Temă Servlet

Silaghi Rareș-Adrian

Grupă 224-2

**Ideea Aplicației**

Aplicația creată, reprezintă un joc care simulează o extragere la loto, utilizatorul va avea posibillitatea să aleagă ce tip de joc vrea să joace 6 din 49 sau 5 din 40 și va completa cu numere un formular care reprezintă un bilet la loto, dupa care se va realiza o extragere random a 6 numere, care va fi comparată cu biletul utilizatorului și se va afișa cate numere a ghicit utilizatorul.

Aplicația se va deschide cu o pagină de start, care va conține un formular de tip select și un buton, care face trimitere spre pagina de joc în care utilizatorul va completa biletul. Dupaă completarea biletului acesta va fi verificat, iar în cazul în care apare o eroare în completare, aceasta pagina va fi reîncarcată, specificând eroarea de completare, care poate fi introducerea unui numar prea mic sau prea mare, repetarea unui numar, sau introducerea altor tipuri de caractere.

Daca biletul este completat corect, utilizatorul va fi redirectionat spre o altă pagină în care îi vor fi prezentate numerele alese de el, numerele extrase la lotto și rezultatul, adica câte numere a ghicit. Totodată, în această pagină va fi prezent un buton, care îi va permite utilizatorului să rejoace jocul.

**Descrierea Aplicației**

Aplicația este alcătuită din 3 servlrturi, reprezentate de clasele: StartGame, ChooseNumbers și LotteryResult.

* StartGame

Este servletul, care întâmpină utilizatorul și îl redirecționează odată cu selectarea tipului de joc pe care vrea să îl joace și acționarea butonului pe pagina de joc. Pagina conține un formular de tip select, cu nu buton de tip submit, din care utilizatorul va alege tipul de joc pe care vrea să îl joace. Formularul va avea valoarea 5 sau 6 în funcție de ce tip de joc alege utilizatorul, această valoare va fi preluată de servletul ChooseNumbers.

* ChooseNumbers

Servletul conține o metodă doGet(HttpServletRequest request, HttpServletResponse response), care va afișa în funcție de tipul jocului pe care îl joacă utilizatorul, care va fi dat de valoarea variabilei number\_of\_lottery\_numbers, preluată din formularul servletului StartGame, formularul cu câmpurile care trebuie completate de utilizator pentru a completa biletul. Datele completate în formular vor fi trimise trumite spre același servlet pentru ca acestea să fie verificate.

Metoda doPost(HttpServletRequest request, HttpServletResponse response), peria datele completate de utilizator și le verifică. În cazul în care găsește o eroare care poate fi de tipul: un număr introdus este prea mare sau prea mic, un număr se repetă sau au fost introduse alte tipuri de caractere, va afișa din nou pagina, specificând și eroarea care a apărut. Dacă datele introduse sunt corecte, numerele utilizatorului vor fi salvate într-o sesiune, iar utilizatorul va fi redirecționat spre servletul LotteryResult.

* LotteryResult

Servletul conține o metodă lotteryExtraction(int nr), care primește ca și parametru lungimea sirului de numere al utilizatorului și care generează un șir format din 6 numere distincte, dupa caz între 1 și 49 sau între 1 și 40.

Metoda doGet(...) care preia numerele utilizatorului din sesiune, apelează funcția care generează șirul de numere extrase la loterie, după care compară cele două șiruri de numere și afișează utilizatorului un mesaj în care se specifică câte numere a ghicit. Totodată, metoda conține și un buton care premite utilizatorului să joace din nou jocul, acest buton redirecționându-l pe utilizator pe pagina de start.

**Structura Aplicației**

Structura aplicației este de forma:

LottoApplication

src

main

java

servlets

StartGame.java

ChooseNumbers.java

LotteryResult.java

webapp

WEB-INF

Web.xml

build.gradle

Această structură este necesară pentru a se putea realiza împachetarea aplicației într-o arhivă war care apoi va putea fi deploy-ată în diverse directoare ale unor servere care vor rula aplicația. Instrumentul folosit pentru impachetarea aplicației este Gradle.

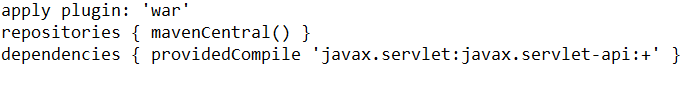
Pentru utilizarea gradle, am descărcat și dezarhivat de pe internet distribuția gradle-6.6.1 și am fixat variabila de mediu $GRADLE\_HOME cu referință sper directorul unde s-a realizat dezarhivarea, și am adăugat $GRADLE\_HOME/bin în variabila PATH.

După efectuarea setăriilor, se execută din linie de comandă, din dierectorul radacină al aplicației, LottoApplication, comanda *gradle clean build* și astfel se va crea arhiva war în directorul nou apărut, build/libs. Aceasta urmând a fi plasată în anumite directoare ale serverelor.

**Modalitățiile de rulare ale aplicației și serverele folosite**

1. **Deploy al arhivei .war pe cele 4 servere**

În primul rând pentru crearea arhivei .war a fost creat fișierul build.gradle, care conține următoarele linii:



a. Rularea pe serverul Tomcat:

- se pornește serverul Tomcat

- arhiva rezultată se plasează în directorul %CATALINA\_HOME%\webapps

- aplicația poate fi accesată din browser de la adresa <http://localhose:8080/LottoApplication>

b. Rularea pe serverul Jetty:

- se pornește serverul Jetty, din cmd, din directorul rădacina cu comanda *java –jar start.jar*

- arhiva se plasează în directorul %JETTY\_HOME%\webapps

- aplicația poate fi accesată din browser de la adresa <http://localhose:8080/LottoApplication>

c. Rularea pe serverul Wildfly

- se pornește serverul Wildfly, dinn cmd, din directorul rădacina\bin cu comanda *standalone.bat*

- arhiva se plasează în directorul %JBOSS\_HOME%\webapps\standalone\deployments

- aplicația poate fi accesată din browser de la adresa <http://localhose:8080/LottoApplication>

d. Rularea pe serverul Glassfish

- se pornește serverul Glassfish, din cmd, din directorul rădacina\bin cu comanda *asadmin start-domain*

- arhiva se plasează în directorul %GF\_HOME%\glassfish\domains\domain1\autodeploy

- aplicația poate fi accesată din browser de la adresa <http://localhose:8080/LottoApplication>

1. **Context extern**
2. Context extern Tomcat

În directotul %CATALINA\_HOME%\conf\Catalina\localhost s-a plasat un fișier aliasTomcat.xml în care se specifică locația unde se găsește conținutul arhivei .war a aplicației. Astfel, aplicația poate fi rulată din browser de la adresa localhost:8080/aliasTomcat.

Conținut fișier aliasTomcat:

<?xml version="1.0" encoding="ISO-8859-1"?>

<Context reloadable="true" path="" docBase="d:/extern/"> </Context>

1. Context extern Jetty

În directotul %JETTY\_HOME%\wabaps s-a plasat un fișier aliasJetty.xml în care se specifică locația unde se găsește conținutul arhivei .war a aplicației. Astfel, aplicația poate fi rulată din browser de la adresa localhost:8080/aliasJetty.

Conținut fișier aliasJetty:

<?xml version="1.0" encoding="ISO-8859-1"?>

[<Configure class="org.eclipse.jetty.webapp.WebAppContext">](file:///C:\jetty-distribution-9.4.33.v20201020\webapps\aliasJetty.xml)<Set name="contextPath">/aliasJetty</Set><Set name="resourceBase">d:/extern/</Set></Configure>

1. **Construirea de aplicații embedded**
2. Tomcat

S-a creat un proiect nou, LottoApplicationTomcat, care spre deosebire de celălalt proiect, conține un fișier build.gradle care va împacheta aplicația într-o arhivă autolansabilă .jar care poate fi rulată din cmd. Fișierul va conține ca apply plugin: ‘java’, în loc de ‘war’ și ca dependințe:

'org.apache.tomcat.embed:tomcat-embed-core:8.5.23',

'org.apache.tomcat.embed:tomcat-embed-jasper:8.5.23',

'org.apache.tomcat:tomcat-jasper:8.5.23'

Proiectul mai conține o clasă suplimentară TomcatServer, care conține o metodă main care crează un obiect de tip Tomcat, la care i se setează portul 8080 și i se adaugă clasele.

După ce proiectul este construit se lansează din directorul bild\libs cu colanda *java –jar LottoApplicationTomcat.jar.*

1. Jetty

S-a creat un proiect nou, LottoApplicationJetty, care conține un fișier build.gradle care are ca și dependințe:

'org.eclipse.jetty:jetty-server:9.4.3.v20170317',

‘org.eclipse.jetty:jetty-servlet:9.4.3.v20170317'

Proiectul mai conține o clasă suplimentară JettyServer, care conține o metodă main care crează un obiect de tip Tomcat, la care i se setează portul 8080 și i se adaugă clasele, dar și o metodă setSession(Server server, ServletHandler handlerServlet) pentru a putea crea sesiunea pe care se vor stoca numerele introduse de utilizator.

După ce proiectul este construit se lansează din directorul bild\libs cu comanda *java –jar LottoApplicationJetty.jar*.