PM2 Java: Reguläre Ausdrücke

Parsen einer Webseite
Quellcode im Modul:
v5-RegulaereAusdrueckeWebseitenParsen

AUSGANGSPUNKT

Die Webseite

https://de.wikipedia.org/wiki/Liste_von_3D-Filmen

Uns interessiert der Inhalt ab der Überschrift 3D-Filme

3D-Filme [Bearbeiten | Quelltext bearbeiten]

Vor 1953 [Bearbeiten | Quelltext bearbeiten]

- The Power of Love – 1922
- Zum Greifen nah – 1936 (Boehner Film/Dresden)
- Koordinatensysteme – 1939
- 6 Mädels rollen ins Wochenend – 1939 (Boehner Film/Dresden)
- Robinzon Kruzo – 1947
- Bwana, der Teufel – 1952

1953 [Bearbeiten | Quelltext bearbeiten]

Beginn der Boom-Phase im 20. Jahrhundert

- Arena
- Fegefeuer
- Cat-Women of the Moon
- Sizilianische Leidenschaft
- Die letzte Patrouille
- Der brennende Pfeil
- Top Banana
- Der letzte Rebell

bis zur Überschrift 4D-Filme

- Kubo – Der tapfere Samurai (27. Oktober 2016)
- Störche – Abenteuer im Anflug (27. Oktober 2016)
- Phantastische Tierwesen und wo sie zu finden sind (16. November 2016) (r
- Sing (8. Dezember 2016)
- Rogue One: A Star Wars Story (15. Dezember 2016) (nachträglich in 3D ko
- Vaiana – Das Paradies hat einen Haken (22. Dezember 2016)
- Assassin's Creed (27. Dezember 2016)

2017 [Bearbeiten | Quelltext bearbeiten]

- Passengers (5. Januar 2017)
- Die irre Heldentour des Billy Lynn (2. Februar 2017)

4D-Filme [Bearbeiten | Quelltext bearbeiten]

In diesem Artikel oder Abschnitt fehlen folgende wichtige Informationen:
vgl. Englische Wikipedia
Du kannst Wikipedia helfen, indem du sie recherchierst und einfügst.

Diese Liste enthält Filme, die in speziellen 4D-Kinos gezeigt wurden und/oder g

- *Captain EO* mit Michael Jackson (wurde von 1986 bis 1998 in sämtlichen D zu sehen.)
- *Liebling, ich habe das Publikum geschrumpft* (gezeigt im Disneyland Paris
- *Pirates 4D* mit Leslie Nielsen und von und mit Eric Idle (derzeit gezeigt im Pl
- *Die „unmögliche" Welt des M. C. Escher* (derzeit gezeigt im Mini Mundus Bo
- *PandaVision* vom WWF (derzeit gezeigt in Efteling, Liseberg, Fårup Somme
- Bionicle-Filme, Drome-Racers-Film, Bob der Baumeister baut eine Achterba
- *Haunted House* (zu Halloween im Europa-Park)

Aufgabenstellung

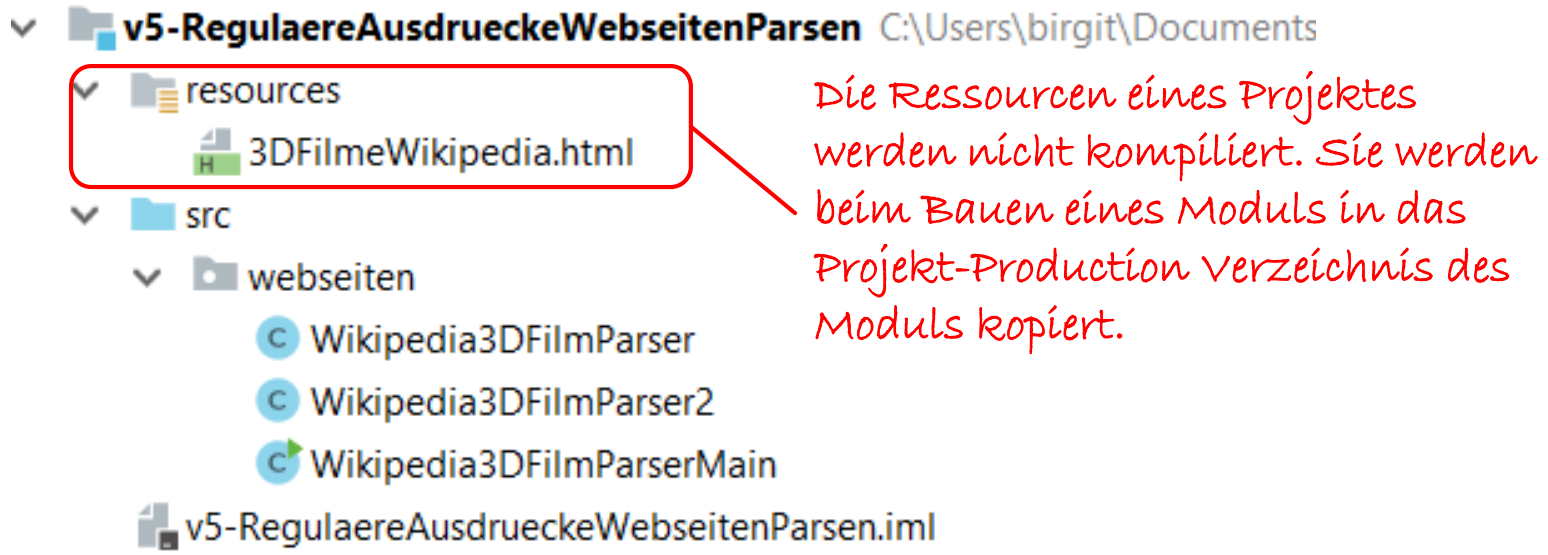
- Das Programm soll die Listen der 3D-Filme pro Jahr extrahieren und diese Filme in einem Verzeichnis sammeln. Dabei sollen alle HTML Tags beseitigt werden.
- Die Klasse soll ***Wikipedia3DFilmParser*** heißen und wird mit einer Referenz auf die Webseite erzeugt.
- Das Verzeichnis bildet Jahresangaben auf eine Liste der 3D-Filme ab.
- Für das Verzeichnis soll die Datenstruktur einer Java-Map verwendet werden.
- **Ergebnis**: siehe rechte Seite (formatierte Ausgabe ist selbst geschrieben):

```
Phantastische Tierwesen und wo sie zu finden sind
(16. November 2016) (nachträglich in 3D konvertiert)
Sing (8. Dezember 2016)
Rogue One: A Star Wars Story (15. Dezember 2016)
(nachträglich in 3D konvertiert)
Vaiana – Das Paradies hat einen Haken (22. Dezember
2016)
Assassin’s Creed (27. Dezember 2016)

2017
Passengers (5. Januar 2017)
Die irre Heldentour des Billy Lynn (2. Februar 2017)

Vor 1953
The Power of Love – 1922
Zum Greifen nah – 1936 (Boehner Film/Dresden)
Koordinatensysteme – 1939
6 Mädels rollen ins Wochenend – 1939 (Boehner
Film/Dresden)
Robinzon Kruzo – 1947
Bwana, der Teufel – 1952
```

VORBEREITUNG

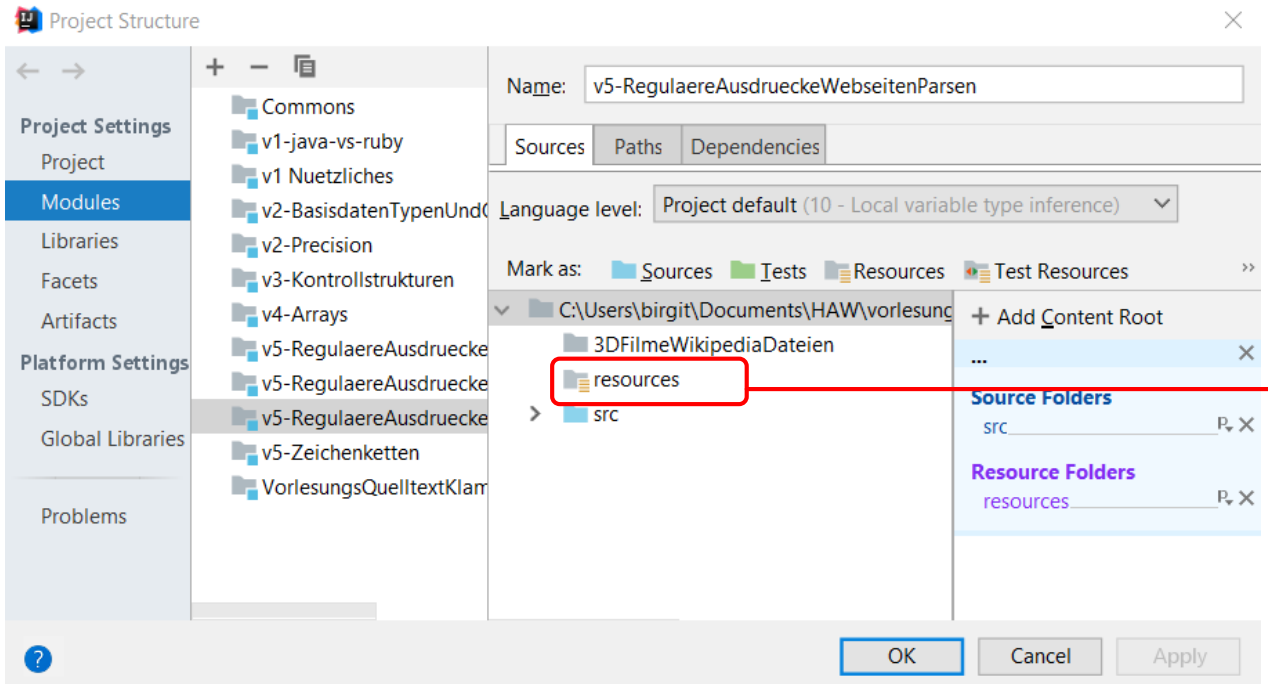
Organisation des Moduls

- Im Verzeichnis ***resources*** liegt die lokale HTML-Datei



Verzeichnisse als Ressource auszeichnen

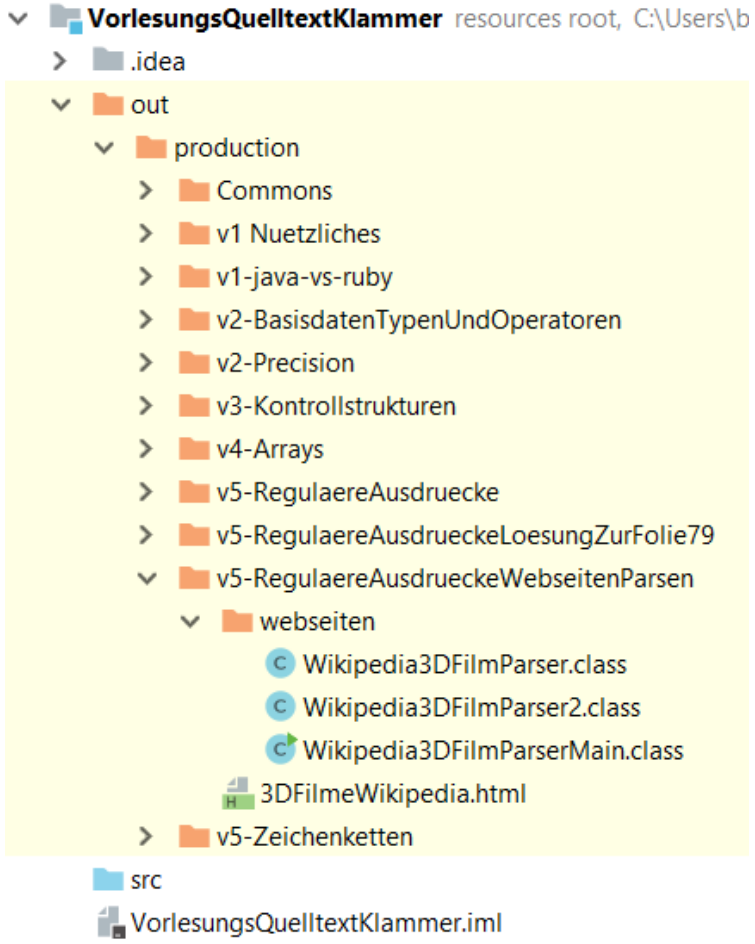
- *Open Modul Settings* im Kontextmenü des Moduls v5-RegulaereAusdrueckeWebseitenParsen dann Open, öffnet das *Project Structure* Fenster.
- Das Verzeichnis ***resources*** wurde als Resources markiert. Die Inhalte werden in das Projektverzeichnis ***out\production\v5-RegulaereAusdrueckeWebseitenParsen*** kopiert.



hier liegen die Dateien des Moduls, die nicht kompiliert werden.

Das Production-Verzeichnis des Projektes

- Das Projekt trägt den Namen ***VorlesungsQuelltextKlammer***
- Unter diesem Verzeichnis befinden sich die kompilierten Klassen der Module. Die Klassen und Package-Struktur wird unterhalb des Modul-Root-Verzeichnisses angelegt.
- Ebenso finden wir hier die Resourcen eines Moduls. Die Resourcen eines Moduls werden in das Root-Verzeichnis des Moduls kopiert.
- Analog zu ***out\production*** werden die Tests unter ***out\tests*** organisiert



Lesen der lokalen HTML-Datei

- Die Datei referenzieren wir über eine File-URI als String.
- Wir könnten auch direkt von der Webseite lesen (Bsp.: siehe Quelltext), aber in der Entwicklungs- und Testphase erzeugt das nur unnötige Zugriffe. Daher arbeiten wir lokal.

```
private static final String RESOURCE_DIR = "out\\production\\v5-
RegulaereAusdrueckeWebseitenParsen";

public static void main(String[] args) throws IOException {
    /*
    * Wir arbeiten mit einer lokalen Datei / Kopie einer Wikipediaseite.
    * Referenz auf die Datei wird als URI String übergeben: file:///<absoluter Pfad zur Datei>
    * Path wiki3DFilmLocal = Paths.get( RESOURCE_DIR +"\\3DFilmeWikipedia.html"); erzeugt einen relativen
    * Pfad zur der Datei (Bezug das aktuelle Projekt)
    * wiki3DFilmLocal.toAbsolutePath() erzeugt den absoluten Pfad, der als
    * Argument übergeben wird.
    * "file:///" beschreibt das Protokoll für das Öffnen eines URI.
    */
    Path wiki3DFilmLocal = Paths.get( RESOURCE_DIR +"\\3DFilmeWikipedia.html");
    Wikipedia3DFilmParser wp1 = new Wikipedia3DFilmParser("file:///" +
          wiki3DFilmLocal.toAbsolutePath());
```

Klasse: Wikipedia3DFilmParserMain

Speichern der URI

```
/**
 * Der Konstruktor merkt sich die uri für die Datei / entfernte Ressource.
 *
 * @param uri eine URI als Zeichenkette. Wird beim Öffnen einer Ressource
 *  vom Scanner benutzt.
 */
public Wikipedia3DFilmParser(String uri) {
   this.uri = uri;
}
```

Öffnen und Schließen der Ressource

- Wir öffnen die HTML-Datei mit einem ***Scanner***.
- Damit später auch von einer entfernten Ressource gelesen werden kann, wird aus der ***uri***, einer Zeichenkette, eine URL erzeugt und darauf ein Lesestrom geöffnet.
- Wichtig ist der Zeichensatz. Wir müssen die Datei egal ob lokal oder remote mit UTF-8 lesen.
- Nach dem Lesen muss die Ressource geschlossen werden. Dazu schließen wir den Scanner mit ***close***.

```
public void echoPage() throws IOException {
   Scanner wiki3DFilmScanner = new Scanner(new URL(uri).openStream(),
                  StandardCharsets.UTF_8);
   while (wiki3DFilmScanner.hasNextLine()) {
      System.out.println(wiki3DFilmScanner.nextLine());
   }
   wiki3DFilmScanner.close();
}
```

Lesestrom

korrekter Zeichensatz

Schließen der Ressource

sich einen Überblick verschaffen

```
/**
 * Liest den Inhalt einer Ressource zeilenweise unter Verwendung eines
 * Scanners und gibt diesen auf der Konsole aus. Dies ist sinnvoll, um sich
 * einen Überblick über den Aufbau der Seite zu verschaffen.
 * @throws IOException wenn die uri nicht korrekt ist oder die Datei /
 * entfernte Resource nicht existiert
 * @return void
 */
public void echoPage() throws IOException {
   Scanner wiki3DFilmScanner = new Scanner(new URL(uri).openStream(),
StandardCharsets.UTF_8);
   while (wiki3DFilmScanner.hasNextLine()) {
      System.out.println(wiki3DFilmScanner.nextLine());
   }
   wiki3DFilmScanner.close();
}
```

Das geht natürlich auch, wenn wir uns den Quelltext im Browser anschauen ☺.
Aber mit diesem Vorgehen können wir auf einfache Weise prüfen, ob wir die Datei öffnen können ☺.
Sinn: In der HTML Seite die Bereiche identifizieren, die Beginn und Ende der Aufzählung der 3D Filme markieren.

LÖSUNGSWEG

Lösungsidee

1. Wir **extrahieren** den Bereich der HTML Seite, der die **Aufzählung der 3D-Filme** enthält. Dazu müssen wir reguläre Ausdrücke für Anfang und Ende des Bereichs definieren.
2. Wir extrahieren **für jedes Jahr** die Jahresangabe und die Liste der 3D-Filme. Auch hierfür müssen wir einen regulären Ausdruck definieren (***THREE_DEE_ENUM***. Der reguläre Ausdruck beschreibt mit der 1‘ten Gruppe die Jahresangabe, mit der 2‘ten Gruppe die Aufzählung der 3D-Filme für das Jahr.
3. Da das Muster von ***THREE_DEE_ENUM*** mehrfach auftritt, verwenden wir die Technik des **partiellen Matchens** (Methode ***Matcher.find*** in einer Schleife).
4. Dann müssen wir aus der 2‘ten Gruppe mit den einzelnen List-Items die **Information einzelner 3D-Filme extrahieren**. Dazu schreiben wir erneut einen regulären Ausdruck (***LI_PATTERN***). Auch hier verwenden wir die Technik des partiellen Matchens. Für jedes Jahr tragen wir die Jahresangabe und die Liste der 3D-Filme in ein Verzeichnis ein (Java-Map Name ***threeDeeMap***).
5. Fertig ☺!

Wir extrahieren den Bereich der HTML Seite, der die Aufzählung der 3D-Filme enthält. Dazu müssen wir reguläre Ausdrücke für Anfang und Ende des Bereichs definieren.

LÖSUNG ZU 1.‘TENS

Beginn und Ende…

- des zu parsenden Bereichs bestimmen,
- diese durch passende reguläre Ausdrücke beschreiben und
- mit den regulären Ausdrücken den relevanten Inhalt der Datei extrahieren.

```
<h2><span class="mw-headline" id="3D-Filme">3D-Filme</span><span class="mw-editsection"><span
class="mw-editsection-bracket">[</span><a
href="https://de.wikipedia.org/w/index.php?title=Liste_von_3D-
Filmen&amp;veaction=edit&amp;section=1" class="mw-editsection-visualeditor" title="Abschnitt
bearbeiten: 3D-Filme">Bearbeiten</a><span class="mw-editsection-divider"> | </span><a
href="https://de.wikipedia.org/w/index.php?title=Liste_von_3D-
Filmen&amp;action=edit&amp;section=1" title="Abschnitt bearbeiten: 3D-Filme">Quelltext
bearbeiten</a><span class="mw-editsection-bracket">]</span></span></h2>
```

```
<h2><span class="mw-headline" id="4D-Filme">4D-Filme</span><span class="mw-editsection"><span
class="mw-editsection-bracket">[</span><a
href="https://de.wikipedia.org/w/index.php?title=Liste_von_3D-
Filmen&amp;veaction=edit&amp;section=23" class="mw-editsection-visualeditor" title="Abschnitt
bearbeiten: 4D-Filme">Bearbeiten</a><span class="mw-editsection-divider"> | </span><a
href="https://de.wikipedia.org/w/index.php?title=Liste_von_3D-
Filmen&amp;action=edit&amp;section=23" title="Abschnitt bearbeiten: 4D-Filme">Quelltext
bearbeiten</a><span class="mw-editsection-bracket">]</span></span></h2>
```

Regulärer Ausdruck für den Start

```
<h2><span class="mw-headline" id="3D-Filme">3D-Filme</span><span class="mw-editsection"><span
class="mw-editsection-bracket">[</span><a
href="https://de.wikipedia.org/w/index.php?title=Liste_von_3D-
Filmen&amp;veaction=edit&amp;section=1" class="mw-editsection-visualeditor" title="Abschnitt
bearbeiten: 3D-Filme">Bearbeiten</a><span class="mw-editsection-divider"> | </span><a
href="https://de.wikipedia.org/w/index.php?title=Liste_von_3D-
Filmen&amp;action=edit&amp;section=1" title="Abschnitt bearbeiten: 3D-Filme">Quelltext
bearbeiten</a><span class="mw-editsection-bracket">]</span></span></h2>
```

```
"<h2><span class=\"mw-headline\" id=\"3D-Filme\">3D-Filme</span>.*?</h2>"
```

Anfangssequenz und das Ende der Zeile werden zur Identifikation des Starts benutzt. Alles was dazwischen steht interessiert uns nicht.

Regulärer Ausdruck für das Ende

```
<h2><span class="mw-headline" id="4D-Filme">4D-Filme</span><span class="mw-editsection"><span
class="mw-editsection-bracket">[</span><a
href="https://de.wikipedia.org/w/index.php?title=Liste_von_3D-
Filmen&amp;veaction=edit&amp;section=23" class="mw-editsection-visualeditor" title="Abschnitt
bearbeiten: 4D-Filme">Bearbeiten</a><span class="mw-editsection-divider"> | </span><a
href="https://de.wikipedia.org/w/index.php?title=Liste_von_3D-
Filmen&amp;action=edit&amp;section=23" title="Abschnitt bearbeiten: 4D-Filme">Quelltext
bearbeiten</a><span class="mw-editsection-bracket">]</span></span></h2>
```

"<h2><span class=\"mw-headline\" id=\"4D-Filme\">4D-Filme.?</h2>"*

Analog verfahren wir für das Ende.

Die Aufzählung aller 3D Filme extrahieren

```
// Erzeugen des Scanners
Scanner wiki3DFilmScanner = new Scanner(new URL(uri).openStream(),
StandardCharsets.UTF_8);
// Erzeugen der Map
Map<String, List<String>> threeDeeMap = new HashMap<>();
// Positionieren des Scanners vor dem Pattern THREE_DEE_BEGIN.
wiki3DFilmScanner.useDelimiter(THREE_DEE_BEGIN);
if (wiki3DFilmScanner.hasNext()) {
   wiki3DFilmScanner.next();
}
// Lesen des Bereichs bis zum Ende der Aufzählung, das durch das Pattern
// THREE_DEE_END markiert wird.
wiki3DFilmScanner.useDelimiter(THREE_DEE_END);
if (wiki3DFilmScanner.hasNext()) {
   String filmsPerYearEnumeration = wiki3DFilmScanner.next();
```

... HIER GEHT ES NOCH WEITER

- In ***filmPerYearEnumeration*** steht jetzt der gesamte HTML-Text für die Aufzählung der Filme nach Jahren.
- Davon überzeugen wir uns mittels einer Ausgabe auf die Konsole.

Wir extrahieren **für jedes Jahr** die Jahresangabe und die Liste der 3D-Filme. Auch hierfür müssen wir einen regulären Ausdruck definieren (***THREE_DEE_ENUM***). Der reguläre Ausdruck beschreibt mit der 1‘ten Gruppe die Jahresangabe, mit der 2‘ten Gruppe die Aufzählung der 3D-Filme für das Jahr.

LÖSUNG ZU 2‘TENS

HTML Struktur für eine Aufzählung
rot: Bereichsidentifikation gelb: Gruppe

```
<h3><span id="1980–1989"></span><span class="mw-headline" id="1980.E2.80.931989">1980–1989</span><span
class="mw-editsection"><span class="mw-editsection-bracket">[</span><a
href="https://de.wikipedia.org/w/index.php?title=Liste_von_3D-Filmen&amp;veaction=edit&amp;section=8"
class="mw-editsection-visualeditor" title="Abschnitt bearbeiten: 1980–1989">Bearbeiten</a><span
class="mw-editsection-divider"> | </span><a
href="https://de.wikipedia.org/w/index.php?title=Liste_von_3D-Filmen&amp;action=edit&amp;section=8"
title="Abschnitt bearbeiten: 1980–1989">Quelltext bearbeiten</a><span class="mw-editsection-
bracket">]</span></span></h3>
<ul>
<li>Es donnert über San Francisco – 1981</li>
<li><a href="https://de.wikipedia.org/wiki/Alles_fliegt_dir_um_die_Ohren" title="Alles fliegt dir um
die Ohren">Alles fliegt dir um die Ohren</a> – 1981</li>
<li><a href="https://de.wikipedia.org/wiki/Der_Killerparasit" title="Der Killerparasit">Der
Killerparasit</a> – 1982</li>
<li><a href="https://de.wikipedia.org/wiki/Und_wieder_ist_Freitag_der_13." title="Und wieder ist
Freitag der 13.">Und wieder ist Freitag der 13.</a> – 1982</li>
<li><a href="https://de.wikipedia.org/wiki/Das_Geheimnis_der_vier_Kronjuwelen" title="Das Geheimnis der
vier Kronjuwelen">Das Geheimnis der vier Kronjuwelen</a> – 1982</li>
<li>Metalstorm – Die Vernichtung des Jared-Syn – 1983</li>
<li><a href="https://de.wikipedia.org/wiki/Amityville_III" title="Amityville III">Amityville III</a> –
1983</li>
<li><a href="https://de.wikipedia.org/wiki/Der_wei%C3%9Fe_Hai_3-D" title="Der weiße Hai 3-D">Der weiße
Hai 3-D</a> – 1983</li>
<li>Spacehunter – Jäger im All – 1983</li>
<li>My Dear Kuttichathan – 1986</li>
</ul>
```

"<h3>.?<span class=\"mw-headline\" id=.*?>(.*?).*?</h3>(.*?)
"*

Achtung

- Da der Punkt in regulären Ausdrücken im Normalfall nicht mit Zeilenumbrüchen matched, müssen wir im Quelltext bei der Erzeugung des Patterns ***THREE_DEEENUM*** Optionen ***Pattern.MULTILINE | Pattern.DOTALL*** spezifizieren.

- Das sieht dann im Quelltext vollständig wie folgt aus:

```
// Die Option Pattern.MULTILINE|Pattern.DOTALL bewirkt, dass auch
// Zeilenumbrüche mit dem . matchen
private static final Pattern THREE_DEE_ENUM = Pattern.compile(
    "<h3>.*?<span class=\"mw-headline\" id=.*?>(.*?)</span>.*?</h3>(.*?)</ul>",
     Pattern.MULTILINE | Pattern.DOTALL);
```

Da das Muster von ***threeDeeEnum*** mehrfach auftritt, verwenden wir die Technik des partiellen Matchens (Methode ***Matcher.find*** in einer Schleife).

LÖSUNG ZU 3‘TENS

Extrahieren der Gruppe Jahresangabe

... Fortsetzung Code Folie 19

```
Matcher matcherEnum = THREE_DEE_ENUM.matcher(filmsPerYearEnumeration);

while (matcherEnum.find()) {
    // Lesen der Jahresangabe (Gruppe 1)
    String currentDate = matcherEnum.group(1);
    if (!threeDeeMap.containsKey(currentDate)) {
        threeDeeMap.put(currentDate, new ArrayList<>());
        // Extrahieren des Bereichs der ListItems, (Gruppe 2)
        String filmListContent = matcherEnum.group(2);

          ... HIER GEHT ES NOCH WEITER

    }
}
```

PARTIELLES MATCHING

Dann müssen wir aus der 2‘ten Gruppe mit den einzelnen List-Items die **Information einzelner 3D-Filme extrahieren**. Dazu schreiben wir erneut einen regulären Ausdruck (***LI_PATTERN***). Auch hier verwenden wir die Technik des partiellen Matchens. Für jedes Jahr tragen wir die Jahresangabe und die Liste der 3D-Filme in ein Verzeichnis ein (Java-Map Name ***threeDeeMap***).

LÖSUNG ZU 4‘TENS

Pattern für die List-Items

- 1-ter Ansatz: Wir extrahieren alles, was zwischen den öffnenden und schließenden List-Item Klammern steht.

```
private static final Pattern LI_PATTERN = Pattern.compile("<li>(.*?)</li>");
```

- ☹ dann enthalten die Zeichenketten noch HTML-Tags, die noch entfernt werden müssen.

Afrika – Das magische Königreich (5. März 2015)
Fußball – Großes Spiel mit kleinen Helden (5. März 2015)
Seventh Son (5. März 2015) (nachträglich in 3D konvertiert)
Die Bestimmung – Insurgent (5. März 2015) (nachträglich in 3D konvertiert)
Home – Ein smektakulärer Trip (26. März 2015)
Avengers: Age of Ultron (23. April 2015) (nachträglich in 3D konvertiert)
Tinkerbell und die Legende vom Nimmerbiest (30. April 2015)
Mad Max: Fury Road (14. Mai 2015)
San Andreas (28. Mai 2015) (nachträglich in 3D konvertiert)
Poltergeist (28. Mai 2015) (nachträglich in 3D konvertiert)
Jurassic World (11. Juni 2015) (nachträglich in 3D konvertiert)
Minions (2. Juli 2015)

Pattern für die List-Items

- 2‘ter Ansatz:
- Wir verfeinern ***liPattern*** und führen sukzessive Gruppen ein, die den Nettotext (der Text ohne HTML-Tags) beschreiben.
  - Verfeinerung 1: Zu Beginn kann **optional** ein Hyperlink stehen (<a ...> ...)

    `"<li>(?:<a .+?>(.+?)</a>)?(.+?)</li>"`
  - Verfeinerung 2: Am Ende kann optional ebenfalls ein Hyperlink stehen, auf den noch Text folgt:

    `"<li>(?:<a .+?>(.+?)</a>)?(.+?)?(?:<a .+?>(.+?)</a>(.+?))?</li>"`
  - Verfeinerung 3: Am Ende können optional zwei Hyperlinks stehen, auf die noch Text folgt:

    `"<li>(?:<a .+?>(.+?)</a>)?(.+?)?(?:<a .+?>(.+?)</a>(.+?))?(?:<a .+?>(.+?)</a>(.+?))?</li>"`
  - **Fertig? Nein**: Jetzt müssen wir noch die Italics <it> </it> löschen und die Sonderzeichen z.B. &amp ersetzen.
- **Alternativ**: lässt sich die Gesamtaufgabe mit zwei ***String.replaceAll*** unter Verwendung eines regulären Ausdrucks erledigen.
- Wir extrahieren die Gruppen eines Matches und konkatenieren diese zu der Gesamtinfo eines Films.

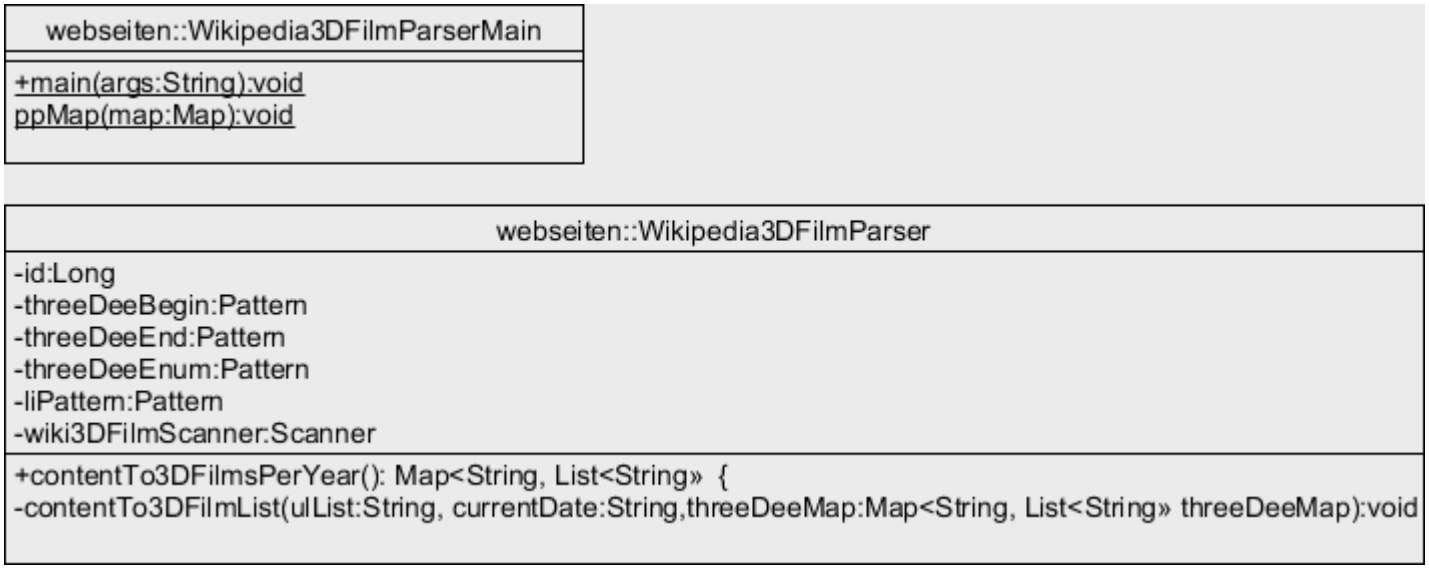
Extrahieren der Film-Info und Eintragen in das Verzeichnis (Variante 1)

```
private void contentTo3DFilmList(String ulList, String currentDate,
                                 Map<String, List<String>> threeDeeMap) {
   // Erzeugen eines Matchers für das Extrahieren der HTML ListItems
   // (Pattern LI_PATTERN)
   // "<li>(.*)?</li>"
   Matcher liMatcher = LI_PATTERN.matcher(ulList);
   // Partielles Matching des liPatterns, für alle Items der Liste
   while (liMatcher.find()) {
      String liContent = liMatcher.group(1);
      liContent = liContent.replaceAll("<a.*?>|</a>|<i>|</i>", "");
      liContent.replaceAll("&amp;", "&");
      threeDeeMap.get(currentDate).add(liContent);
   }
}
```

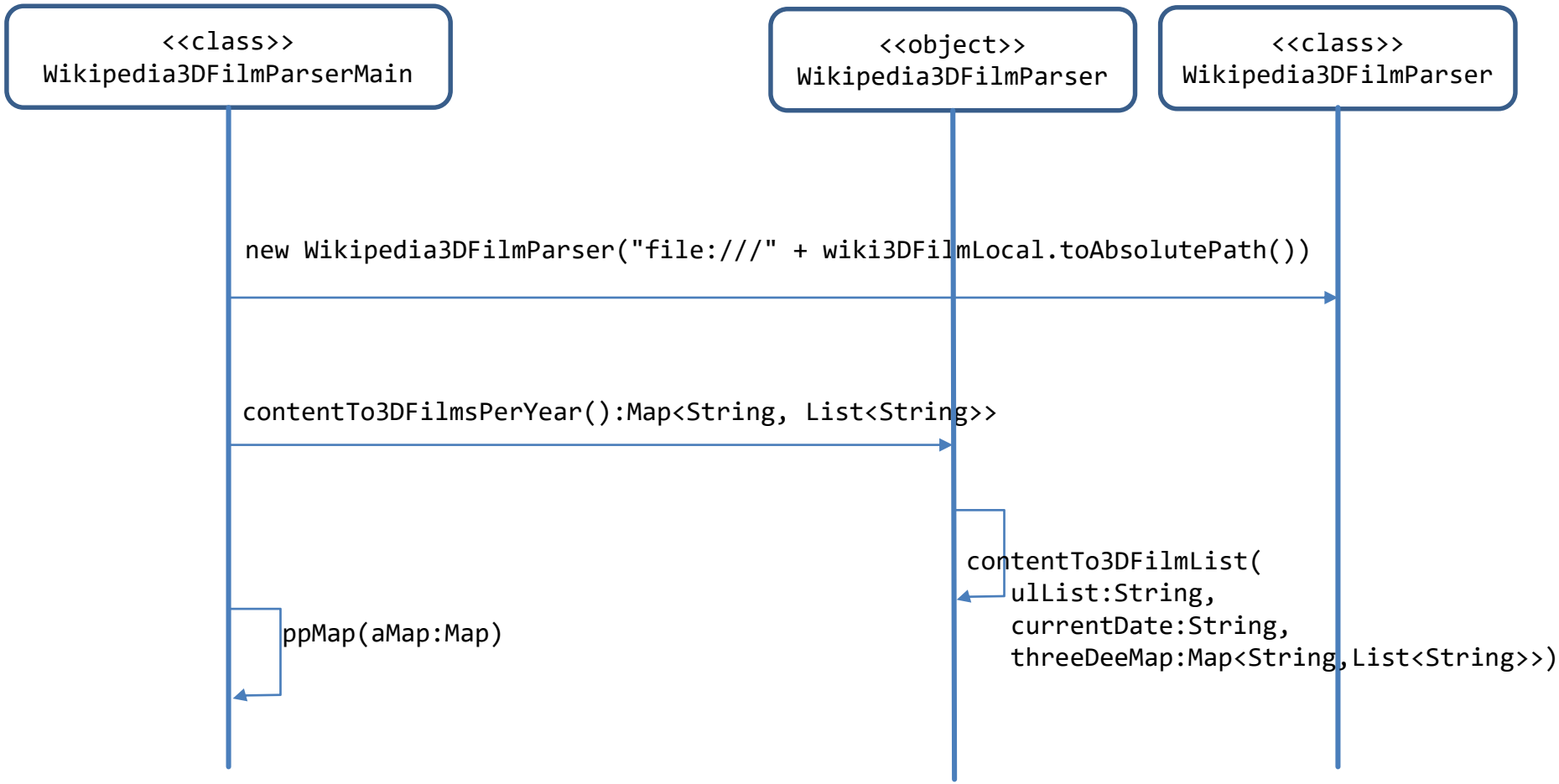
Extrahieren der Film-Info und Eintragen in das Verzeichnis (Variante 2)

```
private void contentTo3DFilmList2(String ulList, String currentDate,
                                  Map<String, List<String>> threeDeeMap) {
   // Erzeugen eines Matchers für das Extrahieren der HTML ListItems
   // (Pattern LI_PATTERN)
   // "<li>(?:<a .+?>(.+?)</a>)?(.+?)?(?:<a .+?>(.+?)</a>(.+?))?(?:<a
   // .+?>(.+?)</a>(.+?))?</li>"
   Matcher liMatcher = LI_PATTERN.matcher(ulList);
   // Partielles Matching des liPatterns, für alle Items der Liste
   while (liMatcher.find()) {
      String concat = "";
      // Extraktion der Info für einen Film mit Hilfe der Gruppen des
      // liPattern
      for (int i = 1; i <= liMatcher.groupCount(); i++) {
         if (liMatcher.group(i) != null) {
            concat += liMatcher.group(i);
         }
      }
      // Hinzufügen eines Films in die Liste der Filme
      threeDeeMap.get(currentDate).add(concat);
   }
}
```

Programmstruktur



Ablaufdiagramm



FIN

Abschlussbemerkung

Eine alternative Lösung enthält die Klasse Wikipedia3DFilmParser2