

Studentska služba - dokumentacija

Sadržaj

1.	UVOD.....	2
1.1	Izgled aplikacije	3
2.	MODEL	4
2.1	MyApp.....	4
2.2	Profesor.....	4
2.3	Predmet	5
2.4	Student.....	5
2.5	Ocena	6
2.6	BazaProfesora	7
2.7	BazaPredmeta	8
2.8	BazaStudenata	9
3.	CONTROLLER.....	10
3.1	ProfessorController	10
3.2	ProfessorFocusListener	11
3.3	SubjectController	11
3.4	SubjectFocusListener	12
3.5	StudentController	12
3.6	StudentFocusListener.....	13
3.7	CheckValue.....	13
3.8	SerializeClass	14
3.9	SearchController	15
4.	VIEW.....	15
4.1	MainFrame	15
4.2	ToolBar	16
4.3	MyMenuBar	17
4.4	MyStatusBar.....	17

4.5	TabbedPane	18
4.6	AbstractTableModelPredmet.....	18
4.7	PredmetiJTable	19
4.8	AbstractTableModelProfesor.....	19
4.9	ProfesoriJTable.....	19
4.10	AbstractTableModelStudent.....	20
4.11	StudentJTable.....	20
5.	VIEW.DIALOGS	20
5.1	AboutDialog	20
5.2	AddProfessorDialog	21
5.3	AddStudentDialog	22
5.4	AddSubjectDialog.....	22
5.5	AddSubjectToProfessor.....	23
5.6	DeleteProfessor	23
5.7	DeleteStudent	24
5.8	DeleteSubject	24
5.9	DodavanjePredmeta	24
5.10	HelpDialog.....	25
5.11	Nepolozenilspiti	25
5.12	OdaberiProfesora.....	26
5.13	Polozenilspiti	26
5.14	SubjectList	27
5.15	UnosOcjene.....	28
6.	O autoru	28

1. UVOD

Aplikacija predstavlja model studentske službe fakulteta. Aplikacija je razvijena u okruženju “Eclipse IDE 2020-0.9”, na programskom jeziku java(**JDK 15.0.1**). Projekat su uradili studenti:

- Anđela Đurić (RA28-2018) – kao student 1
- Petra Jović (RA29-2018) – kao student 2.

Projekat sadrži:




- 5 paketa u okviru "src" foldera:
 - o Controller
 - o Model
 - o View
 - o View.dialogs
 - o Gui.MessageResources – sadrži 2 fajla potrebna za lokalizaciju aplikacije
- 2 foldera sa korištenim fotografijama:
 - o Images
 - o Images1
- Tekstualni fajl:
 - o Objectstream.txt – korišten za serijalizaciju, za upis objekata

U nastavku je dat detaljniji opis sadržaja paketa projekta.

1.1 Izgled aplikacije

Studentska služba

FajlIzmeniPomoćJezik



Student

Profesor

Predmet

INDEKS	IME	PREZIME	GODINA STUDIJA	STATUS	PROSJEK
ra-2-2020	Marko	Milosevic	1	Budžet	0.0
ra-3-2019	Marja	Milić	2	Samofinansiranje	10.0
ra-3-2017	Nikola	Nikolic	1	Budžet	0.0
ra-134-2015	Pera	Peric	3	Samofinansiranje	9.0
ra-5-2019	Sofija	Illic	3	Budžet	0.0
ra-8-2018	Martina	Lukic	3	Samofinansiranje	0.0
ra-10-2017	Stojan	Stojakovic	1	Budžet	0.0
ra-12-2017	Milan	Milanovic	2	Samofinansiranje	0.0
ra-16-2019	Miroslav	Milic	2	Budžet	0.0
ra-21-2015	Stefan	Gojic	3	Samofinansiranje	0.0
ra-9-2020	Anastasija	Jokic	3	Budžet	0.0
ra-4-2017	Bogdan	Bogdanovic	3	Samofinansiranje	0.0
ra-30-2019	Ana	Dabovic	1	Budžet	0.0
ra-1-2020	Mika	Mikic	1	Samofinansiranje	0.0
ra-11-2018	Jovan	Deretic	4	Budžet	9.5
ra-12-2018	Nikola	Miskovic	4	Samofinansiranje	9.0
ra-13-2018	Martin	Stojanovic	4	Samofinansiranje	0.0
ra-14-2018	Tomislav	Novakovic	4	Budžet	0.0
ra-154-2016	Lena	Ivic	4	Budžet	0.0
ra-23-2020	Jovan	Lazic	1	Budžet	0.0
ra-1-2019	Isidora	Mikic	2	Budžet	0.0
sw-4-2014	Vladimir	Illic	4	Budžet	0.0
sw-17-2015	Mirko	Alicic	3	Samofinansiranje	0.0

Studentska služba16:31:3128.01.2021

2. MODEL

Paket sadrži sledeće klase:

2.1 MyApp

Ovo je klasa koja obuhvata samo jednu (main) javnu metodu: `public static void main()`. Ta metoda dobavlja instancu glavnu prozora aplikacije i postavlja je da bude vidljiva. Metoda nema povratnu vrijednost.

2.2 Profesor

Klasa predstavlja pojavu jednog profesora.

Sadrži sledeća privatna polja:

- *String lastName* -> predstavlja prezime profesora
- *String firstName* -> predstavlja ime profesora
- *LocalDate dateOfBirth* -> predstavlja datum rođenja profesora
- *String homeAddress* -> predstavlja adresu stanovanja profesora
- *String phoneNum* -> predstavlja kontakt telefon profesora
- *String email* -> predstavlja email adresu profesora
- *String officeAddress* -> predstavlja adresu kancelarije profesora
- *String idCard* -> predstavlja broj lične karte profesora
- *String title* -> predstavlja titulu profesora
- *String position* -> predstavlja zvanje profesora
- *List<Predmet> listOfSubjects* -> predstavlja listu predmeta koje profesor predaje (lista objekata klase Predmet)

Zatim slijede konstruktori klase:

- *public Profesor()* – u pitanju je prazan konstruktor koji kreira objekat i postavlja polja klase na inicijalne vrijednosti
- *public Profesor(String lastName, String firstName, LocalDate dateOfBirth, String homeAddress, String phoneNum, String email, String officeAddress, String idCard, String title, String position, List<Predmet> listOfSubjects)* – konstruktor koji prima odgovarajuće vrijednosti za sva polja klase i postavlja ih kod objekta koji kreira
- *public Profesor(String lastName, String firstName, LocalDate dateOfBirth, String homeAddress, String phoneNum, String email, String officeAddress, String idCard, String title, String position)* – konstruktor koji prima odgovarajuće vrijednosti za sva polja klase osim za polje „listOfSubjects“ i postavlja ih kod objekta koji kreira

- `public Profesor(String lastName, String firstName, String dateOfBirth, String homeAddress, String phoneNum, String email, String officeAddress, String idCard, String title, String position, List<Predmet> listOfSubjects)` – konstruktor koji prima odgovarajuće vrijednosti za sva polja klase, jedino za polje „dateOfBirth“ prima tip stringa te ga konvertuje u odgovarajući tip(`LocalDate`)

I na kraju imamo metode klase. U pitanju su generisani geteri i seteri redom za sva polja klase.

2.3 Predmet

Klasa predstavlja pojavu jednog predmeta.

Sadrži sledeća privatna polja:

- `String code` -> predstavlja šifru predmeta
- `String name` -> predstavlja ime predmeta
- `String semester` -> predstavlja semestar kom pripada predmet
- `int year` -> predstavlja godinu studija kojoj pripada predmet
- `Profesor professor` -> predstavlja profesora (objekat klase Profesor) koji predaje predmet
- `List<Student> passedSubject` -> predstavlja listu studenata koji su položili predmet
- `List<Student> didntPassSubject` -> predstavlja listu studenata koji nisu položili predmet

Zatim slijede konstruktori klase:

- `public Predmet()` - u pitanju je prazan konstruktor koji kreira objekat i postavlja polja klase na inicijalne vrijednosti
- `public Predmet(String code, String name, int espb, int year, String semester)` - kreira objekat i postavlja odgovarajuća polja klase na zadate vrijednosti
- `public Predmet(String code, String name, int espb, int year, String semester, Profesor p)` - kreira objekat i postavlja odgovarajuća polja klase na zadate vrijednosti
- `public Predmet(String code, String name, String semester, int year, Profesor professor, int espb, List<Student> passedSubject, List<Student> didntPassSubject)` - kreira objekat i postavlja odgovarajuća polja klase na zadate vrijednosti

I na kraju imamo metode klase. U pitanju su generisani geteri i seteri redom za sva polja klase.

2.4 Student

Klasa predstavlja pojavu jednog studenta.

Sadrži sledeća privatna polja:

- `String lastName` -> predstavlja prezime studenta
- `String firstName` -> predstavlja ime studenta
- `LocalDate dateOfBirth` -> predstavlja datum rođenja studenta
- `String homeAddress` -> predstavlja adresu stanovanja studenta
- `String phoneNum` -> predstavlja kontakt telefon studenta
- `String email` -> predstavlja email adresu studenta
- `String numIdx` -> predstavlja indeks studenta
- `int year` -> predstavlja godinu upisa studenta
- `int currentYear` -> predstavlja trenutnu godinu studenta
- `double averageGrade` -> predstavlja prosječnu ocjenu studenta
- `Status studentStatus` -> predstavlja status studenta
- `List<Ocena> passedExams` -> predstavlja listu položenih predmeta
- `List<Predmet> otherExams` -> predstavlja listu nepoloženih predmeta

Zatim slijede konstruktori klase:

- `public Student()` - u pitanju je prazan konstruktor koji kreira objekat i postavlja polja klase na inicijalne vrijednosti
- `public Student(String numIdx, String firstName, String lastName, int currentYear, Status studentStatus, double averageGrade)` - kreira objekat i postavlja odgovarajuća polja klase na zadate vrijednosti
- `public Student(String firstName, String lastName, LocalDate dateOfBirth, String adress, String phone, String email, String numIdx, int year, int currentYear, double averageGrade, Status studentStatus)` - kreira objekat i postavlja odgovarajuća polja klase na zadate vrijednosti
- `public Student(String firstName, String lastName, String dateOfBirth, String adress, String phone, String email, String numIdx, int year, int currentYear, double averageGrade, Status studentStatus)` - kreira objekat i postavlja odgovarajuća polja klase na zadate vrijednosti

I na kraju imamo metode klase. U pitanju su generisani geteri i seteri redom za sva polja klase.

2.5 Ocena

Klasa predstavlja pojavu jedne ocjene.

Sadrži sledeća privatna polja:

- `Student student` -> predstavlja studenta koji je dobio ocjenu

- **Predmet subject** -> predstavlja na koji predmet se ocjena odnosi
- **int value** -> predstavlja ocjenu
- **LocalDate examDate** -> predstavlja datum kada je ocjena dobijena

Zatim slijede konstruktori klase:

- **public Ocena()** – u pitanju je prazan konstruktor koji kreira objekat i postavlja polja klase na inicijalne vrijednost

I na kraju imamo metode klase. U pitanju su generisani geteri i seteri redom za sva polja klase.

2.6 BazaProfesora

Klasa predstavlja bazu profesora u sistemu. Implementirana je kao singleton klasa. Ova baza se prikazuje u tabeli u glavnom prozoru aplikacije.

Sadrži sledeća privatna polja:

- **List<Profesor> professors** -> predstavlja listu profesora koji se nalaze u sistemu
- **List<String> columns** -> lista koja sadrži nazive kolona kod tabele za prikaz baze profesora

Zatim slijede konstruktori klase:

Private BazaProfesora() – u pitanju je prazan konstruktor koji kreira objekat i postavlja polja klase na inicijalne vrijednost

I na kraju imamo metode klase:

- **public List<Profesor> getProfessors()** – getter za polje „professors“
- **public void setProfessors(List<Profesor> professors)** - setter za polje „professors“
- **public int getColumnCount()** – metoda koja vraća broj kolona tabele
- **public String getColumnName(int index)** – metoda koja vraća ime željene kolone
- **public Profesor getRow(int rowIndex)** – metoda koja za proslijeđeni indeks reda vraća objekat klase Profesor
- **public Profesor getProfesorFromList(String rowIndex)** - metoda koja za proslijeđeni indeks reda vraća objekat klase Profesor
- **public String getValueAt(int row, int column)** – metoda koja vraća vrijednosti iz polja tabele za proslijeđeni indeks reda i kolone

- `public void addProfessor(String lastName, String firstName, LocalDate dateOfBirth, String homeAddress, String phoneNum, String email, String officeAddress, String idCard, String title, String position, List<Predmet> listOfSubjects)` – metoda koja kreira objekat Profesora sa proslijeđenim atributima i dodaje kreiranog profesora u bazu profesora
- `public boolean checkIfProfessorExists (String id)` – metoda koja za proslijeđenu vrijednost provjerava da li u bazi postoji profesor sa tim brojem lične karte
- `public void deleteSubject(Predmet p)` – metoda koja prolazi kroz listu predmeta i uklanja proslijeđeni predmet kod svakog profesora kod kog ga pronade
- `public void deleteProfessor(String id)` – metoda koja za proslijeđenu vrijednost broja lične karte pronalazi profesora sa tim brojem i briše ga iz baze profesora (ukoliko postoji)

2.7 BazaPredmeta

Klasa predstavlja bazu predmeta u sistemu. Implementirana je kao singleton klasa. Ova baza se prikazuje u tabeli u glavnom prozoru aplikacije.

Sadrži sledeća privatna polja:

- `List<Predmet> subjects` -> predstavlja listu predmeta koji se nalaze u sistemu
- `List<String> columns` -> lista koja sadrži nazive kolona kod tabele za prikaz baze predmeta

Zatim slijede konstruktori klase:

`Private BazaPredmeta()` – u pitanju je prazan konstruktor koji kreira objekat i postavlja polja klase na inicijalne vrijednosti

I na kraju imamo metode klase:

- `public List<Predmet> getSubjects()` – getter za polje „subjects“
- `public void setSubjects(List<Predmet> subjects)` – setter za polje „subjects“
- `public int getColumnCount()` - metoda koja vraća broj kolona tabele
- `public String getColumnName(int index)` - metoda koja vraća ime željene kolone
- `public Predmet getRow(int rowIndex)` - metoda koja za proslijeđeni indeks reda (int) vraća objekat klase Predmet
- `public Predmet getSubject(String rowIndex)` - metoda koja za proslijeđeni indeks reda (string) vraća objekat klase Predmet
- `public String getValueAt(int row, int column)` – metoda koja vraća vrijednosti iz polja tabele za proslijeđeni indeks reda i kolone

- `public void deleteSubject(String code)` - metoda koja za proslijeđenu vrijednost šifre predmeta pronalazi predmet sa tom šifrom i briše ga iz baze predmeta (ukoliko postoji)
- `public void deleteStudentsFromSubject(Student s)` – metoda koja briše proslijeđenog studenta kod svih predmeta kod kojih postoji u listama studenata koji su položili ili nisu položili taj predmet
- `public void deleteProfessorFromSubject(Profesor p)` – metoda koja za proslijeđenog profesora pronalazi predmet koji taj profesor predaje i uklanja profesora sa tog predmeta
- `public void addSubject(String code, String name, int espb, int year, String semester, Profesor p)` - metoda koja kreira objekat predmeta sa proslijeđenim atributima i dodaje kreirani predmet u bazu predmeta
- `public Predmet checkIfSubjectExists(String ind)` - metoda koja za proslijeđenu vrijednost šifre predmeta provjerava da li u bazi postoji predmet sa tom šifrom
- `public void removeProfessor(String code)` – metoda koja pronalazi predmet sa proslijeđenom šifrom predmeta i uklanja profesora sa tog predmeta tj postavlja polje „professor“ na null, takođe uklanja kod profesora taj predmet sa liste predmeta koje predaje taj profesor

2.8 BazaStudenata

Klasa predstavlja bazu studenata u sistemu. Implementirana je kao singleton klasa. Ova baza se prikazuje u tabeli u glavnom prozoru aplikacije.

Sadrži sledeća privatna polja:

- `List<Student> students` -> predstavlja listu studenata koji se nalaze u sistemu
- `List<String> columns` -> lista koja sadrži nazive kolona kod tabele za prikaz baze studenata

Zatim slijede konstruktori klase:

`Private BazaStudenata()` – u pitanju je prazan konstruktor koji kreira objekat i postavlja polja klase na inicijalne vrijednosti

I na kraju imamo metode klase:

- `public List<Student> getStudents()` – getter za polje „students“
- `public void setStudents(List<Student> students)` – setter za polje „students“
- `public int getColumnCount()` - metoda koja vraća broj kolona tabele
- `public String getColumnName(int index)` - metoda koja vraća ime željene kolone

- `public Student getStudentFromList(String rowIdx)` - metoda koja za proslijeđeni indeks reda(string) vraća objekat klase Student
- `public Object getValueAt(int row, int column)` – metoda koja vraća vrijednosti iz polja tabele za proslijeđeni indeks reda i kolone
- `public void addStudent(String firstName, String lastName, String birthday, String adress, String phone, String email, String numOfIndex, int year, int currentYear, double avgGrade, Student.Status status)` - metoda koja kreira objekat klase Student sa proslijeđenim atributima i dodaje kreiranog studenta u bazu studenata
- `public Student checkIfStudentExists(String ind)` - metoda koja za proslijeđenu vrijednost indeksa provjerava da li u bazi postoji student sa tim brojem indeksa
- `public void deleteSubject(Predmet p)` – metoda koja proslijeđeni predmet briše iz lista položenih i nepoloženih ispita kod svakog studenta iz baze kod koga pronađe taj predmet
- `public void deleteStudent(String ind)` – metoda koja briše studenta sa proslijeđenim brojem indeksa iz baze
- `public void deleteStudentWithGrade(Student s)`
- `public void addGradeToSTudent(Student s, Ocena o)`- metoda koja pronalazi proslijeđenog studenta i dodaje mu proslijeđenu ocjenu u listu položenih predmeta i kod proslijeđene ocjene dodaje proslijeđenog studenta u listu studenata koji su položili predmet
- `public void removeSubject(Student s, Predmet p)` – metoda koja pronalazi proslijeđenog studenta i uklanja proslijeđeni predmet iz liste nepoloženih predmeta ako postoji

3. CONTROLLER

Paket sadrži sledeće klase:

3.1 ProfessorController

Klasa je implementirana kao singleton i služi za rad sa bazom profesora. Polja nema.

Metode klase su:

- `public boolean addProfessor(Profesor professor)` – metoda koja provjerava da li zadati profesor postoji u bazi, ako ne postoji dodaće ga i osvježiti prikaz tabele profesora u glavnom prozoru a kao povratnu vrijednost vraća uspjehnost operacije

- `public List<Profesor> getProfessors()` – metoda koja vraća listu profesora koji postoje u bazi
- `public Profesor getProfessor(int selectedIndex)` – metoda koja će da vrati objekat klase Profesor koji se nalazi na selektovanoj poziciji u tabeli profesora
- `public void deleteSubject(Predmet p)` – metoda koja poziva metodu iz baze koja će da obriše zadati predmet kod profesora kod kojih se on nalazi
- `public Profesor findProfessor(int row)` – metoda koja će da vrati objekat klase Profesor koji se nalazi na selektovanoj poziciji u tabeli profesora
- `public void deleteProfessor(int row)` – metoda koja će da obriše selektovanog profesora iz baze profesora, a to uključuje i brisanje profesora sa predmeta koje predaje. Osvježiće i prikaz tabela predmeta i profesora.

3.2 ProfessorFocusListener

Klasa implementira FocusListener i služi za validaciju unijetih podataka od strane korisnika prilikom dodavanja novog entiteta ili izmjene postojećeg. Polja nema.

Metode klase su:

- `public void focusGained(FocusEvent e)` – metoda koja određuje šta će da se desi kada polje dobije fokus : u slučaju da je tekst polja bio crven postaće crn
- `public void focusLost(FocusEvent e)` - metoda koja određuje šta će da se desi kada polje izgubi fokus: pokupiće tekst unijet u polje i provjeriće da li je u odgovarajućem formatu, u slučaju da nije boja tog teksta postavlja se na crvenu, zatim gleda sva polja na dijalogu i omogućuje dugme za potvrdu u slučaju da nema neispravnih ili praznih tekst polja(u suprotnom dugme za potvrdu je onemogućeno)
- `public boolean checkFormat(String inputText, String fieldName)` – metoda koja provjerava ispravnost formata teksta unijetog u zadato polje

3.3 SubjectController

Klasa je implementirana kao singleton i služi za rad sa bazom predmeta. Polja nema. Ima prazan konstruktor: `private SubjectController()`.

Metode klase su:

- `public void deleteSubject(int rowSelectedIndex)` – metoda koja obezbjeđuje brisanje selektovanog predmeta(u tabeli) iz baze predmeta i osvježava prikaz tabele predmeta

- `public boolean addSubject(String code, String name, int espb, int year, String semester, Profesor p)` – metoda koja dodaje novi predmet sa zadatim atributima u bazu predmeta i osvježava prikaz tabele predmeta
- `public Predmet findSubject(int row)` – metoda koja vraća objekat predmeta za selektovano polje u tabeli predmeta
- `public Predmet getSubject(String code)` – metoda koja vraća objekat predmeta koji ima zadatu šifru predmeta
- `public void removeProfessor(int row)` – metoda koja uklanja profesora sa predmeta za selektovani predmet u tabeli(dešava se prilikom pritiska na dugme „ - “)

3.4 SubjectFocusListener

Klasa implementira FocusListener i služi za validaciju unijetih podataka od strane korisnika prilikom dodavanja novog entiteta ili izmjene postojećeg. Polja nema.

Metode klase su:

- `public void focusGained(FocusEvent e)` – metoda koja određuje šta će da se desi kada polje dobije fokus : u slučaju da je tekst polja bio crven postaće crn
- `public void focusLost(FocusEvent e)` - metoda koja određuje šta će da se desi kada polje izgubi fokus: pokupiće tekst unijet u polje i provjeriće da li je u odgovarajućem formatu, u slučaju da nije boja tog teksta postavlja se na crvenu, zatim gleda sva polja na dijalogu i omogućuje dugme za potvrdu u slučaju da nema neispravnih ili praznih tekst polja(u suprotnom dugme za potvrdu je onemogućeno)

3.5 StudentController

Klasa je implementirana kao singleton i služi za rad sa bazom studenata. Polja nema. Ima prazan konstruktor: `private StudentController()`.

Metode klase su:

- `public boolean addStudent(String firstName, String lastName, String birthday, String adress, String phone, String email, String numOfIndex, int year, int currentYear, double avgGrade, Student.Status status)` – metoda koja dodaje novog studenta sa zadatim atributima u slučaju da već ne postoji
- `public void deleteSubject(Predmet p)` – metoda koja uklanja zadati predmet iz lista položenih i nepoloženih predmeta kod studenata kod kojih postoji i osvježava prikaz tabele studenata

- `public Student findStudent(int row)` – metoda koja vraća objekat klase Student za selektovani red u tabeli studenata
- `public void deleteStudent(int row)` – metoda koja briše selektovanog studenta iz baze studenata i osvježava prikaz tabele studenata
- `public boolean checkSubject(Student s, Predmet pr)` – metoda koja provjerava da li postoji zadati predmet kod zadatog studenta u njegovim listama položenih i nepoloženih ispita kao i da li je student još imao taj predmet
- `public void addGrade(Student s, Ocena o)` – metoda koja zadatom studentu upisuje zadatu ocjenu i osvježava prikaz tabele studenata
- `public void removeSubject(Student s, Predmet p)` – metoda koja će kod zadatog studenta iz njegovih lista položenih ili nepoloženih predmeta izbrisati zadati predmet ukoliko postoji

3.6 StudentFocusListener

Klasa implementira FocusListener i služi za validaciju unijetih podataka od strane korisnika prilikom dodavanja novog entiteta ili izmjene postojećeg. Polja i konstruktore nema.

Metode klase su:

- `public void focusGained(FocusEvent e)` – metoda koja određuje šta će da se desi kada polje dobije fokus : u slučaju da je tekst polja bio crven postaće crn
- `public void focusLost(FocusEvent e)` - metoda koja određuje šta će da se desi kada polje izgubi fokus: pokupiće tekst unijet u polje i provjeriće da li je u odgovarajućem formatu, u slučaju da nije boja tog teksta postavlja se na crvenu, zatim gleda sva polja na dijalogu i omogućuje dugme za potvrdu u slučaju da nema neispravnih ili praznih tekst polja(u suprotnom dugme za potvrdu je onemogućeno)

3.7 CheckValue

Ovo je klasa koja sadrži metode koje provjeravaju ispravnost unijetog teksta od strane korisnika prilikom rada sa aplikacijom. Sav tekst mora da bude u skladu sa zadatim regularnim izrazima. Klasa ne sadrži polja i konstruktore.

Metode klase su:

- `public static boolean checkNameAndSurname(String s)` – metoda koja provjerava ispravnost unijetog teksta za ime i prezime

- `public static boolean checkDate(String s)` - metoda koja provjerava ispravnost unijetog teksta za datum
- `public static boolean checkAdress(String s)` – metoda koja provjerava ispravnost unijetog teksta za adresu
- `public static boolean checkPhone(String s)` – metoda koja provjerava ispravnost unijetog teksta za broj telefona
- `public static boolean checkEmail(String s)` – metoda koja provjerava ispravnost unijetog teksta za email
- `public static boolean checkIndex(String s)` – metoda koja provjerava ispravnost unijetog teksta za broj indeksa
- `public static boolean checkYear(String s)` – metoda koja provjerava ispravnost unijetog teksta za godinu
- `public static boolean checkAvg(String s)` – metoda koja provjerava ispravnost unijetog teksta za prosječnu ocjenu
- `public static boolean checkIdCard(String s)` – metoda koja provjerava ispravnost unijetog teksta za broj lične karte
- `public static boolean checkSubName(String s)` – metoda koja provjerava ispravnost unijetog teksta za naziv predmeta
- `public static boolean checkCode(String s)` – metoda koja provjerava ispravnost unijetog teksta za šifru predmeta
- `public static boolean checkESPB(String s)` – metoda koja provjerava ispravnost unijetog teksta za broj ESPB bodova

3.8 SerializeClass

Ovo je klasa koja služi za serijalizaciju sistema. Implemetirana je kao singleton.

Polja klase su:

- `private List<Predmet> subjects` – lista predmeta koji se nalaze u sistemu
- `private List<Student> students` – lista studenata koji se nalaze u sistemu
- `private List<Profesor> professors` – lista profesora koji se nalaze u sistemu

Konstruktor:

- `public SerializeClass()` – konstruktor koji kreira objekat klase i postavlja ga na inicijalne vrijednosti

Metode klase su:

- `public void serialize()` – metoda koja je zadužena za serijalizaciju: uzima sve entitete iz baza i upisuje ih u "objectstream.txt" fajl prilikom zatvaranja aplikacije

- `public void deserialize()` – metoda koja je zadužena za deserijalizaciju
- `public List<Predmet> getSubjects()` – generisani getter za listu predmeta
- `public void setSubjects(List<Predmet> subjects)` – generisani setter za listu predmeta
- `public List<Student> getStudents()` – generisani getter za listu studenata
- `public void setStudents(List<Student> students)` – generisani setter za listu studenata
- `public List<Profesor> getProfessors()` – generisani getter za listu profesora
- `public void setProfessors(List<Profesor> professors)` – generisani setter za listu profesora

3.9 SearchController

U pitanju je klasa koja je zadužena za pretraživanje entiteta u tabelama glavnog prozora. Implementirana je kao singleton. Nema polja i konstrukture.

Metode klase su:

- `public void searchStud()` – metoda za pretragu studenata
- `public void searchProf()` – metoda za pretragu profesora gdje moguć unos dijela prezimena, odnosno imena kao kriterijuma pretrage
- `public void searchSub()` – metoda za pretragu predmeta gdje je moguć unos dijela naziva predmeta kao kriterijuma pretrage

4. VIEW

4.1 MainFrame

Klasa nasleđuje Jframe i predstavlja glavni prozor aplikacije koji se otvara prilikom pokretanja aplikacije. Glavni prozor je $\frac{3}{4}$ visine i širine veličine ekrana. Implementiran je kao singleton.

Polja klase su:

- `private ResourceBundle resourceBundle` – za lokalizaciju
- `privateToolBar tb` – toolbar aplikacije koji se nalazi ispod menija
- `private MyMenuBar menu` – meni aplikacije
- `private MyStatusBar status` – statusbar aplikacije, nalazi se u dnu
- `private TabbedPane tabbedPane` – paneli sa tabelama za entitete, glavni dio aplikacije

- `private static MainFrame instance` – instanca glavnog prozora

Konstruktor:

- `public MainFrame()` – kreira se glavni prozor sa svim svojim komponentama, ovdje se poziva i metoda za deserijalizaciju, kreira se i klasa `WindowListener` gdje je implementirana metoda „`public void windowClosing(WindowEvent e)`“ koja je zadužena za zatvaranje aplikacije i serijalizaciju

Metode aplikacije su:

- `public void changeLanguage()` – metoda koja ponovo inicijalizuje komponente glavnog prozora u cilju mijanjanja jezika
- `public void initComponents()` – metoda koja postavlja naziv aplikacije u zavisnosti od željenog jezika
- `public ResourceBundle getResourceBundle()` – generisani setter za polje „resourceBundle“

4.2 ToolBar

Klasa predstavlja tool bar aplikacije koji sadrži 4 dugmeta i jedno polje za unos teksta.

Polja klase su:

- `private JButton btnAdd` - dugme koje služi za otvaranje dijaloga za kreiranje novog entiteta informacionog sistema
- `private JButton btnEdit` - dugme koje služi za otvaranje dijaloga za izmjenu označenog entiteta
- `private JButton btnDelete` - dugme koje briše označeni enitet iz sistema
- `private JButton btnSearch` – dugme za pretragu tabele aktivnog taba
- `public static JTextField searchField` – polje za unos kriterijuma pretrage

Konstruktor klase:

- `public ToolBar()` – kreira izgled toolbar-a: postavlja ikonice, tooltip-ove i akcije

Metode klase su:

- `private Object dodavanjeEntiteta()` – metoda zadužena za otvaranje dijaloga za dodavanje odgovarajućeg novog entiteta
- `private Object izmjenaEntiteta()` – metoda zadužena za otvaranje dijaloga za izmjenu označenog entiteta
- `private Object brisanjeEntiteta()` – metoda zadužena za otvaranje dijaloga za brisanje označenog entiteta

- `public void initComponents()` – metoda koja prevodi sve tooltip-ove u slučaju promjene jezika aplikacije

4.3 MyMenuBar

Klasa predstavlja menu bar aplikacije. Sadrži 4 menija sa njihovim stavkama.

Polja klase su:

- `private JMenu mnuAdministracija` - meni za promjenu jezika
- `private JMenu file` – File meni
- `private JMenu edit` – Edit meni
- `private JMenu help` – Help meni
- `JMenuItem menuItemNew` - Dodavanje novog entiteta u sistem
- `JMenuItem menuItemClose` - Zatvaranje aplikacije
- `JMenuItem menuItemEdit` - Izmjena postojećeg entiteta
- `JMenuItem menuItemDelete` - Brisanje postojećeg entiteta
- `JMenuItem menuItemHelp` - sadrži detaljan opis o načinu korišćenja aplikacije
- `JMenuItem menuItemAbout` - Prikaz verzije aplikacije, kao i kratak opis iste i biografija svakog autora
- `private JCheckBoxMenuItem mniSrpski` – prevodi aplikaciju na srpski
- `private JCheckBoxMenuItem mniEngleski` – prevodi aplikaciju na engleski

Konstruktor:

- `public MyMenuBar()` – kreira menu bar, postavlja sve menije kao i odgovarajuće akcije na svaku stavku menija

Metode klase su:

- `public void initComponents()` – metoda koja prevodi sve stavke klase na izabrani jezik

4.4 MyStatusBar

Klasa koja predstavlja status bar aplikacije. Sadrži trenutno vrijeme, datum i naslov aplikacije.

Polje:

- `JLabel appName` – ime aplikacije

Konstruktor:

- `public MyStatusBar()` – kreira status bar u dnu glavnog prozora aplikacije

Metode:

- `public void initComponents()` – metoda koja prevodi sve stavke klase na izabrani jezik

4.5 TabbedPane

Klasa koja kreira potrebne tabove aplikacije. To su tabovi za profesore, studente i predmete koji se prikazuju ispod toolbar-a na glavnom prozoru aplikacije. Implementirana je kao singleton.

Polja klase su:

- `private static TabbedPane instance` – instanca klase
- `private JScrollPane scrollPaneStud` – scrollpane na koji ide tabela studenata
- `private JScrollPane scrollPaneProf` – scrollpane na koji ide tabela profesora
- `private JScrollPane scrollPaneSub` – scrollpane na koji ide tabela predmeta

Konstruktor:

- `public TabbedPane()` – kreira tabove, i implemetirano je sortiranje tabela

Metode klase su:

- `public void refreshStudentTable()` – metoda koja osvježava prikaz tabele studenata
- `public void refreshProfessorTable()` – metoda koja osvježava prikaz tabele profesora
- `public void refreshSubjectTable()` – metoda koja osvježava prikaz tabele predmeta
- `public void initComponents()` – metoda koja prevodi komponente klase na zadati jezik

4.6 AbstractTableModelPredmet

Klasa nasleđuje klasu AbstractTableModel i implementira njene metode. Sadrži prazan konstruktor. Služi kao model za tabelu predmeta.

Metode:

- `public int getColumnCount()` – vraća broj kolona u tabeli predmeta
- `public int getRowCount()` – vraća broj redova u tabeli predmeta

- `public String getColumnName(int column)` – vraća ime željene kolone tabele predmeta
- `public Object getValueAt(int rowIndex, int columnIndex)` – vraća objekat u željenom polju tabele predmeta

4.7 PredmetiJTable

Klasa koja kreira tabelu predmeta koja se postavlja u odgovarajući tab na glavnom prozoru. Implementirana je kao singleton. Sadrži konstruktor: `public PredmetiJTable()`, koji postavlja model tabele, izgled i mogućnost selektovanja. Od metoda ima: `public Component prepareRenderer(TableCellRenderer renderer, int row, int column)`, metoda koja je zadužena da selektovani red ima drugačiju boju od ostalih.

4.8 AbstractTableModelProfesor

Klasa nasleđuje klasu AbstractTableModel i implementira njene metode. Sadrži prazan konstruktor. Služi kao model za tabelu profesora.

Metode:

- `public int getColumnCount()` – vraća broj kolona u tabeli profesora
- `public int getRowCount()` – vraća broj redova u tabeli profesora
- `public String getColumnName(int column)` – vraća ime željene kolone tabele profesora
- `public Object getValueAt(int rowIndex, int columnIndex)` – vraća objekat u željenom polju tabele profesora

4.9 ProfesoriJTable

Klasa koja kreira tabelu profesora koja se postavlja u odgovarajući tab na glavnom prozoru. Implementirana je kao singleton. Sadrži konstruktor: `public PofesoriJTable()`, koji postavlja model tabele, izgled i mogućnost selektovanja. Od metoda ima: `public Component prepareRenderer(TableCellRenderer renderer, int row, int column)`, metoda koja je zadužena da selektovani red ima drugačiju boju od ostalih.

4.10 AbstractTableModelStudent

Klasa nasleđuje klasu AbstractTableModel i implementira njene metode. Sadrži prazan konstruktor. Služi kao model za tabelu studenata.

Metode:

- `public int getColumnCount()` – vraća broj kolona u tabeli studenata
- `public int getRowCount()` – vraća broj redova u tabeli studenata
- `public String getColumnName(int column)` – vraća ime željene kolone tabele studenata
- `public Object getValueAt(int rowIndex, int columnIndex)` – vraća objekat u željenom polju tabele studenata

4.11 StudentJTable

Klasa koja kreira tabelu studenata koja se postavlja u odgovarajući tab na glavnom prozoru. Implementirana je kao singleton. Sadrži konstruktor: `public StudentJTable()`, koji postavlja model tabele, izgled i mogućnost selektovanja. Od metoda ima: `public Component prepareRenderer(TableCellRenderer renderer, int row, int column)`, metoda koja je zadužena da selektovani red ima drugačiju boju od ostalih.

5. VIEW.DIALOGS

5.1 AboutDialog

Klasa koja kreira dijalog koji se otvara nakon izbora opcije „About“ iz menija aplikacije. Prozor sadrži kratak opis aplikacije, verziju aplikacije i sažetu biografiju autora aplikacije.

Konstruktor:

- `public AboutDialog()` – zadužen za kreiranje samog dijaloga koji se centira u odnosu na glavni prozor

5.2 AddProfessorDialog

Klasa koja kreira dijalog za dodavanje novog profesora u sistem(mode 1) ili za izmjenu izabranog profesora iz tabele(mode 2). Da bi se dijalog otvorio potrebno je da aktivni tab bude tab za profesore i zatim izborom odgovarajućih opcija iz menija ili toolbar-a otvara se dijalog koji je modalan i centriran u odnosu na glavni prozor koji ga poziva. Sadrži formu za unos(ili izmjenu) informacija o profesoru, dugme za potvrdu unosa „Potvrdi“ i dugme za odustanak od unosa „Odustani“. Da bi bilo moguće pritisnuti dugme za potvrdu sva polja forme moraju biti validno popunjena. Nakon potvrde unosa sadržaj tabele profesora se ažurira. U slučaju izmjene profesora pojavljuje se i dodatni tab na dijalogu (pored taba sa formom za izmmjenu informacija) koji prikazuje listu predmeta koje taj profesor predaje.

Polja klase su:

- `public static AddProfessorDialog instance` – instanca klase
- `public static JTextField firstName` – polje za unos imena
- `public static JTextField lastName` – polje za unos prezimena
- `public static JTextField dateOfBirth` – polje za unos datuma rođenja
- `public static JTextField homeAddress` – polje za unos adrese stanovanja
- `public static JTextField phoneNum` – polje za unos kontakt telefona
- `public static JTextField email` – polje za unos email adrese
- `public static JTextField officeAddress` – polje za unos adrese kancelarije
- `public static JTextField idCard` – polje za unos broja lične karte
- `public static JComboBox<String> position, title` – polje sa mogućim opcijama za zvanje i poziciju profesora
- `public static JButton btnOk` – dugme za potvrdu unosa
- `public Profesor professor` – kreirani profesor
- `public SubjectList predaje` – panel sa listom predmeta koje profesor predaje

Konstruktor:

- `public AddProfessorDialog(int mode)` – kreira odgovarajući dijalog i postavlja sve akcije

Metode klase su:

- `private GridBagConstraints constraintLbl(int x,int y)` – metoda za kreiranje ograničenja GridBagLayout-a prilikom kreiranja labela
- `private GridBagConstraints constraintTF(int x,int y)` – metoda za kreiranje ograničenja GridBagLayout-a prilikom kreiranja tekstualnih polja

5.3 AddStudentDialog

Klasa koja kreira dijalog za dodavanje novog studenta u sistem(mode 1) ili za izmjenu izabranog studenta iz tabele(mode 2). Da bi se dijalog otvorio potrebno je da aktivni tab bude tab za studente i zatim izborom odgovarajućih opcija iz menija ili toolbar-a otvara se dijalog koji je modalan i centriran u odnosu na glavni prozor koji ga poziva. Sadrži formu za unos(ili izmjenu) informacija o studentu, dugme za potvrdu unosa „Potvrdi“ i dugme za odustanak od unosa „Odustani“. Da bi bilo moguće pritisnuti dugme za potvrdu sva polja forme moraju biti validno popunjena. Nakon potvrde unosa sadržaj tabele studenata se ažurira. U slučaju izmjene studenta pojavljuje se i dva dodatna taba: jedan koji prikazuje položene predmete izabranog studenta i drugi koji prikazuje nepoložene ispite.

Polja klase su:

- public static JPanel firstName, lastName, dateOfBirth, adress, phoneNum, email, numOfIndex, year, currentYear, studentStatus, avgGrade,centralPanel
- private JLabel labFirstName, labLastName, labBirthday, labAdress, labPhoneNum, labEmail, labNumOfIndex, labYear, labCurrYear, labStatus, labAvg
- public static JTextField textFName, textLName, textDate, textAdress, textPhone, textEmail, textIndex, textYear, textAvg
- public static JComboBox<String> yearsComboBox, statusComboBox
- public static JButton okButton
- public Student stud
- public NepolozeniIspiti nepolozeni
- public PolozeniIspiti polozeni

Konstruktor:

- public AddStudentDialog(int mode) – zadužen za kreiranje odgovarajućeg dijaloga i postavljanje akcija na dugmad i polja

5.4 AddSubjectDialog

Klasa koja kreira dijalog za dodavanje novog predmtea u sistem(mode 1) ili za izmjenu izabranog predmeta iz tabele(mode 2). Da bi se dijalog otvorio potrebno je da aktivni tab bude tab za predmte i zatim izborom odgovarajućih opcija iz menija ili toolbar-a otvara se dijalog koji je modalan i centriran u odnosu na glavni prozor koji ga poziva. Sadrži formu za unos(ili izmjenu) informacija o predmetu, dugme za potvrdu unosa „Potvrdi“ i dugme za odustanak od

unosa „Odustani“. Da bi bilo moguće pritisnuti dugme za potvrdu sva polja forme moraju biti validno popunjena. Nakon potvrde unosa sadržaj tabele predmeta se ažurira.

Polja klase su:

- private JPanel subName, subCode, semester, year, espb, professor
- private JLabel labName, labCode, labSemester, labYear, labESPB, labProf
- public static JTextField textName, textCode, textESPB, textProf
- public static JButton okButton, cancelButton, addProf, removeProf
- public static JComboBox<String> semesterComboBox, yearsComboBox
- private Predmet sub

Konstruktor:

- public AddSubjectDialog(int mode) - zadužen za kreiranje odgovarajućeg dijaloga i postavljanje akcija na dugmad i polja

5.5 AddSubjectToProfessor

Prilikom selektovanja nekog profesora iz tabele i odgovarajuće opcije za izmjenu entiteta dobijamo dijalog za izmjenu selektovanog profesora sa 2 taba, zatim izborom drugog taba „Predmeti“ vidimo dugme „Dodaj predmet“ čijim izborom dobijamo novi dijalog koji kreira ova klasa. Na dijalogu će biti prikazana lista predmeta koje je moguće dodati selektovanom profesoru. Na dijalogu postoje dugme za potvrdu i odustanak od dodavanja. Potvrdom izabrani predmet se prikazuje u tabu predmeta koje profesor predaje. Ponuđeni su predmeti koje ne predaje nijedan drugi profesor. Dijalog se otvara centriran u odnosu na dijalog za izmjenu profesora koji ga poziva.

Polja klase su:

- public static JButton btn1, btn2 – dugme za potvrdu i odustanak od dodavanja

Konstruktor:

- public AddSubjectToProfessor() – kreira željeni dijalog, popunjava odgovarajućim predmetima i implementira odgovarajuće akcije za dugmad dijaloga

5.6 DeleteProfessor

Klasa koja kreira dijalog za brisanje entiteta. Do otvaranja ovog dijaloga dolazi prilikom izbora odgovarajućih opcija za brisanje entiteta iz menija ili toolbar-a ili upotrebom

mnemonika i akceleratora za brisanje. Potrebno je da bude selektovan jedan profesor iz tabele. Dijalog nudi opcije potvrde brisanja entiteta i odustanak od brisanja. Izborom dugmeta za potvrdu dolazi do brisanja izabranog profesora iz sistema i ažuriranje prikaza tabele profesora.

Konstruktor:

- `public DeleteProfessor()` – kreira dijalog

5.7 DeleteStudent

Klasa koja kreira dijalog za brisanje entiteta. Do otvaranja ovog dijaloga dolazi prilikom izbora odgovarajućih opcija za brisanje entiteta iz menija ili toolbar-a ili upotrebom mnemonika i akceleratora za brisanje. Potrebno je da bude selektovan jedan student iz tabele. Dijalog nudi opcije potvrde brisanja entiteta i odustanak od brisanja. Izborom dugmeta za potvrdu dolazi do brisanja izabranog studenta iz sistema i ažuriranje prikaza tabele studenata.

Konstruktor:

- `public DeleteStudent()` – kreira dijalog

5.8 DeleteSubject

Klasa koja kreira dijalog za brisanje entiteta. Do otvaranja ovog dijaloga dolazi prilikom izbora odgovarajućih opcija za brisanje entiteta iz menija ili toolbar-a ili upotrebom mnemonika i akceleratora za brisanje. Potrebno je da bude selektovan jedan predmet iz tabele. Dijalog nudi opcije potvrde brisanja entiteta i odustanak od brisanja. Izborom dugmeta za potvrdu dolazi do brisanja izabranog predmeta iz sistema i ažuriranje prikaza tabele predmeta.

Konstruktor:

- `public DeleteSubject()` – kreira dijalog

5.9 DodavanjePredmeta

Klasa koja kreira dijalog koji se iscrtava nakon izbora dugmeta za dodavanje predmeta u tabelu kod nepoloženih predmeta studenta u okviru dijaloga za izmjenu izabranog studenta. Dijalog se otvara centriran u odnosu na dijalog za izmjenu koji ga poziva. Sadrži listu predmeta koje je moguće dodati izabranom studentu, vodeći računa o godini studija tog studenta. U dnu se nalaze dugme za potvrdu izbora i odustanak. Potvrdom se predmet prikazuje u tabeli nepoloženih predmeta kod izabranog studenta.

Polja klase su:

- `private JButton okButton, cancelButton` – dugme za potvrdu i odustanak
- `public DefaultListModel<String> model` – model liste
- `public String row`
- `private Student s`
- `public JList<String> subjects` – lista predmeta

Konstruktor:

- `public DodavanjePredmeta()` – zadužen za kreiranje željenog dijaloga i postavljanje odgovarajućih akcija

5.10 HelpDialog

Klasa zadužena za kreiranje dijaloga koji se iscrtava prilikom izbora „Help“ opcije iz menija aplikacije. Prozor sadrži detaljan opis o načinu korišćenja aplikacije. Objašnjava kako se svaka od funkcionalnosti aplikacije može sprovesti u djelo i to u vidu niza korisničkih akcija. Takođe, navodi i prečice (akceleratori) koje naprednim korisnicima mogu olakšati rad. Dijalog se iscrtava centriran u odnosu na glavni prozor koji ga poziva.

Konstruktor:

- `Public HelpDialog()` – kreira željeni dijalog

5.11 NepolozeniIspiti

Klasa koja kreira panel koji prikazuje nepoložene ispite izabranog studenta. Panel se nalazi u okviru dijaloga za izmjenu studenta na trećem tabu („Nepoloženi“). Panel u centralnu dijelu sadrži tabelu sa nepoloženim predmetima a iznad tabele nalaze se 3 dugmeta: „Dodaj“, „Obriši“ i „Polaganje“. Prije izbora bilo kog dugmeta neophodno je da izaberemo predmet. Dijalog se otvara centriran u odnosu na dijalog za izmjenu studenta koji ga poziva.

Polja klase su:

- `private static Nepolozenilspiti instance` – instanca klase
- `private Predmet pr` – predmet koji želimo da prikazemo u tabeli
- `public static JButton btn1` – dugme koje služi da dodamo novi predmet u tabelu
- `public static JButton btn2` – dugme koje služi da obrišemo neki predmet iz tabele
- `public static JButton btn3` – dugme koje otvara dijalog za unos ocjene za selektovani predmet
- `public JTable nepolozeniJTable` – tabela koju kreiramo

Konstruktor:

- `public Nepolozenilspiti()` – kreira željeni dijalog i postavlja odgovarajuće akcije na svako dugme

5.12 OdaberiProfesora

Prilikom dodavanja ili izmjene predmeta dobijamo dijalog koji ima dugme „+“ koje omogućava dodavanje profesora na predmet. Odabirom tog dugmeta otvara se novi dijalog za čije je kreiranje zadužena ova klasa. Ovaj dijalog za odabir profesora iscrtava se centiran u odnosu na glavni prostor. U centralnom dijelu nalazi se lista profesora koje je moguće staviti na taj predmet a u dnu se nalaze dva dugmeta za potvrdu ili odustanak od odabira. U slučaju potvrde profesor se prikazuje u polju za profesora a predmet se dodaje u tabelu predmeta koje taj profesor predaje.

Polja klase su:

- `private JButton okButton, cancelButton` – dugme za potvrdu i odustanak
- `public JList<String> professors` – lista profesora
- `public DefaultListModel<String> model` – model liste
- `public String row`
- `private Predmet pr`

Konstruktor:

- `public OdaberiProfesora(int mode)` – kreira dijalog i postavlja odgovarajuće akcije na dugmad

5.13 Polozenilspiti

Klasa koja kreira panel koji prikazuje položene ispite izabranog studenta. Panel se nalazi u okviru dijaloga za izmjenu studenta na drugom tabu („Položeni“). Panel u centralnu dijelu sadrži tabelu sa položenim predmetima a iznad tabele nalazi se dugme: „Poništi ocenu“. Prije izbora dugmeta neophodno je da izaberemo predmet koji želimo da poništimo. Kada poništimo ocjenu predmet se prebacuje iz tanele položenih predmeta u tabelu nepoloženih predmeta izabranog studenta. U donjem desnom uglu panela nalaze se informacije o prosječnoj ocjeni, odnosno o ukupnom broju osvojenih ESPB bodova u toku studija. Dijalog se otvara centiran u odnosu na dijalog za izmjenu studenta koji ga poziva.

Polja klase su:

- `public JTable polozeniJTable` – tabela koja sadrži položene predmete
- `public static JTextArea t` – dio za prosječnu ocjenu
- `private static PolozeniIspiti instance` – instanca klase

Konstruktor:

- `public PolozeniIspiti()` – kreira željeni panel, postavlja odgovarajuću akciju na dugme i popunjava tabelu odgovarajućim informacijama

5.14 SubjectList

Klasa predstavlja panel koji prikazuje tabelu predmeta koje izabrani profesor predaje. Panel sadrži dugme za dodavanje i uklanjanje predmeta iz te tabele. Panel se nalazi u tabu „Predmeti“ kod dijaloga za izmjenu profesora.

Polja klase su:

- `public static JButton btn1, btn2` – dugme za dodavanje novog predmeta profesoru i dugme za uklanjanje postojećeg predmeta profesoru
- `public static JTable table` – tabela koju popunjavamo predmetima i prikazujemo
- `public DefaultTableModel model` – model tabele koji koristimo
- `private Predmet pr` – predmet koji upisujemo u tabelu
- `private Profesor prof` – profesor od kog uzimamo predmete za prikaz

Konstruktor:

- `public SubjectList()` – konstruktor koji je zadužen za kreiranje željenog panela, implementira i odgovarajuće akcije prilikom izbora dugmadi

5.15 UnosOcjene

Klasa kreira dijalog sa formom za upis ocjene izabranom studentu za neki njegov nepoloženi predmet. Da bi došli do ovog dijaloga potrebno je izabrati izmjenu nekog studenta i pozicionirati se u tab sa njegovim nepoloženim predmetima, zatim treba od prikazanih predmeta izabrati onaj za koji želimo upisati ocjenu i izborom dugmeta „Polaganje“ otvara nam se dijalog za unos ocjene. Referent ima mogućnost unosa brojčane vrijednosti ocjene u intervalu od 6 do 10, kao i datuma polaganja ispita u odgovarajuća polja. Naziv i šifru predmeta nije moguće mijanjati. Na samom dnu dijaloga nalaze se dugmad za potvrdu i odustanak od unosa. Samo ukoliko su sva polja validno unijeta moguće je potvrdi unos. Rezultat akcije je da se izabrani predmet prebaci u tabelu položenih ispita kod tog studenta.

Polja klase su:

- `public static JButton btn1, btn2` – dugme za potvrdu unosa i dugme za odustanak od unosa
- `public static JTextField code, name, date` – tekstualna polja za šifru predmeta, naziv predmeta i datum polaganje
- `public JComboBox<String> ocena` – moguće opcije za vrijednost ocjene
- `public static JLabel codeLbl, nameLbl, ocenaLbl, dateLbl` – labela za šifru predmeta, naziv predmeta, ocjenu i datum polaganja

Konstruktor:

- `public UnosOcjene(Predmet p)` – kreira željeni dijalog

6. O autoru

Dokumentaciju je napisala Petra Jović (**RA 29-2018**), student Fakulteta tehničkih nauka u Novom Sadu.

Kontakt: jovic.petra99@gmail.com

