

## Zadanie 1

Napisz program analizujący napis zawierający ujęte w nawiasy okrągłe pary słów oddzielonych przecinkiem (być może z pewną liczbą spacji po obu jego stronach). Program ma wyświetlić (w osobnych liniach) dla każdej takiej pary oba znalezione słowa — najpierw drugie z nich, potem, po dowolnym separatorze, pierwsze.

Na przykład program

```
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class RegParen {
    public static void main (String[] args) {
        String str = "Lisboa (Lisbon , Portugal), " +
                    "Warszawa (Warsaw, Poland), and " +
                    "Roma (Rome,Italy)";

        String pat = ...

        Matcher m = Pattern.compile(pat).matcher(str);
        while (m.find()) {
            // ...
        }
    }
}
```

[download RegParen.java](#)

powinien wypisać

```
Portugal -> Lisbon
Poland -> Warsaw
Italy -> Rome
```

## Zadanie 2

Uzupełnij poniższy program tak aby wypisywał wszystkie słowa znalezione we wczytanym tekście w których pierwsza i ostatnia litera jest taka sama lub różni się tylko wielkością (czyli na przykład słowo *Angola* spełnia ten warunek):

```
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import static java.nio.charset.StandardCharsets.UTF_8;
```

[download FirstLast.java](#)

```

public class FirstLast {
    public static void main (String[] args) {
        String text = null;
        try {
            byte[] bytes =
                Files.readAllBytes(Paths.get("FirstLast.txt"));
            text = new String(bytes, UTF_8);
        } catch (IOException e) {
            System.out.println("Problems...");
            System.exit(1);
        }
        String reg = /* regex */;
        Matcher m = Pattern.compile(reg).matcher(text);
        while (m.find())
            System.out.println(m.group());
    }
}

```

Na przykład dla pliku zawierającego text

```

Angola, Bahama, Zanzibar, baobab,
égalité, ćwierć, pół, Kraśnik

```

wynik powinien być

```

Angola
baobab
égalité
ćwierć
Kraśnik

```

### Zadanie 3

Zakładamy, że data ma format: dzień (jedna lub dwie cyfry), miesiąc (jedna lub dwie cyfry) i rok (dokładnie cztery cyfry). Numer dnia i miesiąca oraz numer miesiąca i roku oddzielone są *tym samym* separatorem, którym może być *tylko* kropka, ukośnik lub myślnik.

Uzupełnij poniższy program tak, aby w podanym tekście wszystkie prawidłowo zapisane daty zamienione zostały na postać rok-miesiąc-dzień z separatorem takim samym jak w oryginalnej formie:

[download DayMonthYear.java](#)

```

import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class DayMonthYear {
    public static void main (String[] args) {
        String text = "good 2-01-2020 abc \nwrong " +
            "4/7-2010\ngood 11/11/2011 " +

```

```

        "wrong 211/3/1998 jkl\n" +
        "good 13.9.1999 wrong 1.1.20121\n" +
        "good 11.05.1823, wrong 9_10_2000";
    System.out.println("*** Before\n" + text);
    String r = /* regex */;
    String s = /* replacement */;
    text = text.replaceAll(r, s);
    System.out.println("\n*** After\n" + text);
}
}

```

Dla tekstu podanego w przykładzie, wynik powinien być

```

*** Before
good 2-01-2020 abc
wrong 4/7-2010
good 11/11/2011 wrong 211/3/1998 jkl
good 13.9.1999 wrong 1.1.20121
good 11.05.1823, wrong 9_10_2000

*** After
good 2020-01-2 abc
wrong 4/7-2010
good 2011/11/11 wrong 211/3/1998 jkl
good 1999.9.13 wrong 1.1.20121
good 1823.05.11, wrong 9_10_2000

```

#### Zadanie 4

W programie  $\text{\LaTeX}$ , używanym powszechnie do składania tekstów, szczególnie naukowych i technicznych, instrukcje (komendy) mają zwykle postać `\command{argument}`, gdzie `argument` może składać się z wielu słów. W szczególności `\index{words}` powoduje umieszczenie `words` jako hasła w indeksie na końcu książki czy artykułu.

Napisz program wczytujący plik  $\text{\LaTeX}$ 'owy i znajdujący w nim, za pomocą wyrażeń regularnych, wszystkie hasła umieszczone w komendach `\index{...}` i zapisujący znalezione wyrażenia do pliku tekstowego, każde hasło w osobnej linii. Zwróć uwagę, że hasło może się składać z wielu słów i zawierać znaki niealfanumeryczne. Może też być tak, że jeden fragment hasła wypada na końcu linii, a jego kontynuacja znajduje się już w linii następnej.

Na przykład dla pliku wejściowego zawierającego polski tekst w kodowaniu UTF-8 (nie zmieniaj podziału tekstu na linie)

```

Za początek okresu hellenistycznego uznaje się śmierć
\emph{Stagiryty}.\index{Stagiryta} To okres powstania
do dziś żywych koncepcji etycznych: \emph{epikureizmu}
(Epikur,\index{Epikur} ur. 341, zm. 270 p.n.e.)
mówiącego, że sensem życia jest doznawanie przyjemności

```

(por. \emph{hedonizm}) i \emph{stoicyzmu} (IV wiek p.n.e.  
- II n.e.) dającego wzór postawy mędrca wobec śmierci  
(Zenon z Kition,\index{Zenon  
z Kition} Chryzyp, Seneka, mistrz retoryki Cyceeron  
i Marek Aureliusz\index{Marek  
Aureliusz}, filozof na tronie). Sceptycy starożytni (Pirron,  
\index{Pirron} Sekstus Empiryk)\index{Sekstus  
Empiryk} z pobudek etycznych (uspokojenie umysłu) podawali  
w wątpliwość możliwość wszelkiej wiedzy i dali początek wszelkim  
nurtom sceptycznym. Tomasz z Akwinu\index{Tomasz  
z Akwinu} (ur. 1225, zm. 1274)

wynik powinien być

Stagiryta  
Epikur  
Zenon z Kition  
Marek Aureliusz  
Pirron  
Sekstus Empiryk  
Tomasz z Akwinu

---