Regularni izrazi v javi

Programiranje 2, Tomaž Dobravec

Kaj je regularni izraz?

Regularni izraz je zaporedje znakov, s katerim opišemo družino nizov.

Primera:

.	Niz, sestavljen iz poljubnega (enega) znaka (primeri: 'x', 'e', '3', '#', '.', ...)
a.a	niz, sestavljen iz treh črk: a + katerakoli črka + a (primeri: 'aaa', 'a a', 'a.a', 'aAa', 'a%a', ...)

VIR: http://www.ocpsoft.org/opensource/guide-to-regular-expressions-in-java-part-1/

Razredi znakov

```
.       Dot, any character (may or may not match line terminators, read on)
\d      A digit: [0-9]
\D      A non-digit: [^0-9]
\s      A whitespace character: [ \t\n\x0B\f\r]
\S      A non-whitespace character: [^\s]
\w      A word character: [a-zA-Z_0-9]
\W      A non-word character: [^\w]
```

Pozor - za uporabo znakovnih razredov je potrebno uporabiti dvojno "predznačenje":

```
String regularniIzraz = "\\d";    // ena števka
```

Kvantifikatorji

```
*       Match 0 or more times
+       Match 1 or more times
?       Match 1 or 0 times
{n}     Match exactly n times
{n,}    Match at least n times
{n,m}   Match at least n but not more than m times
```

Meta znaki

```
\       Escape the next meta-character (it becomes a normal/literal character)
^       Match the beginning of the line
.       Match any character (except newline)
$       Match the end of the line (or before newline at the end)
|       Alternation ('or' statement)
()      Grouping
[]      Custom character class
```

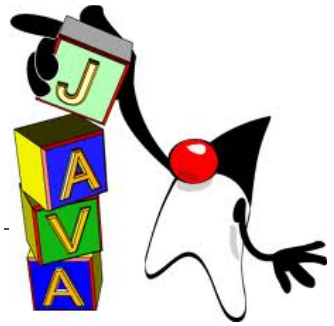
Primer uporabe v metodi `split()`

```
String ocene = "Mojca Novak:10:9";
ocene.split(":");       // ->  {"Mojca Novak", "10", "9"}
ocene.split(": ");      // ->  {"Mojca Novak:10:9"}
ocene.split("[: ]");    // ->  {"Mojca", "Novak", "10", "9"}
```

```
String racun = "1 + 3.3 = 4.3";
racun.split(" ");       // -> {"1", "+", "3.3", "=", "4.3"}
racun.split("[ +=]+"); // -> {"1", "3.3", "4.3"}
```

Naloga

Uporaba metode `matches()`



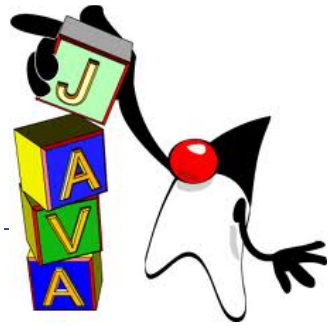
`regex/Telefonska.java`

Napiši program, ki prebere niz in preveri, ali gre za mobilno telefonsko številko.

```
String niz = "031 / 123 212”;
String regex = "0(3|4)(0|1)[/ ]*\\d{3} *\\d{3}";

if (niz.matches(regex))
   System.out.println("OK");
```

Naloga

Uporaba razredov `Pattern` in `Matcher`



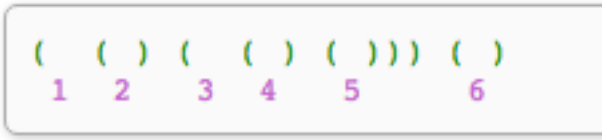
`regex/Podatki.java`

Napiši program, ki izpiše vse telefonske številke nekega uporabnika.

Primer: v nizu `"Ime: Miha, Tel: 01234567, Mobi: 040123456, Mesto: Ljubljana"`; naj porgram najde številki `Tel: 01234567` in `Mobi:040123456`.

```
String input =
  "Ime: Miha, Tel: 01234567, Mobi:040123456, Mesto: Ljubljana";

Pattern p = Pattern.compile("(Tel|Mobi): *\\d+");
Matcher m = p.matcher(input);
while (m.find()) {
  System.out.println("Nasel: " + m.group());
}
```

Skupine (groups)



- Dele niza lahko razbijemo v skupine
- Ena skupina je vse, kar je znotraj oklepajev

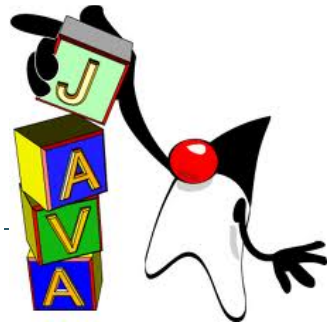
- Primer: niz `EMSO=1509978505123` lahko razbijemo, na primer, v dve skupini: `EMSO=` in `1509978505123`.

```
String input = "EMSO=1509978505123”;

Pattern p = Pattern.compile("(EMSO=)(\\d+)");

Matcher m = p.matcher(input);

if (m.find()) {

  System.out.println("Prva  skupina: " + m.group(1));

  System.out.println("Druga skupina: " + m.group(2));
```

Naloga

Obrni besede - uporaba skupin

regex/Obrni.java



Napiši program, ki zamenja prvo in zadnjo besedo vsakega stavka v podanem nizu.

```
Pattern p = Pattern.compile("([^ ]+)( [^.]* )([^. ]+)\\.?");
Matcher m = p.matcher(input);
while (m.find()) {
  System.out.println(m.group(3) + m.group(2) + m.group(1));
}
```