Testat

Objektorientierte Programmierung Praktikum

Gruppe 1 (08:30)

Studiengang

Wirtschaftsinformatik und E-Business (Bachelor) WI PLUS

Prof. Dr. Thomas Bayer

Wintersemester 2021

(Bitte in Druckbuchstaben ausfüllen!)

Matrikel-Nr:	Datum: 03.02.2022
Sitzplatz:	Semester: **WS 2021**
Hilfsmittel: **keine**	Bearbeitungszeit: **90 Minuten**

(Bitte nicht ausfüllen!)

	Unterschrift Prüfer:					Note:				
Aufgabe	CL	Prep	Add	getTC	Cont	Rem	rT	getT		Gesamt
max. Punkte	**15**	**25**	**20**	**10**	**10**	**10**	**15**	**5**		**110**
Punkte										

Ausgangssituation: Analyse von Tweets

Der Analyzer erhält Tweets (als TweetObject) als Eingabe und extrahiert die Tags aus dem Text. Er zählt die Anzahl an Tags, die in den Tweets enthalten sind. Die Tweets können mit einem Filter ausgefiltert werden.
Beispiel:
`TweetObject.text = „#corona #COVID19“`: Es werden die Tags „corona“ und „rwu“ extrahiert

Tag	count
corona	1
covid19	1

`TweetObject.text = „#Corona #rwu“`: Es werden die Tags „corona“, „rwu“, „covid19“ extrahiert

Tag	count
corona	2
covid19	1
rwu	1

Beachten Sie: Das Extrahieren der Tags und das Aufnehmen in die Zuordnung wird in unterschiedlichen Methoden durchgeführt! Die Tags müssen in allen Methoden in Kleinbuchstaben umgewandelt werden!

Absrakte Klasse ATweetAnalyzer

```
public abstract class ATweetTagAnalyser {
// Extrahiert die Tags aus dem Text des Tweet-Objektes in eine Liste und fügt
// diese dem Tweet-Objekt hinzu, falls dass Tweet-Objekt den filter erfüllt
// (Methode filter.test(...)) muss true liefern. Achtung: Das "#"-Zeichen muss
// entfernt werden. Die Tags werden nicht in den Analyzer aufgenommen!
// Rückgabe: 0 falls tweet == null oder filter.test(tweet)==False
// Sonst: Anzahl an gefundenen Tags
// Beachten Sie die Hinweise
public abstract int prepareTweetObject(TweetObject tweet,
Predicate<TweetObject> filter);

// Fügt die Tags eines tweet-Objektes zum Analyzer hinzu. Das Tweet-Objekt muss
// die Tags als Liste enthalten (im Attribut tagList).
// Rückgabe: Anzahl an Tags, die noch nicht im Analyzer enthalten waren
// 0 falls tweet==null
public abstract int addTags(TweetObject tweet);

// Liefert die Anzahl, wie oft der Tag tag zum Analyzer hinzugefügt wurde
// 0 falls der Tag nicht enthalten ist (Kleinbuchstaben)
public abstract int getTagCount(String tag);

// Ist der Tag im Analyzer enthalten? (Kleinbuchstaben)
public abstract boolean contains(String tag);

// Löscht einen Tag und gibt die Anzahl zurück, wie oft er addiert wurde
public abstract int remove(String tag); (Kleinbuchstaben)

// Undo-Operation für addTags(TweetObject tweet).
public abstract int removeTags(TweetObject tweet);

// Liefert die Tags zurück
public abstract Set<String> getTags();
```

Vorbereitung

1. Starten Sie Eclipse mit einem passenden Workspace: Java Version 11 (default)
2. Importieren Sie das Projekt 2021-WS Testat-G1-TagAnalyzer-eclipse.zip (Import existing project)!
3. Alternativ: Starten Sie IntelliJ und importieren Sie 2021-WS Testat-G1-TagAnalyzer-intellij.zip
4. Erstellen Sie die Klasse **`TweetAnalyzerTest#####`**, die von **`ATweetAnalyzerTest`** erbt
5. Lesen Sie die Hinweise!

Aufgaben

- Benennen Sie das Paket G1_solution in G1_solution##### um (intelliJ: Paket anlegen)
- Implementieren Sie die abstrakten Methoden der Klasse **`ATweetTagAnalyzer`**
- ##### = Ihre Matrikelnummer

Vorgehen

- Implementieren Sie die Klasse **`TweetTagAnalyzer#####`**, die von **`ATweetTagAnalyzer`** erbt.
- Passen Sie Ihre Testklasse so an, dass eine Instanz Ihrer Klasse übergeben wird.
    a. Methode **`getInstance()`** – Geben Sie eine Instanz der Klasse **`TweetTagAnalyzer#####`** zurück

Abgabe

- Exportieren Sie Ihr Projekt (Testat-G1-#####) als Archiv in das Verzeichnis C:\InsightFiles
- Vor Abgabe melden Sie sich!

Hinweise

- Sie benötigen eine Collection
- Das TweetObject ist eine vereinfachte Version im Vergleich zum Projekt. Verwenden Sie die passenden Methoden!
- Wandeln Sie alle Tags in Kleinbuchstaben mit der Methode toLowerCase() um!
- Bearbeitung der Zeichenkette: Splitten Sie das String mit dem Leerzeichen in einzelne Wörter.
- Extraktion von Tags: (Methoden der Klasse String)
    - Erkennen von Tags mit `contains("#")`
    - Entfernen von „#“ mit `replace("#","")`
- Verwenden von Integer in einer Map<K, Integer>: get(k) liefert null, falls der Schlüssel nicht enthalten ist. Sie müssen null in 0 umwandeln. Beispiel (aus dem Projekt):
    - `Integer tweetCount = tweetsPerUser.get(tweet.getUser());`
    - `if(tweetCount == null) {tweetCount = 0;}`

Hinweise zur Benotung (110 Punkte)

- **Es reichen 50 Punkte zum Bestehen!**
- **(Versuchte) Manipulation der gegebenen Klassen/Interfaces: max. 25 Punkte**
- **Kopieren der gegebenen Klassen/Interfaces in ein eigenes Paket: 0 Punkte**
- **Programm nicht lauffähig (Syntaxfehler): max. 25 Punkte**
- **Artefakte aus der anderen Gruppe tauchen auf: 0 Punkte!**
- Sowohl Unit-Tests als auch der Quelltext werden für die Benotung herangezogen
- Unit-Tests dienen nur zur Orientierung!

Collections

Interface/Klasse	Implementierende Klasse	Wichtige Methoden
Map	HashMap	put(k,v), get(k), size(), remove(k), keySet(), values(), clear()
Set	HashSet	add(e), contains(e), remove(e), clear()
List	ArrayList	add(e), get(i), clear() add(i,e): Fügt e an Position i ein, vorher wird die Liste ab i nach rechts verschoben, set(i, e):Überschreibt den Eintrag an Position i mit e contains(e), remove(e): true, falls e (einmal) entfernt wurde, sonst false remove(i): Löscht den Eintrag an Position i
Date	-	new Date()// aktuelles Datum
Consumer<T>	-	void accept(T t)
String		String[] split(separator), boolean contains(string), String replace(x,y) // ersetzt x durch y int length(), String toLowerCase(), toUpperCase()
Predicate<T>		boolean test(T t)

Anhang

Klasse TweetObjekt: Enthält alle Getter und Setter

```
public class TweetObject {
    private String id;
    private String created_at;
    private String text;
    private String user;
    private int wordCount;
    private List<String> tags;
    public TweetObject(String id, String created_at, String text) {
        this.id = id;
        this.created_at = created_at;
        this.text = text;
        this.tags = new LinkedList<>();
    }
```