

Gestionarea unei biblioteci **-Proiect-**

Proiect Java: Biblioteca

Grupa 253

Membrii echipa:

Ghita Alexandru

Marinescu George

Profesori coordonatori:

Curs: Dascalescu Cristina

Seminar & Laborator: Ceausescu Ciprian

Cuprins:

<u>Descriere.....</u>	<u>3</u>
<u>Etapele de dezvoltare.....</u>	<u>3</u>
<u>Funcțiile aplicației.....</u>	<u>4</u>
<u>Persistența datelor.....</u>	<u>5</u>
<u>Exemple grafice.....</u>	<u>5</u>
<u>Interfața principală.....</u>	<u>5</u>
<u>Interfața de adăugare carte.....</u>	<u>6</u>
<u>Interfața de adăugare cititor.....</u>	<u>6</u>
<u>Interfața de ștergere.....</u>	<u>7</u>
<u>Pasi de dezvoltare, piedici, concluzii educationale.....</u>	<u>8</u>
<u>Lucrurile create/utilizate cu ușurință.....</u>	<u>8</u>
<u>Piedici.....</u>	<u>8</u>
<u>Concluzii educationale.....</u>	<u>9</u>

Descriere:

Tema proiectului este un program java ce gestioneaza o biblioteca universitara cu carti din diferite domenii, dar lipsa unor utilitati pentru un domeniu separat (nicio cerere pentru afisare etc.) acesta nu exista separat. Includerea unui domeniu caracteristic se poate face prin titlu sau id, acesta fiind de 6 caractere care sunt un cod unic, iar primele 2 pot reprezenta domeniul cu usurinta. De asemenea, programul vizeaza si se concentreaza in totalitate pe carti, nu pe studentii ce le utilizeaza. De aceea, cele doua tabele stocate in baza de date sunt “carti” (date ale cartii) si “cititori” (care reprezinta nu studentii abonati la biblioteca, ci cei ce au in mod curent carti de la noi). Gandirea aceasta diferita ii ofera unicitate.

Etapele de dezvoltare:

- I. Prima etapa a fost realizarea unui plan al aplicatiei si a ce ar trebui sa realizeze. Pe ce se va concentra si ce va contine.
- II. Apoi am creat baza de date si tabelele necesare. Am adaugat legatura intre cele doua tabele (id-ul cartii) si adaugat exemple.
- III. In urma bazei de date am format o interfata pe care am incercat sa o customizam sa analizam posibilitatile.
- IV. Etapa urmatoare a fost cea de a conecta proiectul la baza de date.
- V. Apoi a inceput crearea efectiva a functiilor dorite si utilitatii lor in ordinea aceasta: buton iesire, imagine background pentru estetica, adaugare carte, adaugare cititor, stergere carte, stergere cititor, afisarea tabelelor, functia de refresh, afisare carte dupa nume/autor, afisare cititor dupa nume/prenume/cnp, afisare carti nereturnate la timp.
- VI. Verificarea tuturor functiilor si rezolvarea bug-urilor.
- VII. Crearea documentatiei.

Funcțiile aplicației:

- Adaugarea și stergerea cartilor (dacă cartea deja există, adică id-ul se regăsește deja în tabelă, atunci va fi modificată cea existentă). Adaugarea unei cărți va face automat datele de împrumut și returnare ale cărții nule. În momentul stingerii unei cărți, dacă aceasta este împrumutată în momentul actual, va fi sters și cititorul care o deținea în momentul respectiv.
 - Adaugarea și stergerea de cititori (dacă cititorul deja există, adică id-ul cărții luate se regăsește deja în tabelă, atunci va fi modificat cel existent).
 - Se afișează în dreapta cele 2 tabele, prin urmare user-ul poate vedea conținutul tabelelor.
 - Cautarea unei cărți după titlu sau autor. Cautarea se face simultan și după titlu și după autor. Rezultatul apare în tabelele din dreapta. Cartile găsite dacă sunt împrumutate vor apărea în tabelă de jos și detinatorii actuali.
 - Cautarea unui cititor după nume, prenume sau cnp. Cautarea se face simultan după toate trei. Rezultatul apare în tabelele din dreapta. În tabelă de jos apar cititorii și în cea de sus apar cartile împrumutate.
 - Împrumutarea/returnarea unei cărți unui cititor se face efectiv prin adaugarea/stergerea cititorilor. Datorită faptului că tabelă cititori reprezintă doar posesorii actuali de cărți de la noi, adaugarea unui cititor (id-ul pus la coloana carte trebuie să existe și să fie unic) va face în tabelă cărți să apară data curentă ca dată de împrumut și dată cu o lună mai târziu ca dată de returnare. La stergerea unui cititor, coloanele de împrumut și returnare ale cărții vor deveni iar nule.
 - Afișarea listei de cărți nereturnate la timp. Acestea se compară pe baza datei curente de la computerul ce utilizează programul.
 - Refresh la tabelele din dreapta. Această funcție este utilă pentru a reseta ce se vizualizează pe ele, util în momentul adăugării/stingerii/modificării de cărți/cititori sau revinerea la vizionarea tuturor datelor tabelelor în urma căutărilor.
 - Tabelele sunt puse în spații scroll. Prin urmare, oricâte date sunt în tabele, user-ul va putea mereu să le vadă pe toate cu ușurință.
- Obs.: La stergeri butoanele sunt inversate pentru a atrage atenția user-ului să fim siguri că a vrut să ștergă acea informație (multi useri apăsând din instinct primul buton din dreapta când voiau să iasă fără să mai ștergă ...)

Obs.: Nu se pot deschide mai multe ferestre simultan subalterne celei principale. Altfel spus, in momentul apasarii bunotului de adauga carte, el creeaza o fereastră, dacă reapasati butonul sau oricare din cele 4 care creeaza fereastră separata fereastră subalterna deja creata va fi inchisa si creata o noua fereastră care va corespunde butonului apasat.

Persistenta datelor:

Datele bazei de date sunt stocate local cu ajutorul unui local host creat de xampp. Iar ambele tabele utilizate si datele acestora sunt luate din baza de date “biblioteca” si tabelele au numele “carti” si “cititori”. Intre cele doua tabele exista conexiunea de foreign key de la cititori (campul carte) la carti (campul id).

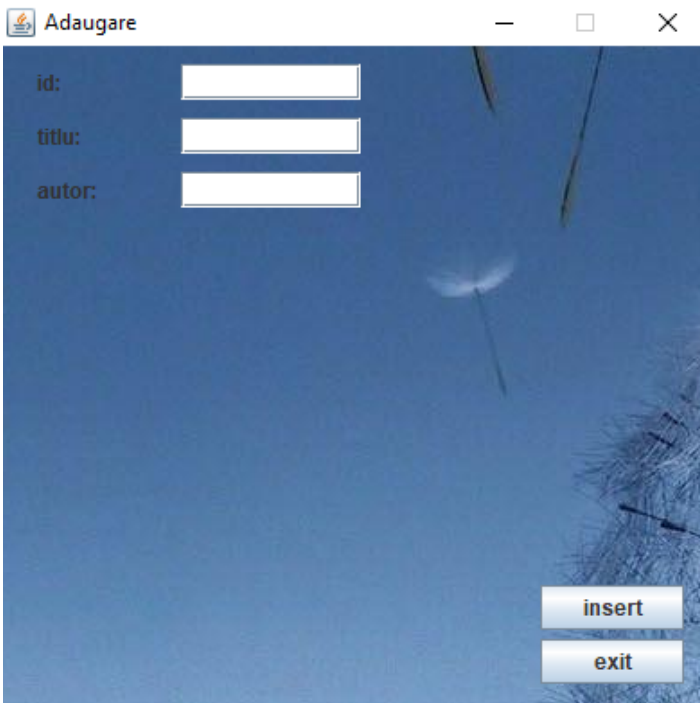
Exemple grafice:

Interfata principala:

ID	Titlu	Autor	Imprumutare	Returnare
100001	caruta	george	2018-05-21	2018-05-25
100002	diana	maximilian	2018-05-21	2018-05-29
100003	tenta	wow	2018-05-23	2018-05-30
100004	altceva	maisafie	2018-05-24	2018-06-24
100005	tentant	wow	2018-05-23	2018-05-30
100006	tes	tess	2018-04-23	2018-05-21
100007	george	testam	2018-05-24	2018-06-24
100008	altceva1	maisafie1	2018-05-21	2018-05-23

Nume	Prenume	CNP	Carte
george	marineaca	1160497022132	100001
antonia	marasina	2132397524524	100002
altcineva	really	1121299054376	100003
george	tre	4321432142212	100004
test	testare	123123123	100005
ana	bere	1231231231233	100006
cineva	george	4321432143212	100007
altcineva1	really1	1121299054376	100008

Interfata de adaugare carte:



Adaugare

id:

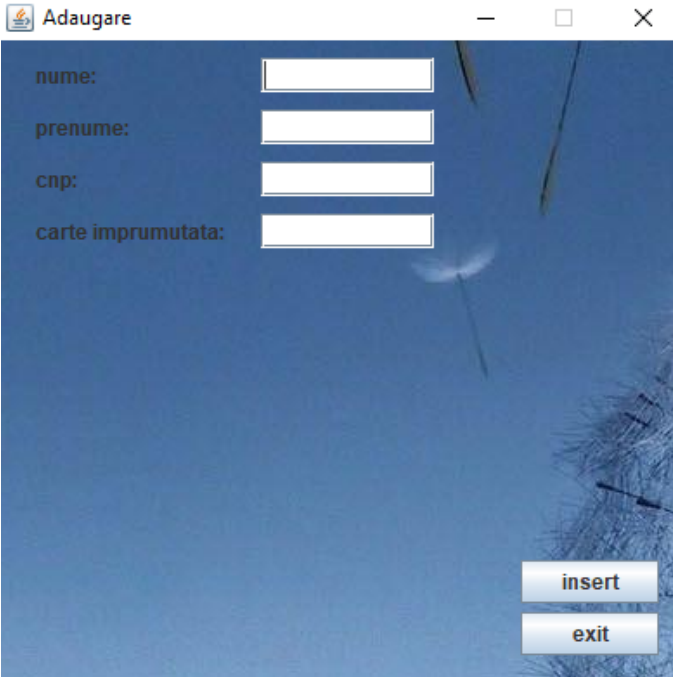
titlu:

autor:

insert

exit

Interfata de adaugare cititor:



Adaugare

nume:

prenume:

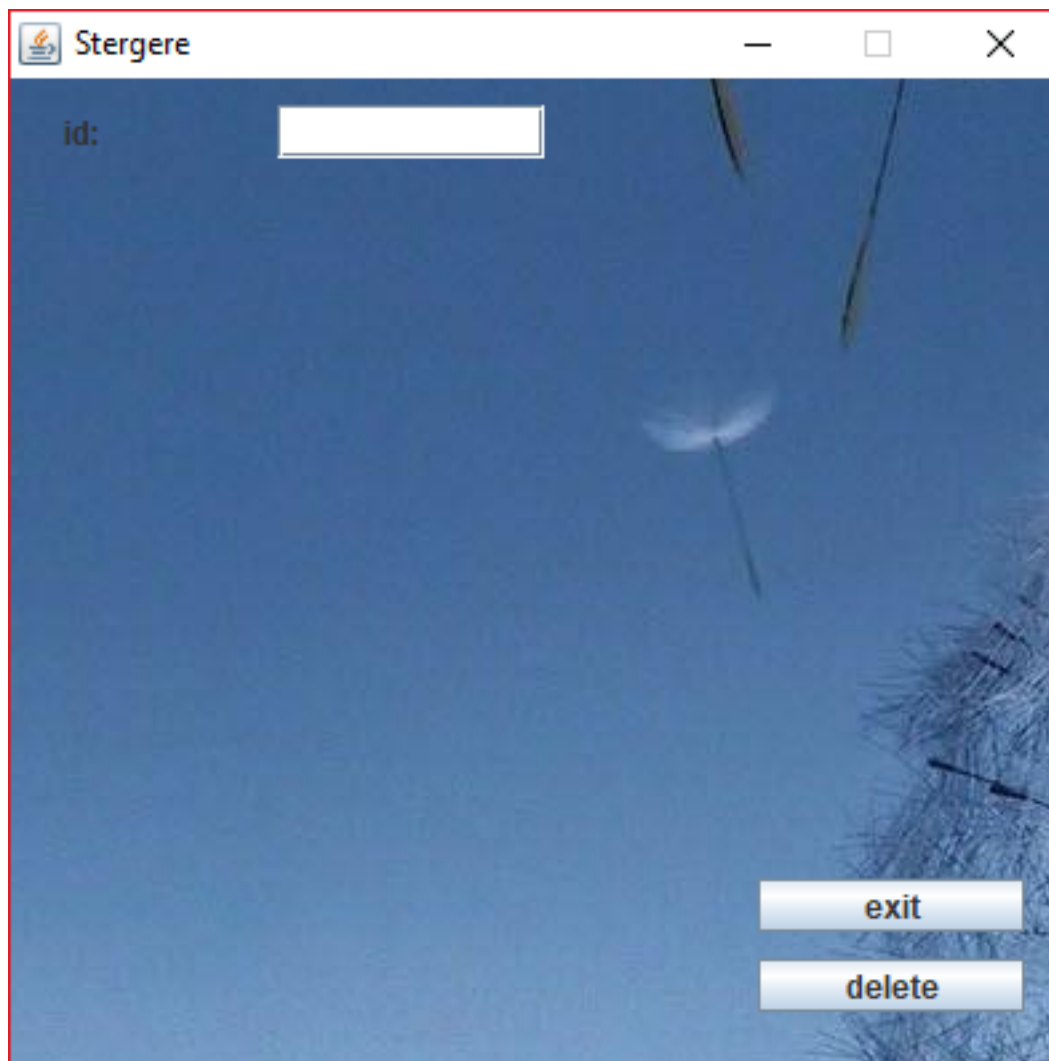
cnp:

carte imprumutata:

insert

exit

Interfata pentru stergeri:



Obs.: Ambele tipuri de stergeri au o interfata complet identica deoarece stergerea de date este un lucru important pe care user-ul trebuie sa fie sigur. Daca nu este sigur unde a intrat atunci sa inchida fereastra si sa reapele pe buton. Lucru intentionat pentru a asigura faptul ca se vor sterge cat mai putine date din greseala prin metode psihologice (necesitatea de autoconfirmare).

Pasi de dezvoltare, piedici, concluzii educationale:

Lucrurile create/utilizate cu usurinta:

Crearea bazei de date si conectarea acesteia a fost relativ usor. Utilizarea acesteia prin browser pentru crearea de tabele a fost foarte intuitiva.

Adaugarea mai multor functii cu fereastra dupa prima a fost mult copy paste si modificari minore, ceea ce a grabit procesul.

Partea de sql query.

Utilizarea frame-urilor.

Aranjarea lor in interfata (frame) a fost prin “trial by fire”, prin urmare a fost usor chiar daca a ocupat putin timp, a fost si amuzant in momentele de suprapunere agresiva (coordonate data fara luarea de seama a height si width) a elementelor.

Piedici (lucruri ce au ocupat din timp prin erori numeroase):

Adaugarea tabelelor in scrollpane si utilizarea acestora. La inceput tabela incercam sa o initializam prin null la parametrul de obiect, iar eroarea ce ne aparea nu era de mare ajutor, astfel insinuand ca de vina este scrollpane-ul. Dupa multiple incercari am reusit sa includem jTable in scrollpane (am dat obiect gol la parametru) si am realizat ca jTable nu are functii de adaugare de coloane sau randuri. Prin urmare am adaugat defaulttablemodel care este apelat de jTable care este apelat de scrollpane. Astfel putem sterge, adauga si modifica datele afisate in tabele.

Pe Ghita l-a “frecat pe creier” posibilitatea user-ului de a spama ferestre de adaugare si stergere. A folosit principiul de mostenire dupa cautarea unor variante pe net si negasirii uneia aplicabile cu usurinta. Exista un JFrame gol la inceput care este inchis cand se apasa primul buton si variabila de tip JFrame va lua valori de tip clasa ce extinde clasa JFrame.

Obtinerea datei intr-un mod editabil. Initial a fost cu date, dar functiile ei sunt dezactivate (am gasit pe net ca sunt prost facute si nu faceau ce doream). Si am facut trecerea la metoda de calendar, iar cautat ce si cum despre el, etc..

Concluzii educationale:

1. Proiectul a fost o metoda foarte utila de a invata cum sa utilizam baze de date impreuna cu java si interfata. A fost relativ usor si plin de incercari pentru a utiliza mai optim metodele si a nu umple cu clase cand pur si simplu nu este nevoie.

2. Macrourele din java sunt foarte utile si importurile automate la fel.

3. Am observat ca un cod mare de interfata nu inseamna ca e si complicat, doar multi parametri si astfel de lucruri usor de inteles si am mai exersat si utilizarea blocurilor de tip try catch.

4. Utilitatea XAMPP-ului a fost scoasa in evidenta (noi nici nu auzisem de el). Este un program foarte usor de folosit si destul de intuitiv care poate fi de mare folos.