

Softwaretechnikpraktikum 2016/17

Gruppe APRT17

# Entwurfsbeschreibung

19. Mai 2017

## Inhalt

- 1 Allgemeines
- 2 Produktübersicht
- 3 Grundsätzliche Struktur- und Entwurfsprinzipien
- 4 Pakete und Klassen
- 5 Bibliotheken
- 6 Datenmodell
- 7 Programmwartung und -erweiterung
- 8 Direkte Präsentationserstellung durch Dateiübergabe
- 9 Quellen
- 10 Bilder
- 11 Glossar

## 1 Allgemeines

Das vorliegende Programm soll automatisch Präsentationen im PPTX-Format erzeugen. Im Sinne des Auftraggebers geht es darum, inhaltlich und formell korrekte Präsentationen als Testobjekte für eine Datenbank schnell und leicht erstellen zu können. Eine Weiterentwicklung des Programms für eine allgemeinere Anwendung, z.B. für Schüler, ist vorstellbar.

## 2 Produktübersicht

Das in Java geschriebene Programm erstellt Präsentationen zu Städten und Schlachten, die vom Nutzer vorgegeben werden müssen. Bedient wird das Programm über die Kommandozeile. Der Nutzer wählt eine oder mehrere Städte bzw. Schlachten, zu denen dann eine Präsentation erstellt wird. Das Format der Folien ist im Präsentationsstandard (siehe Anhang zum Arbeitsplan) festgelegt und entspricht gängigen Formatierungsstandards. Die Folien enthalten Informationen zur gewählten Stadt oder Schlacht. Die nötigen Informationen werden aus dem Internet von Wikipedia-verwandten Webseiten und Datenbanken zusammengetragen

## 3 Grundsätzliche Struktur- und Entwurfsprinzipien

Das Programm gliedert sich in fünf Pakete: *Controller*, *Presentation*, *Sources*, und *Topics* sowie die Klasse *Main.java*. *Controller* steht dabei im Zentrum des Programms; von hier aus werden die Klassen der anderen Packages angesprochen und der Programmablauf koordiniert. In *Sources* werden Daten zur Nutzereingabe im Internet gesucht, im Programm zwischengespeichert und bereinigt. Das Paket unterteilt sich weiter in *DBPedia*, *Flickr*, *OpenStreetMap*, *SimplePedia*, *Wikipedia* und *WikiMedia*. Diese stehen für die verschiedenen Quellen, in denen gesucht wird. *Topics* ist unterteilt in *Data* und *SlideFactories*. Ersteres

spezifiziert die zu suchenden Informationen der einzelnen Themen, letzteres verteilt die gefundenen Informationen auf entsprechende Folien. In *Presentation* werden die einzelnen Folien zu einer PPTX-Datei zusammengefügt.

Inspiziert wurde diese Gliederung in Pakete durch die MVC-Architektur. Sie wurde gewählt, da sie die verschiedenen Aufgabenbereiche voneinander trennt und das Programm in über Schnittstellen interagierende, doch sonst unabhängige Programmsegmente teilt. Das erleichtert sowohl paralleles Arbeiten in der Entwicklung als auch später das Hinzufügen neuer Themen und Quellen nach Bedarf.

## 4 Pakete und Klassen

### 4.1 Controller

Erste Hauptschnittstelle des Programms.

Controller

Instanzen anderer Klassen werden angelegt und deren zentrale Methoden nacheinander aufgerufen, und so das Programm schrittweise durchgeführt.

FileInputScanner

Liest Titel aus einer Datei, um das Erstellen mehrerer Präsentationen zu beschleunigen.

TimedPeriodOutputter

Ladebalken in Form regelmäßig ausgegebener Punkte.

UserInputScanner

Nutzerschnittstelle. Diese Klasse enthält das Nutzermenü inklusive der Abfrage nach dem Suchbegriff.

#### 4.1.1 Finder

Finder, City-, BattleFinder (extend Finder)

Zuständig für die Suche nach DBPedia-Seiten und das Prüfen derer Existenz. Diese Klassen ermöglichen Eingaben durch den Nutzer.

#### 4.1.2 ZipArchiver

Erstellt und speichert die ZIP-Datei, in der die PPTX-Datei(en) ausgegeben werden.

Zip

Komprimiert die PPTX-Präsentation und speichert sie als ZIP-Datei.

ZipObject

Erstellt Objekte, die Thema und zugehörige PPTX-Datei speichern.

## 4.2 Presentation

Formatiert Folienvorlagen und erstellt die Präsentation.

### SlideGroup

Verwaltet Slide-Liste, die einen Unterpunkt der aktuellen Thematik zusammenfasst; entspricht einem Kapitel in der Präsentation. Die Folien werden in der SlideFactory formatiert. SlideGroup enthält Methoden zum Erstellen neuer Folien sowie zur folienübergreifenden Formatierung (z.B. viele Stichpunkte auf mehrere Slides aufteilen). DBPedia-Informationen zum Unterpunkt werden in den Klassenvariablen als eine Liste von Properties gespeichert.

### PowerPoint

Erstellt die eigentliche Präsentation im PPTX-Format aus allen in SlideGroups gespeicherten Slides.

### 4.2.1 Slides

#### Slide (abstract)

Erstellt Basisfolie für Präsentation mit Kopf- und Fußzeile. Stellt Basismethoden zur Verfügung, um Titel, Text, Bild etc. einzufügen.

BulletsSlide, ContentsSlide, ImpressionSlide, IntroPicSlide, IntroSlide, PictureSlide, SourceSlide, TextSlide, TextPicSlide, TitleSlide (all extend Slide)

Erstellen verschiedene Folientypen für die Präsentation.

## 4.3 Sources

In diesen Klassen werden Informationen aus verschiedenen Quellen extrahiert.

### Content

Datentyp zur Verwaltung von Inhalten aus Simple- und Wikipedia.

### 4.3.1 DBPedia

#### RedirectGetter

Bietet Funktionen um die Daten, auf welche von Links weitergeleitet wird, zu erhalten.

#### StringUtils

Entfernt HTML-Tags aus extrahierten Texten.

#### ValueGetter

Information zu vorgegebenen Stichpunkten wird mittels SPARQL aus der DBPedia abgerufen und in Strings gespeichert. Diese Datenbankstrings müssen anschließend bereinigt werden.

#### ValueCleaner

Mittels String-Operationen werden die Datenbankstrings aus der Klasse ValueGetter durchsucht, bereinigt und die gefundenen Werte in den Properties eines Topics gespeichert.

#### 4.3.1.1 Properties

##### Property

Eine Property ist ein DBPedia-Eintrag (z.B. "areaTotal") mit dazugehörigen Werten in einer ArrayList. Properties können komplex sein (Werte werden zu neuen Suchbegriffen) oder invers, womit man Einträge der Form „is ... of“ in DBPedia abfragen kann. Außerdem gibt es spezielle Attribute, die die spätere Ausgabe manipulieren.

##### TextProperty (extends Property)

Erbt von Property und dient dazu Werte zu verwalten, die mehrere Sätze lang sind (z.B. abstract).

#### 4.3.2 Flickr

##### ExtractFlickrImages

Findet Bilder auf Flickr.com.

#### 4.3.3 OpenStreetMap

##### MapImageGetter

Enthält Funktion, um die Koordinaten eines Ortes zu suchen und damit eine Karte zu erstellen.

#### 4.3.4 SimplePedia

##### SimplePedia

Extrahiert Inhaltsverzeichnis und Inhalte zum Thema von simple.Wikipedia.org.

##### SimplePediaReader

Speichert alle Informationen eines Themas, die auf SimplePedia zu finden sind, in einer Liste von Contents (siehe Content).

#### 4.3.5 WikiMedia

##### ExtractBeautifulWikiImage

Findet visuell ansprechende Grafiken auf Wikimedia mit API google.gson.

##### ExtractWikiImage

Findet Bilder auf Wikimedia mit API Jsoup.

#### 4.3.6 WikiPedia

##### ExtractBattleMap

Sucht nach einer strategischen Karte einer Schlacht.

#### ExtractCoordinate

Extrahiert Koordinaten zu Schlachten aus Wikipedia-Artikeln.

#### WikiPedia

Extrahiert Inhaltsverzeichnis und Inhalte zum Thema von en.Wikipedia.org.

#### WikiPediaReader

Speichert alle Informationen eines Themas, die auf SimplePedia zu finden sind, in einer Liste von Contents (siehe Content).

### 4.4 Topics

In diesen Paketen werden die themenspezifischen Details festgelegt.

#### 4.4.1 Data

Zweite Hauptschnittstelle. Definiert und verwaltet den Inhalt der Präsentation.

##### MainData (abstract)

Verwaltet und speichert die Information zu einer Präsentation in ihren Klassenvariablen. Enthält Methoden, die durch Aufrufen der Quellen (via Package Sources) die Klassenvariablen mit Information füllen. Eine Ausnahme stellen die Informationen aus DBPedia dar, die in Instanzen SlideGroups gespeichert werden.

##### CityData (extends MainData)

Definiert für das Thema Stadt, welche Quellen auf welche Informationen hin durchsucht werden, und speichert diese Information vorsortiert in Stadt-spezifischen SlideGroups ab.

##### BattleData (extends MainData)

Wie CityData, nur für Schlacht.

#### 4.4.2 SlideFactories

Spezifiziert die Formatierung der Folien innerhalb der einzelnen SlideGroups.

##### SlideFactory (abstract)

Enthält Methoden zum Erstellen von allen Themen gemeinsamen Folien sowie allgemeine Hilfsmethoden.

##### CitySlideFactory (extends SlideFactory)

Enthält Methoden zum Erstellen von Folien spezifisch für Städte.

##### BattleSlideFactory (extends SlideFactory)

Wie CitySlideFactory, nur für Schlachten.

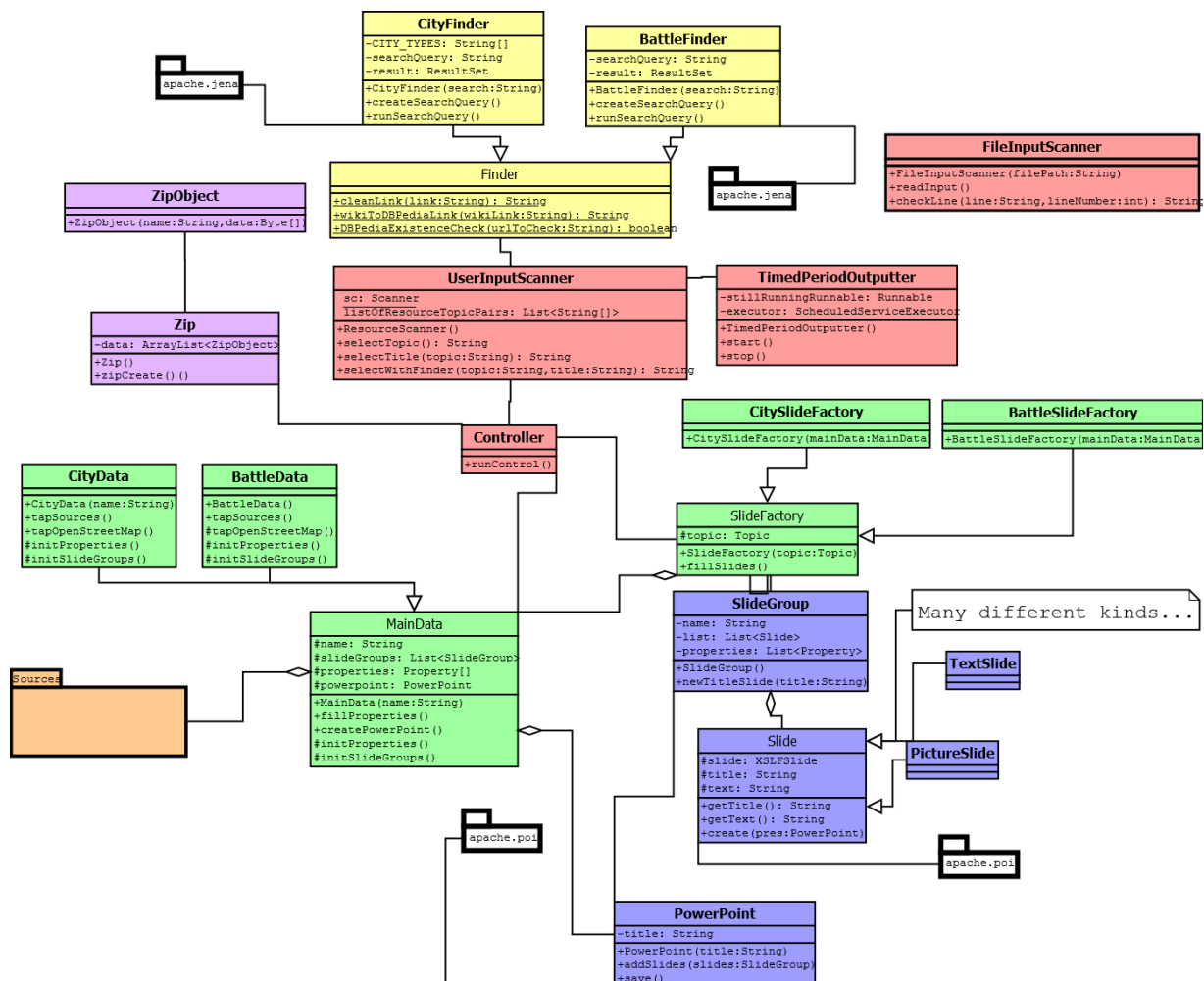
## 5 Bibliotheken

Im Projekt werden einige Bibliotheken verwendet. Apache Jena wird für Datenbankabfragen genutzt und Apache POI zur Präsentationserstellung. Für die Zusammenfassung von Texten wird das NLP-API (NLP: Natural Language Processing) Classifier4J verwendet. Beim Extrahieren der Bilder werden vor allem Google/Gson und jsoup verwendet. Mithilfe von Google/Gson ist es möglich, Java-Objekte in JSON-Dateien (Textdokumente, die sowohl für Mensch als auch Maschine relativ schnell lesbar sind) zu konvertieren und umgekehrt; jsoup ermöglicht es, mit HTML-Dateien zu interagieren. Bei der Bildersuche wird unter anderem Flickr.com als Quelle verwendet. Dafür wird ein generierter Key verwendet und die Bilder werden nach Lizenz gefiltert.

## 6 Datenmodell

Das vorliegende UML-Diagramm enthält die meisten Klassen des Programms (aus Übersichtsgründen wurden einige Kinderklassen von Slide ausgelassen) und eine gekürzte Liste der Klassenvariablen und Methoden. Details zu Klassenvariablen und Funktionen sind der Java-Dokumentation zu entnehmen.

Die Pakete sind farbig gekennzeichnet:







Austausch in Topic mit Parameterübergaben zu machen; die Quellen selbst sollen sich nicht gegenseitig aufrufen. Es ist empfohlen, alle Information in Form von (Collections von) Strings an andere Programmteile zu übergeben.

In Paket Topics ist folgendes zu tun:

Aktualisiere *MainData*

Klassenvariable einfügen, in der Information aus Quelle gespeichert wird

Methode einfügen, die *NewSource*-Methoden aufruft und Klassenvariable mit Wert belegt

get-Methode für Klassenvariable einfügen

Aktualisiere Kindklassen von *MainData*

SlideGroups zur neuen Quelle einfügen

Aktualisiere *SlideFactory* und ihre Kindklassen

fill...Slides()-Methoden einfügen

## 7.2 Neues Thema hinzufügen

### 7.2.1 Thema wählen

Bei der Wahl eines neuen Themas für das Programm ist folgendes zu bedenken:

- Bieten die vorhandenen Quellen Material zum Thema?
- Ergibt das Thema (potentiell) genug verschiedene Präsentationen?
- Hat das Thema Relevanz?

### 7.2.2 Thema hinzufügen

Um dem Programm ein neues Thema (z.B. *NewTopic*) hinzuzufügen, sind folgende Schritte nötig:

Topics.Data

Schreibe Klasse *NewTopicData* (extends *MainData*)

Topics.SlideFactories

Schreibe Klasse *NewTopicSlideFactory* (extends *SlideFactory*)

Controller.Finder

Schreibe Klasse *NewTopicFinder* (extends *Finder*)

Controller

Aktualisiere Controller: Methode *runControl()* ergänzen

Aktualisiere *UserInputScanner*: Methoden *selectTopic()* und *selectWithFinder(...)* ergänzen

Aktualisiere FileInputScanner: Klassenvariable TOPICS und Methode checkType() ergänzen.

### 7.3 Präsentation bearbeiten

#### Topics.Data

Welche Informationen aus welchen Quellen

Kapitelüberschriften (Namen der SlideGroups)

Kapitelreihenfolge (Reihenfolge der SlideGroups in Klassenvariable slideGroups)

Namen der Attribute, wie sie auf der Präsentation erscheinen

#### Topics.SlideFactories

Anordnung der Informationen innerhalb eines Kapitels auf Folien (fill...Slides())

#### Presentation.Slides

Folienvorlagen: Kopf- und Fußzeile, Schriftart und -größe, Anordnung von Text und Bildern, ... (siehe Präsentationsstandard)

#### Controller.ZipArchiver

Speicherort der ZIP-Datei

## 8 Direkte Präsentationserstellung durch Dateiübergabe

Es ist möglich beim Ausführen des Programms den Pfad zu einer Textdatei als Übergabe anzuhängen.

Dies sähe beispielsweise so aus:

```
java -jar Auto_Slides-2.3.jar C:\Directory\data.txt
```

Legt man eine Datei an, in welcher, in korrekter Art und Weise, gewünschte Titel für Präsentationen aufgelistet sind, so kann man den Pfad übergeben und so die Eingabe überspringen. Das Programm wird direkt zur Erstellung übergehen.

Es sind Themen und die dazu gewünschten Titel einzutragen. Die Themen entsprechen jenen aus der normalen Nutzereingabe. Richtigkeit Titel ist durch Kopieren der Endungen von Wikipedia- bzw. DBPedia-Links leicht sicherzustellen.

Hier durch Fettschreiben gekennzeichnet:

[http://dbpedia.org/page/\*\*Battle\\_of\\_Leipzig\*\*](http://dbpedia.org/page/Battle_of_Leipzig)

[https://en.wikipedia.org/wiki/\*\*Battle\\_of\\_Leipzig\*\*](https://en.wikipedia.org/wiki/Battle_of_Leipzig)

Titel können auch mit Leerzeichen anstatt von Unterstrichen eingetragen werden und werden auf Richtigkeit geprüft.

Die Formatierung der Datei unterliegt gewissen Auflagen und bietet Funktionen, wie folgend beschrieben:

- Nur eine Buchstabenfolge pro Zeile
- Vor einer Zeile mit Titel muss ein Thema angegeben worden sein
- Doppelte Titel werden im Programm nur einmal erstellt
- Leere Zeilen werden übersprungen
- ,//‘ zu Beginn einer Zeile lässt diese übersprungen werden
- ,#‘ verlangt die direkte Folge eines Themas und gibt jenes Thema für darunter folgende Titel an
- Titel werden ohne Zusatz in eine Zeile unter einem Thema eingetragen

Beispiel für den Inhalt einer solchen Datei:

```
#city
```

```
Leipzig
```

```
//London
```

```
#battle
```

```
Battle_of_Leipzig
```

Es ist ratsam die Last für den Arbeitsspeicher nicht zu unterschätzen und mit 15 MB pro zu erstellender Präsentation zu rechnen.

## 9 Quellen

### **DBPedia (DBPedia)**

<http://wiki.dbpedia.org/>

Die Datenbank hinter Wikipedia und Hauptinformationsquelle des Programms. Diverse Fakten können über Schlagworte gezielt abgefragt und bei Bedarf via Link weiter verfolgt werden. Geeignete Schlagworte werden für jedes Thema einzeln festgelegt. Eine zentrale Rolle für die Präsentation spielen “thumbnail” (Bild, im Wikipedia-Artikel oben rechts) und “abstract” (Einleitungstext eines Wikipedia-Artikels; auf den Folien werden die ersten Sätze verwendet).

### **Simple English Wikipedia (SimplePedia)**

[https://simple.wikipedia.org/wiki/Main\\_Page](https://simple.wikipedia.org/wiki/Main_Page)

Wikipedia auf “Simple English”: die Artikel der englischen Wikipedia in stark gekürzter und vereinfachter Form. Das Format der Artikelabschnitte – kurze Texte, viele Stichpunkte – erlaubt es, die Abschnitte mit wenig Nachbearbeitung auf Folien bringen. Ausgenommen explizit ausgeschlossener Abschnitte, werden dabei ganze Artikel in Stichpunkte

umgewandelt und für die Folien verwendet. Eine gezielte Suche nach Abschnitten ist mangels Standardisierung schwer. Eignet sich zur Faktensuche in Textform und Umfangserweiterung.

### **Wikipedia (WikiPedia)**

[https://en.wikipedia.org/wiki/Main\\_Page](https://en.wikipedia.org/wiki/Main_Page)

Website, auf der Artikel zu weitgefächerten Themen von Freiwilligen gesammelt und aktuell gehalten werden. Da sich Wikipedia und SimplePedia inhaltlich überschneiden wenn beide existieren, wird Wikipedia nur verwendet, wenn auf SimplePedia keine Informationen gefunden wurden oder die Suche dort nicht aufgerufen wurde. Dabei werden die Texte mit NLP überarbeitet und so in drei Sätzen zusammengefasst, die auf eine Folie kommen.

### **Wikimedia Commons (WikiMedia)**

[https://commons.wikimedia.org/wiki/Main\\_Page](https://commons.wikimedia.org/wiki/Main_Page)

Mediensammlung mit Bild-, Audio- und Videodateien. Aufgebaut wie Wikipedia. Medien zu einem Artikel lassen sich leicht finden, aber Bilder eines bestimmten Typs (z.B. Sehenswürdigkeiten) innerhalb des Artikels nur schwer gezielt suchen. Eignet sich z.B. für Bildfolie mit Impressionen.

### **Flickr (Flickr)**

<https://www.flickr.com/>

Fotoplattform, auf der Amateur- und Profifotografen ihre Werke teilen. Die Bilder sind von hoher Qualität und eher künstlerischer als wissenschaftlicher Natur. Eignet sich z.B. für Bildfolie mit Impressionen.

### **OpenStreetMap (OpenStreetMap)**

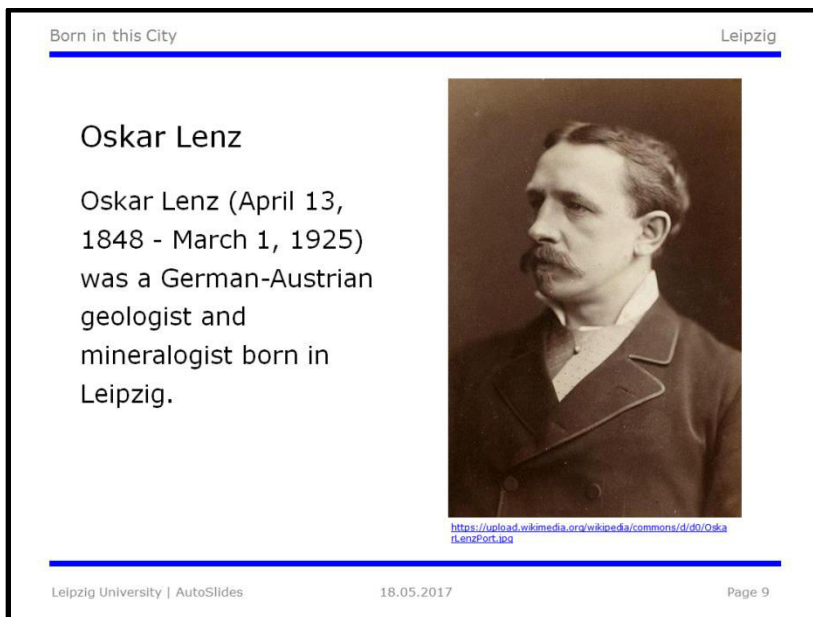
<https://www.openstreetmap.org>

Plattform für Freiwillige, um Kartendaten zu Straßen, Schienen, Cafés und ähnlichen reiserelevanten Informationen online zu stellen, die dann gesammelt und kostenlos und so aktuell wie möglich zur Verfügung gestellt werden können. Eignet sich, um Orte auf einer Landkarte zu finden

## 10 Bilder

```
ca. B:\Windows\system32\cmd.exe
H:\>\target>java -jar Auto_Slides-2.3.jar
Welcome to AutoSlides.
Please select a topic.
[0] city
[1] battle
Your choice: 0
Choose title.
Enter a search term (separate words with spaces, use umlauts); or
[0] enter a Wikipedia/DBPedia link
[1] return to main menu
Your choice: Leipzig
Searching .
Is Leipzig <Germany> the city you are looking for?
[0] yes
[1] no, try again
Your choice: 0
The selected city was successfully saved.
[0] create more presentations
[1] end selection
Your choice: 1
Working .....
Added 'Leipzig' 1/1
Success.
Path: H:\>\target\
File: pres_20170518.204811.zip
```

Screenshot der Konsolenausgabe nach Ausführung des Programms



Folie aus der erstellten Präsentation

## 11 Glossar

**Apache Jena** — Java-Bibliothek zur Arbeit mit SPARQL u.a.

**Bug** — Fehler im Code, der unerwartetes Verhalten des Programms verursacht

**Collection** — Sammlung von Daten des selben Typs in Java, z.B. in Form einer Liste

**Datenbank** — strukturierte Sammlung von Information; Daten sind systematisch katalogisiert und können gezielt abgerufen werden

**DBpedia** — über Spenden finanziertes Gruppenprojekt, das strukturierte Informationen aus Wikipedia online zugänglich machen soll

**Elternklasse** — siehe Vererbung

**get-Methode** — eine Methode, die eine Klassenvariable zurückgibt

**Instanz (einer Klasse)** — Exemplar einer Klasse; konkrete Realisierung mit Werten

**Java** — objektorientierte Programmiersprache

**Kindklasse** — siehe Vererbung

**Klasse** — abstraktes Modell für eine Reihe von ähnlichen Objekten

**Klassenvariable** — Daten einer Klasse, die mit Klassenmethoden manipuliert werden können

**Kommandozeile** — Eingabezeile im Terminal, über das das Programm vom Benutzer bedient wird

**Konstruktor** — erstellt eine Instanz einer Klasse

**Methode** — über Methoden können Klassen kommunizieren und z.B. Werte manipulieren

**Package** — Programmstrukturelement in Java; enthält Klassen und/oder Schnittstellen

**PPTX-Format** — Dateiformat von Microsoft PowerPoint

**SPARQL** — graphenbasierte Abfragesprache für Datenbanken

**String** — Folge von Zeichen aus einem definierten Zeichensatz; Datentyp in Java

**Stringoperationen** — Manipulation von Zeichenketten

**Vererbung** — Konzept in Java, bei dem Funktionalitäten einer Elternklasse in deren Kindklassen nutzbar sind

**ZIP-Format** — Dateiformat, komprimiert eine oder mehrere Dateien