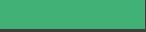


# Vývoj Aplikácií s Viacvrstvovou Architektúrou

## Regulárne výrazy

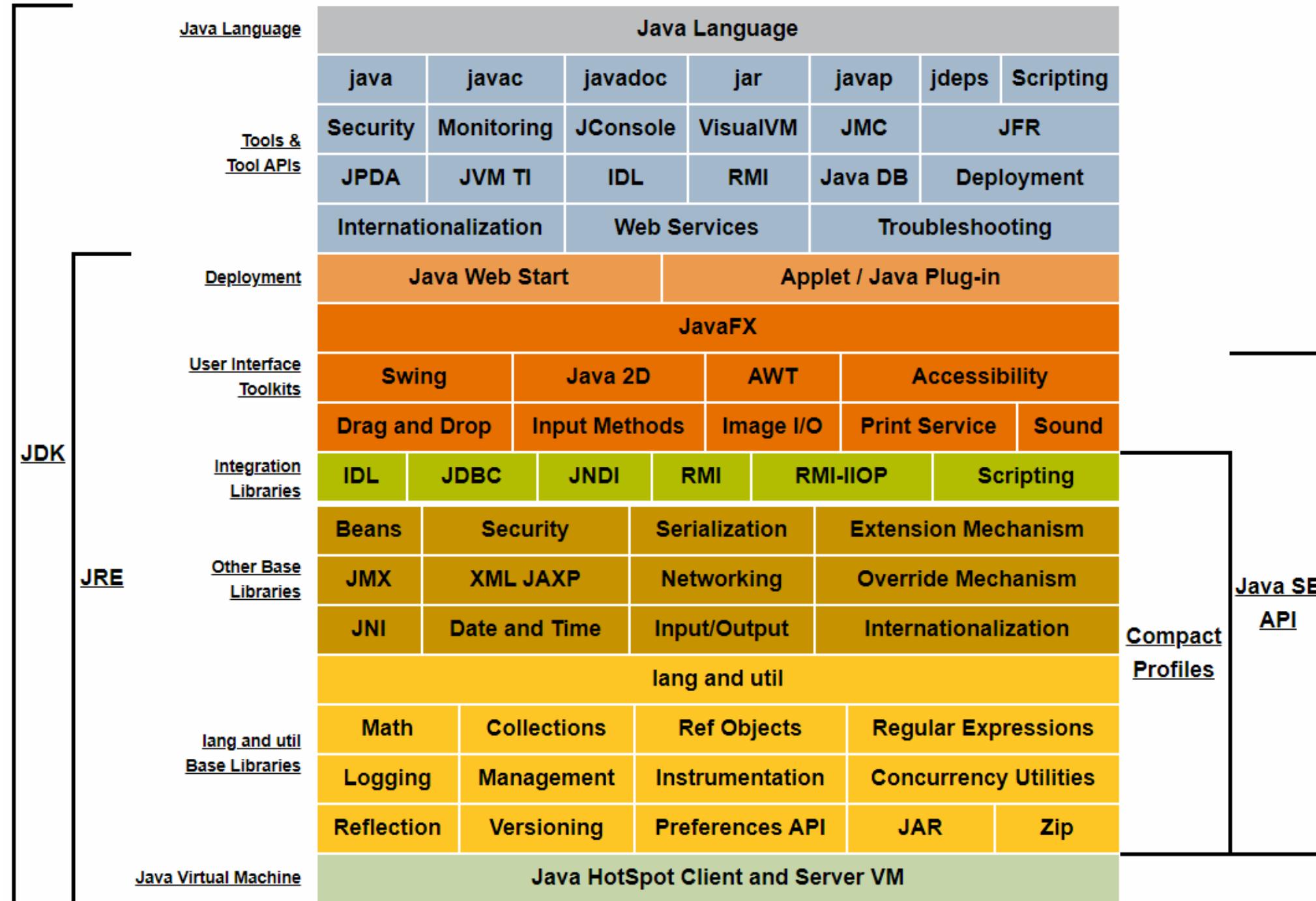




**Čo sú regulárne  
výrazy a ako ich  
efektívne využívať?**

---





# Čo sú regulárne výrazy?

Retázec popisujúci celú množinu retázcov,  
konkrétny regulárny jazyk

```
s/^𓁃𓁃*𓁄-𓁅*(&𓁆??𓁃 )𓁃$/\ ${1}\$/mg
```

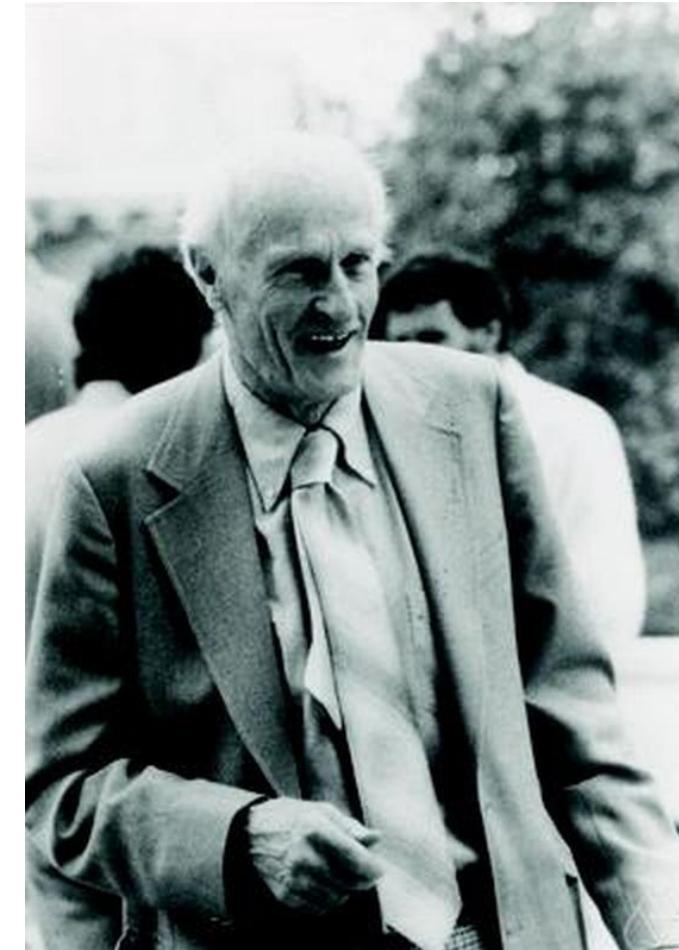
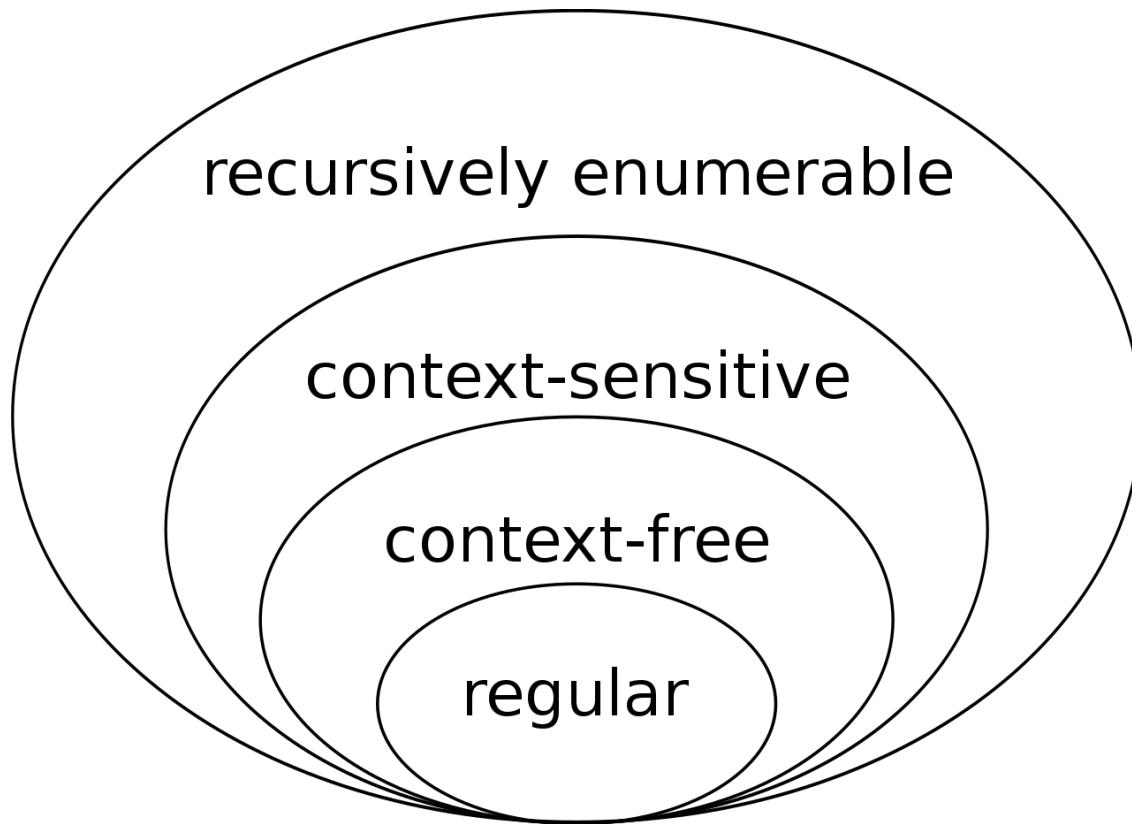


geek and poke

*ANCIENT EGYPTIAN REGEXP*

regexp | regex | re

# Stephen Cole Kleene



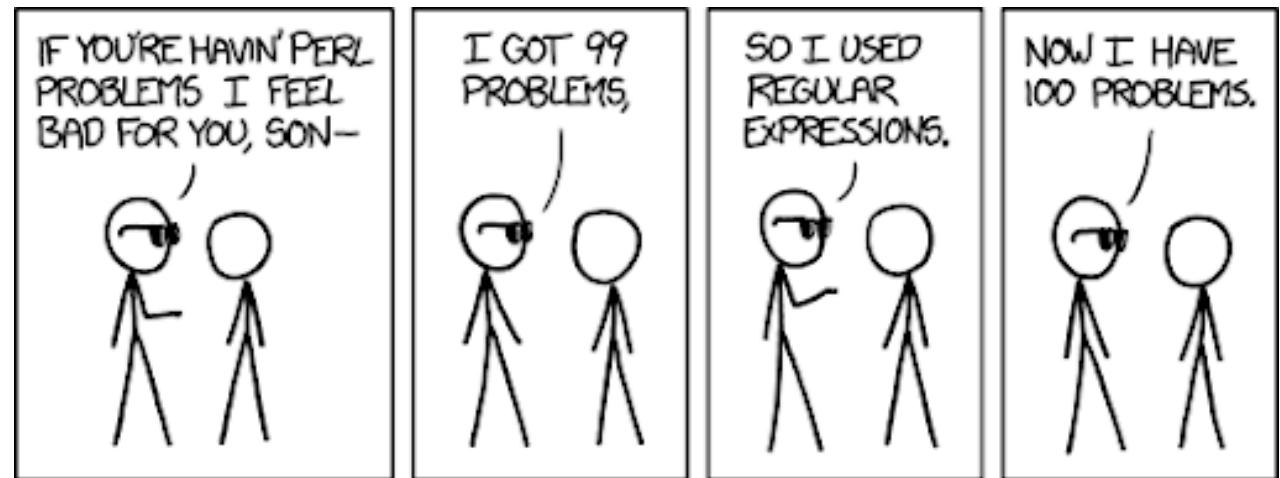
# Načo je to dobré?

## Vyhľadávanie textu

- Zistenie, či vstupný text vyhovuje zadanému regulárному výrazu
- Zistenie pozície vo vstupnom teste, kde sa nachádza zhoda s regulárnym výrazom

## Manipulácia s textom

- Zámena resp. zmena textu
- Výber všetkých zhodných výskytov so zadaným regulárnym výrazom



# Definované triedy alebo množiny znakov

Trieda znakov	Význam
\w	Písmeno alebo číslica z množiny (a-zA-Z_0-9)
\W	Iný znak ako písmeno alebo číslica
\s	"biely znak" (Whitespace) čiže znak, ktorý v informatike predstavuje biele miesto, jednoducho nie je priamo viditeľný. Príkladom takéhoto znaku môže byť napr. znak medzery, znak tabulátora alebo prípadne iného riadiaceho znaku z <a href="#">ASCII</a> . <b>medzera</b> a \f \n\ \r \t \v vid' nižšie v tabuľke
\S	Iný ako tzv. "biely znak" (Non whitespaces)
\d	Číslica (Decimal Digit) čiže znaky 0 až 9
\D	Iný znako ako číslica (Non decimal digit)
\t	znak tabulátora
\r	Carriage Return je špeciálnym riadiacim znakom slúžiacim na návrat kurzora na začiatok riadku
\v	Vertical Tab
\f	Form Feed
\n	Nový riadok (New Line)
\e	Escape

## Regular expression boundary

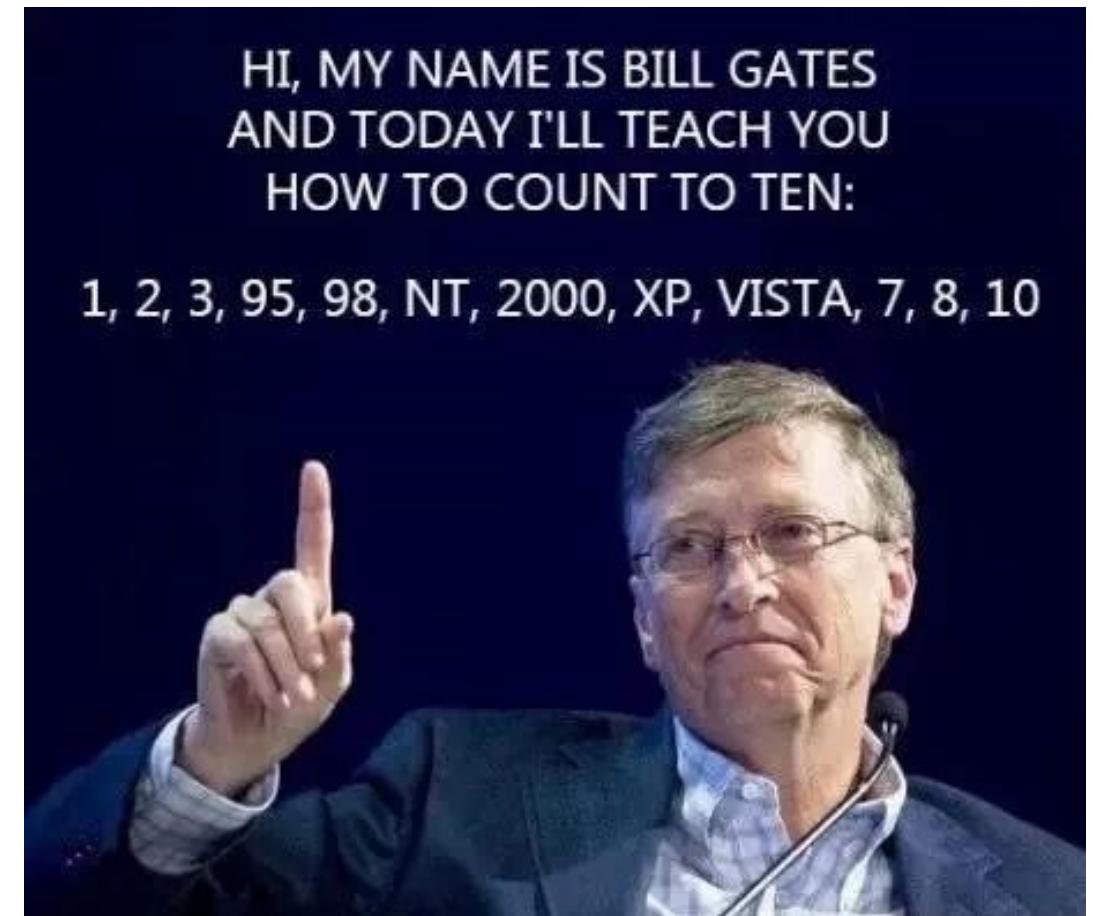
The diagram illustrates a regular expression boundary with various annotations:

- match anything contained within brackets**: Points to the first set of brackets `(\w._%+-)+`.
- as many times as possible**: Points to the quantifier `+`` after the first set of brackets.
- match the @ symbol**: Points to the character `@`.
- match upper and lower case A through Z**: Points to the character class `[a-zA-Z]`.
- match at least two times but no more than four times**: Points to the quantifier `{2,4}`.
- match the . symbol**: Points to the character `.`.
- match any .,\_%+-**: Points to the character class `(\w._%+-)+`.
- match any character A-Z upper or lower case and any number 0 to 9**: Points to the character class `\w`.

The regular expression itself is: `/[\w._%+-]+@[\\w.-]+\.\[a-zA-Z]{2,4}/`

# Hľadáme slovo Bill

- Regulárny výraz
- Bii?ll?.
- Tento výraz nájde Bill, Biil, Bil a Biill



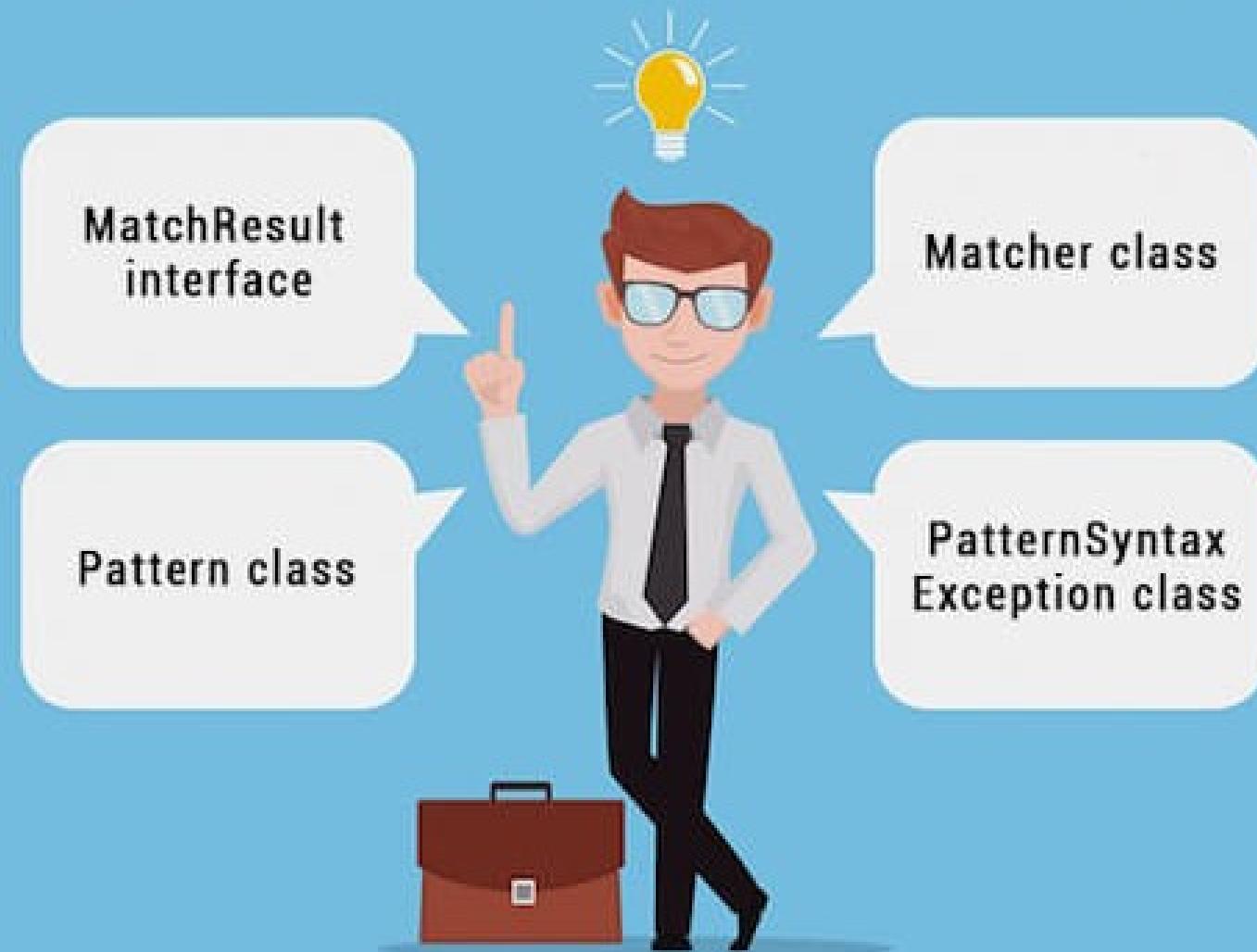
# Príklady regulárnych výrazov

- **A(d|l)am** reťazec "Adam" a "Alam"
- **Ba\*f** reťazce "Bf", "Baf", "BAAF", "Baaaf"
- **\d{3} \d{2}** Postupnosť 3 číslic, medzera a 2 číslice PSČ
- **<[^>]\*>** Tag v jazyku HTML - ľubovoľný text uzavretý medzi špicaté zátvorky (poznámka: tento výraz je zjednodušený, v skutočnom HTML situácii komplikujú reťazca v úvodzovkách)
- **[0-9a-fA-F]+(, ?[0-9a-fA-F]+)\*** Zoznam hexadecimálnych čísel, oddelených čiarkami a nepovinnými medzerami

# Java Regex

## java.util.regex package

The **Matcher** and **Pattern** classes provide the facility of Java regular expression. The **java.util.regex** package provides following classes and interfaces for regular expressions.



# Triedy pre regulárne výrazy

## Patern

- Umožňuje vytvoriť regulárny výraz, s ktorým je potom ďalej možné pracovať
- Vytvoríme si novú inštanciu paternu a odovzdáme mu samotný regulárny výraz

## Matcher

- Overuje či nejaký textový reťazec splňa kritériá daná regulárny výrazom

```
// Regulárny výraz pre overenie emailovej adresy
Pattern p = Pattern.compile
("[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.\[a-zA-Z]{2,4}");
Matcher m = p.matcher("Text na overenie");
```

# Jednoduché overenie

```
if (m.find()) {  
    System.out.println("Výraz splňuje pravidlá.");  
}  
else {  
    System.out.println("Výraz nesplňuje pravidlá.");  
}
```

# Používanie regulárnych výrazov

```
1. import java.util.regex.Matcher;
2. import java.util.regex.Pattern;
3. public class Main {
4.     public static void main(String[] args) {
5.         Pattern pattern = Pattern.compile("STU FIIT",
Pattern.CASE_INSENSITIVE);
6.         Matcher matcher = pattern.matcher("Pod' na STU FIIT!");
7.         boolean matchFound = matcher.find();
8.         if(matchFound) {
9.             System.out.println("Výraz splňuje pravidlá.");
10.        } else {
11.            System.out.println("Výraz nesplňuje pravidlá.");
12.        }
13.    }
14. }
```

```
import java.util.regex.*;

class RegTest {
    public static void main(String [] args) {
        Pattern p = Pattern.compile(args[0]);
        Matcher m = p.matcher(args[1]);
        System.out.println("source: " + args[1]);
        System.out.println(" index: 01234567890123456\n");
        System.out.println("expression: " + m.pattern());
        System.out.print("match positions: ");

        while(m.find()) {
            System.out.print(m.start() + " ");
        }

        System.out.println("");
    }
}
```

- Vyvolanie z CMD:  
java RegTest "ab"  
"abaaaaba"
- Vyprodukuje  
zdroj: abaaaaba  
source: abaaaaba  
index:  
01234567890123456  
expression: ab  
match positions: 0 4

PREV PACKAGE NEXT PACKAGE

FRAMES NO FRAMES

ALL CLASSES

## Package java.util.regex

Classes for matching character sequences against patterns specified by regular expressions.

See: [Description](#)

### Interface Summary

Interface	Description
<a href="#">MatchResult</a>	The result of a match operation.

### Class Summary

Class	Description
<a href="#">Matcher</a>	An engine that performs match operations on a <a href="#">character sequence</a> by interpreting a <a href="#">Pattern</a> .
<a href="#">Pattern</a>	A compiled representation of a regular expression.

### Exception Summary

Exception	Description
<a href="#">PatternSyntaxException</a>	Unchecked exception thrown to indicate a syntax error in a regular-expression pattern.

## Package java.util.regex Description

Classes for matching character sequences against patterns specified by regular expressions.

An instance of the [Pattern](#) class represents a regular expression that is specified in string form in a syntax similar to that used by Perl.

Instances of the [Matcher](#) class are used to match character sequences against a given pattern. Input is provided to matchers via the [CharSequence](#) interface in order to support matching against characters from a wide variety of input sources.

compact1, compact2, compact3

java.util.regex

## Class Pattern

java.lang.Object  
java.util.regex.Pattern

All Implemented Interfaces:

Serializable

---

```
public final class Pattern
extends Object
implements Serializable
```

A compiled representation of a regular expression.

A regular expression, specified as a string, must first be compiled into an instance of this class. The resulting pattern can then be used to create a **Matcher** object that can match arbitrary character sequences against the regular expression. All of the state involved in performing a match resides in the matcher, so many matchers can share the same pattern.

A typical invocation sequence is thus

```
Pattern p = Pattern.compile("a*b");
Matcher m = p.matcher("aaaaab");
boolean b = m.matches();
```

A **matches** method is defined by this class as a convenience for when a regular expression is used just once. This method compiles an expression and matches an input sequence against it in a single invocation. The statement

```
boolean b = Pattern.matches("a*b", "aaaaab");
```

is equivalent to the three statements above, though for repeated matches it is less efficient since it does not allow the compiled pattern to be reused.

Instances of this class are immutable and are safe for use by multiple concurrent threads. Instances of the Matcher class are not safe for such use.

# Flags (Príznaky)

- Príznaky v metóde **compile()** menia spôsob vykonávania vyhľadávania:
- **Pattern.CASE\_INSENSITIVE** – Veľké písmená budú pri vyhľadávaní **ignorované**
- **Pattern.LITERAL** - **Špeciálne znaky** vo vzore **nebudú mať žiadny zvláštny význam** a budú sa pri vyhľadávaní považovať za bežné znaky
- **Pattern.UNICODE\_CASE** - Použite ho spolu s príznakom CASE\_INSENSITIVE na **ignorovanie malých a veľkých písmen mimo anglickej abecedy**

# RegexBuddy



RegexBuddy

Java 8   Case sensitive   Exact spacing   Dot doesn't match line breaks   ^\$ don't match at line breaks   Default line breaks   Reset

"[^\\n]\*(?:\\n[^\\n]\*)\*" Match Replace Split Copy Paste

History

Programming: String  
Programming: String (escape quotes)  
Programming: String (multiline; escape quotes)

Create Convert Debug Use Library Test GREP Forum

Detailed Explain Token Insert Token Compare Java 13–14 Export Print RegexMagic Whole file LF only

Java 8 & Java 13–14

All selected applications handle your regular expression in the same way

A Match the character "" literally

Match any single character NOT present in the list below

Between zero and unlimited times, as many times as possible, giving backslash character

The literal character ""

The backslash character

Match the regular expression below

Between zero and unlimited times, as many times as possible, giving backslash character

Match the backslash character

Match any single character that is NOT a line break character (line feed)

Match any single character NOT present in the list below

Between zero and unlimited times, as many times as possible, giving backslash character

The literal character ""

The backslash character

A Match the character "" literally

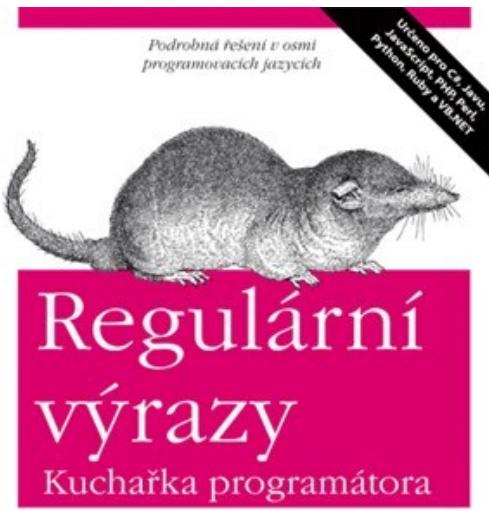
string  
two" "strings"  
multi  
line  
string  
"string with \"quotes\""  
multi  
line  
string  
with  
\"quotes\""

Start Length

Match 1 of 6: "string" 0 8

# RegexBuddy

- Jan Goyvaerts
- Steven Levithan



The screenshot shows the RegexBuddy application window. The title bar says 'RegexBuddy'. The menu bar includes 'Java 8', 'Helpful', 'Match', 'Replace', 'Split', 'Copy', 'Paste', and various tool icons. The toolbar below has buttons for 'Create', 'Convert', 'Test', 'Debug', 'Use', 'Library', 'GREP', and 'Forum'. The main area has tabs for 'Java 8; Match; Case sensitive; Exact spacing; Dot doesn't match line breaks; ^\$ don't match at line breaks; Default line breaks' and 'Reset'. The search bar contains the regex pattern: "[^\r\n]\*". The status bar at the bottom says 'Quotes may appear in the string when escaped with a backslash. The string may span multiple lines.' The bottom pane shows the results of the regex match applied to the book cover image, with parts of the text highlighted in blue and yellow.

# Regular Expression Test Page

for Java

[JavaDoc](#)Share: 

Expression to test

Regular expression:

Options:

- Force canonical equivalence (CANON\_EQ)
- Case insensitive (CASE\_INSENSITIVE)
- Allow comments in regex (COMMENTS)
- Dot matches line terminator (DOTALL)
- Treat as a sequence of literal characters (LITERAL)
- ^ and \$ match EOL (MULTILINE)
- Unicode case matching (UNICODE\_CASE)
- Only consider '\n' as line terminator (UNIX\_LINES)

Replacement:

[Test](#)[More Inputs](#)

Input 1:



**Formatters**[JSON Formatter](#)  
[HTML Formatter](#)  
[XML Formatter](#)  
[SQL Formatter](#)**Validators**[JSON Validator](#)  
[HTML Validator](#)  
[XML Validator - XSD](#)  
[XPath Tester](#)  
[Credit Card Number Generator & Validator](#)  
[Regular Expression Tester \(RegEx\)](#)  
**Java Regular Expression Tester (RegEx)**  
[Cron Expression Generator - Quartz](#)**Encoders & Decoders**[Url Encoder & Decoder](#)  
[Base 64 Encoder & Decoder](#)  
[Convert File Encoding](#)  
[QR Code Generator](#)**Code Minifiers / Beautifier**[JavaScript Beautifier](#)  
[CSS Beautifier](#)  
[JavaScript Minifier](#)  
[CSS Minifier](#)**Converters**[XSD Generator](#)  
[XSLT \(XSL Transformer\)](#)  
[XML to JSON Converter](#)  
[JSON to XML Converter](#)

## Java Regular Expression Tester

This **free Java regular expression tester** lets you test your regular expressions against any entry of your choice and clearly highlights all matches. It is based on the [Pattern class of Java 8.0](#).

Consult the [regular expression documentation](#) or the [regular expression solutions to common problems](#) section of this page for examples. If you need more examples or solutions, please [contact me](#).

**Java Regular Expression :****Entry to test against :****Replace with (Optional):**

You can make use of \$1, \$2, \$3 and so on if you are using parenthesis groups in your regular expression. \t \n \r are supported.

**Flags:**

- |                                     |  |
|-------------------------------------|--|
| <input type="checkbox"/> Dotall     | <input type="checkbox"/> Comments                |
| <input type="checkbox"/> Multiline  | <input type="checkbox"/> Literal                 |
| <input type="checkbox"/> Unix lines | <input type="checkbox"/> Unicode case            |
| <input type="checkbox"/> Canon EQ   | <input type="checkbox"/> Unicode character class |

**TEST MATCH****REPLACE FIRST****REPLACE ALL**

# Regulárne výrazy

## Borders

^ start of Exp  
\$ end of Exp

^Hello  
bye.\$

Hello....  
... bye.

## Quantifiers

?	0x or 1x	ab?c	“abc”, “ac”
*	0x to $\infty$	ab*c	“ac”, “abc”, “abbbc”
+	1x to $\infty$	ab+c	“abc”, “abbbc”
{n}	Nx	ab{3}c	“abbcc”
{n,m}	form Nx to Mx	ab{1,3}c	“abc”, “abbc”, “abbcc”
{n, }	from Nx to $\infty$	ab{3}c	“abbbcc”

# Regulárne výrazy

## Brackets

[0-9]	any character from 0 to 9
[a-z]	any character from a - z
[A-Z]	any character from A - Z
[abc]	any character in the given set
[^abc]	any character outside the given set
(red blue green)	any of the alternatives

## Metacharacters

.	single character
/	normal char from metachar
/w	word character [a-zA-Z0-9_]
/d	digit character [0-9]

# Validácia čísel

1. Kladné celé čísla nedefinovanej dĺžky:

➤  $^{\wedge} \backslash d+ \$$

2. Kladné celé čísla maximálnej dĺžky (v našom príklade 10)

➤  $^{\wedge} \backslash d\{1,10\} \$$

3. Pozitívne celé čísla pevnej dĺžky (v našom príklade 5):

•  $^{\wedge} \backslash d\{5\} \$$

4. Záporné celé čísla nedefinovanej dĺžky:

•  $^{\wedge} - \backslash d+ \$$

5. Záporné celé čísla maximálnej dĺžky (v našom príklade 10)

•  $^{\wedge} - \backslash d\{1,10\} \$$

# Validácia čísel 2

1. Záporné celé čísla pevnej dĺžky (v našom príklade 5)
  - $^-\backslash\text{d}\{5\}\$$
2. Celé čísla nedefinovanej dĺžky:
  - $^-\text{?}\backslash\text{d}+\$$
3. Celé čísla maximálnej dĺžky (v našom príklade 10)
  - $^-\text{?}\backslash\text{d}\{1,10\}\$$
4. Celé čísla pevnej dĺžky (v našom príklade 5)
  - $^-\text{?}\backslash\text{d}\{5\}\$$
5. Počet nedefinovaných dĺžok s desatinnými miestami alebo bez nich (1234.1234)
  - $^-\text{?}\backslash\text{d}*\backslash\text{.}\{\text{0,1}\}\backslash\text{d}+\$$
6. Čísla s 2 desatinnými miestami (0,00)
  - $^-\text{?}\backslash\text{d}*\backslash\text{.}\backslash\text{d}\{2\}\$$

# Čísla mien s voliteľným znakom dolára a oddelovačmi tisícov a voliteľnými 2 miestami

- `^$?\\-?([1-9]{1}[0-9]{0,2}(\\,\\d{3})*(.\\d{0,2})?|[1-9]{1}\\d{0,}(\\.\\d{0,2})?|0(\\.\\d{0,2})?|(\\.\\d{1,2}))$|^\\-?\\$?([1-9]{1}\\d{0,2}(\\,\\d{3})*(.\\d{0,2})?|[1-9]{1}\\d{0,}(\\.\\d{0,2})?|0(\\.\\d{0,2})?|(\\.\\d{1,2}))$|^\\($?([1-9]{1}\\d{0,2}(\\,\\d{3})*(.\\d{0,2})?|[1-9]{1}\\d{0,}(\\.\\d{0,2})?|0(\\.\\d{0,2})?|(\\.\\d{1,2})))\\)$`

# Percento od 0 do 100 s voliteľnými 2 miestami a znamienkom % na konci

- ^-?[0-9]{0,2}(\.[0-9]{1,2})?%?\$ | ^-?(100)(\.[0]{1,2})?%?\$

# Hexadecimálny farebný kód (#FFFFFF)

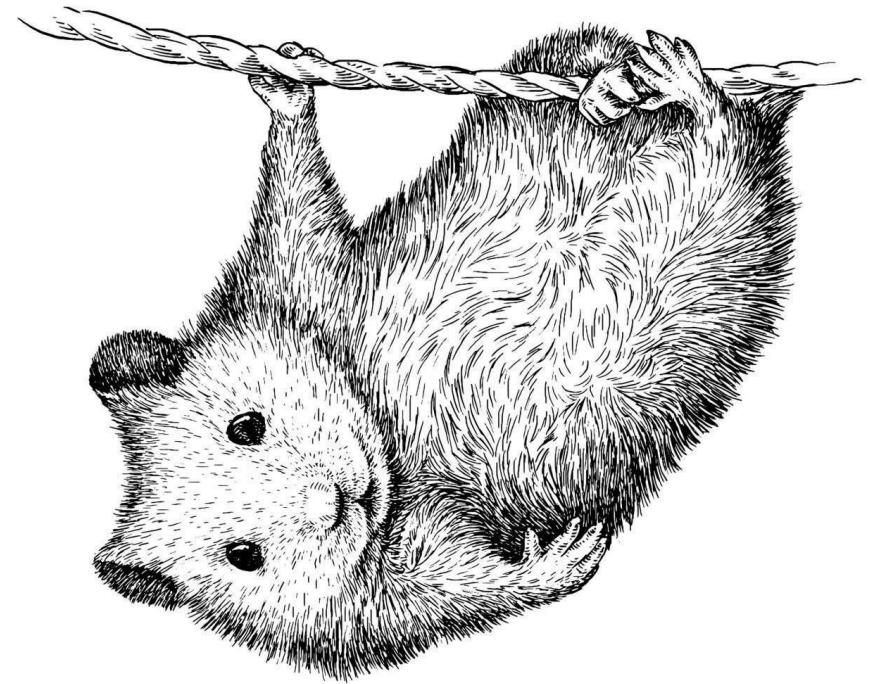
- ^#?([a-f0-9]{6}|[a-f0-9]{3})\$

# Alfanumerické hodnoty

- ^[ a-zA-Z0-9 ]+\$

# Emailová adresa

- ^[-a-zA-Z0-9~!\$%^&\*\_+=]{\'.}[ -a-zA-Z0-9~!\$%^&\*\_+=]{\'.}+@[ -a-zA-Z0-9\_-][-a-zA-Z0-9\_-]\*(\.[-a-zA-Z0-9\_-]+)\*\.(aero|arpa|biz|com|coop|edu|gov|info|int|mil|museum|name|net|org|pro|travel|mobi|[a-zA-Z][a-zA-Z])|([0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}))(:[0-9]{1,5})?\$\_



Googling for  
the Regex

*Every. Damn. Time.*

# Regulárny výraz email

```
public String validateEmail(String email) {  
    var re = "^(\\^<>()\\[\\]\\.,;:\\s@\\"]+\\(.\\^<>()\\[\\]\\.,;:\\s@\\"]+\\))*\\(\\\".+\\\")\\)@\\((\\[[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\])|(([a-zA-Z\\-0-9]+\\.)+[a-zA-Z]{2,}))$";  
    return re.test(email);  
}
```

# IP adresa (IPV4)

- `^(?:(?:25[0-5]|2[0-4][0-9]| [01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]| [01]?[0-9][0-9]?)$`

# Formát dátumu ISO (rrrr-mm-dd)

- ^[0-9]{4}-(((0[13578] | (10|12))- (0[1-9] | [1-2][0-9] | 3[0-1])) | (02- (0[1-9] | [1-2][0-9])) | ((0[469] | 11)- (0[1-9] | [1-2][0-9] | 30)))\$

# Formát dátumu ISO (rrrr-mm-dd) s oddelovačmi - / . " "

- `^[0-9]{4}([- /.])(((0[13578] | (10|12))\1(0[1-9] | [1-2][0-9] | 3[0-1])) | (02\1(0[1-9] | [1-2][0-9])) | ((0[469] | 11)\1(0[1-9] | [1-2][0-9] | 30)))$`

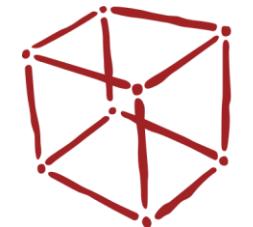
# Formát dátumu v USA (mm / dd / rrrr)

- ^(((0[13578] | (10|12)) / (0[1-9] | [1-2][0-9] | 3[0-1])) | (02 / (0[1-9] | [1-2][0-9]))) | ((0[469] | 11) / (0[1-9] | [1-2][0-9] | 30))) / [0-9]{4}\$

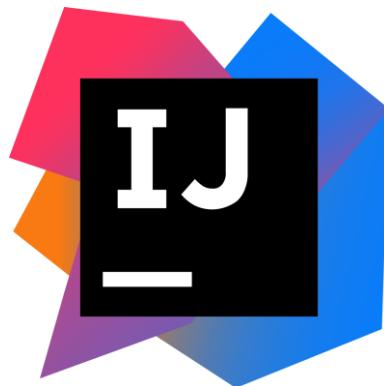
# Hodiny a minúty, 24-hodinový formát (HH:MM)

- ^ (20|21|22|23|[01]\d|\d) ((:[0-5]\d){1,2})\$

# Aké IDE mám použiť?



**NetBeans**

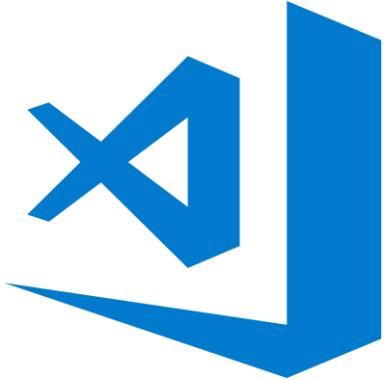
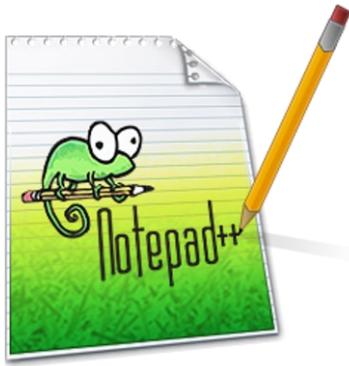


**Visual Studio**

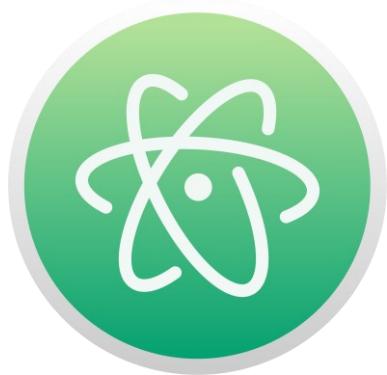


Integrated development environment

# Aký editor mám použiť?



```
:::  
iLE880j. :jD888880j:  
.LGitE888D.f8GjjjL8888E;  
iE :8888Et. .G888.  
;i E888, ,8888,  
D888, :8888:  
D888, :8888:  
D888, :8888:  
D888, :8888:  
888W, :8888:  
W88W, :8888:  
W88W: :8888:  
DGGD: :8888:  
:8888:  
:W888:  
:8888:  
E888i  
tW88D
```



# Updaty a aktualizácie



jdk1.6.0\_21



jdk1.6.0\_29



jre6



jdk1.6.0\_26



jdk1.7.0\_21



jre7

**JAVA DEVELOPERS NEVER RIP,  
THEY JUST GET GARBAGE COLLECTED.**

- I LIKE NITTY-WITTY.COM

# Čo sa oplatí prečítať?

## Slovensko a česko

- Albatrosmedia
- Kopp
- Grada
- Wolters Kluwer
- BEN
- Veda

## Zahraničie

- O'Reilly
- Manning
- Packt
- Apress
- Wiley
- No Starch Press

## YouTube tutoriály

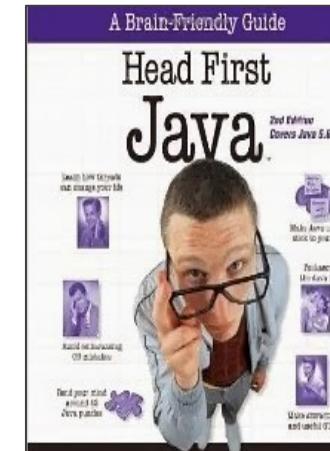
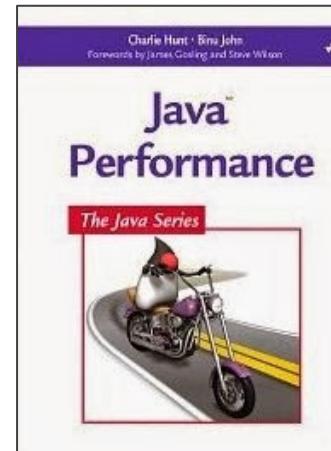
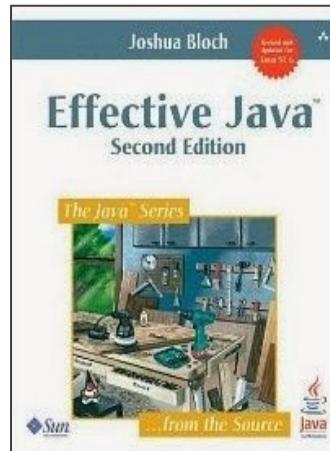
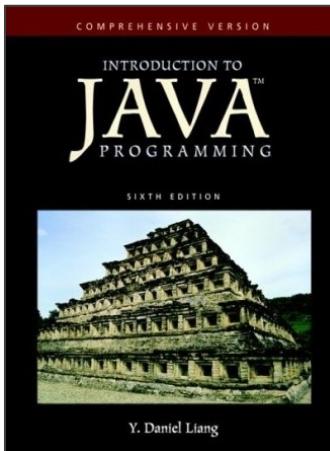
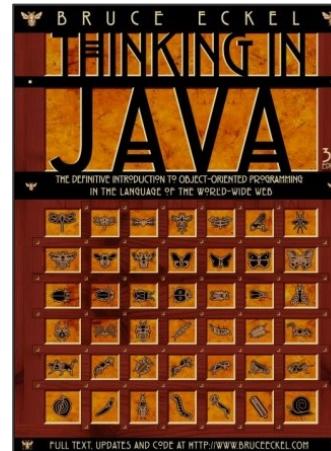
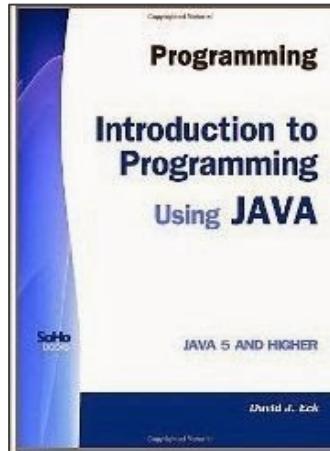
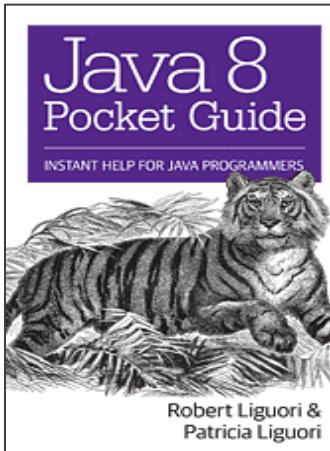
O'Reilly

# Čo sa oplatí/neoplatí prečítať SK/CZ



Mistrovství a Kuchárka

# Čo sa oplatí/neoplatí prečítať EN



Head First

## Vývojári

Verejná skupina

Informácie

## Diskusia

Oznámenia

Členovia

Podujatia

Videá

Fotky

Súbory

Hľadať v tejto skupine



Ste člen ▾

Upozornenia

Zdieľať

... Viac



Napísat' príspěvok...

Pridať fotku/vi...

Živé video

Viac



Napište niečo...



Fotka/video



Divácka páry



Označiť priat...

...

NOVÁ AKTIVITA ▾



Roland Mondek

10 h

## POZVAŤ ČLENOV

+ Zadajte meno alebo e-mailovú adresu...



## ČLENOVIA

5 505 členov



## POPIS

Skupina softvérových vývojárov. Táto skupina by mala byť miestom... [Zobraziť viac](#)

## TYP SKUPINY

Všeobecné

## VAŠE STRÁNKY



IT Academy



VITA - Virtual It Academy

## KONTAKTY



Evka Rybárska



Jarmila Palenčárová



Stefan Orosi



Ivana Ivka Jasaňová



Hrá Word Blitz



Ivana Pavlíková



Martin Vanko



Lucia Kovačičová

4 h



Lošák Filip



Andrej Nejedlik



Gabika Zubrikova

## SKUPINOVÉ KONVERZÁCIE



Vytvoriť novú skupinu

Hľadať



...



...

Home

PUBLIC

Questions

**Tags**

Users

COLLECTIVES

Explore Collectives

FIND A JOB

Jobs

Companies

TEAMS

Create free Team

# Tags

A tag is a keyword or label that categorizes your question with other, similar questions. Using the right tags makes it easier for others to find and answer your question.

[Show all tag synonyms](#)

java

java

Java is a high-level object oriented programming language. Use this tag when you're having problems using or understanding the language itself. Thi...

1827413 questions

419 asked today, 2408 this week

javascript

For questions regarding programming in ECMAScript (JavaScript/JS) and its various dialects/implementations (excluding ActionScript). Note...

2335556 questions

779 asked today, 4877 this week

javafx

The JavaFX platform enables developers to create and deploy Graphical User Interface (GUI) applications that behave consistently...

36355 questions

6 asked today, 50 this week

java-8

for questions specific to Java 8 which is version 8 (internal number 1.8) of the Java platform, released on 18 March 2014. In most cases, you should also...

22076 questions

9 asked today, 40 this week

java-stream

for questions related to the use of the Stream API. It was introduced in Java 8 and supports functional-style operations on streams of values, such...

10293 questions

5 asked today, 26 this week

java-native-interface

The Java Native Interface (JNI) gives both the ability for JVM implementations to run system native code and the ability for native code t...

9404 questions

12 asked this week, 38 this month

rx-java

RxJava – Reactive Extensions for the JVM – a library for composing asynchronous and event-based programs using observable sequence...

6796 questions

6 asked this week, 27 this month

javascript-objects

for questions related to JavaScript objects.

6151 questions

20 asked this week, 118 this month

java.util.scanner

A simple text scanner in the JDK which can parse primitive types and strings using regular expressions.

javafx-8

JavaFX 8 (previously named JavaFX 3) introduces a new API for JavaFX technology. JavaFX 8 supports 3D and brings up a Retina-Display Support. It ...

java-me

Java Platform, Micro Edition, or Java ME, is a Java platform designed for embedded systems.

facebook-javascript-sdk

Facebook's JavaScript SDK provides a rich set of client-side functionality for accessing Facebook's server-side API calls. It can collaborate with any SDK...

Mrkni na náš YouTube kanál a daj odber

 [WWW.YOUTUBE.COM/C/IT-ACADEMYSK](https://www.youtube.com/c/IT-ACADEMYSK) 