

Wiederholklausur
Programmierung II – WWIBE120/WWIBE220 – Q4 2021

Matrikelnummer

--

Hinweise

- Bearbeitungszeit: 60 Minuten
- Maximale Punktzahl: 50 + 1
- Hilfsmittel: keine

Punkte

Aufgabe	Maximale Punktzahl	Erreichte Punktzahl
1	14	
2	19	
3	18	
Gesamt	51	

Viel Erfolg

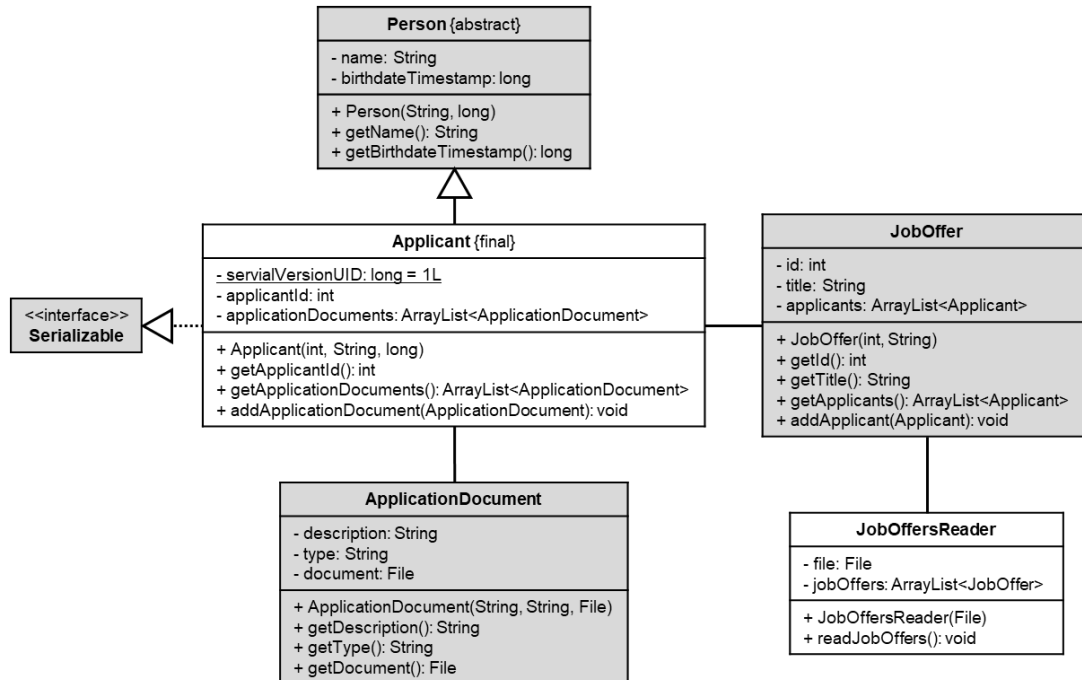
Aufgabe 1

- a) Skizziere und Erläutere kurz die verschiedenen Lebens-Phasen einer JavaFX-Anwendung (4 Punkte)
- b) Erläutere kurz, was man unter Serialisierung versteht (2 Punkte)
- c) Benenne die Datenstromklasse zum zeichenbasierten Schreiben von Daten in eine Datei (1 Punkt)
- d) Erläutere kurz den wesentlichen Unterschied zwischen einer absoluten und einer relativen Pfadangabe (2 Punkte)
- e) Skizziere und Erläutere kurz das MVC-Pattern (3 Punkte)
- f) Erläutere kurz, was man unter dem Begriff *Separation of Concerns* versteht (2 Punkte)

Aufgabe 2

- a) Erstelle die Klasse *Applicant* anhand des abgebildeten Klassendiagramms (8 Punkte)
- b) Erstelle die Klasse *JobOffersReader* anhand des abgebildeten Klassendiagramms (11 Punkte)

Klassendiagramm



Hinweis zur Klasse Applicant

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

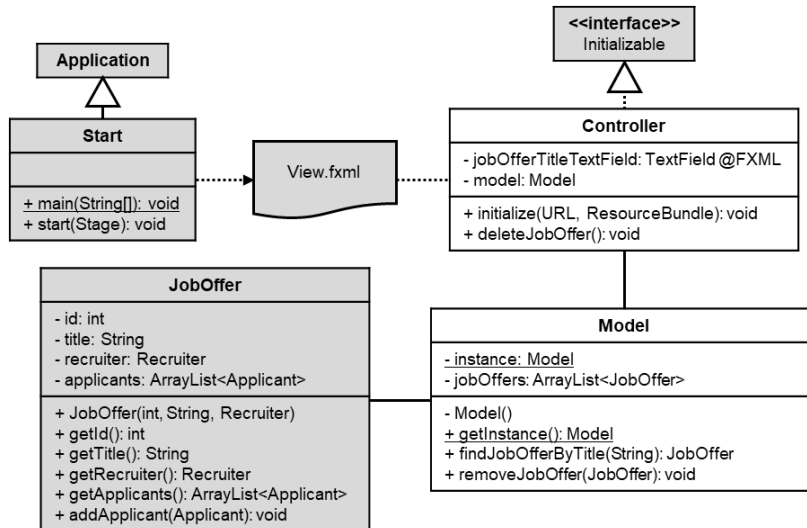
Hinweise zur Klasse JobOffersReader

- der Konstruktor soll alle Instanztribute initialisieren bzw. mit den eingehenden Werten belegen
- die Methode `readJobOffers()` soll alle Einträge aus der Datei (Attribut `file`) lesen, daraus entsprechende Objekte der Klasse `JobOffer` erstellen und diese den Stellenangeboten (Attribut `jobOffers`) hinzufügen. Zudem soll die Ausnahme `IOException` abgefangen und als Fehlerbehandlung die Ausnahme `JobOfferIOException` ausgelöst werden
- die Datei mit den Stellenangeboten ist wie folgt aufgebaut: `<id>;<title>`

Aufgabe 3

- a) Erstelle die Klasse *Model* anhand des abgebildeten Klassendiagramms (9 Punkte)
- b) Erstelle die Klasse *Controller* anhand des abgebildeten Klassendiagramms (9 Punkte)

Klassendiagramm



Grafische Darstellung

Bewerberportal	
<input type="text" value="Stellenangebotsbezeichnung"/>	<input type="button" value="Stellenangebot löschen"/>

Hinweise zur Klasse Model

- der Konstruktor soll alle Instanzattribute initialisieren. Hierzu soll die statische Methode `getJobOffers()` der Klasse `JobOffersReader` verwendet werden
- die Methode `findJobOfferByTitle(String)` soll zur eingehenden Stellenangebotsbezeichnung das dazugehörige Stellenangebot in den Stellenangeboten (Attribut `jobOffers`) suchen und dieses im Erfolgsfall zurückgeben. Im Fehlerfall soll der Wert `null` zurückgegeben werden
- die Methode `removeJobOffer(JobOffer)` soll das eingehende Stellenangebot aus den Stellenangeboten (Attribut `jobOffers`) entfernen
- die Klasse `Model` soll als Singleton implementiert werden

Hinweise zur Klasse Controller

- die Methode `initialize(URL, ResourceBundle)` soll das `Model` initialisieren
- die Methode `deleteJobOffer()` soll zur eingegebenen Stellenangebotsbezeichnung das dazugehörige Stellenangebot suchen (Methode `findJobOfferByTitle(String)`) und dieses anschließend aus den Stellenangeboten entfernen (Methode `removeJobOffer(JobOffer)`). Für den Fall, dass zur eingegebenen Stellenangebotsbezeichnung kein dazugehöriges Stellenangebot gefunden wird, soll ein Dialog (`AlertType.ERROR`) mit der Fehlermeldung „Bitte eine gültige Stellenangebotsbezeichnung eingeben“ angezeigt und die Verarbeitung der Methode abgebrochen werden