

Präsenzaufgaben 3

23./24.10.2024

Die Lösung der Aufgaben wird am Ende der Übung von Ihnen vorgestellt.

HashMaps

In vielen Fachbüchern findet man am Ende ein Sachwortverzeichnis, auch unter der Bezeichnung Index bekannt. Ein Index besteht aus einer Übersicht aller wichtigen Begriffe im Buch und den Seitenzahlen, auf denen sich die Begriffe finden lassen. Auf dem folgenden Bild sehen Sie ein Beispiel für einen Index: Implementieren Sie eine Klasse `Index`, die ein solches Sachwortverzeichnis repräsentiert.

W

While-Schleife 135, 149

Wrapperklassen 16, 35, 70

- Schreiben Sie einen Default-Konstruktor, der einen leeren Index erzeugt.
- Schreiben Sie eine Methode `wortHinzufuegen`, die als Parameter einen Begriff (`String`) und eine Seitenzahl (`int`) übergeben bekommt. Begriff und Seitenzahlen sollen durch die Methode dem Index hinzugefügt werden. Beachten Sie: Jeder Begriff darf nur einmal im Index stehen, aber mit mehreren (unterschiedlichen) Seitenzahlen.
- Schreiben Sie eine `main`, um die Klasse zu testen. Fügen Sie folgende Zeilen hinzu:

```
Index index = new Index();
index.wortHinzufuegen("boolean", 1);
index.wortHinzufuegen("boolean", 15);
index.wortHinzufuegen("boolean", 3);
index.wortHinzufuegen("boolean", 15);
```

```
index.wortHinzufuegen("if", 45);
index.wortHinzufuegen("if", 2);
index.wortHinzufuegen("if", 10);
index.wortHinzufuegen("if", 45);
```

- Schreiben Sie eine `toString`-Methode, die den aktuellen Status des Index zurückgibt. Der Status soll wie folgt aufgebaut sein:
<begriff1>: [<seite1>, <seite2>, ..., <seiteN>]
<begriff2>: [<seite1>, <seite2>, ..., <seiteN>]

Testen Sie die Methode, indem Sie der `main` folgende Zeilen hinzufügen:

```
System.out.println("index");
System.out.println(index);
```

- Schreiben Sie eine Methode `getAlleFundstellen(String word)`, die eine `ArrayList<Integer>` mit allen Fundstellen von `word` zurückgibt. Achten Sie auf die Datenkapselung. Ist `word` nicht vorhanden, wird eine leere `ArrayList` zurückgegeben.

Testen Sie die Methode, indem Sie der `main` folgende Zeilen hinzufügen:

```
ArrayList<Integer> seiten = index.getAlleFundstellen("boolean");
System.out.println("seiten = " + seiten);
```

- Schreiben Sie eine Methode `getErsteFundstelle(String word)`, die die erste Seite zurückgibt, auf der das Wort gefunden wurde. Ist `word` nicht vorhanden, wird `-1` zurückgegeben.

Testen Sie die Methode, indem Sie der `main` folgende Zeilen hinzufügen:

```
int ersteStelle = index.getErsteFundstelle("if");
System.out.println("ersteStelle = " + ersteStelle);
```

Am Ende sollte folgende Ausgabe zu sehen sein:

```
index
boolean: [1, 15, 3]
if: [45, 2, 10]

seiten = [1, 15, 3]
ersteStelle = 2
```

Benutzen Sie zur Implementation der Aufgabe die Klasse `HashMap`.

Zusatzaufgabe

- a) Schreiben Sie eine weitere Testklasse. Lesen Sie darin die Datei „zauberlehrling.txt“ ein. Sie enthält die Ballade „Der Zauberlehrling“ von Johann Wolfgang von Goethe. Sämtliche Satzzeichen sind bereits entfernt. Speichern Sie alle Wörter, die erstens groß geschrieben sind und zweitens mehr als 3 Buchstaben besitzen, mit der zugehörigen Zeilennummer im Index. Die erste Zeile hat die Zeilennummer 1.
- b) Geben Sie anschließend den Index aus.
- c) Fügen Sie neue Elemente in eine `ArrayList` sortiert ein, so dass die Zeilennummern geordnet ausgegeben werden.
- d) Benutzen Sie statt einer `HashMap` eine `TreeMap`, um die Wörter auch alphabetisch geordnet auszugeben.