

Aufgabe 1 (~12 Punkte)

Bearbeite eine der beiden folgenden Optionen:

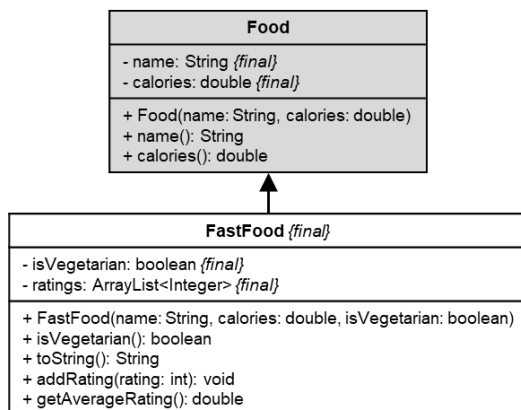
Option 1

- a) Erläutere kurz, was man unter einer **Schnittstelle (Interface)** versteht (2 Punkte)
- b) Benenne die **4 Varianztypen** bei der generischen Programmierung (2 Punkte)
- c) Erläutere kurz den wesentlichen Zweck von **inneren Klassen** (2 Punkte)
- d) Erläutere kurz den wesentlichen Unterschied zwischen **intermediären und terminalen Operationen** (2 Punkte)
- e) Erläutere kurz, was man unter der **Red-Green-Refactor-Methode** versteht (2 Punkte)
- f) Skizziere die **Testpyramide** (2 Punkte)

Option 2

Erstelle die Klasse **FastFood** anhand des abgebildeten Klassendiagramms.

Klassendiagramm



Hinweise zur Klasse FastFood

- Der Konstruktor soll alle Attribute mit den eingehenden Werten initialisieren
- Die Methode **boolean isVegetarian()** soll den Wert des Attributs **isVegetarian** zurückgeben
- Die Methode **String toString()** soll das Fast-Food in der Form *FastFood [name=[Name], calories=[Kalorien]kcal, isVegetarian=[Vegetarisch?], averageRating=[Durchschnittsbewertung]]* zurückgeben
- Die Methode **void addRating(rating: int)** der der Bewertungsliste (**ratings**) die eingehende Bewertung hinzufügen
- Die Methode **double getAverageRating()** soll die durchschnittliche Bewertung zurückgeben

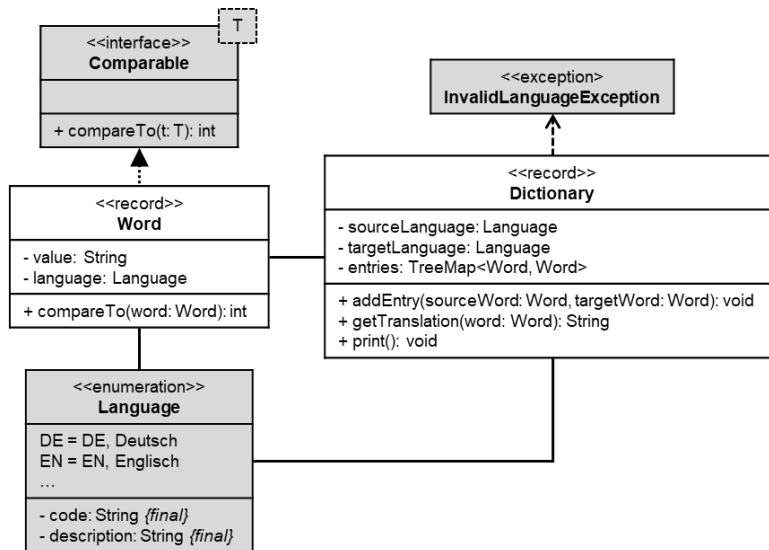
Beispielhafte Konsolenausgabe

```
FastFood [name=Veggie Burger Deluxe, calories=177kcal, isVegetarian=true, averageRating=4.8]
```

Aufgabe 2 (~15 Punkte)

Erstelle die Klassen **Word** (~3 Punkte) und **Dictionary** (~12 Punkte) anhand des abgebildeten Klassendiagramms

Klassendiagramm



Allgemeine Hinweise zum Klassendiagramm

- Der Stereotyp **record** impliziert, dass die Datenklasse einen entsprechenden Konstruktor, Getter zu allen Attributen sowie entsprechende Implementierungen für die Object-Methoden besitzt
- Der Stereotyp **enumeration** impliziert, dass die Aufzählung einen privaten Konstruktor sowie ggbf. passende Setter und Getter besitzt

Hinweis zur Klasse **Word**

Die Methode **int compareTo(word: Word)** soll so implementiert werden, dass damit Wörter aufsteigend nach ihrem Wert (**value**) sortiert werden können.

Hinweise zur Klasse **Dictionary**

- Die Methode **void addEntry(sourceWord: Word, targetWord: Word)** soll dem Wörterbuch die beiden eingehenden Wörter hinzufügen. Für den Fall, dass die Sprache des ersten eingehenden Wortes (**language**) nicht der Quellsprache des Wörterbuchs (**sourceLanguage**) entspricht, oder dass die Sprache des zweiten eingehenden Wortes (**language**) nicht der Zielsprache des Wörterbuchs (**targetLanguage**) entspricht, soll die Ausnahme **InvalidLanguageException** ausgelöst werden
- Die Methode **String getTranslation(word: Word)** soll zum eingehenden Wort die dazugehörige Übersetzung als Zeichenkette zurückgeben. Für den Fall, dass zum eingehenden Wort keine Übersetzung gefunden werden kann, soll eine leere Zeichenkette zurückgegeben werden. Achte darauf, dass das eingehende Wort entweder als Schlüssel oder als Wert vorhanden sein kann
- Die Methode **void print()** soll alle Attribute wie abgebildet auf der Konsole ausgeben

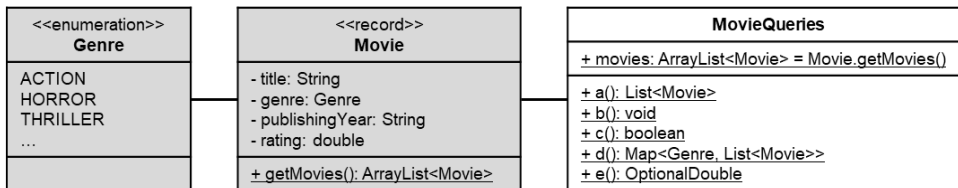
Beispielhafte Konsolenausgabe

```
Wörterbuch Deutsch - Englisch
Haus - house
Maus - mouse
...
```

Aufgabe 3 (~15 Punkte)

Erstelle die Klasse **MovieQueries** anhand des abgebildeten Klassendiagramms.

Klassendiagramm



Allgemeine Hinweise zum Klassendiagramm

- Der Stereotyp **record** impliziert, dass die Datenklasse einen entsprechenden Konstruktor, Getter zu allen Attributen sowie entsprechende Implementierungen für die Object-Methoden besitzt
- Der Stereotyp **enumeration** impliziert, dass die Aufzählung einen privaten Konstruktor sowie ggffs. passende Setter und Getter besitzt

Hinweise zur Klasse **MovieQueries**

- Die statische Methode **List<Movie> a()** soll alle Filme zwischen 1980 und 1990 als Liste zurückgeben
- Die statische Methode **void b()** soll die 5 am besten bewerteten Horror-Filme absteigend sortiert nach der Bewertung in der Form **[FILMTITEL] ([Erscheinungsjahr])** ausgeben
- Die statische Methode **boolean c()** soll zurückgeben, ob es einen Thriller aus dem Jahr 2022 mit einer Bewertung von 5,5 (inklusive) bis 6,4 (inklusive) gibt
- Die statische Methode **Map<Genre, List<Movie>> d()** soll alle Filme je Genre als Assoziativspeicher zurückgeben
- Die statische Methode **OptionalDouble e()** soll die durchschnittliche Bewertung aller Action-Filme als Gleitpunktzahl zurückgeben

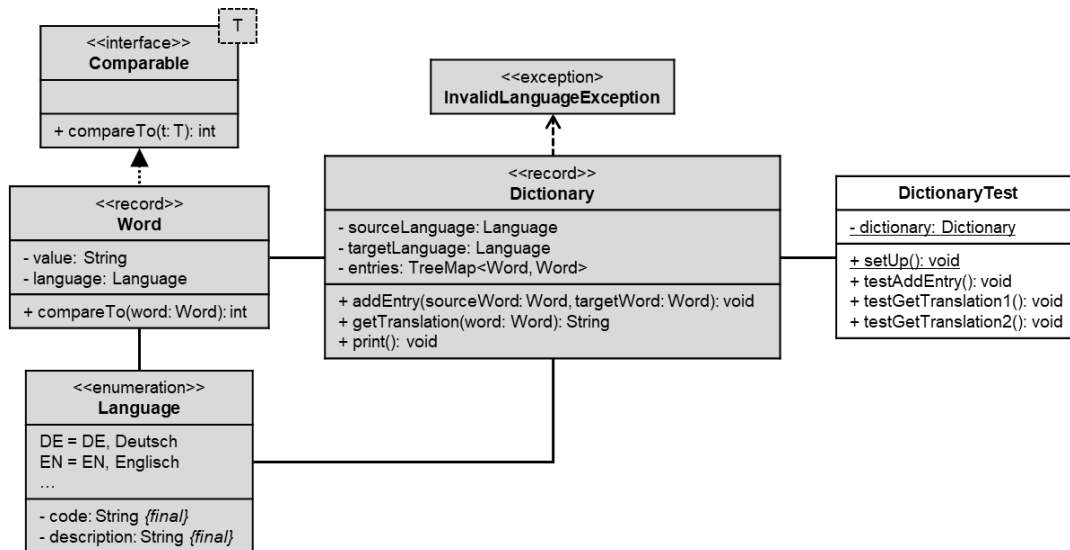
Beispielhafte Konsolenausgabe

```
a():The Matrix (1999)
```

Aufgabe 4 (~10 Punkte)

Erstelle die Testklasse **DictionaryTest** anhand des abgebildeten Klassendiagramms.

Klassendiagramm



Allgemeine Hinweise zum Klassendiagramm

- Der Stereotyp **record** impliziert, dass die Datenklasse einen entsprechenden Konstruktor, Getter zu allen Attributen sowie entsprechende Implementierungen für die Object-Methoden besitzt
- Der Stereotyp **enumeration** impliziert, dass die Aufzählung einen privaten Konstruktor sowie ggffs. passende Setter und Getter besitzt

Hinweise zur Testklasse **DictionaryTest**

- Die statische Lebenszyklus-Methode **void setUp()** soll das nachfolgende Testszenario aufbauen:
 - Es soll ein Wörterbuch (**dictionary**) mit der Quellsprache **DE** und der Zielsprache **EN** sowie einem Assoziativspeicher erstellt werden
 - Dem Assoziativspeicher soll ein Eintrag mit einem Wort mit dem Wert (**value**) **Baum** und der Sprache (**language**) **DE** als Schlüssel und einem Wort mit dem Wert (**value**) **tree** und der Sprache (**language**) **EN** als Wert hinzugefügt werden
- Die Testmethode **void testAddEntry()** soll den nachfolgenden Testfall abdecken: Beim Aufruf der Methode **void addEntry(sourceWord: Word, targetWord: Word)** mit einem Wort für Parameter 1, dessen Sprache ungleich der Quellsprache ist, und bei einem Aufruf der Methode mit einem Wort für Parameter 2, dessen Sprache ungleich der Zielsprache ist, wird die Ausnahme **InvalidLanguageException** erwartet
- Die Testmethode **void testGetTranslation1()** soll den nachfolgenden Testfall abdecken: Beim Aufruf der Methode **String getTranslation(word: Word)** mit einem Wort, zu dem eine Übersetzung in der Quellsprache vorhanden ist, und bei einem Aufruf mit einem Wort, zu dem eine Übersetzung in der Zielsprache vorhanden ist, wird als Rückgabe die entsprechende Übersetzung als Zeichenkette erwartet
- Die Testmethode **void testGetTranslation2()** soll den nachfolgenden Testfall abdecken: Beim Aufruf der Methode **String getTranslation(word: Word)** mit einem Wort, zu dem keine Übersetzung in der Quellsprache vorhanden ist, und bei einem Aufruf mit einem Wort, zu dem keine Übersetzung in der Zielsprache vorhanden ist, wird als Rückgabe eine leere Zeichenkette erwartet