TASKS

1. Download și instalare JDK 12

<https://www.oracle.com/technetwork/java/javase/downloads/index.html>



Să se creeze un proiect

Laborator-06

Pachetul principal va fi com.fmi.laborator6

Se va crea pachetul models ca subpachet al lui com.fmi.laborator6 și în el se vor afla Authenticable, AuthenticableDB. AuthenticableLDAP, User.

Să se implementeze 3 interfețe

Fiecare din metode va conține o linie de genul

Pseudo cod

System.out.println(< Curent time> : <className> -> <metodName>)

Authenticable

default void performAuthentication() throws Exception

AuthenticableLDAP extends Authenticable

private void performLDAPAuth()

default void performAuthentication() throws Exception

t

AuthenticableDB extends Authenticable

private void performDBAuth()

default void performAuthentication()

throws Exception

Metoda performAuthentication() va apela performLDAPAuth sau performDBAuth in funcție de interfața in care implementam.

Să se implementeze clasa User ce implementează ambele interfețe AuthenticableLDAP și AuthenticableDB. Se va suprascrie metoda performAuthentication pentru a evita erorile de compilare

Clasa user va avea un enum în interior AuthType cu două posibile valori DB și LDAP.

Constructorul lui User va conține un parametru enum AuthType.

User implements AuthenticableDB, AuthenticableLDAP

String name

String hashPassword

AuthType authType

public void performAuthentication()

throws Exception

Se va crea un pachet exceptions ca sub pachet al lui com.fmi.laborator6.

În pachetul deja creat se va implementa o clasă customizată de excepție UserAuthException. E ava avea un constructor cu parametru String msg ce va fi trimis constructorului din clasa părinte.

În metoda public void performAuthentication() din clasa User se implementa un switch ce va alege în functie de parametrul authType să apeleze o metoda sau alta de performAuthentication bazat pe parametru (DB sau LDAP).

1. Să se creeze o interfață

Drawable

private void internalDraw

default void draw() în afara de linia sout va conține un apel către internalDraw

void refresh() – metoda abstractă

Fiecare din metode va conține o linie de genul

Pseudo cod

System.out.println(< Curent time> : <className> -> <metodName>)

Să se creeze o clasă

Window implements Drawable

Ce va supracscrie metoda refresh() cu

Pseudo cod

System.out.println(< Curent time> : <className> -> <metodName>)

În clasa Window să se creeze un enum Event {EV1, EV2, EV3} și o clasă statică imbricată EventListener cu o metoda abstracta onEvent cu parametrul Event.

În clasa event să se creeze o metoda doStuff iar in metodă să se utilizeze ca o clasă anonima o implementare de EventListener.