Regexid Elus

Vesal Vojdani
(TÜ Arvutiteaduse Instituut)

Korraldusest

- Tähtajad nihkuvad ühe päeva võrra edasi!
- Meie soov on siiski, et alustaksite kodutööga enne praktikumi.
- Teisipäevases 5. rühmas on veel vaba ruumi…
- Tagasiside anname rühmade järgi, aga me ei takista teid käimas teises praktikumiajas, kui seal piisavalt ruumi on!

Kodutöö

1. Proovige ise dokumentatsiooni põhjal aru saada, kuidas peab kasutama…

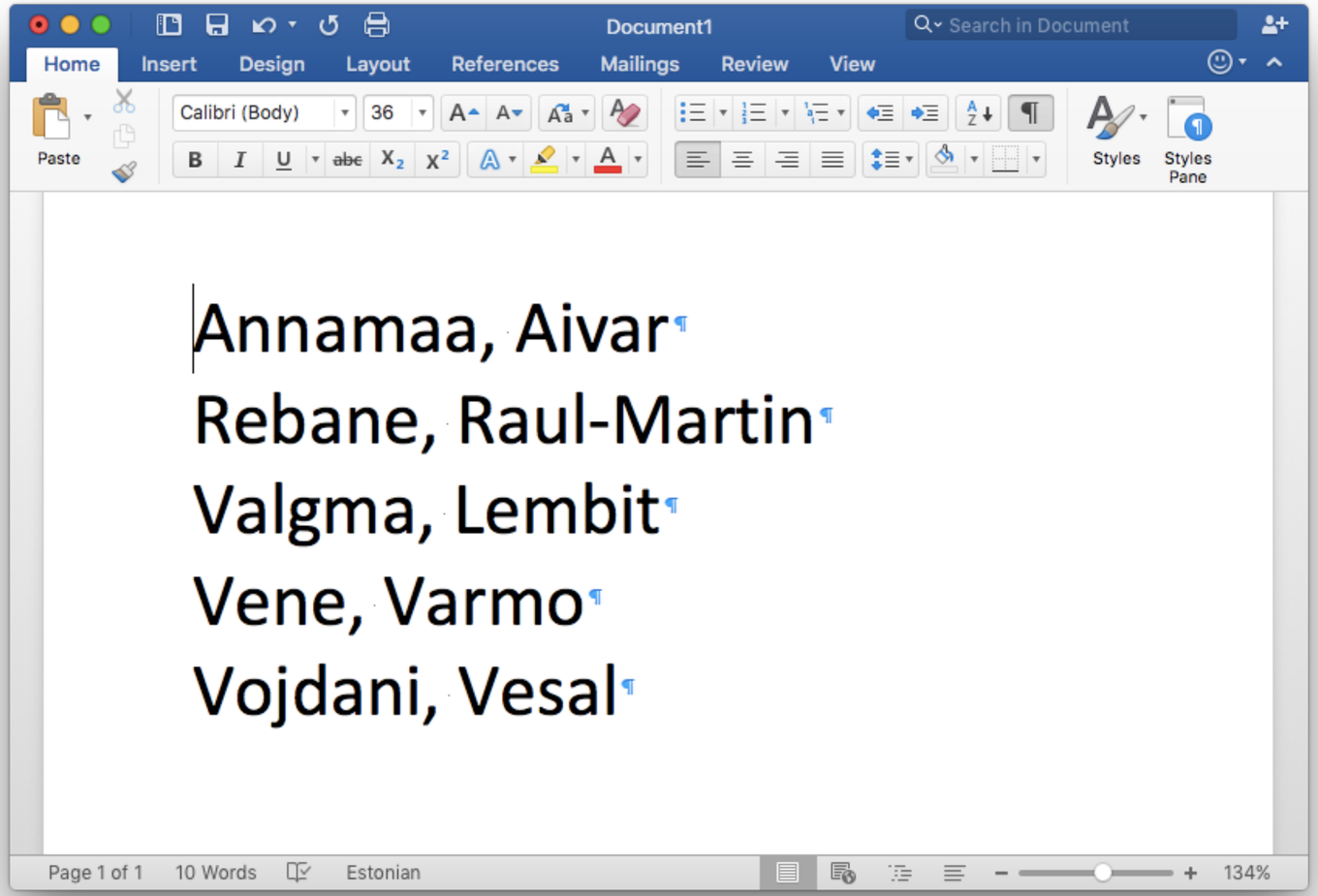
2. Lisame homseks mõned kasutusnäited.

3. Praktikumis rohkem scaffolding, et abistada neid, kes kuidagi ise hakkama ei saa.

```
public abstract class Regex {

    public static Regex repetition(Regex regex) { ... }

    public static Regex concatenation(Regex left, Regex right) { ... }

    . . .

    public abstract boolean matchesEmptyWord();
    public abstract boolean matchesInfinitelyManyWords();

    public static void main(String[] args) {
        Regex r = repetition(concatenation(letter('a'), letter('b')));
        . . .
    }
}
```

Defend Civil Liberties!

Otsustage ise, kas teha üks klass, või kas jääb abstract.
Mis siis tegema peab? Kus luua alamklassid?

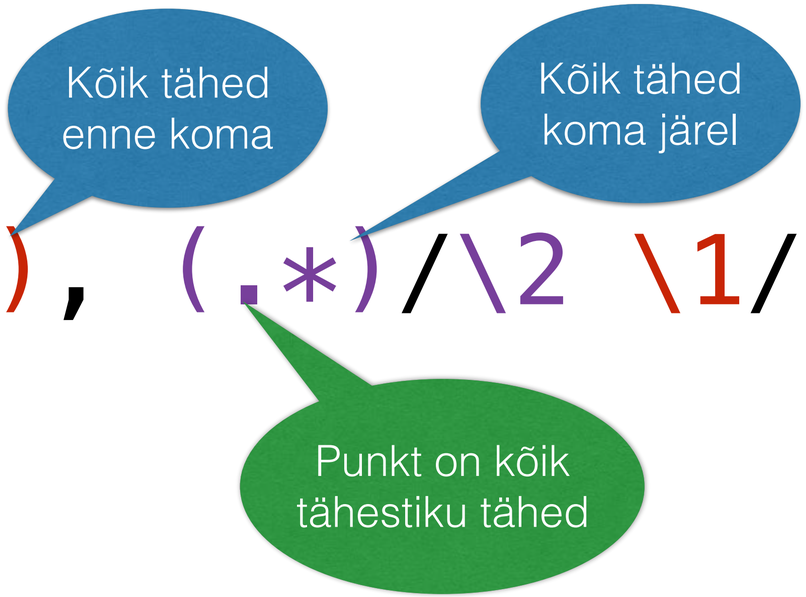


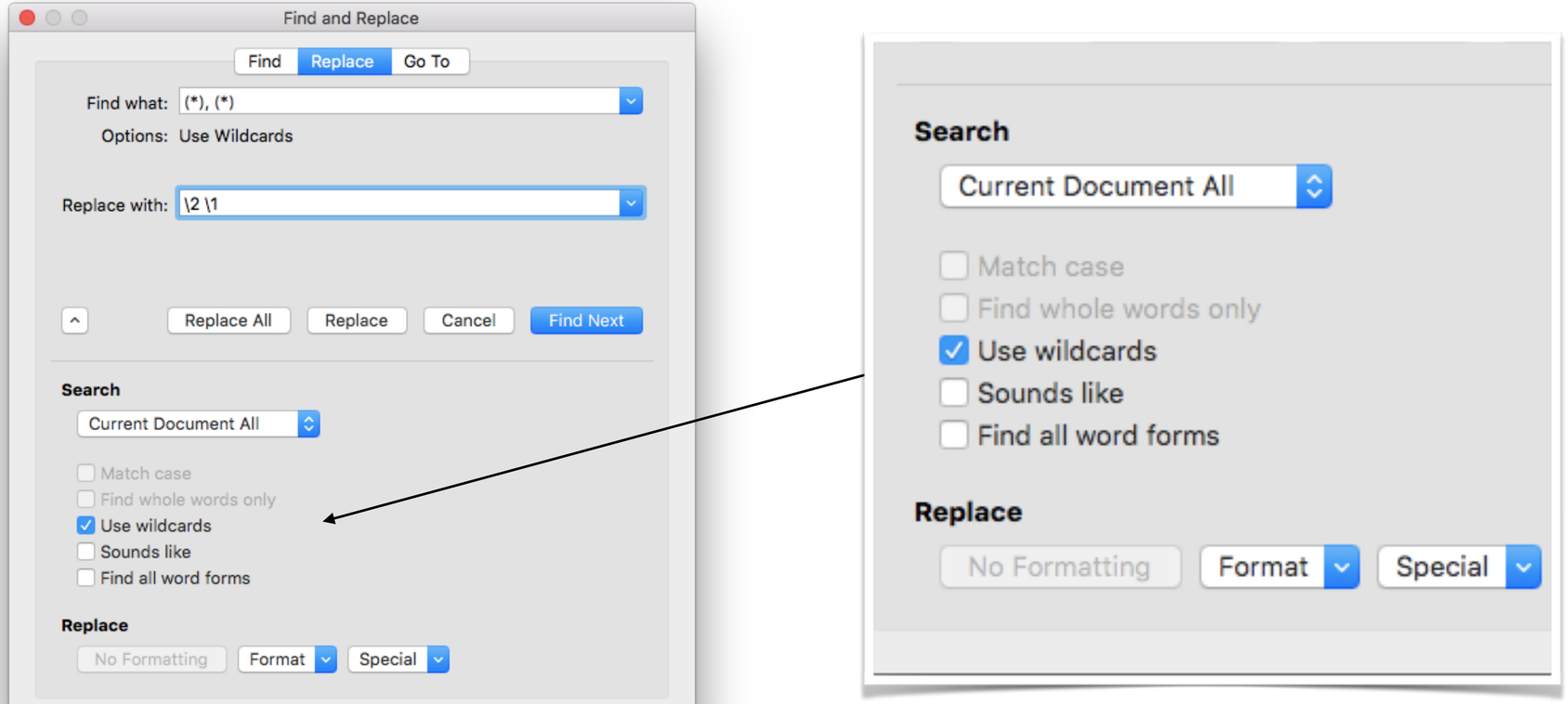
Teisendus

Perekonnanimi, Nimi → Nimi Perekonnanimi

Capturing Groups

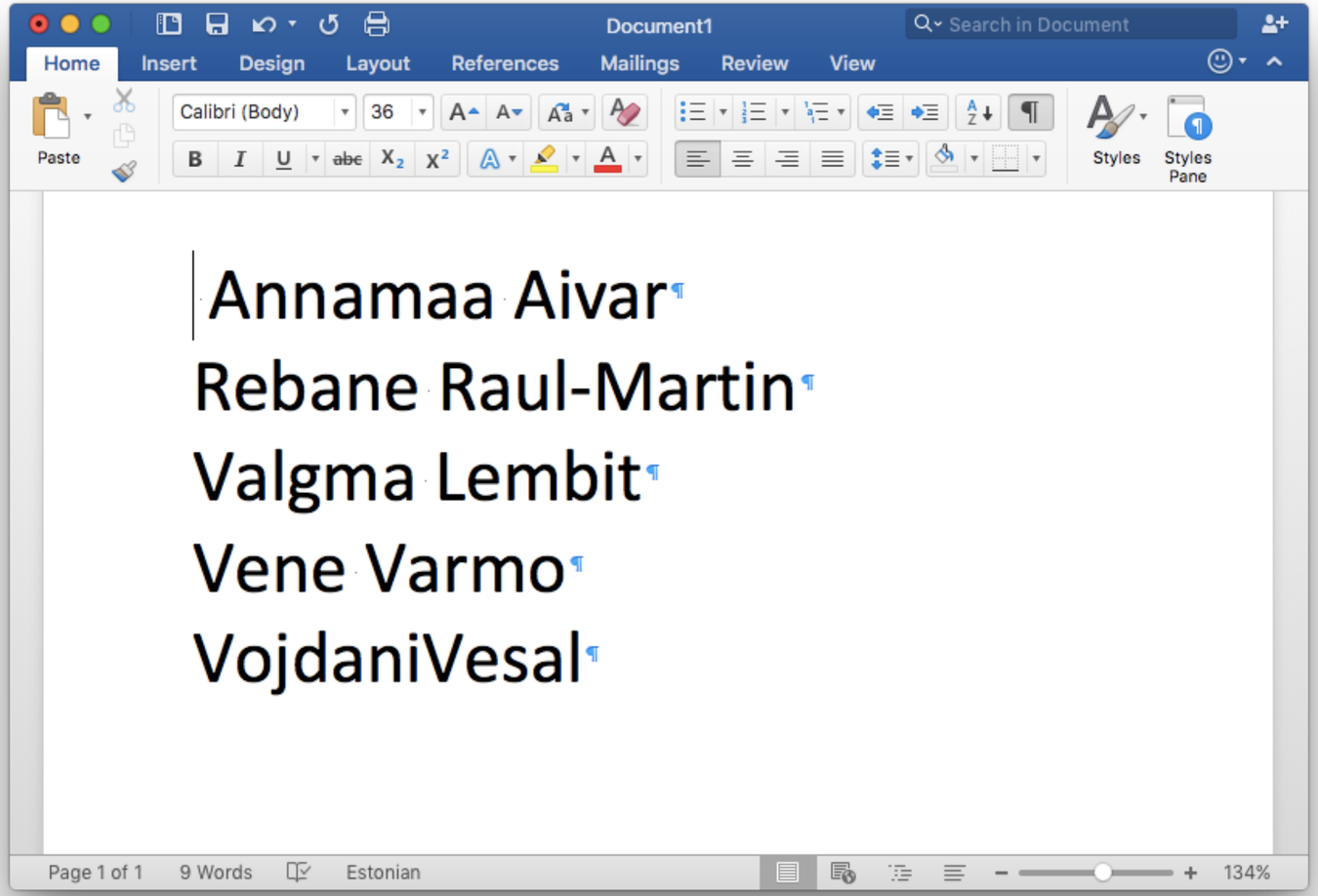
- Tihti on sõnes alamrühmitusi vaja kätte saada.
- Rühmitamine: “Vene, Varmo” ära tunda kui “(Vene)[1], (Varmo)[2]”
- Väljastamine: “`\2 \1`”
- Näiteks: `sed –E 's/(.*), (.*)/\2 \1/'`





Word: “Use Wildcards”

(Advanced Find & Replace…)



Peaaegu…

Ma ei oska Wordile selgitada, et võtaks terve rida.
Selleks on muidu operaatorid ^ ja $.

Proovime Javaga

- Java regexite teek on `java.util.regex`
- Nende funktsionaalsus on ka otse sõnede kaudu kättesaadavad.
- `str.split(regex) = Pattern.compile(regex).split(str)`

str.split(“,”) on aga kiirem.
(*fast-track* lihtsate juhtumite jaoks)

```
public class Demo {
    public static void main(String[] args) throws IOException {
        Path path = Paths.get(args[0]);
        Files.lines(path).forEachOrdered(Demo::printline);
    }

    private static void printline(String line) {
        String edited = line.replaceAll("(.*), (.*)", "$2 $1");
        System.out.println(edited);
    }
}
```

Java kood

```
line.replaceAll("(.*), (.*)", "$2 $1")
```

```java
Pattern pattern = Pattern.compile("\\d+");
Matcher matcher = pattern.matcher(input);
while (matcher.find()) {
    String match = matcher.group();
    . . .
}
```

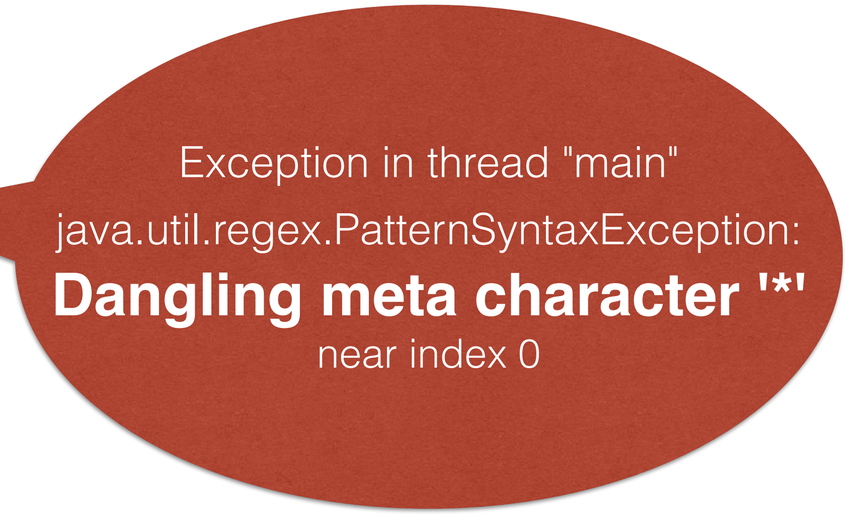
Sisend (sõne)
“Tere, maailm!”

Regex API

Kohtadele (compile) — valmis-olla (matcher) — läks (find)

Sõne tükeldamine

- Oletame, et meil on vaja rühmitada tärniga eraldatud sõnu.
- “`üks *kaks *  kolm`” → [“`üks`”, “`kaks`”, “`kolm`”]
- Selleks on `split()` meetod.
- Peame lihtsalt kirjutama `str.split("*")`?



Maskeerimistähed!?
(*Escape characters / Maskierungszeichen*)

- Regulaaravaldiste “kompilaatori” jaoks on tärn erilise tähendusega.
- Kui me lihtsalt tahame seda sümbolit regexis kasutada, siis peab kaldkriips ette panema.
- Proovime `str.split("\*").`
- Nüüd Java viriseb: illegal escape character :(

System.out.println(" ??? ")

- Midagi tuleks kirjutada küsimärkide asemele, et ekraanile ilmuks järgmist kaks tähte:

```
\*
```

- Sest selline peaks olema sisend regexi kompilaatorile!

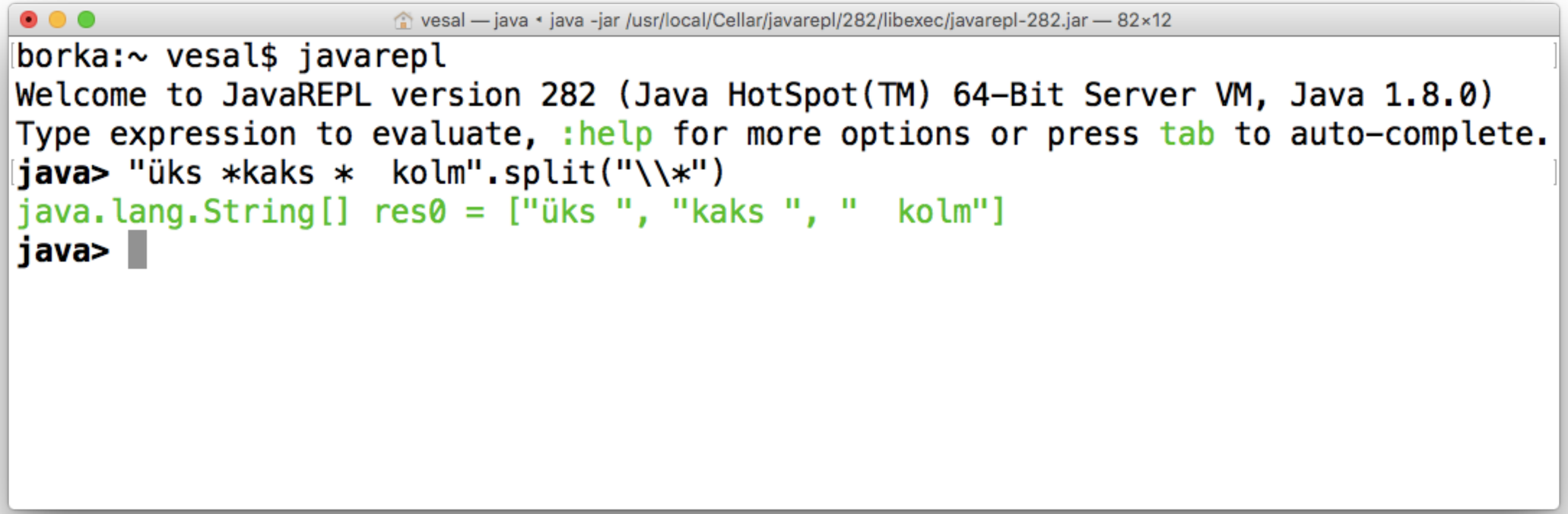
IDE on abiks…

- Kellel on IntelliJ võib lihtsalt seda teksti lõigata ja jutumärkide vahele kleepida.
- Eclipse’il on kuskil seadistus “Escape text when pasting into a string literal”

str.split("*")

- Me üritame Java literaalina kirja panna regex kompilaatori sisend.
- Temal oli vaja maskeeritud sisend, aga Java vajab omakorda kaldkriipsude maskeerimist…
- Oluline on meeles pidada, et “*” koosneb kahest tähest ja kehtib

```
"\\*".charAt(0) == '\\'
```

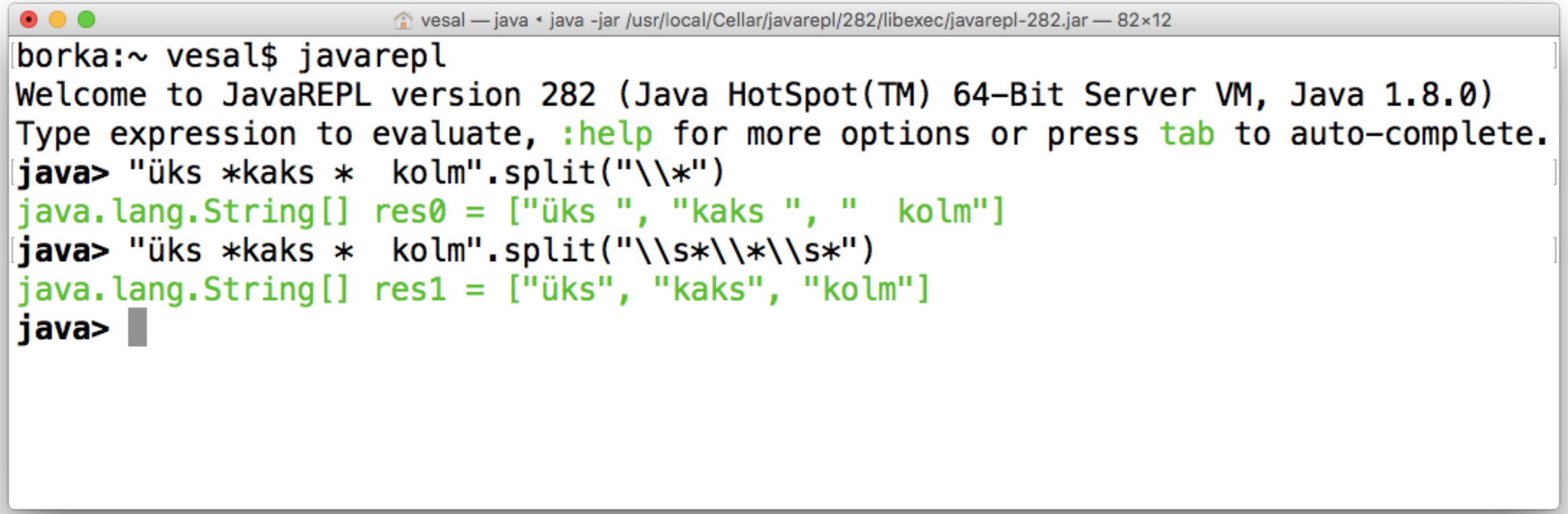


[“üks ”, “kaks ”, “ kolm”]

http://www.javarepl.com/console.html

Teine katse

- “`üks *kaks *  kolm`” → [“`üks`”, “`kaks`”, “`kolm`”]
- Meil oleks vaja whitespace ka eraldajaks.
- Selleks on spetsiaalne grupp ‘`\s`’.
- Meil on vaja regex `\s*\*\s*` Java literaalina.



[“üks”, “kaks”, “kolm”]

```
str.split("\\s*\\*\\s*")
```

Kodutöö moodi tükeldamine

- Oletame, et meil on vaja rühmitada tärniga eraldatud sõnu, aga tärn peaks jääma alles.
- “`üks *kaks * kolm`” → [“`üks`”, “`*`”, “`kaks`”, “`*`”, “`kolm`”]
- Siin võisite valida, kuidas lahendada, aga vaatame, kas saab huvitavamalt seda teha?
- Võiks kogu aeg küsida: kas saame paremini, lihtsamini, ilusamini?

Esimene katse

- String `str` = “`üks *kaks * kolm`”
- Me võiks proovida `str.split("\\s*")`
- Proovime siis `str.split("\\s+")`
- See on parem, aga [“`üks`”, “`*kaks`”, “`*`”, “`kolm`”]
- Kui ainult oleks tärni ümber alati tühikud…

Siin võiks ise javarepl’is järgi proovida!

Igav lahendus: eeltöötlusega asenda(“*”, “ * ”)

Lõbusam Lahendus

- Kui tahame ainult ühe regexiga seda teha?
  (See on tegelikult mõnusalt raske, et äkki isegi natuke areneme…)
- Lookahead: `(?=\\*)`
- Lookback: `(?<=\\*)`
- Proovige näiteks sisendit “`Test **123`”niimoodi tükeldada, et saaks [“`Test`”, “`*`”, “`*`”, “`123`”]
- Kasuks võib tulla ‘`\b`’… (proovige ja küsige slacki teooriakanalis!)