

Ausgangssituation: TweetGenerator

Es soll ein TweetGenerator erstellt werden, der CSV-Strings in Objekte umwandelt (TweetObject). Diese Objekte werden im Generator gespeichert und können auch getweetet werden. Jeder Tweet enthält einen User. Wird der Tweet aufgenommen, wird zum User die Anzahl an Worte des Tweets addiert (Worte zählen mehrfach). Nach dem Twitern werden die Tweets und die Zuordnung User-Anzahl an Wörtern zurückgesetzt (Methode clear() der Collection)

Beispiel: CSV="1234;2020-12-22T13:28:30.000Z;Testat trotz #corona;99"

TwitterObjekt: id="1234", created_at="2020-12-22T13:28:30.000Z", user="99"

text="Testat trotz #corona"

Tabelle (User „10“ schon vorhanden):

User	nrWords
„99“	3
„10“	20

Das Erstellen der Tweet-Objekte und das Aufnehmen wird in unterschiedlichen Methoden durchgeführt!

Absrakte Klasse ATweetGenerator

```
public abstract class ATweetGenerator {
// Umwandlung der Zeichenkette csvTweet in ein TweetObject
// Die Tags können mit extractTags(String text) extrahiert werden
// Rückgabe: TweetObjekt, null falls csvString == null oder syntaktisch nicht
// korrekt ist. Format csvString: "id;created_at;text;user"
// Hinweis: text enthält kein ";" (split bei ";")
// Hinweis: Das TweetObject wird nicht in den Generator aufgenommen
public abstract TweetObject addTweetCSV(String csvTweet);

// Fügt das tweetObjekt hinzu sofern es nicht null oder bereits enthalten ist
// Falls das TweetObjekt hinzugefügt wird, wird die Anzahl an Wörtern im Text
// zu der bereits gespeicherten Anzahl der Wörter des Users tweet.getUser()
// hinzugefügt. Mehrfach vorkommende Wörter werden berücksichtigt
// Falls tweet keine Wörter enthält, nicht aufnehmen (Länge text = 0)
// Rückgabe: Anzahl der neu aufgenommenen Wörter
// 0 falls tweet bereits enthalten ist
public abstract int addTweet(TweetObject tweet);

// Gesamte Anzahl an Wörtern (für alle User)
public abstract int getNrOfWords();
// Gesamte Anzahl an Wörtern von User user
// Sind keine Wörter zum User gespeichert, wird 0 zurückgegeb (siehe Hinweise)
public abstract int getNrOfWords(String user);

// Tweet enthalten?
public abstract boolean contains(TweetObject tweet);

// Löscht den Tweet und entfernt die entsprechende Anzahl an Wörtern bei User
// tweet.getUser(). Rückgabe: Die Anzahl an verbleibenden Worten des User
// Ist tweet == null oder der Tweet nicht enthalten, wird 0 zurückgegeben
public abstract int remove(TweetObject tweet);

// Postet alle gesammelten Tweets, sofern testConnection() true ergibt
// Rückgabe true, falls postTweet() true liefert. danach ist der Client leer
public abstract int postTweets(ITwitterConnector connector);
```

Vorbereitung

1. Starten Sie Eclipse mit einem passenden Workspace: Java Version 11 (default)
2. Importieren Sie das Projekt 2021-WS Testat-G2-TweetGen-eclipse.zip (Import existing project)!
3. Alternativ: Starten Sie IntelliJ und importieren Sie die intelliJ-Version
4. Erstellen Sie die Klasse **TweetGeneratorTest#####**, die von **ATweetGeneratorTest** erbt
5. Lesen Sie die Hinweise!

Aufgaben

- Benennen Sie das Paket G2_solution in G2_solution##### um (intelliJ: Paket anlegen)
- Implementieren Sie die abstrakten Methode der Klasse **ATweetGenerator**
- ##### = Ihre Matrikelnummer

Vorgehen

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

Abgabe

- Exportieren Sie Ihr Projekt (testat-G2-#####) als Archiv in das Verzeichnis C:\InsightFiles
- Vor Abgabe melden Sie sich!

Hinweise

- Sie benötigen mindestens eine Collection
- Das TweetObject ist eine vereinfachte Version im Vergleich zum Projekt. Verwenden Sie die passenden Methoden!
- Bearbeitung des csvString:
 - Splitten Sie das String mit dem Semikolon „;“ in die Bestandteile.
 - Ist die Anzahl < 4, ist der String nicht syntaktisch korrekt
- Bearbeitung der Zeichenkette text: Splitten Sie das String mit dem Leerzeichen in einzelne Wörter.
- Extraktion von Tags: Verwenden Sie die passende Methode der abstrakten Klasse
- 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:
 - `Integer tagCount = tagsPerTweet.get(tag);`
 - `if(tagCount == null) {tagCount = 0;}`
- Interface **ITwitterConnector** ist in der abstrakten Klasse enthalten

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<>();
    }
}
```

Interface ITwitterConnector ist in der abstrakten Klasse enthalten

```
public interface ITwitterConnector {
    public boolean testConnection();
    public boolean postTweets(Collection<TweetObject> tweetObjects);
}
```