

Synonym Handler

Mål

- Komplettera några metoder i SynonymHandler. Så att filen SynonymsUser kan exekveras.

Skriven Kod

```
107     // removeSynonymLine accepts synonym data, and removes  
108     // the synonym line corresponding to a given word.  
109     // If the given word is not present, an exception of  
110     // the type IllegalArgumentException is thrown.  
111    public static String[] removeSynonymLine (String[] synonymData, String word) throws IllegalArgumentException  
112    {  
113        int indexWord = synonymLineIndex(synonymData, word);  
114        String[] synData = new String[synonymData.length - 1];  
115        for (int i = 0; i < indexWord; i++)  
116        {  
117            synData[i] = synonymData[i];  
118        }  
119  
120        for (int i = indexWord+1; i < synonymData.length; i++)  
121        {  
122            synData[i-1] = synonymData[i];  
123        }  
124        return synData;  
125    }
```

- Metoden tar in en mängd data av typen `String` (som array) Samt ett ord.
Den kommer då hitta efterSökt raden i datamängden och tar bort den.
- Rad 113: Hittar indexet av den efterSökt raden.
- Rad 114: Skapar en tom array med längden som är en plats mindre än `SynonymData`.
- Tanken med for-loopen är att kopiera alla rader från `SynonymData` till nya arrayn (`synData`) förutom den raden som skulle tas bort.
- Rad 115-118: Kopierar alla rader (från första raden fram till den efterSökt) till den nya arrayn.
- Rad 120-123: Kopierar alla rader (från raden efter den efterSökt till slutet av data) till den nya arrayn.
- TillSlut returneras den nya datamängden.

```

127     // addSynonym accepts synonym data, and adds a given
128     // synonym for a given word.
129     // If the given word is not present, an exception of
130     // the type IllegalArgumentException is thrown.
131     public static void addSynonym (String[] synonymData, String word, String synonym)
132     {
133         int indexWord = synonymLineIndex(synonymData, word);
134         synonymData[indexWord] += ", " + synonym;
135     }

```

Metoden tar in en mängd data av typen `String` (Som array) ett ord samt en synonym.

Den kommer då hitta efterSökt raden i data mängden och adderar en synonym till slutet av raden.

Rad 133: Hittar indexet av den efterSökt raden.

Rad 134: Adderar Synonymet i Slutet av efterSökt raden och uppdaterar raden.

```

137     // removeSynonym accepts synonym data, and removes a given
138     // synonym for a given word.
139     // If the given word or the given synonym is not present, an
140     // exception of the type IllegalArgumentException is thrown.
141     public static void removeSynonym (String[] synonymData, String word, String synonym)
142     {
143         String synLine = getSynonymLine(synonymData, word);
144         String [] tempSyn = synLine.split(regex: "[| ,]+");
145         String [] finalSyn = new String[tempSyn.length - 1];
146         int counter = 0;
147         for(int i = 0; i < tempSyn.length; i++)
148         {
149             if(!tempSyn[i].equalsIgnoreCase(synonym))
150             {
151                 finalSyn[counter] = tempSyn[i];
152                 counter++;
153             }
154         }
155         finalSyn[0] += "|";
156         String synString = Arrays.toString(finalSyn).replace(target: "[", replacement: "")
157             .replace(target: "]", replacement: "").replace(target: "|", replacement: "|");
158
159         synonymData[synonymLineIndex(synonymData, word)] = synString;
160     }

```

Metoden tar in en mängd data av typen `String` (Som array) ett ord samt en synonym.

Den kommer då hitta efterSökt raden i data mängden och tar bort en vis synonym.

Rad 143: Tar fram den efterSökt raden som sträng.

Rad 144: Splittar strängen `SynLine` efter givna tecken och lagrar i en array `tempSyn`

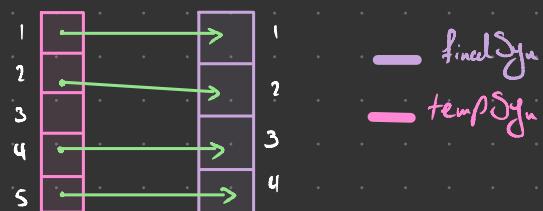
Rad 145: Skapar en tom array med en längd som har en plats mindre än `tempSyn`.

Rad 146: definieras en Counter som används senare i koden.

for-loopen gör igenom hela arrayn tempSyn, och jämför elementer i tempSyn med synonym. Om tempSyn[i] inte är synonym, så läggs det i finalSyn i platsen counter och till slut plussas counter.

// Counter behövs i for-loopen då finalSyn är en kortare array än tempSyn.

Tex: tempSyn har längd 5 och finalSyn har längd 4. Om synonymet i platsen 3 i tempSyn ska tas bort, så kopierar vi tempSyn[1] → finalSyn[1], tempSyn[2] → finalSyn[2]. När vi kommer till plats 3 ska vi hoppa över (d.v.s. i++), dock ska inte counter hoppa över



. Rad 155: lägger till | mellan ordet och synonymer (efter ordet).

. Rad 156 & 157: Omvandlar finalSyn till sträng och tar bort skräftecken m.h.a. replace().
Och sparar den i variabeln synString.

. Rad 159: uppdaterar data mängden med den nya raden som är nu redigerad.

```
161     // sortIgnoreCase sorts an array of strings, using
162     // the selection sort algorithm
163     private static void sortIgnoreCase (String[] strings)
164     {
165         for(int i = 0; i < strings.length-1; i++)
166         {
167             int min = i;
168             for(int j = i + 1; j<strings.length; j++)
169             {
170                 if(strings[j].compareToIgnoreCase(strings[min]) < 0)
171                 {
172                     min = j;
173                 }
174             }
175             if(i != min)
176             {
177                 String temp = strings[i];
178                 strings[i] = strings[min];
179                 strings[min] = temp;
180             }
181         }
182     }
```

Implementation av Selection Sort algoritmen.

```

184     // sortSynonymLine accepts a synonym line, and sorts
185     // the synonyms in this line
186     private static String sortSynonymLine (String synonymLine)
187     {
188
189         String[] synLine = synonymLine.split(regex: "[|]");
190         String word = synLine[0].trim();
191         String [] synonym = synLine[1].trim().split(regex: "[ ,]+");
192         sortIgnoreCase(synonym);
193         String synTemp = Arrays.toString(synonym).replace(target: "[",replacement: "")
194             .replace(target: "]", replacement: "");
195         String wordLine = word + " | " + synTemp;
196         return wordLine;
197     }

```

- Metoden tar in en sträng, sorterar efter bokstavsordning, och returnerar en sträng.
- Rad 189: Splittrar synonymline efter | och lagrar i synLine.
- Rad 190: lagrar strängen till vänster om | som word.
- Rad 191: lagrar strängen till höger om | som Synonym och splittrar den efter given tecken.
- Rad 192: Sorterar Synonym genom att anropa sortIgnoreCase().
- Rad 193 & 194: Omvandlar Synonym till sträng och tar bort skäftecken m.h.a. replace(), och lagrar i synTemp.
- Rad 195: Slår ihop word och synTemp och lagrar i wordLine.
- Rad 196: returnerar wordLine

```

199     // sortSynonymData accepts synonym data, and sorts its
200     // synonym lines and the synonyms in these lines
201     public static void sortSynonymData (String[] synonymData)
202     {
203         for(int line = 0; line < synonymData.length; line++)
204         {
205             synonymData[line] = sortSynonymLine(synonymData[line]);
206         }
207         sortIgnoreCase(synonymData);
208     }

```

- Metoden tar in en array av strängar och sorterar den efter bokstavsordningen.
- for-loopet används för att sortera synonymer varje rad som finns i datamängden.

Rad 207: anropar sortIgnoreCase() för att sortera hela mängden efter bokstavsordning.