Kennziffer

Hochschule des Bundes für öffentliche Verwaltung
Fachbereich Finanzen
Studiengang Verwaltungsinformatik

Klausurteil Programmieren

Grundlagen der Programmierung (mit Java)

(M08, Hauptstudium)

Prüfungstag: 22.07.2016

Bearbeitungszeit: 60 Minuten

Umfang: 1 Aufgabe, max. 60 Punkte

Hilfsmittel: VIT-Notebook mit deaktiviertem Netzwerkzugriff

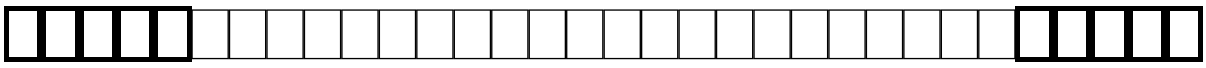
Gewicht für Modulprüfung: 27%

Hinweise

- Die Verbindung mit jeder Art von Netzwerk (z.B. WLAN, Bluetooth oder LAN) wird als Täuschungsversuch gewertet und führt zum Ausschluss von der Klausur.
  - Trennen Sie Ihr Notebook durch Entfernen des Netzwerkkabels vom LAN.
  - Deaktivieren Sie die Funknetzwerke mit dem entsprechenden Hardware-Schalter Ihres Notebooks (roter Hintergrund am Schalter sichtbar).
- Die einzigen zulässigen USB-Geräte, die mit Ihrem Notebook verbunden sein dürfen, sind eine kabelgebundene Maus sowie der zur Verfügung gestellte USB-Speicher-Stick.

  Das Verbinden anderer als der zuvor angegebenen USB-Geräte wird als Täuschungsversuch gewertet und führt zum Ausschluss von der Klausur.

- Bearbeitung der Aufgabe
  - Nutzen Sie die Entwicklungsumgebung *Eclipse* zur Erstellung des Java-Codes.
  - Legen Sie zur Bearbeitung ein Projekt an und benennen dieses nach Ihrer Kennziffer.
    - *Datei – Neu – Java-Projekt*
- Exportieren der Lösung
  - Am Ende der Bearbeitungszeit exportieren Sie das Eclipse-Projekt in eine ZIP-Datei, die Ihre Kennziffer als Namen trägt.
    - *Datei – Exportieren – Allgemein – Archivdatei*
    - Exportieren Sie auch die Eclipse-Projekt-Datei `.project`
  - Kopieren Sie die ZIP-Datei auf den bereitgestellten USB-Stick.
- Erstellen des digitalen Fingerabdrucks
  - Erzeugen Sie mithilfe des Programms *HashCheck* die MD5-Prüfsumme der ZIP-Datei.
    - *Rechtsklick – Eigenschaften – Prüfsummen*
  - Notieren Sie mindestens die ersten fünf und die letzten fünf Stellen der MD5-Prüfsumme



Notieren Sie bitte oben sowie auf allen weiteren Blättern Ihre Kennziffer.

Wir wünschen Ihnen viel Erfolg!

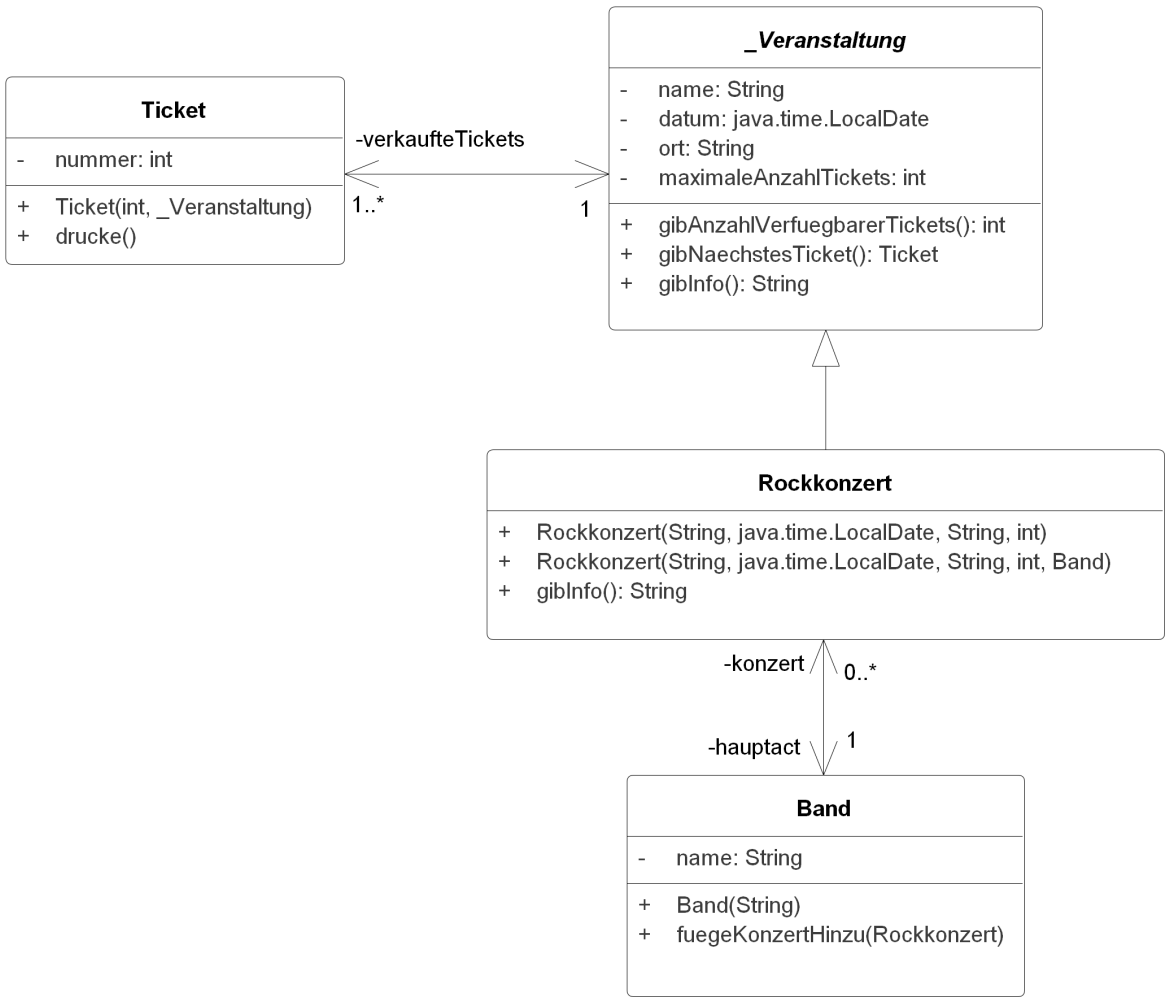


Abbildung 1: UML-KLassendiagramm eines Ticketsystems

Aufgabe 1 (60 Punkte)

(a) Implementieren Sie das in Abbildung 1 gezeigte UML-Klassendiagramm unter Berücksichtigung der auf Seite 4 f. genannten Hinweise.

(b) Implementieren Sie die Klasse `Vorverkaufsstelle`, die eine `main`-Methode enthält. In der `main`-Methode sollen folgende Anweisungen implementiert werden:

1. Erzeugen Sie ein Rockkonzert der Band „Die zwei lustigen Drei“, für das es 5000 Eintrittskarten gibt. Das Konzert findet am 22.7.2016 im Rahmen der Tour „Wir rocken die Welt“ in Münster statt.
2. Lassen Sie in einer `for`-Schleife 4999 Tickets generieren, die Sie in einer Liste speichern.
3. Prüfen Sie, ob tatsächlich nur noch ein Ticket verfügbar ist.
4. Lassen Sie sich auch dieses Ticket generieren und lassen Sie es auf der Konsole ausgeben.
5. Zeigen Sie, dass das nächste Generieren eines Tickets `null` zurück gibt.

Hinweise zur Implementierung

- Alle Konstruktoren initialisieren die Attribute mit den Werten der Parameter
- Erzeugen Sie nur die Getter- und Setter-Methoden, die zur fehlerfreien Ausführung des Codes notwendig sind.
- Mit Ausnahme der Getter- und Setter-Methoden sind alle Methoden und alle Attribute, die nicht im UML-Klassendiagramm genannt sind, mit Javadoc-Kommentaren zu beschreiben.
- Verwenden Sie zur Repräsentation eines Datums die Klasse `LocalDate` aus dem Package `java.time`.
  - Eine Instanz dieser Klasse erzeugen Sie über den Aufruf der statischen Methode `of()` der Klasse `LocalDate`:
    `LocalDate.of(2000, Month.JANUARY, 1)`

Hinweise zum UML-Diagramm

- `_Veranstaltung`
  - Diese **abstrakte** Klasse repräsentiert Veranstaltungen wie Konzerte, Filmaufführungen oder Opernbälle. Für jede Veranstaltung gibt es eine begrenzte Menge Eintrittskarten (Tickets).
  - Die Methoden der Klasse sind nicht abstrakt!
  - `maximaleAnzahlTickets` Die Anzahl von Tickets, die insgesamt für diese Veranstaltung verkauft werden können.
  - `gibAnzahlVerfuegbarerTickets` Gibt die Anzahl noch verfügbarer Tickets zurück.
  - `gibNaechstesTicket` Generiert unter Verwendung der nächsten Ticketnummer ein Ticket für diese Veranstaltung.
  - Für eine Veranstaltung können nicht mehr als die maximal zulässige Anzahl von Eintrittskarten generiert werden. Sollten dennoch weitere Tickets angefordert werden, so gibt `gibNaechstesTicket` `null` zurück.
  - `gibInfo` Gibt die Infos zur Veranstaltung (Name, Datum, Ort) als Zeichenkette zurück, die zum Aufdruck auf die Eintrittskarte benötigt werden.
- `Rockkonzert`
  - Die Klasse repräsentiert Veranstaltungen, bei denen eine Band auftritt.
  - `gibInfo` Gibt die Infos zum Konzert als Zeichenkette zurück, die zum Aufdruck auf die Eintrittskarte benötigt werden. In Ergänzung zur `_Veranstaltung` ist hier auch der Name der Band wichtig.

- `Band`
  - Die Klasse repräsentiert Musikgruppen, die einen Namen haben und Musik machen.
  - `fuegeKonzertHinzu` fügt dieser Band einen Auftritt auf einem Rockkonzert hinzu.
- Ticket
  - Die Klasse repräsentiert Eintrittskarten zu einer Veranstaltung. Alle Eintrittskarten einer Veranstaltung sind mit eindeutigen Nummern versehen. Die Tickets werden aufsteigend nummeriert. Die Nummerierung beginnt bei 1.
  - `drucke` gibt diese Eintrittskarte auf der Konsole aus. Dies umschließt die allgemeinen Infos der Veranstaltung bzw. zum Konzert sowie die Nummer der Eintrittskarte.