Arbeitsblatt 12: Mehr OOP und RegExp

Aufgaben

1. Implementieren Sie eine `Bruch()` Klasse mit den Attributen `Zähler` und `Nenner` und den folgenden Methoden:
   - alle Grundrechenarten mit anderen Brüchen
   - `erweitern(<zahl>)`
   - `kuerzen(<zahl>)` (wenn `<zahl>` nicht gegeben ist, nutze GCD)
   - `dezimal(<stellenanzahl>)`
2. Entfernen Sie mittels geeigneter RegExp-Methoden die Seitenzahl aus der Textdatei, die Sie unter `https://github.com/UweZiegenhagen/Skriptsprachen_WS1920/tree/master/VL_10` finden.
3. In dieser Textdatei finden sich auch Wörter, bei denen auch der zweite Buchstabe groß geschrieben ist. Nutzen Sie geeignete RegExp-Methoden, um diese Fehler zu beseitigen.

   Hinweis: Erhöhter Schwierigkeitsgrad, siehe `https://stackoverflow.com/questions/8934477/making-letters-uppercase-using-re-sub-in-python`.

`re.` **sub**(*pattern*, *repl*, *string*, *count=0*, *flags=0*)

Return the string obtained by replacing the leftmost non-overlapping occurrences of *pattern* in *string* by the replacement *repl*. If the pattern isn't found, *string* is returned unchanged. *repl* can be a string or a function; if it is a string, any backslash escapes in it are processed. That is, `\n` is converted to a single newline character, `\r` is converted to a carriage return, and so forth. Unknown escapes of ASCII letters are reserved for future use and treated as errors. Other unknown escapes such as `\&` are left alone. Backreferences, such as `\6`, are replaced with the substring matched by group 6 in the pattern. For example:

```
>>> re.sub(r'def\s+([a-zA-Z_][a-zA-Z_0-9]*)\s*\(\s*\):',
...         r'static PyObject*\npy_\1(void)\n{',
...         'def myfunc():')
'static PyObject*\npy_myfunc(void)\n{'
```

If *repl* is a function, it is called for every non-overlapping occurrence of *pattern*. The function takes a single match object argument, and returns the replacement string. For example:

```
>>> def dashrepl(matchobj):
...     if matchobj.group(0) == '-': return ' '
...     else: return '-'
>>> re.sub('-{1,2}', dashrepl, 'pro----gram-files')
'pro--gram files'
>>> re.sub(r'\sAND\s', ' & ', 'Baked Beans And Spam', flags=re.IGNORECASE)
'Baked Beans & Spam'
```