

SARCINA PENTRU PROIECTUL INDIVIDUAL

Tema „PROIECTAREA ȘI DEZVOLTAREA APLICAȚIILOR CU BAZE DE DATE”

Elaborați o aplicație Web, utilizând Java și Javascript:

1. Efectuați o analiză a domeniului subiectului. Proiectați și creați o structură a bazei de date (tabele și relații). Rezultatul este o bază de date finalizată și structura acesteia sub forma unei diagrame de tabele și relații. Numărul minim de tabele este de 4-5. **Resurse:** <https://www.postgresql.org/docs/>
2. Creați un simplu web proiect dinamic în IDE. Utilizați web serverul Tomcat sau Jetty pentru lansarea proiectului. Adăugați în WEB-INF/lib un JDBC driver în formatul jar corespunzător SGBD folosit pentru crearea bazei de date. Verificați dacă aplicația e conectată la baza de date. Creați un simplu Servlet, utilizați web.xml. **Resurse:** Crearea unui web proiect dinamic <https://www.jetbrains.com/help/idea/creating-and-running-your-first-java-ee-application.html#what-next>, Despre JDBC <http://www.tutorialspoint.com/jdbc/>, Crearea conexiunii la bd <http://www.tutorialspoint.com/jdbc/jdbc-db-connections.htm>, web.xml mapping <http://tutorials.jenkov.com/java-servlets/web-xml.html>, <https://stackoverflow.com/questions/8198312/servlet-mapping-using-web-xml>.
3. Construiți o ierarhie de clase, folosind o clasă abstractă și o interfață conform variantei. Implementați în cadrul proiectului principiul polimorfismului.
4. Elaborați o diagramă de caz de utilizare și o descriere detaliată a fiecărui caz de utilizare (ce informații sunt necesare pentru a finaliza cazul de utilizare, care este rezultatul). Aranjați cazurile de utilizare obținute sub forma unui tabel. Numărul de opțiuni de utilizare este de cel puțin 5. **Resurse:** <https://www.jetbrains.com/help/idea/class-diagram.html>, http://discipline.elcom.pub.ro/isc/2009_ISC_Lab_2_vers02.pdf.
5. Scrieți o aplicație de testare care verifică funcționalitatea aplicației dezvoltate.
6. Aplicația trebuie să conțină următoarele funcții: adăugare, ștergere, căutare după parametri și modificare înregistrări (CRUD). Pentru a căuta, utilizați interogări parametrizate (PreparedStatement). **Resurse:** <http://www.javatpoint.com/Statement-interface>, <http://www.javatpoint.com/PreparedStatement-interface>.
7. Implementați în cadrul aplicației **Modelul MVC** - <http://www.javatpoint.com/MVC-in-jsp>. Separați nivelul de model în niveluri **de serviciu** și **DAO**. Scrieți un modul DAO pentru domeniul selectat. Pentru a face acest lucru, creați un pachet „DAO” separat în proiect și plasați fișierele principale ale modulului în el: interfața și clasa de implementare a interfeței cu toate metodele – pentru fiecare entitate câte o clasă dao. În DAO utilizați numai metode pentru operațiuni elementare (toate interogările le executați în DAO). În serviciu includeți business logica aplicației. **Resurse:** http://www.tutorialspoint.com/design_pattern/data_access_object_pattern.htm, <http://www.oracle.com/technetwork/java/dataaccessobject-138824.html>.
8. Elaborați o interfață grafică corespunzătoare variantei selectate.
9. Creați un **servlet** care servi în calitate de un **controller** al aplicației. Atrageți atenție la metodele **doGet** și **doPost**. Folosiți un **enum** pentru acțiunile din interfața grafică a utilizatorului GUI. **Resurse:** <http://www.tutorialspoint.com/servlets/>, <http://www.javatpoint.com/requestdispatcher-in-servlet>, [http://www.javatpoint.com/sendRedirect\(\)-method](http://www.javatpoint.com/sendRedirect()-method), Enums <http://tutorials.jenkov.com/java/enums.html>.

TEMA CONFORM VARIANTEI

1. Biblioteca
2. Spital
3. Depozit
4. Policlinica
5. Magazin alimentar
6. Școala
7. Universitatea
8. Firma de construcții
9. Atelier de haine
10. Agenție de turism
11. Agenție de asigurări
12. Un sector de poliție
13. Farmacie
14. Restaurant
15. Atelier foto