**LitoDSA**

Nume: Hîncu Ștefan Data: 12.04.2022

CUPRINS

[1.INTRODUCERE 3](#_Toc132185419)

[2 .INFORMAȚII GENERALE 3](#_Toc132185420)

[2.1. CLASA MAIN 4](#_Toc132185421)

[2.2. CLASA GUEST 4](#_Toc132185422)

[2.3. CLASA LOGINPAGE 5](#_Toc132185423)

[2.4. CLASA MAINPAGE 5](#_Toc132185424)

[2.5. CLASA RULESPAGE 5](#_Toc132185425)

[2.6. CLASA QUESTPAGE 6](#_Toc132185426)

[2.7. CLASA QUESTS 6](#_Toc132185427)

[2.8. CLASA CREATEQUEST PAGE 6](#_Toc132185428)

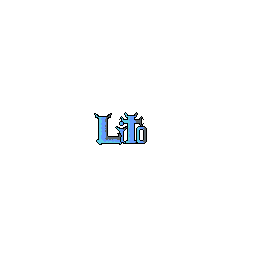
[2.9. CLASA BADGESPAGE 7](#_Toc132185429)

[2.10. CLASA BADGES 7](#_Toc132185430)

[3.SISTEMUL DE RANKING 8](#_Toc132185431)

[4.INFORMAȚII FINALE 8](#_Toc132185432)

[5.BIBLIOGRAFIE 9](#_Toc132185433)

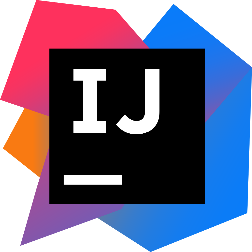


# INTRODUCERE

În zilele noastre, majoritatea studentilor îsi petrec majoritatea timpului jucandu-se diverse jocuri video, iar timpul alocat pentru invatat este din ce in ce mai mic. Pentru a face invatatul mai amuzant este nevoie de o aplicatie ce te ajuta sa inveti, o aplicatie care nu are o interfata monotona si totodata o aplicatie care sa aduca cateva “mecanici” folosite in jocurile video, asa ca am venit cu aceasta idee iar in urmatoarele randuri va voi prezenta ideea mea a unei aplicatii ce combina structurile de date si algoritmi cu jocurile video.

“LitoDsa” (Lito Data Structures & Algorithms) este o aplicatie care face ca invatatul sa fie mai amuzant si sa ii atraga pe studenti, totodata motivandu-I sa invete.

Tehnologia folosita pentru realizarea aceste aplicatii este limbajul de programare “Java” iar IDE-ul folosit este IntelliJ IDEA de la JetBrains.



# 2 .INFORMATII GENERALE

Aplicatia este alcatuita din 10 clase principale:

* Main
* Guest
* LoginPage
* MainPage
* RulesPage
* QuestPage
* Quests
* CreateQuestPage
* BadgesPage
* Badges

## 2.1. CLASA MAIN

Clasa Main contine initializarea unui obiect din clasa LoginPage cu numele loginPage si reprezinta punctul de plecare al executiei programului.

## 2.2. CLASA GUEST

Clasa Guest reprezinta utilizatorul nostru. Acest utilizator are parte un numar de monede numite “tokens”, numarul medaliilor primite de-a lungul completarii misiunilor, numarul misiunilor completate si un rank.

Constructor-ul “Guest()” are parte de cateva instunctiuni ce se executa odata cu initializarea unui obiect de timp Guest. Aceste instructiuni seteaza numarul monedelor si numarul medaliilor obtinute cu 0, iar rank-ul la “UNRANKED”.

Totodata aceasta clasa are parte de cateva metode precum:

* getTokens() = returneaza numarul monedelor utilizatorului
* setTokens(int tokens) = cu ajutorul parametrului “tokens” aceasta metoda creste numarul monedelor utilizatorului cu “tokens”.
* getBadgesNumber() = returneaza numarul medaliilor obtinute
* setBadgesNumber(int badgeNumber) = creste numarul medaliilor obtinute cu ajutorul parametrului “badgeNumber”
* getRank() = returneaza rank-ul utilizatorului
* setRank(String rank) = modifica rank-ul utilizatorului, acesta fiind “rank”
* getQuests() = returneaza numarul misiunilor completate
* setQuestsCounting() = modifica numarul misiunilor completate

## 2.3. CLASA LOGINPAGE

Clasa LoginPage, respectiv obiectul din clasa LoginPage, reprezinta interfata ce apare la deschiderea aplicatiei. Cand aplicatia este executata, utilizatorul intampina o interfata ce contine un sistem de login in functie de un username si parola.

Contul pentru a accesa aplicatia are username-ul ‚guest” si parola „12345”. Dupa scrierea datelor se afla un un button de submit care permite conectarea utilizatorului doar in cazul in care datele introduse sunt corecte. Daca datele introduse nu sunt corecte, utilizatorul este intampinat de un mesaj de eroare ce ii spune ca datele introduse sunt incorecte si nu permite conectarea.

## 2.4. CLASA MAINPAGE

Daca utilizatorul a trecut de faza de conectare, fereastra de login se inchide si un obiect de tipul MainPage va fi creat ce reprezinta interfata principala care face conexiunea intre toate celelalte.

La intrarea in Main Page se pot observa 2 panouri. Panoul din stanga reprezinta panoul unde sunt afisate datele utilizatorului precum: poza, numele, monedele si rank-ul. Panoul din dreapta contine 4 butoane:Create Quest, Rules, Quests si Badges.

## 2.5. CLASA RULESPAGE

La apasarea butonului de Rules, utilizatorul este intampinat de o interfata ce prezinta regulile de folosire ale aplicatiei, cea mai importanta fiind ca utilizatorul sa se distreze invatand. Celelalte reguli aduc in evidenta faptul ca utilizatorul trebuie sa primeasca permisiunea profesorului de a putea bifa completarea unei misiuni. Nerespectarea regulilor duce la sanctionarea studentului de catre profesor.

## 

## 2.6. CLASA QUESTPAGE

Una dintre cele mai importante clase ale aplicatiei este clasa QuestPage. In aceasta clasa se afla misiunile pe care utilizatorul trebuie sa le faca cu scopul de a primi monede, care vor aduce diferite recompense din partea profesorului pentru student, si de a creste in rank. Totodata se afla si misiunea creata de utilizator.

Fiecare misiune este recompensata cu 20 de monede. Cum este scris si inainte, cu cat studentul rezolva cat mai multe misiuni, cu atat mai multe primeste mai multe monede ce ii aduc recompense din partea profesorului.

Utilizatorul are voie sa apese pe butonul de completare a misiunii doar in cazul in care profesorul a verificat faptul ca acesta si-a indeplinit sarcina.

## 2.7. CLASA QUESTS

Clasa Quests are ca si continut numele, descrierea si numarul de minisuni completate. Constructor-ul Quests este initializat in functie de numele si descrierea acestuia, iar pe langa acestea se afla si initializarea numarului de misiuni rezolvate cu 0.

Aceasta clasa contine ca si metode:

* getQuestName() = returneaza numele misiunii
* setQuestName(String name) = metoda folosita pentru a seta numele unei misiuni
* questDes() = returneaza descrierea misiunii
* setQuestDes(String des) = seteaza descrierea unei misiuni

Aceste metode au un rol important, intrucat acestea ajuta la crearea unor misiuni.

## 

## 2.8. CLASA CREATEQUEST PAGE

Clasa CreateQuestPage reprezinta interfata ce are ca scop crearea unei misiuni pe baza unui nume si a unei descrieri, unde vin in ajutor metodele clasei Quests. Pe baza numelui si a descrierii, daca ambele campuri sunt completate se va trimite misiunea direct in QuestPage in partea stanga (despre care am vorbit anterior). Daca unul dintre cele 2 campuri nu este completat, sau ambele, utilizatorul va primi o eroare cu privire la completarea campurile.

Cand vine vorba de crearea misiunilor, pentru a accesa aceasta pagina, utilizatorul (studentul) trebuie sa indeplineasca o conditie, aceea fiind de a avea rank-ul „GOLD” . Fiecare utilizator cu acest rank sau mai ridicat are dreptul de a crea o singura misiune pana aceasta este rezolvata.

## 2.9. CLASA BADGESPAGE

Aceasta clasa are un rol important deoarece reprezinta una dintre sistemele esentiale ale aplicatiei. Fiecare utilizator rezolvand misiuni sau obtinand un numar mai ridicat de monede poate primi medalii numite „badges”. La obtinerea unei medalii, utilizatorul cand va accesa pagina de medalii va fi intampinat de un mesaj ce ii va transmite ca a castigat o noua medalie.

Pentru a vedea ce medalie a obtinut, numele (EARNED) va aparea scris in dreptul numelui medaliei. Ultima medalie „RICH LEARNER” ce necesita obtinerea numarului de 500 de monede este cea mai importanta intrucat numele presupune faptul ca utilizatorul detine multe monede si multe notiuni despre conceptele studiate pentru rezolvarea misiunilor.

## 2.10. CLASA BADGES

Aceasta clasa are un rol important cand vine vorba de medalii fiindca datorita ei a fost posibila crearea medaliilor. Aceasta clasa contine numele, descrierea si o variabila de tip boolean ce ne va indica daca medalia a fost primita sau nu. Aceste campuri vor ajuta la crearea unei medalii in constructor-ul clasei.

Metodele pe care aceasta clasa le are sunt:

* getBadgeName() = returneaza numele medaliei
* setBadgeName(String name) = seteaza numele unei medalii
* getBadgeDec() = returneaza descrierea medaliei
* isEarned() = returneaza true sau false daca moneda a fost sau nu primita de catre utilizator
* setEarned(boolean earned) = aceasta metoda seteaza faptul ca moneda a fost sau nu primita datorita variabilei earned

# SISTEMUL DE RANKING

Acest sistem permite utilizatorului deblocarea unor optiuni precum crearea unei misiuni. Pentru a obtine un rank din ce in ce mai mare utilizatorul trebuie sa rezolve cat mai multe misiuni, iar pe baza unui numar de misiuni completate, acesta va fi intampinat de un mesaj ce indica faptul ca a crescut in rank. Cu cat rank-ul este mai mare si monedele mai multe atunci si recompensa studentului este mai mare la final.

# INFORMATII FINALE

Aceasta aplicatie are la baza limbajul de programare JAVA realizat in IDE-ul de la Jet Brains numit IntelliJ IDEA

Toate elementele ce se aflat pe interfatele aplicatiei sunt obiecte ale unor clase precum JFrame, JLabel, JTextField si multe altele.

Un rol important in dezvoltarea aplicatiei este datorat ActionListener-ului ce permite conexiunea elementelor cu utilizatorul pentru a putea folosi aplicatie cum se cuvine. Totodata , in majoritatea claselor, este prezenta metoda updateUserInfo() ce are ca scop revizurirea rank-ului, a monedelor, a misiunilor si a medaliilor pentru a le modifica in timp real cand au loc diferite actiuni asupra acestor elemente.

# BIBLIOGRAFIE

<https://www.youtube.com/watch?v=xk4_1vDrzzo&t=31348s>

<https://www.javatpoint.com/>

<https://docs.oracle.com/>

https://stackoverflow.com/