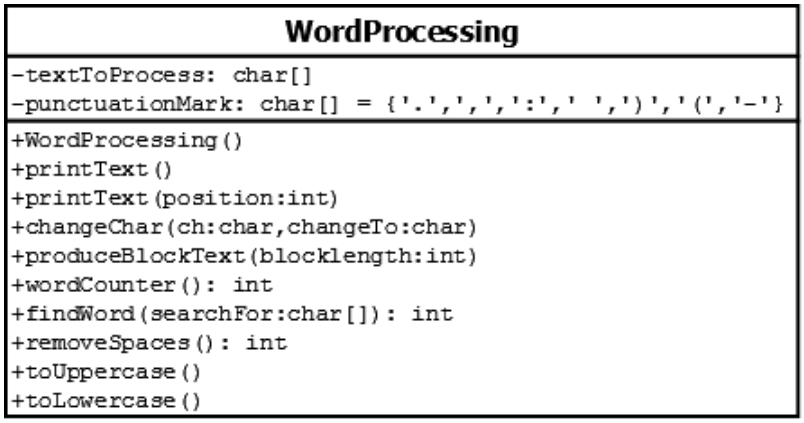
Word processing – character Array

You should implement a class which processes text.

Class diagram:



The (given) text which should be processed is stored in the char-Array in the constructor. Any text file can be stored in the char-Array. You will find the text (name: **18_laborordnung.txt**) for our example in the same folder as your task. Copy the file into your BlueJ-project directory.

```
import java.io.*;

public class WordProcessing{
    // hold the text
    private char[] textToProcess;
    // punctuation marks
    private char[] punctuationMark={'.',',',':',' ',')','(','-'};
    /**
     * Constructor for objects of class WordProcessing
     * The constructor reads a given text file into the
     * character array textToProcess
     */
    public WordProcessing()throws IOException{
           // String to hold the text
           String text="";
           // connect to the file for reading
           FileReader fr=new FileReader("17_laborordnung.txt");
           // read character by character till end of file (returns -1)
           int is=fr.read();
           while (is!=-1){
               text=text+(char)is;
               is=fr.read();
           }
          // textToProcess gets the text
           textToProcess=text.toCharArray();
    }
}
```

Implement the following methods:

- **printText()** – the text is output to the console

- **printText(position:int)** – the text is output to the console starting from a given position
  changeChar(ch:char, changeTo:char) – the character **ch** is replaced with **changeTo**.
- **produceBlockText(blocklength:int)** – produces a text with justification (=Blocksatz) therefore you have to insert '\n' after blocklength-characters. **Attention:** As a result the char-Array **textToProcess** increases in size. **Hint:** Use a local String-variable in your algorithm.

After calling **produceBlockText(10)** and afterwards **printText()** the following output is produced:

```
Werkstätte
n und Labo
ratorienor
dnung


VERHALTENS
-  UND SIC
HERHEITSVO
RSCHRIFTEN
 FÜR WERKS
TÄTTEN UND
 LABORATOR
IEN  DER H
TL SPENGER
GASSE (Gül
tig ab Win
tersemeste
r 2012/201
usw.
```

- **wordCounter():int** – counts the number of words in the text and returns this number. Use the array **punctuationMark** for this method. The punctuation characters are stored in that array.
- **findWord(searchFor: char[]): int** – returns the index of the first occurrence of **searchFor** in the text.
  Example:
  searchFor: {'1','.'} returns **203** By calling **printText(203)** you can check if the algorithm works correctly.
  Please note: In BlueJ you have to use curly brackets for a char-Array.
- **removeSpaces():int** – removes unnecessary spaces in the text and returns the number of deleted spaces.
  Attention: As a result the char-Array **textToProcess** decreases in size. **Hint:** Use a local String-variable in your algorithm.
- **toUppercase()** – replaces all the lowercase letters with uppercase letters in textToProcess
  Hint: Use the static method Character.toUpperCase(ch:char):char in order to do so.
- **toLowercase()** - replaces all the lowercase letters with uppercase letters in textToProcess
- **Hint:** Use the static method Character.toUpperCase(ch:char):char in order to do so.

K**eywords:** Virtual machine, syntax, compiler, editor, object, object attribute, object behaviour, get-/set-methods, class, attributes, data type, String, int, parameter, default-constructor, constructor with parameter, this, method: System.out.println(….), data encapsulation, boolean, byte, mathematical operators (+,-,*,/), private, public, comparative operators (<, <=, >, >=, !=, ==), logical operators (&&, ||, ^, &, |, !), algorithms, building blocks/algorithms (sequence, selection, iteration/loop), modulo operator, prime number, array, length, for, while, switch.