



- **b)** In diesem Fall kann man sich einfach das Maximum der Anzahl der Terminal-Symbole jedes Nicht-Terminals nehmen, also **10**. Bei jedem Durchlauf kann man dann ein anderes dieser 10 Knoten durchlaufen, wobei man bei den anderen Nicht-Terminalen ebenfalls immer ein anderes nimmt.
- c) Wie in a), nur dass jetzt noch die optionale Kante mitgenommen werden muss (die, die die Nicht-Terminale überspringen), da sonst für jedes Nicht-Terminal die Anzahl der Knoten und die Anzahl der Kanten gleich ist. Pro Nicht-Terminal hat man dann (Anzahl\_der\_Knoten + 1) Kanten und das Maximum ist 11.
- d) Alle Pfad-Möglichkeiten: 6\*11\*11\*11 = 7.986

```
/** Method to analyse the elements of a given list of strings. */
01
02
      public boolean analyseList(List<String> in_lstList, int[] out_aiResults) {
03
          boolean bResult = false;
04
          if (in_lstList != null && !in_lstList.isEmpty()) {
05
              // initialize counters
06
              int iNumberCounter = 0;
07
              int iEvenNumberCounter = 0;
80
              int iWordCounter = 0;
09
              int iEvenWordCounter = 0;
              // go through all elements of the given list
10
11
              for (String strElement : in_lstList) {
                  // ignore 'null' and empty strings
12
                  if (strElement != null && !strElement.isEmpty()) {
13
14
                       int iNumber = 0;
15
                      boolean bNumberFound = false;
                       // try to parse the current list element to an int value
16
17
18
                           iNumber = Integer.parseInt(strElement);
19
                           bNumberFound = true;
20
                       } catch (NumberFormatException e) {
21
                       // do nothing
                      }
22
23
                       // check if a number was found
24
                       if (bNumberFound) {
25
                           iNumberCounter++;
                           // check if number is even or odd
26
                           if (iNumber % 2 == 0) {
27
28
                               iEvenNumberCounter++;
29
                               System.out.println(iEvenNumberCounter +
30
                                       ". even number found: " + iNumber);
31
32
                               System.out.println(iNumberCounter +
33
                                       ". number found: " + iNumber);
                           }
34
35
                       } else { // no number found
                           iWordCounter++;
36
37
                           // check if the number of characters of the current
                           // list element is even or odd
38
                           if (strElement.length() % 2 == 0) {
39
40
                               iEvenWordCounter++;
                               System.out.println(iEvenWordCounter + ". word " +
41
42
                                       "with even number of characters found: " +
43
                                       strElement);
44
                           } else {
                               System.out.println(iWordCounter + ". word found: " +
45
                                       strElement);
46
47
                           } // end if
                      } // end if
48
                  } // end if
49
              } // end for
50
              // put the results into output array
51
              if (out_aiResults != null && out_aiResults.length >= 4) {
52
                  out_aiResults[0] = iNumberCounter;
                                                          // numbers found
53
54
                  out_aiResults[1] = iEvenNumberCounter; // even numbers found
                  out_aiResults[2] = iWordCounter;
                                                          // words found
55
                  out_aiResults[3] = iEvenWordCounter;
                                                          // words with even number
56
                                                           // ...of characters found
                  bResult = true;
57
              } // end if
58
59
60
          return bResult;
61
      }
```