
Programmieren in Java<http://proglang.informatik.uni-freiburg.de/teaching/java/2017/>

phone-book*Ein Telephonbuch*

Woche 04 Aufgabe 3/4

Herausgabe: 2017-05-15

Abgabe: 2017-05-26

Achtung: beachten Sie unbedingt die allgemeinen Hinweise zur Abgabe auf der Homepage.

Project **phone-book**Package **phonebook**

Klassen

Main
<code>public static void main(String[])</code>

Das Programm liest zuerst eine bestimmte Menge an Telefonbucheinträgen ein, bestehend aus Name und Telefonnummer. Anschließend kann der Benutzer Telefonnummern durch Angabe von Namen durchsuchen.

Das Eingabeformat für Telefonbucheinträge ist:

- Zunächst wird die Anzahl der Einträge n als Integer gelesen.
- Dann folgen n Eingaben des Formats `<Name> <Nummer>`
- `<Name>` hat das Format `<Vorname> <Nachname>` oder `<Vorname. <Vorname>` und `<Nachname>` sind nicht-leere Wörter, die nur aus Buchstaben bestehen.
- `<Nummer>` ist eine nicht-leeres Wort, dass aus den Ziffern 0-9 besteht.

Entspricht die Eingabe des Telefonbuchs nicht diesem Format, soll das Programm mit der Fehlermeldung

Error: cannot parse phone book.

(auf `stdout`) abbrechen.

Im Telefonbuch gilt immer die letzte eingegebene Nummer für einen Namen. Wenn während der Eingabe des Telefonbuchs ein Name mehrfach vorkommt, soll bei den weiteren Vorkommen beim Einlesen die Meldung

Warning: overwriting entry for: <Name>

In der Meldung soll `<Name>` durch den entsprechenden Namen ersetzt werden.

Nachdem das Telefonbuch eingelesen ist, liest das Programm zeilenweise Namen ein und gibt die entsprechenden Telefonnummern im Format

`<Name> = <Nummer>`

aus. Zeilen, die nur aus *Whitespace* bestehen, sollen ignoriert werden. Befindet sich ein Name nicht im Telefonbuch, soll die Meldung

Not found: <Name>

erscheinen. Bei Zeilen, die keine korrekten Namen enthalten, soll die Meldung

Not a name: <Eingabe>

erscheinen.

Beispieleingabe 01:

```
4
uncle sam 99912222
tom
  11122222
harry
12299933
harry 12299934
uncle sam
uncle tom
t$m
harry
```

Erwartete Ausgabe 01:

```
Warning: overwriting entry for: harry
uncle sam = 99912222
Not found: uncle tom
Not a name: t$m
harry = 12299934
```

Beispieleingabe 02:

```
1
number of mike is 0123
```

Erwartete Ausgabe 02:

```
Error: cannot parse phone book.
```

Hinweise Hier können wieder Reguläre Ausdrücke nützlich sein (vgl. `list-operations`)