

Documentatie Proiect Java

Modalitate de implementare

Pentru a implementa aceasta tema (Test grila), m-am folosit de limbajul de programare Java, impreuna cu biblioteca JavaFX. Ca IDE am folosit IntelliJ Idea, deoarece este cel mai bine pus la punct editor de cod din prezent si ofera cele mai multe functionalitati ajutatoare (Scene builder pentru JavaFX, generare automata de JavaDoc, etc.).

Pentru a realiza fereastra de Log In m-am folosit de Scene builder si pot spune ca am facut alegerea corecta, deoarece interfata are un aspect placut si este foarte usor de utilizat. Pentru a-i oferi un plus de functionalitate, am adaugat logica si pentru datele in care utilizatorul greseste parola, fiind afisata o fereastra popup cu un mesaj de atentionare.

Meniul principal are urmatoarele functionalitati:

- Alegerea dintre doua dificultati: usor si greu
- Alegerea dintre trei domenii pentru intrebari, fiecare avand intrebari separate

Structura meniului este una expandabila, putand fi adaugate foarte usor noi functionalitati precum adaugarea de noi dificultati si domenii.

Fereastra principala a fost realizata prin suprapunera unui AnchorPane cu un Pane pentru a elimina clipping-ul. Aici se afla aplicatia propriu-zisa fiind reprezentata de o portiunea in care este afisata intrebarea, mai jos urmand cele trei variante de raspuns cu o singura varianta corecta si un buton in partea dreapta jos pentru a inregistra raspunsul utilizatorului si a-l verifica.

Pentru a intregi design-ul ferestrei am adaugat si un ProgressBar care se actualizeaza de fiecare data cand utilizatorul trece la o noua intrebare, si un indicator ce afiseaza numarul de raspunsuri corecte alese de utilizator. Pentru a tine cont ce raspuns este corect, intrebarile sunt stocate intr-un mod specific si anume:

{intrebare}:{raspuns corect}:{raspuns gresit}:{raspuns gresit}

Acest format face posibila accesarea intrebarilor si raspunsurilor folosit expresii regulate (regex) deoarece caracterul ':' este folosit ca delimitator.

La final utilizatorul este intampinat de o fereastră popup in care i se comunica scorul final, urmand mai apoi ca aplicatia sa fie inchisa.

Realizarea tranzitiilor dintre ferestre am facut-o prin utilizarea a doua Scene separate pe care le-am schimbat in cadrul aceleiasi ferestre.

Pentru a face posibila alegerea a mai multor optiuni in cadrul aceleiasi ferestre, cum este in cazul ferestrei de meniu si cea principala, am folosit tipul RadioButton, pe care l-am grupat cu ajutorul obiectelor de tipul ToggleGroup. Mai apoi am accesat valoarea selectata din ToggleGroup pentru a afla ce optiune s-a ales.

Actualizarea continutului ferestrei principale am facut-o pur si simplu prin inlocuirea a fiecarui label in parte atunci cand utilizatorul raspundea la o intrebare. Partea dificila a fost la actualizarea ProgressBar-ului, unde a fost nevoie de o metoda pentru a genera un nou fir de executie. Daca actualizarea ProgressBar-ului nu era facuta pe un thread diferit, atunci aplicatia dadea crash.

Structura aplicatiei

Aplicatie este structurata pe mai multe clase, in felul urmator:

Hierarchy For All Packages

Package Hierarchies:

main.aplicatie

Class Hierarchy

- java.lang.Object
- javafx.application.Application
- main.aplicatie.LoginController
- main.aplicatie.MainAppController
- main.aplicatie.MainMenuController

Fisierele pentru validare a datelor de inregistrare, respectiv cele unde se afla intrebarile si raspunsurile se afla in directorul de resurse

Comentariu JavaDoc

Comentariul JavaDoc a fost generat in format HTML, folosind editorul de text si se afla fisierele cu codul sursa al aplicatiei, mai exact in directorul “Documentatie”.
