Programowanie w języku Java (projekt)

Temat: E-dziennik

Mateusz Kosior II rok studiów stacjonarnych
Hubert Kowalczyk GR. 2ID13B

DOKUMENTACJA TECHNICZNA

Opis projektu:

Projekt e-dziennik wykonaliśmy korzystając ze środowiska Intelij IDEA, jako projekt Maven-a w Javie wersji 8, z wykorzystaniem JavaFX do zbudowania interfejsu graficznego. Do przetrzymywania wszystkich danych wprowadzanych w programie posłużyliśmy się bazą danych SQL lite, a operacje na niej wykonywaliśmy za pomocą narzędzi ORM lite. Do klas kontrolerów stworzyliśmy testy z wykorzystaniem frameworka JUnit. Sceny budowaliśmy z pomocą SceneBuilder-a.

Nasz program pozwala na stworzenie konta dla ucznia oraz nauczyciela. Nauczyciel może przeglądać grupy ze swojej szkoły oraz ich uczniów. Ma też możliwość wstawienia oceny uczniowi, oraz przeglądania i ewentualnego usunięcia oceny. Uczeń może przeglądać swoje oceny, sprawdzić datę wystawienia, opis oraz imię i nazwisko nauczyciela, który tą ocenę wystawił. Program jest zabezpieczony przed stworzeniem kont o takim samym adresie e-mail.

Program podzieliliśmy na paczki:

- *controllers* – zawiera klasy kontrolerów do obsługi zdarzeń w programie
- *converters* – zawiera klasy konwerterów modeli
- *database* – zawiera klasy niezbędne do łączności z bazą danych
- *modelFX* – zawiera klasy odwzorowujące modele z baz danych
- *resource* – zawiera pliki fxml
- *test* – zawiera klasy testów

controllers:

Klasa LoginController – zawiera pola oraz metody do obsługi logowania się do programu.

Metody:

logIn() – sprawdza czy mail podany w formularzu istnieje w bazie danych oraz czy wprowadzone hasło jest prawidłowe, a następnie wprowadza użytkownika do programu. Nic nie zwraca.

goBack() – powraca do poprzedniej sceny

Klasa RegistrationController – zawiera pola oraz metody do obsługi rejestrowania się w programie.

Metody:

Initialize() – w tej metodzie ustawiane są listenery oraz ładowane są choiceboxy odpowiednimi obiektami z bazy danych.

initBindings() – ustawia bindy uniemożliwiające stworzenie konta bez wprowadzenia wszystkich danych.

confirmNewUser() – sprawdza czy podanego maila nie ma w bazie, oraz czy hasło zgadza się z powtórzonym, a następnie rejestruje użytkownika w bazie danych.

goBack() – wraca do poprzedniej sceny.

Klasa StackPaneController – zawiera pola oraz metody do obsługi głównego StackPane-a.

Metody:

openStartScreen() – metoda służąca do otwarcia pierwszej sceny w programie.

setScreen(Parent parent) – czyści stage i ustawia nową scenę podaną przez parametr.

Klasa StartScreenController – zawiera pola oraz metody do obsługi startowej sceny.

Metody:

openLoginPanel() – otwiera panel logowania.

openRegistrationPanel() – otwiera panel rejestracji.

Klasa StudentScreenController – zawiera pola oraz metody do obsługi sceny dla zalogowanego ucznia.

Metody:

logOut() - pozwala na wylogowanie się użytkownika z programu.

setCenter(String fxmlPath) – ustawia formularz podany, jako parametr w centralnej części programu.

openGrades() – otwiera okno z ocenami ucznia.

Klasa StudentGradeScreenController – zawiera pola oraz metody do obsługi okna z ocenami ucznia.

initialize() – ładuje obiekty do comboboxów oraz ustawia listenery.

displayAllGrades() – wyświetla wszystkie oceny ucznia, pomijając filtry.

addStudentGradesToTableView() – ładuje obiekty ocen do tabeli.

Klasa TeacherScreenController – zawiera pola oraz metody do obsługi sceny dla zalogowanego nauczyciela.

Metody:

logOut() - pozwala na wylogowanie się użytkownika z programu.

setCenter(String fxmlPath) – ustawia formularz podany, jako parametr w centralnej części programu.

openGrades() – otwiera okno z ocenami uczniów.

openDiary() – otwiera okno z grupami oraz uczniami ze szkoły nauczyciela.

Klasa TeacherGradeScreenController – zawiera pola oraz metody do obsługi okna z ocenami uczniów.

initialize() – ładuje obiekty do comboboxów oraz ustawia listenery.

displayAllGrades() – wyświetla wszystkie oceny ucznia, pomijając filtry.

addGradesToTableView() – ładuje obiekty ocen do tabeli.

deleteGrade() – pozwala na usunięcie zaznaczonej oceny.

Klasa TeacherGroupScreenController – zawiera pola i metody do zarządzania grupami i uczniami

Metody:

initialize() – ładuje obiekty do comboboxów oraz ustawia listenery.

addGrade() – pozwala na dodanie oceny dla wybranego ucznia.

converters:

Klasa ConverterGrade

Metody:

convertToGrade(GradeFx gradeFx) – zamienia obiekt GradeFx na obiekt Grade.

convertToGradeFx(Grade grade) – zamienia obiekt Grade na GradeFx.

Klasa ConverterGroup

Metody:

convertToGroup(GroupFx groupFx) – zamienia obiekt GroupFx na obiekt Group.

convertToGroupFx(Group group) – zamienia obiekt Group na GroupFx.

Klasa ConverterSchool

Metody:

convertToSchool(SchoolFx schoolFx) – zamienia obiekt SchoolFx na obiekt School.

convertToSchoolFx(School school) – zamienia obiekt School na SchoolFx.

Klasa ConverterStudent

Metody:

convertToStudent(StudentFx studentFx) – zamienia obiekt StudentFx na obiekt Student.

convertToStudentFx(Student student) – zamienia obiekt Student na StudentFx.

Klasa ConverterSubject

Metody:

convertToSubject(SubjectFx subjectFx) – zamienia obiekt SubjectFx na obiekt Subject.

convertToSubjectFx(Subject subject) – zamienia obiekt Subject na SubjectFx.

Klasa ConverterTeacher

Metody:

convertToTeacher(TeacherFx teacherFx) – zamienia obiekt TeacherFx na obiekt Teacher.

convertToTeacherFx(Teacher teacher) – zamienia obiekt Teacher na TeacherFx.

database:

- models:

Interfejs BaseModel – wspólny interfejs dla każdego modelu.

Klasa Grade – zawiera pola opisujące dane dla oceny w bazie danych.

Klasa Group – zawiera pola opisujące dane dla grup w bazie danych.

Klasa School – zawiera pola opisujące dane dla szkoły w bazie danych.

Klasa Student – zawiera pola opisujące dane dla studenta w bazie danych.

Klasa Subject – zawiera pola opisujące dane dla przedmiotu w bazie danych.

Klasa Teacher – zawiera pola opisujące dane dla nauczyciela w bazie danych.

-dao:

Abstrakcyjna klasa CommonDao – zawiera metody umożliwiające podstawowe operacje na bazie danych.

Metody:

createOrUpdate() – tworzy albo aktualizuje obiekt w bazie danych

findById(Class<T> cls, Integer id) – szuka elementu po podanym ID

deleteByID(Class<T> cls, Integer id) – usuwa obiekt o podanym ID

closeDbConnection() – zamyka połączenie z bazą danych

queryForAll(Class<T> cls) – zbiera dane o obiektach z danego modelu

Klasy: GradeDao, GroupDao, SchoolDao, StudentDao, SubjectDao, TeacherDao – wykorzystują abstrakcyjną klasę CommonDao oraz jej metody.

-utils:

Klasa DbManager - zawiera metody umożliwiające zainicjowanie połączenia z bazą danych, stworzenie tabel, oraz usunięcie tabel

Metody:

initDatabse() – inicjuje połączenie z bazą danych oraz tworzy tabele.

createConnectionSource() – ustanawia połączenie z bazą.

closeConnectionSource() – kończy połączenie z bazą.

createTables() – tworzy tabele w bazie danych na podstawie modeli, jeżeli nie istnieją.

dropTables() – usuwa wszystkie tabele z bazy danych.

main:

Klasa Main – klasa umożliwiająca stworzenie stage-a i scen oraz uruchomienie programu.

modelFx:

Klasy GradeFx, GroupFx, SchoolFx, StudentFx, SubjectFx, TeacherFx – odzwierciedlają klasy tworzące modele w bazie danych, do obsługi z poziomu JavyFx

Klasa GradeModel

Metody:

saveGradeInDatabase(int grade, String desc, String added_date, StudentFx studentFx, SubjectFx subjectFx, TeacherFx teacherFx) – umożliwia zapisanie oceny o podanych parametrach w bazie danych.

init() – pobiera obiekty ocen z bazy danych.

initGradesForLoggedStudent(Student student) – pobiera dane o ocenach dla zalogowanego studenta (przekazanego w argumencie).

initGradesForLoggedTeacher(Teacher teacher) – pobiera dane o ocenach dla zalogowanego nauczyciela (przekazanego w argumencie).

initGradesBySubjectForStudent(Student student) – pobiera dane o ocenach dla zalogowanego studenta (przekazanego w argumencie), z filtrem dla odpowiedniego przedmiotu.

initGradesBySubjectForTeacher(Teacher teacher) – pobiera dane o ocenach dla zalogowanego nauczyciela (przekazanego w argumencie), z filtrem dla odpowiedniego przedmiotu.

deleteGradeById() – usuwa zaznaczoną ocenę z bazy danych.

Klasa GroupModel

Metody:

init() – pobiera dane o grupach z bazy danych.

Klasa SchoolModel

Metody:

init() – pobiera dane o grupach z bazy danych.

Klasa StudentModel

Metody:

init() – pobiera dane o studentach z bazy danych.

initGrups() – pobiera dane o grupie danego studenta.

InitSchools() – pobiera dane o szkole danego studenta.

emailIsInDatabase(String email) – sprawdza czy mail podany w argumencie znajduje się w bazie danych. Jeżeli tak, to zwraca true, w przeciwnym wypadku zwraca false.

checkLogIn(String email, String password) – sprawdza czy podany login i hasło zgadza się z zawartymi w bazie danych.

saveStudentInDatabase(String name, String surname, String dateOfBirth,
String phoneNumber, String address,
String email, String password,SchoolFx schoolFx,
GroupFx groupFx) – zapisuje studenta w bazie danych.

Klasa SubjectModel

Metody:

initSubjects() – pobiera dane o przedmiotach.

Klasa TeacherModel

Metody:

init() – pobiera dane o nauczycielach z bazy danych.

InitSubjects() – pobiera dane o przedmiotach danego nauczyciela.

emailIsInDatabase(String email) – sprawdza czy mail podany w argumencie znajduje się w bazie danych. Jeżeli tak, to zwraca true, w przeciwnym wypadku zwraca false.

checkLogIn(String email, String password) – sprawdza czy podany login i hasło zgadza się z zawartymi w bazie danych.

initStudentsByGroup() – pobiera dane o studentach z wybranej grupy.

initGroupsByLoggedTeacherSchool() - pobiera dane o grupach danego nauczyciela.

saveTeacherInDatabase(String name, String surname, String dateOfBirth,

String phoneNumber, String address,

String email, String password,SchoolFx schoolFx,

) – zapisuje nauczyciela w bazie danych.

Testy

Testy zostały stworzone z wykorzystaniem frameworka JUnit oraz biblioteki TestFx udostępnionej na licencji open source. Głównym zadaniem testów, było przetestowanie aplikacji pod kątem zdarzeń wyjątkowych t.j. wyjątki oraz zbadanie dostępności do komponentów zawartych w katalogu fxml. Więcej informacji o TestFx można znaleść pod linkiem https://github.com/TestFX/TestFX/wiki.

Klasa JavaFxThreadingRule

Jej zadaniem jest stworzenie (symulacja) głównego wątku na jakim uruchamiana jest biblioteka FX. Wątek ten jest niezbędny, gdyż bez niego nie można uzyskać dostępu do komponentów.

Pozostałe klasy

Klasy te dziedziczą po TestFx, co pozwala na symulację działania danej sceny(panelu). Zawarto w niej symulację inicjalizacji komponentów oraz zabezpieczono przed nieoczekiwanymi wyjątkami. Klasy te testują także poszczególne funkcje zawarte w odpowiadających im kontolerach.

Informacja na temat ilości włożonej pracy

Każdy członek naszego zespołu poświęcał cały swój wolny czas oraz przyczynił się w tym samym stopniu w powstanie projektu.