

Klausur
Programmierung II – WWIBE120/WWIBE220 – Q2 2021

Matrikelnummer

--

Hinweise

- Bearbeitungszeit: 60 Minuten
- Maximale Punktzahl: 50 + 2
- Hilfsmittel: keine

Punkte

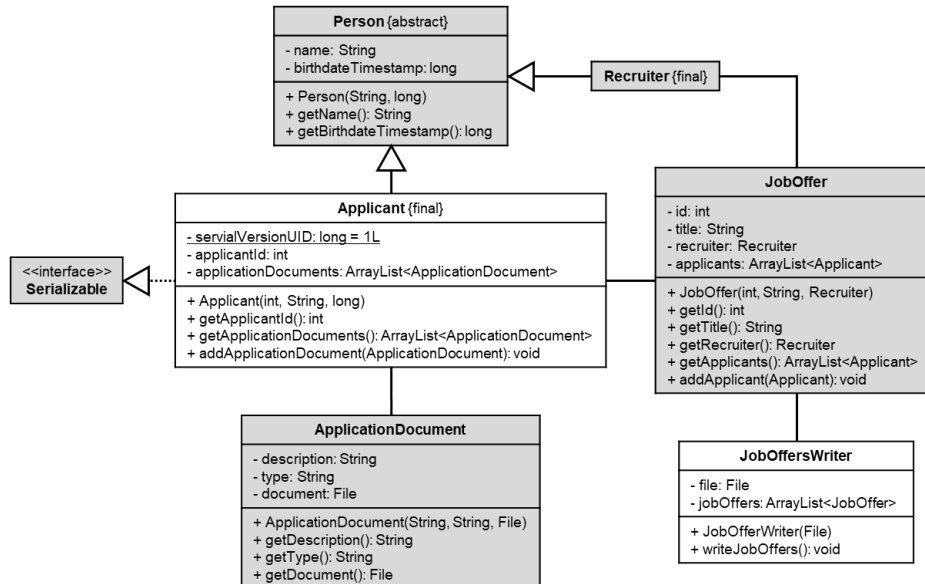
Aufgabe	Maximale Punktzahl	Erreichte Punktzahl
1	14	
2	18	
3	20	
Gesamt	52	

Viel Erfolg

Aufgabe 1

- a) Erläutern Sie kurz was man unter einer Oberklasse und einer Unterklasse versteht (2 Punkte)
- b) Erläutern Sie kurz, wozu sich Klassen verpflichten, die eine Schnittstelle realisieren (2 Punkte)
- c) Benennen Sie die Datenstromklasse zum bytebasierten Lesen von Daten aus einer Datei (1 Punkt)
- d) Erläutern Sie kurz, was man unter der Catch-or-Throw-Regel versteht (2 Punkte)
- e) Erläutern Sie kurz den wesentlichen Unterschied zwischen einer absoluten und einer relativen Pfadangabe (2 Punkte)
- f) Skizzieren und erläutern Sie kurz den Aufbau einer JavaFX-Oberfläche (3 Punkte)
- g) Erläutern Sie kurz, was die Besonderheit der Schnittstelle *Serializable* darstellt (2 Punkte)

- Erstellen Sie die Klasse *Applicant* anhand des abgebildeten Klassendiagramms (8 Punkte)
- Erstellen Sie die Klasse *JobOffersWriter* anhand des abgebildeten Klassendiagramms (10 Punkte)



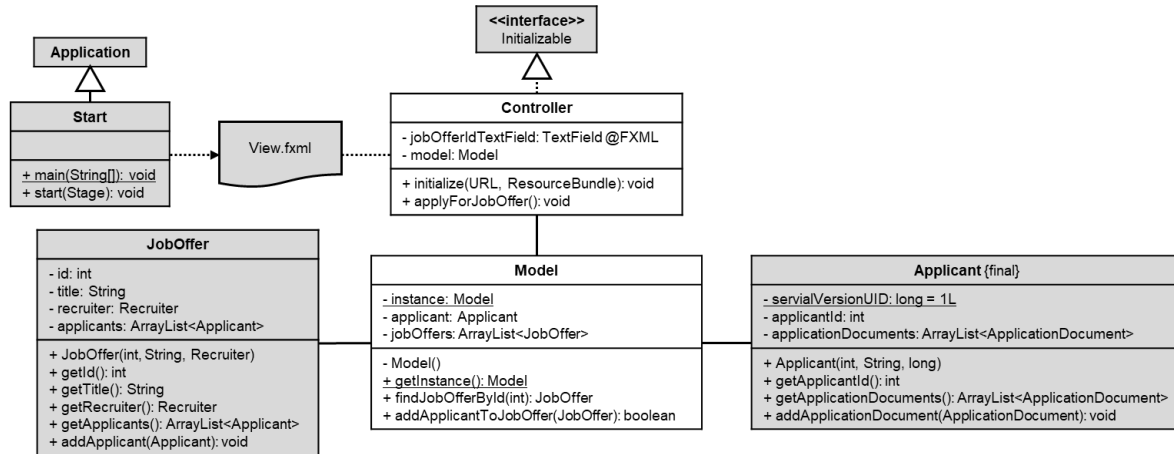
- der Konstruktor soll alle Instanztribute initialisieren bzw. mit den eingehenden Werten belegen

- der Konstruktor soll alle Instanzattribute initialisieren bzw. mit den eingehenden Werten belegen
- die Methode `writeJobOffers()` soll alle Stellenangebote (Attribut `jobOffers`) gemäß der Formel `<id>;<title>;<recruiter.name>` in die Datei (Attribut `file`) schreiben. Fangen Sie die Ausnahme `IOException` ab und lösen Sie als Fehlerbehandlung die Ausnahme `JobOfferIOException` aus. Hinweis: der Inhalt der Datei soll bei jedem Methodenaufruf überschrieben werden

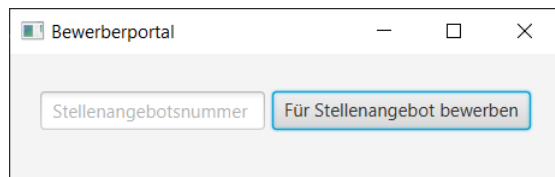
Aufgabe 3

- Erstellen Sie die Klasse *Model* anhand des abgebildeten Klassendiagramms (9 Punkte)
- Erstellen Sie die Klasse *Controller* anhand des abgebildeten Klassendiagramms (11 Punkte)

Klassendiagramm



Grafische Darstellung



Hinweise zur Klasse Model

- der Konstruktor soll alle Instanzattribute initialisieren. Verwenden Sie hierzu die Methode *getApplicant()* der Klasse *ApplicantReader* sowie die Methode *getJobOffers()* der Klasse *JobOffersReader*
- die Methode *findJobOfferById(int)* soll zur eingehenden Stellenangebotsnummer das dazugehörige Stellenangebot in den Stellenangeboten (Attribut *jobOffers*) suchen und dieses im Erfolgsfall zurückgeben. Im Fehlerfall soll der Wert *null* zurückgegeben werden
- die Methode *addApplicantToJobOffer(JobOffer)* soll den Bewerber (Attribut *applicant*) zum eingehenden Stellenangebot hinzufügen
- die Klasse *Model* soll als Singleton implementiert werden

Hinweise zur Klasse Controller

- die Methode *initialize(URL, ResourceBundle)* soll das Model initialisieren
- die Methode *applyForJobOffer()* soll zur eingegebenen Stellenangebotsnummer das dazugehörige Stellenangebot suchen (Methode *findJobOfferById(int)*) und diesem anschließend den Bewerber hinzufügen (Methode *addApplicantToJobOffer(JobOffer)*). Geben Sie für den Fall, dass es sich bei der Eingabe um keine ganze Zahl handelt einen Dialog (*AlertType.ERROR*) mit der Fehlermeldung „Bitte eine Ganzzahl eingeben“ aus und für den Fall, dass zur eingegebenen Stellenangebotsnummer kein dazugehöriges Stellenangebot gefunden werden konnte, einen Dialog (*AlertType.ERROR*) mit der Fehlermeldung „Bitte eine gültige Stellenangebotsnummer eingeben“. Brechen Sie in beiden Fehlerfällen die Verarbeitung der Methode ab. Hinweis: die Methode *valueOf(String)* der Klasse *Integer* kann die Ausnahme *NumberFormatException* auslösen.