnymi błędami wynikającymi z zacommitowanych transakcji, które powinny być rollbackowane, najlepiej zbudować własną strukturę exceptionów projektowych, które domyślnie cofają transakcje.

W przypadku, kiedy w programie potrzebne są wyjątki, których rzucenie nie powoduje rollbacku transakcji, należy je specjalnie, osobno zdefiniować.

Czyli reguła jest następująca: cofam transakcje, wyjątek commituje.

Przykład (z życia) błędu wynikłego z braku znajomości zasad cofania transakcji.
w pseudo kodzie:

```
rejestrujUsera(User user){
zapiszUserDoBazy(user);
wyslijEMailLinkAktywacyjny(user); //komunikacja z serwerem poczty
}
```

Gdy podczas komunikacji z serwerem poczty zostanie rzucony wyjątek, na który nie jesteśmy gotowi, prawdopodobnie będzie to exception commitujący, co w efekcie spowoduje, że w systemie użytkownik zostanie dodany do bazy danych, jednak trzecia metoda z przykładu nie wykona się. Do użytkownika nie zostanie wysłany mail z linkiem aktywacyjnym, w związku z czym nie będzie on mógł potwierdzić rejestracji.

POWODZENIA!