Rozřazovací příklad

Intro

Príklad obsahuje niekoľko levelov. Prosím, riešenie nám pošlite nezávisle na tom, do ktorého levelu sa až dostanete. Ultimátnym cieľom je vypracovať levely všetky, ale určite začnite prvým a postupne riešte ďalšie a ďalšie. Kde sa dostanete, tam sa dostanete. Test má za úlohu ukázať nám vaše znalosti, splniť určitý počet levelov nie je podmienkou pre pokračovanie v prijímacom konaní. Ale čím viac dokážete a čím krajšie bude riešenie, tým lepšie pre vás. Príklad nám prosím odovzdajte formou odkazu na GitHub repozitár. Prípadné poznámky k implementácii nám ideálne napíšte priamo do README súboru (viď level 1). Veľa šťastia!

Level 1 - Git

Vytvorte si vlastný GitHub repozitár, do ktorého budete ukladať svoju prácu. Svoju prácu priebežne commitujte a po väčších celkoch (minimálne po naplnení úrovne) odporúčame aj pushovať do repozitára aby ste o svoju prácu neprišli. Je nutné aby bolo z commitov vidieť váš progress, čo znamená minimálne jeden commit na jeden level. V repozitári si vytvorte súbor README.md (resp. Github ho generuje sám), do ktorého si môžete priebežne ukladať progress či zapisovať pre seba alebo pre nás nejaké poznámky k úlohe.

Návody, ktoré vám môžu pomôcť:

- ako vytvoriť repozitár - https://guides.github.com/activities/hello-world/
- interaktívny tutoriál pre prácu s GIT - https://try.github.io/levels/1/challenges/1

Level 2 - Maven, Java, Servlet

Vytvorte si vlastný Maven projekt, v ktorom budete ďalej vytvárať svoju webovú aplikáciu (pozn.: pre vytvorenie projektu je možné použiť niektorý z Maven archetypov pre "simple web application"). Cieľom je vytvorenie webovej aplikácie, ktorá pri prístupe na jej URL na stránke zobrazí text "Hello IBA!". Ako webový kontajner použite Apache Tomcat. Odporúčame využiť standalone kontajner, nie jeho vbudovanú verziu do IDE (napr. v NetBeans). Inštalácia zaberie pár minút, na obsluhu vám postačia prakticky dva príkazy a tým, že budete s kontajnerom pracovať napriamo tak sa naučíte lepšie ako funguje.

Návody, ktoré vám môžu pomôcť:

- úvod do Maven, na odkazovanej stránke sú dva návody: päťminútový základ alebo tridsaťminútová rozšírená verzia - http://maven.apache.org/guides/
- tutoriál pre vytvorenie servletu - http://www.mkyong.com/servlet/a-simple-servlet-example-write-deploy-run/

Level 3 - Vylepšený servlet, JSP, JSTL

Vytvorený servlet obohatíte o rozpoznávanie parametru "x", ktorý bude určovať, koľko krát sa má na stránke nápis "Hello IBA!" vypísať. Pokiaľ parameter nie je zadaný, vypíše sa nápis 1-krát (pozor na prípadné chyby v prípade, keď do URL užívateľ nezadá číslo). Samotný niekoľkonásobný výpis bude zabezpečený v JSP, do ktorého si musíte predať hodnotu získanú z URL parametru. Pre výpis použite JSTL knižnicu, ktorú si vďaka Maven môžete ľahko do projektu pridať.

Level 4 - Využitie Spring MVC

V tomto leveli nie je nutné pridávanie ďalšej funkcionality. Cieľom je refaktorovať projekt tak, aby pre implementáciu servletu využíval Spring Web MVC knižnicu. Fungovanie aplikácie však musí zostať identické ako v treťom leveli.

Návody, ktoré vám môžu pomôcť:

- ako inšpirácia môže poslúžiť tutoriál na vytvorenie servletu s pomocou Spring MVC - http://www.mkyong.com/spring3/spring-3-mvc-hello-world-example/

Level 5 - Obohatenie o vstupný formulár

Vytvorte jednoduchý POJO model (napr. "Student.java") obsahujúci nasledujúce polia:

Meno poľa	Príklad hodnoty
Meno	Jozef
Priezvisko	Mrkvička
Dátum narodenia	22. 8. 1982
Pohlavie	muž

Vytvorte separátne JSP v ktorom zobrazíte užívateľovi formulár pre vyplnenie týchto polí. Po odoslaní formuláru zobrazte odoslané údaje užívateľovi v druhom JSP. Pozn.: nie je nutná žiadna persistencia alebo validácia dát formuláru.

Level 6 - Validácia

Pridajte backendovú (Java) validáciu vstupov formuláru pomocou JSR303 validácií. Jednotlivé polia validujte podľa nasledujúcich pravidiel:

Pole	Pravidlo	Dĺžka
Meno	Text bez čísel	min. 1 znak, max. 60 znakov
Priezvisko	Text bez čísel	min. 1 znak, max. 60 znakov
Dátum narodenia	Dátum vo formáte dd.MM.yyyy v minulosti (vrátane dneška)	

Pri chybe vstupu vypíšte užívateľovi pri formulári informáciu o chybe spolu s poľom (poľami), ktorých sa chyba týka.

Návody, ktoré vám môžu pomôcť:

- ukážka JSR303 - http://www.mkyong.com/spring-mvc/spring-3-mvc-and-jsr303-valid-example/

Level 7 - Servisná vrstva

Pridajte do projektu rozhranie - interface (napr. "StudentService.java"), v ktorom sa budú nachádzať metódy na vytvorenie, odstránenie, editáciu a zobrazenie zoznamu študentov (CRUD). Vytvorte implementáciu tohto rozhrania (napr. "StudentServiceListImpl.java"), ktoré bude implementovať CRUD logiku nad objektami držanými v pamäti (bez perzistencie, napríklad v obyčajnom List). Upravte servlet tak, aby používal metódy StudentService. Vytvorte JSP pohľady - zoznam študentov s preklikom na detail konkrétneho záznamu, detail študenta, formulár na ukladanie nových študentov a formulár na editáciu existujúcich záznamov.

Level 8 - Databáza

Vytvorte novú implementáciu rozhrania StudentService, nazvať ju môžete napr. "StudentServiceDbImpl.java", ktorá bude namiesto ukladania v pamäti ukladať záznamy do DB a z DB záznamy aj čítať. Na tento účel použite vhodný ORM framework (odporúčame použitie Hibernate). Prosím ako databázu použite niektorú z Java embedded databáz, tak aby bol váš projekt spustiteľný aj u nás bez dodatočných konfigurácií.

Návody, ktoré vám môžu pomôcť:

- ukážka Hibernate so Springom - http://www.mkyong.com/spring/maven-spring-hibernate-mysql-example/
- príklady embedded databáz - http://www.mkyong.com/spring/spring-embedded-database-examples/

Level 9 - Vylepšený frontend

Upravte JSP na vytváranie a editáciu študentov tak, aby na zadávanie dátumu narodenia používala JS komponentu datepicker (odporúčame použiť knižnicu jQuery UI). Doplňte do formuláru na vytváranie a editáciu študentov javascriptové validácie, ktoré pri odoslaní formuláru upozornia užívateľa v prípade nevalídnych polí (viď level 6) a neumožnia odoslať formulár (odporúčame použitie jQuery Validation Plugin).

Návody, ktoré vám môžu pomôcť:

- dokumentácia jQuery Datepicker - https://jqueryui.com/datepicker/
- dokumentácia jQuery Validation Plugin - https://jqueryvalidation.org/files/demo/

Level 10 - REST

Vytvorte triedu (napr. "StudentServiceRest.java"), ktorá bude používať metódy StudentServiceDbImpl a bude vystavovať CRUD metódy ako REST webové služby.

Návody, ktoré vám môžu pomôcť:

- príklad na implementáciu REST pomocou Jersey - http://www.mkyong.com/webservices/jax-rs/jersey-hello-world-example/

Level 11 - Unit testy

Do projektu doplňte triedu (napr. "StudentServiceTest.java"), ktorá bude obsahovať jednotkové testy jednotlivých metód StudentServiceDbImpl pomocou JUnit.

Návody, ktoré vám môžu pomôcť:

- JUnit tutoriál - http://www.mkyong.com/unittest/junit-4-tutorial-1-basic-usage