

Modellierung und Programmierung 1

Übung 5

Stefan Preußner

30. November / 1. Dezember 2020

Zufallszahlen

- Die Klasse `Random` aus dem Paket `java.util` ermöglicht die Generierung von (Pseudo-)Zufallszahlen
 - Alle Klassen, die sich nicht im Paket `java.lang` befinden, müssen importiert werden. Hier ist also der Befehl `import java.util.Random` erforderlich.
- Der Zufallszahlengenerator wird mit einem *seed* initialisiert
 - Wenn dieser nicht explizit angegeben wird, dann wird die Systemzeit zur Berechnung des seeds genutzt
 - Mit der Methode `setSeed(long seed)` kann der seed explizit gesetzt werden
 - Dies vereinfacht die Fehlersuche, da die Zufallszahlen dann reproduzierbar generiert werden

Zufallszahlen

- `nextInt()` erzeugt einen neuen `int` zwischen `Integer.MIN_VALUE` und `Integer.MAX_VALUE` (zwei Konstanten, die in Java den kleinsten bzw. größten durch einen `int` darstellbaren Wert speichern)
- `nextInt(int wert)` erzeugt einen neuen `int` zwischen 0 (eingeschlossen) und `wert` (nicht eingeschlossen)
- `nextFloat()` erzeugt einen neuen `float` zwischen 0.0 (eingeschlossen) und 1.0 (nicht eingeschlossen)
- `nextDouble()` erzeugt einen neuen `double` zwischen 0.0 (eingeschlossen) und 1.0 (nicht eingeschlossen)
 - Die Auflösung ist hier höher als bei `nextFloat()`

Zufallszahlen

- Erzeugung einer zufälligen Ganzzahl zwischen untereGrenze und obereGrenze:
 - `nextInt(obereGrenze - untereGrenze) + untereGrenze`
 - `(int)(nextDouble() * (obereGrenze - untereGrenze)) + untereGrenze`

Formatierte Strings

- Formatierte Strings können mit der statischen Methode `String.format` erzeugt werden
- Syntax: `String.format(Formatstring, Wert1, Wert2, ...)`
- `Formatstring` ist hierbei ein `String`, welcher neben beliebigen Text auch Formatparameter enthalten kann - diese werden in der Reihenfolge, in der sie im `Formatstring` auftreten, durch die angegebenen Werte ersetzt
 - Formatparameter beginnen mit einem Prozentzeichen `%`
 - Eine ausführliche Beschreibung findet sich in der Java-Dokumentation der Klasse `java.util.Formatter`

Formatierte Strings

■ Beispiel 1:

- `String.format("%s %s %s", "MuP", "ist", "toll")`
- Rückgabe: `"MuP ist toll"`

■ Beispiel 2:

- `String.format("Ein Würfel hat %2d Seiten", 6)`
- Rückgabe: `"Ein Würfel hat 6 Seiten"`
- `d` steht für eine Ganzzahl, die Zahl davor ist optional und gibt die Mindestzahl an auszugebenden Zeichen an

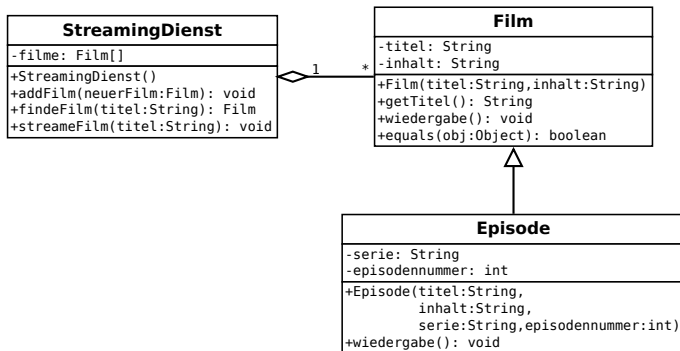
Formatierte Strings

■ Beispiel 3:

- `String.format("2 durch 3 ist %10.8f", 2.0/3)`
- Rückgabe: 2 durch 3 ist 0.66666667
- `f` steht für eine Fließkommazahl
- Die Zahl vor dem Punkt gibt die Gesamtzahl an auszugebenden Zeichen, einschließlich des Dezimalpunkts, an
- Die Zahl nach dem Punkt gibt die Anzahl an auszugebenden Nachkommastellen an
- Es wird automatisch gerundet

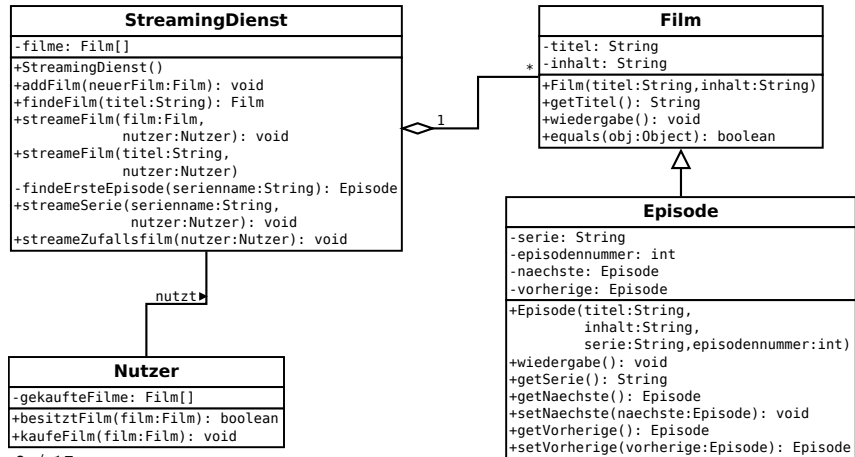
Übung - Programmierung

UML-Diagramm aus Übung 4:



Übung - Programmierung

Erweitertes UML-Diagramm:



Klasse StreamingDienst

- `filme` speichert alle Filme des Streamingdienstes
- Der Konstruktor soll ein neues `Film`-Array der Länge 100 erzeugen
- `addFilm` soll den übergebenen Film an der ersten freien Stelle in `filme` speichern
- `findeFilm` soll den Film mit dem übergebenen Titel zurückgeben. Existiert kein solcher Film, dann soll `null` zurückgegeben werden
- `streameFilm(Film film, Nutzer nutzer)` soll einen Film kaufen lassen und dann wiedergeben

Klasse StreamingDienst

- `streameFilm(String titel, Nutzer nutzer)` sucht einen Film mit dem angegebenen Titel. Ist der Film vorhanden, dann wird er gekauft und wiedergegeben.
- `findeErsteEpisode` findet die erste Episode einer Serie mit dem angegebenen Seriennamen. Existiert keine solche Serie, dann wird `null` zurückgegeben.
- `streameSerie` gibt alle Episoden einer Serie wieder
- `streameZufallsfilm` gibt einen zufälligen Film wieder

Klasse Film

- `titel` speichert den Titel, `inhalt` den Inhalt eines Films
- `wiedergabe` soll erst den Titel und dann den Inhalt ausgeben
- `equals` soll dann `true` zurückgeben, wenn das Vergleichsobjekt ein `Film` mit dem gleichen Titel ist

Klasse Episode

- `serie` speichert den Namen der Serie, zu der die Episode gehört
- `episodennummer` speichert die Nummer der Episode innerhalb der Serie
- `vorherige` bzw. `naechste` geben die vorherige bzw. nächste Episode innerhalb einer Serie an
- `wiedergabe` soll vor der Ausgabe des Titels und des Inhalts (wie in der Klasse `Film`) die Serie und die Episodennummer ausgeben
- `setNaechste` soll nicht nur `naechste` setzen, sondern auch das Attribut `vorherige` bei der nächsten Episode

Klasse Nutzer

- `gekaufteFilme` speichert bis zu 100 vom Nutzer gekaufte Filme
- `besitztFilm` gibt zurück, ob ein Nutzer einen Film bereits gekauft hat
- `kaufeFilm` fügt den angegebenen Film zur Liste der gekauften Filme hinzu, falls der Nutzer den Film nicht bereits besitzt

Klasse Main

- In der `main`-Methode soll ein neuer Streamingdienst erstellt werden
- Neue Filme und Episoden sollen erstellt und dem Streamingdienst hinzugefügt werden
- Anschließend sollen ein Film mit bekanntem Titel, eine Serie sowie ein zufälliger Film wiedergegeben werden