

Name:

Platznummer:

Aufgabe verfügbar auf:

## Java

### 1

- Bringen Sie die folgenden Begriffe in eine sinnvolle Reihenfolge (Entstehung bis Ausführung eines Java Programms)
  - Quelltext
  - Java virtual machine (JVM)
  - realer Prozessor
  - Bytecode
  - Java Compiler

Antwort:

```
1. Quelltext
2.
3.
4.
5. realer Prozessor
```

### 2

- Erläutern Sie die Begriffe `Syntax`, `Semantik`, `Pragmatik` anhand eines Beispiels. (java oder it oder natürliche Sprache)

Antwort:

```
- Semantik:

- Syntax:

- Pragmatik:
```

## Klassen und Objekte / UML

### 1

- Gegeben seien zwei Klassen: `Soldat` und `Personalakte`
- Soldaten haben einen `Namen` (`String`) und eine `Personalakte`
  - Ein Soldat besitzt 2 Konstruktoren: `Soldat()`, `Soldat(String initName, Personalakte initPersonalakte)`
- Eine `Personalakte` besteht aus einem `Inhalt` (`String`) und einer `Personal Nummer` (ganze Zahl)
  - Eine `Personalakte` besitzt einen Konstruktor: `Personalakte(int initPersNu)`
- Methoden (Verhalten) besitzen diese Klassen erstmal keine, allerdings vergessen Sie nicht die Konstruktoren abzubilden
- Nur Ihre Soldaten Klasse soll später mit der Hauptklasse (`public class Main` und `public void main(String[] args)`) kommunizieren können.
  - Die Klasse `Soldat` und seine Konstruktoren besitzen die Sichtbarkeit: `public`
  - Die Sichtbarkeit einer `Personalakte` ist: `package`
  - Alle Attribute haben die Sichtbarkeit: `private`

1. Modellieren Sie diese Klassen in UML, beachten Sie die Beziehung zwischen den 2 Klassen (Klassendiagramm)
2. Überlegen Sie sich sinnvolle Attribut Namen (gerne auf Englisch) / achten Sie dabei besonders auf die Groß- und Kleinschreibung

Antwort:

## 1.1

- Wie wird ein Konstruktor bezeichnet, der keinen Parameter übergeben bekommt?
  - z.B.: `Soldat()`

Antwort:

## 2

- Gegeben sei ein Objekt: `Erika Musterfrau`.
- Das Objekt entsteht aus der Klasse `Soldat`
- In ihrer Personalakte steht `tadellos`
- Ihre Personal Nummer lautet: `123`

1) Modellieren Sie diese(s) Objekt(e) in UML (Objektdiagramm):

Antwort:

## 3

Erläutern Sie die Begriffe `Objekt` und `Klasse`. Verwenden Sie dabei bitte folgende Stichwörter: `Zur Laufzeit`; `Zur Zeit der Entwicklung`; `statisches Programmelement`; `dynamisches Programmelement`

Antwort:

## 4

Gegeben ist folgendes Codefragment:

```
1 /**
2  *
3  * Enumeration Dienstgrad.
4  * Hier werden ein Teil der moeglichen Dienstgrade festgelegt.
5  *
6  * @author paul
7  */
8 public enum Dienstgrad {
9     UNTEROFFIZIER, OBERGEFREITER, OBERSTABSGEFREITER, GEFREITER;
10 }
```

### 4.1

- Wie wird der Inhalt in Zeile 1-7 genannt?

Antwort:

### 4.2

- Bilden Sie dieses Enumeration korrekt in UML ab

Antwort:

# Klassen

## 1.1 (4 Punkte)

- Korrigieren Sie die Fehler des nachstehenden Quellcodes
  - 4 Fehler haben sich eingeschlichen

```
1 public clas Soldat
2
3     private String name;
4     private _____ meinePersonalakte
5
6
7
8 //Hier ist der Platz für Konstruktoren
9
10 // Hier werden Später Methoden geschrieben/implementiert
11
12 }
```

## 1.2

- Schreiben Sie für die Klasse `Soldat` 2 Konstruktoren
  - Standard Kontruktor (default constructor):
    - Initialisiert den Namen mit "Erika Musterfrau"
    - Das Attribut `meinePersonalakte` vom Datentyp `Personalakte` wird wie folgt initialisiert:
      - `this.meinePersonalakte = new Personalakte(123);`
  - Ein Konstruktor mit 2 Parametern: Namen und Personalakte
    - `Soldat(String initName, Personalakte initPersonalakte)`
- Beide Konstruktoren sind öffentlich sichtbar (`public`)

# Methoden

## 1

- Für Ihre Klasse `Soldat` schreiben Sie bitte eine Ausführende Methode (ohne Rückgabewert) mit folgenden Eigenschaften:
  - Sie ist öffentlich zugänglich (`public`)
  - Methoden Name: `halloAnAlle`
  - Es werden keine Parameter übergeben
- Die Methode soll auf der Konsole folgendes ausgeben:
  - `Hallo ich heiße (Name des Soldaten).`
  - `(Name des Soldaten)` soll dabei das Attribut `name` der Klasse `Soldat` entsprechen
    - z.B.: `Hallo ich heiße Erika Musterfrau`

# Hauptklasse/Main

## 1

- Erstellen Sie ein Objekt: `erika` mit Hilfe des Standard Konstruktor (default constructor)
- Das Objekt `erika` soll die Methode `halloAnAlle()` aufrufen
- Erstellen Sie ein Objekt: `testSoldat`
  - Überlegen Sie sich selbst: Namen und Personalnummer
  - zur Erinnerung: `Soldat(String initName, Personalakte initPersonalakte)`
- Das Objekt `testSoldat` soll die Methode `halloAnAlle()` aufrufen
- Weisen Sie dem Objekt `testSoldat`, mit Hilfe des Standard Konstruktor, neue Werte zu

```
public class Main {
```

```
public static void main(String[] args) {  
  
    }  
}
```