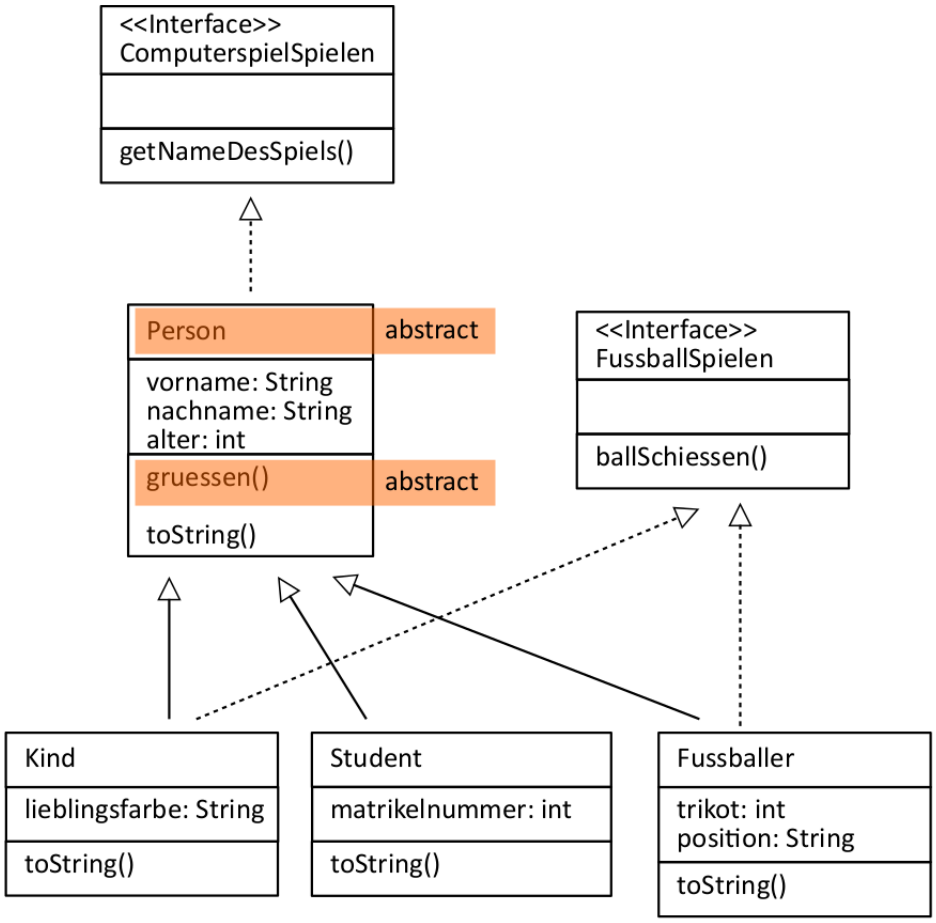
Übung 22 – Kinder, Studenten, Fußballer

In dieser Übung schreiben Sie ein paar Klassen und Interfaces, damit Sie die folgenden Konzepte bzw. Techniken erleben können:

- `abstract` Klasse
- Interface
- `ArrayList` mit `Comparable` anordnen
- `ArrayList` mit `Comparator` anordnen
- `ArrayList` mit `Iterator` durchlaufen

Anleitungen



Das obige Diagramm stellt unsere Klassen und Interfaces dar. Schreiben Sie den entsprechenden Code, um die Klassen und Interfaces zu implementieren. Natürlich sollen Sie auch ein MainProgramm schreiben, um Ihren Code zu testen.

Allgemeine Anleitungen (z.B. Attribute, Konstruktoren, `toString()`-Methoden, usw.) werden nicht in jedem Absatz unten beschrieben. Unten werden nur die besonderen Merkmale und Hinweise detailliert.

Teil 0: Vorbereitung

- Laden Sie die Eclipse-Projekt-Zip-Datei aus Moodle herunter.
- Achtung: Bitte die neue Version 1.1.11 vom AutoBewertungs-Programm herunterladen und benutzen.

Teil 1: Interfaces und Klassen schreiben

Alle Codes, die Sie schreiben, sollen in dem Package `personen` sein.

`ComputerspielSpielen`-Interface

- `getNameDesSpiels()`-Methode:
  - Eingabe: Keine.
  - Rückgabe: ein `String` – Name des Spiels.

`Person`-Klasse

- Die Klasse ist `abstract`.
- Wegen AutoBewertung bitte die folgende Reigenfolge der Eingaben beim Konstruktor benutzen: Vorname, Nachname, Alter.
- `gruessen()`-Methode:
  - Die Methode ist `abstract`.
  - Eingabe: Keine.
  - Rückgabe: Keine.

`FussballSpielen`-Interface

- `ballSchiessen()`-Methode:
  - Eingabe: ein `int` – eine Länge in Metern.
  - Rückgabe: Keine.

`Kind`-Klasse

- Wegen AutoBewertung bitte die folgende Reigenfolge der Eingaben beim Konstruktor benutzen: Vorname, Nachname, Alter, Lieblingsfarbe.
- `gruessen()`-Methode:

- Die Methode druckt in der Console einen Gruß, in dem den Namen sowie die Lieblingsfarbe erwähnt werden, z.B.:

  ```
  Hallo! Mein Name ist Anna Alt und ich liiiiiebe die Farbe rot!
  ```

- `ballSchiessen()`-Methode:
  - Wenn die Eingabe n Meter ist, wird der Ball für nur $0.7n$ Meter geschossen.
  - Die Farbe des Balls ist natürlich die Lieblingsfarbe des Kindes.
  - Die Methode druckt in der Console eine Zeile, in der die Farbe des Balls sowie die Distanz des Schusses erwähnt werden, z.B. wenn $n = 12$:

    ```
    Ein roter Ball ist fuer 8.4m geschossen.
    ```

    Drucken Sie die Distanz mit 1 Nachkommastelle. Dafür benutzen Sie nochmal `DecimalFormat`, aber mit einem anderen Format:

    Beispiel(e)

    ```
    DecimalFormat decimalFormat = new DecimalFormat("#.0");
    // # --> Ziffer optional
    // 0 --> Ziffer erforderlich
    ```

- `getNameDesSpiels()`-Methode:
  - Die Methode gibt den Namen "Super Mario" zurück.

`Student`-Klasse

- `gruessen()`-Methode:
  - Die Methode druckt in der Console einen Gruß mit dem Namen, z.B.:

    ```
    Hallo! Mein Name ist Bob Berg.
    ```

- `getNameDesSpiels()`-Methode:
  - Die Methode gibt den Namen "Minecraft" zurück.

`Fussballer`-Klasse

- Wegen AutoBewertung bitte die folgende Reigenfolge der Eingaben beim Konstruktor benutzen: Vorname, Nachname, Alter, Trikot, Position.
- `gruessen()`-Methode:
  - Die Methode druckt in der Console einen Gruß mit einem (explizit erwähnten) Faustst oß , z.B.:

    ```
    Hallo! Mein Name ist Manuel Neuer. *Fauststoss!*
    ```

- `ballSchiessen()`-Methode:
  - Wenn die Eingabe n Meter ist, wird der Ball für genau n Meter geschossen.
  - Die Methode druckt in der Console eine Zeile, in der die Distanz des Schusses erwähnt wird, z.B. wenn $n = 12$:

    ```
    Ein Ball ist fuer 12.0m geschossen.
    ```

    Drucken Sie die Distanz mit 1 Nachkommastelle.

- `getNameDesSpiels()`-Methode:
  - Die Methode gibt den Namen "FIFA 21" zurück.

Methoden testen

Fragen Sie sich selbst:

- Welche Objekte mit welchen Attribut-Werten sollen Sie erzeugen, um die verschiedenen Methoden gründlich zu testen?

Teil 2: `Person`-Objekte nach Default-Ordnung anordnen

Wir möchten `Person`-Objekte in einer `ArrayList` nach der folgenden Default-Ordnung (natural order) anordnen:

- 1. Kriterium: Alter.
- 2. Kriterium: Nachname.
- 3. Kriterium: Vorname.

Fragen Sie sich selbst:

- Welches Interface – `Comparable` oder `Comparator` – ist hier geeignet? Und damit welche Methode bzw. Klasse sollen Sie schreiben?
- Welche Objekte mit welchen Attribut-Werten sollen Sie erzeugen, um Ihren Code gründlich zu testen?

Weitere Tipps:

- Benutzen Sie die `Collections.shuffle()`-Methode, um die `ArrayList` zu mischen, bevor Sie Ihren Code testen.
- Nach dem Anordnen drucken Sie jedes Objekt in der `ArrayList` "direkt", damit die entsprechende `toString()`-Methode automatisch benutzt wird.
- Üben Sie die zwei Formen, die die Objekte in der `ArrayList` durchläuft:
  - Mit einer `for`-Schleife
  - Mit `Iterator`

Teil 3: `Fussballer`-Objekte anordnen

Wir möchten `Fussballer`-Objekte in einer `ArrayList` nach der folgenden drei Ordnungen anordnen.

1. Ordnung:

- Kriterium: Trikot.

2. Ordnung:

- 1. Kriterium: Position nach alphabetischer Anordnung, d.h. Abwehr, Mittelfeld, Stürmer, Torwart.
- 2. Kriterium: Nachname.

3. Ordnung:

- 1. Kriterium: Alter in absteigender Ordnung. Beispielsweise ist das Folgende eine gültige Anordnung:

  Neuer mit 35, Hummels mit 32, Müller mit 31

- 2. Kriterium: Vorname.

Fragen Sie sich selbst:

- Welches Interface – `Comparable` oder `Comparator` – ist hier geeignet? Und damit welche Methode bzw. Klasse sollen Sie schreiben?

  Wegen AutoBewertung müssen leider hier die Antworten verraten werden. Bitte benutzen Sie genau die folgenden Klassen-Namen:

  - `FussballerComparatorTrikot`
  - `FussballerComparatorPositionNachname`
  - `FussballerComparatorAlterVorname`

- Welche Objekte mit welchen Attribut-Werten sollen Sie erzeugen, um Ihren Code gründlich zu testen?

Die Tipps vom obigen "Personen nach Default-Ordnung anordnen" Ansatz gelten hier immer noch.

Teil 4: MainProgramm

Ihr MainProgramm muss die folgenden Leistungen erfüllen:

- Benutzen Sie die gegebenen Daten. Sie finden die Daten in der `Daten`-Klasse. Rufen Sie die entsprechende Methode auf, um die erwünschten Daten abzuholen.
- Rufen Sie die entsprechenden Methoden (`gruessen()`, `getNameDesSpiels()` bzw. `ballSchiessen()`) jeweils mit einem `Kind`-, einem `Student`- und einem `Fussballer`-Objekt.
- Erstellen Sie eine `ArrayList` von Personen mit `Kind`-, `Student`- und `Fussballer`-Objekten. Mit dieser `ArrayList` ordnen Sie die Objekte nach Person-Default-Ordnung an und drücken Sie die angeordnete Reihenfolge in der Console. Dafür müssen Sie einen `Iterator` benutzen.
- Erstellen Sie eine zweite `ArrayList` nur von den Fußballern. Mit dieser `ArrayList` ordnen Sie die Objekte nach den 3 verschiedenen Ordnungen und drücken Sie die angeordneten Reihenfolgen in der Console. (Dafür ist die Art der Schleife egal – `Iterator` oder "einfache" `for`-Schleife sind beides in Ordnung.)

Achtung für AutoBewertung:

Diesmal betrachtet die AutoBewertung die Console je nach dem Test-Fall bzw. der Methode, die gerade zu testen ist. Aus diesem Grund ist das Folgende zu beachten:

- Nur die folgenden Methoden sollen in der Console drucken:
  - Die verschiedenen `gruessen()`- und `ballSchiessen()`-Methoden.
  - Alle Methoden im MainProgramm. Die AutoBewertung kontrolliert das MainProgramm nicht.
- Die andere Methoden sollen überhaupt nichts in der Console drucken.
- Zum Debugging bleibt die `System.out.println()`-Strategie nützlich, aber die entsprechenden Zeilen Code müssen vor Abgabe gelöscht oder deaktiviert werden. Der Debugger ist natürlich jederzeit eine gute Idee.

Code-Konventionen beachten:

- Kommentare
- Sinnvolle Variablennamen, auf singular/plural achten
- Kein Hardcoding
- Einrückung (indentation)
- Keine if-Bedingung wie: `if (booleanWert == true)`
- Möglichst Code-Duplizierung vermeiden bzw. Variable wiederverwenden
- Code-Effizienz
- Reihenfolge der Programm-Bestandteile (d.h. Attribute, Konstruktoren, Methoden) sowie Reihenfolge der Methoden
- Klassen, Enum und Neu: Interfaces beginnen mit Großbuchstaben, Objekte mit Kleinbuchstaben
- Klassen per MainProgramm testen
- Geheimnis-Prinzip berücksichtigen
- Vererbung verstanden
- Programming to the interface
- Neu: Iterator verstanden

Abgabe

Laden Sie im Moodle die folgenden Dateien hoch:

- `ComputerspielSpielen.java`
- `Fussballer.java`
- `FussballerComparatorAlterVorname.java`
- `FussballerComparatorPositionNachname.java`
- `FussballerComparatorTrikot.java`

- FussballSpielen.java
- Kind.java
- MainProgramm.java
- Person.java
- Student.java