

Vývoj Aplikácií s Viacvrstvovou Architektúrou

8. Regulárne Výrazy (Regexy)



Čo nás čaká a neminie...

1. časť

Úvod do Javy

Štruktúra platformy

Vývojové technológie

Kolekcie

Logovanie

Lokalizácia

2. časť

NIO.2, IO, XML

Regulárne výrazy

Modularita

JDBC

Bezpečnosť

Prehľad JEE a .NET

| | | Java Language | | | | | | | | | | | |
|----------------------|-------------------------|-----------------------------------|---------------|---------------|-----------------------|-----------------|----------------------|-----------------------|--|--|--|--|--|
| | | java | javac | javadoc | jar | javap | jdeps | Scripting | | | | | |
| JDK | Tools & Tool APIs | Security | Monitoring | JConsole | VisualVM | JMC | JFR | | | | | | |
| | | JPDA | JVM TI | IDL | RMI | Java DB | Deployment | | | | | | |
| | | Internationalization | | Web Services | | Troubleshooting | | | | | | | |
| JRE | Deployment | Java Web Start | | | Applet / Java Plug-in | | | | | | | | |
| | | JavaFX | | | | | | | | | | | |
| | | Swing | | Java 2D | | AWT | Accessibility | | | | | | |
| JRE | User Interface Toolkits | Drag and Drop | | Input Methods | | Image I/O | Print Service | Sound | | | | | |
| | | IDL | JDBC | JNDI | RMI | RMI-IIOP | | Scripting | | | | | |
| | | Beans | Security | | Serialization | | Extension Mechanism | | | | | | |
| JRE | Integration Libraries | JMX | XML JAXP | | Networking | | Override Mechanism | | | | | | |
| | | JNI | Date and Time | | Input/Output | | Internationalization | | | | | | |
| | | lang and util | | | | | | | | | | | |
| JRE | Base Libraries | Math | | Collections | | Ref Objects | | Regular Expressions | | | | | |
| | | Logging | | Management | | Instrumentation | | Concurrency Utilities | | | | | |
| | | Reflection | Versioning | | Preferences API | | JAR | Zip | | | | | |
| Java Virtual Machine | | Java HotSpot Client and Server VM | | | | | | | | | | | |

Java SE

API

Compact Profiles

Používanie regulárnych výrazov

Untitled Pattern Save (ctrl-s) New

by gskinner GitHub Sign In

Menu

Pattern Settings

My Patterns

Cheatsheet

RegEx Reference

Community Patterns

Help

Expression

/([A-Z])\w+/g

Text Tests NEW

29 matches (0.9ms)

RegExr was created by gskinner.com, and is proudly hosted by Media Temple.

Edit the Expression & Text to see matches. Roll over matches or the expression for details. PCRE & JavaScript flavors of RegEx are supported. Validate your expression with Tests mode.

The sidebar includes a Cheatsheet, full Reference, and Help. You can also Save & Share with the Community, and view patterns you create or favorite in My Patterns.

Explore results with the Tools below. Replace & List output custom results. Details lists capture groups. Explain describes your expression in plain English.

RegExr is an online tool to **learn, build, & test** Regular Expressions (RegEx / RegExp).

- Supports **JavaScript & PHP/PCRE** RegEx.
- Results update in **real-time** as you type.
- **Roll over** a match or expression for details.
- Validate patterns with suites of **Tests**.
- **Save** & share expressions with others.
- Use **Tools** to explore your results.
- Full **RegEx Reference** with help & examples.
- **Undo & Redo** with ctrl-Z / Y in editors.
- Search for & rate **Community Patterns**.

Tools

Replace List Details Explain

Roll-over elements below to highlight in the Expression above. Click to open in Reference.

Capturing group #1. Groups multiple tokens together and creates a capture group for extracting a substring or using a backreference.

Character set. Match any character in the set.

A-Z Range. Matches a character in the range "A" to "Z" (char code 65 to 90). Case sensitive.

)

\w Word. Matches any word character (alphanumeric & underscore).

+ **Quantifier.** Match 1 or more of the preceding token.

Want to support RegExr? Consider disabling your ad-blocker for this domain. We'll show a non-intrusive, dev-oriented ad in this area.



SAVE & SHARE

[Save Regex](#) ctrl+s

FLAVOR



</> PCRE2 (PHP >=7.3)

</> PCRE (PHP <7.3)

</> ECMAScript (JavaScr...)

</> Python

</> Golang

</> Java 8



</> .NET (C#)

FUNCTION

>_ Match



☒ Substitution

☒ List

☒ Unit Tests

TOOLS

[Code Examples](#)**DOPPLER**

All your environment variables, in one place

If you're running an ad blocker, consider whitelisting regex101 to support the website. [Read more.](#)

REGULAR EXPRESSION

no match

" insert your regular expression here

" gm



TEST STRING

insert your test string here

EXPLANATION

An explanation of your regex will be automatically generated as you type.

MATCH INFORMATION

Detailed match information will be displayed here automatically.

QUICK REFERENCE

Search reference

☒ All Tokens

★ Common Tokens



○ General Tokens

☒ Anchors

☒ Meta Sequences

A single character of: a, b or c

[abc]

A single character of: a, b, c o... [[ab][cd]]

A character except: a, b or c

[^abc]

A character in the range: a-z

[a-z]

A character not in the range: a-z

[^a-z]

A character in the range: a-z or ... [a-zA-Z]

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
- CSV to XML Converter
- CSV to JSON Converter
- Epoch Timestamp To Date

Cryptography & Security

Regular Expression Tester

This free regular expression tester lets you test your regular expressions against any entry of your choice and clearly highlights all matches. It is JavaScript based and uses [XRegExp library](#) for enhanced features.

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](#).

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:

- i - Case-insensitive
- m - Multiline
- g - Global (don't stop at first match)
- s - Dot matches all INCLUDING line breaks (XRegExp only).

TEST MATCH**REPLACE**

Regular Expression - Documentation

Metacharacters

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](#)
[CSV to XML Converter](#)
[CSV to JSON Converter](#)
[Epoch Timestamp To Date](#)**Cryptography & Security**

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**

Regular Expression - Documentation

Metacharacters

Related Tools

Text Editor Online

Regex Tester

Regex Replace

Word Counter

Character Count

Case Converter

Reverse Text

Number to Words

Other Tools By Category



Image Utilities



Formatter & Minifier



Internet Testing



Hash & Cryptography



Number Conversion



Text & String Utilities

Regex Tester Online Tool

Regular Expression

([A-Z])\w+

Flags(Part 1)

g ON(global)



i OFF(ignore case)



m OFF(multi-line)



Flags(Part 2)

s OFF(dotAll)



u OFF(Unicode)



y OFF(sticky)



- 1 This online **Regex Tester** tool helps you to test if your regular expression is working correctly. It support Matching h2-highlight and 6 different Flags, powered by **Javascript RegExp**.

About Regex Tester Online Tool:

This online Regex Tester tool helps you to test if your regular expression is working correctly. It support Matching h2-highlight and 6 different Flags, powered by Javascript RegExp.

| Flags Type | Description |
|------------|---|
| g | (global match), find all matches |
| i | ignore case |
| m | multiline; treat beginning and end characters (^ and \$) as working over multiple lines |
| s | allows . to match newlines |
| u | Unicode; treat pattern as a sequence of Unicode |
| y | sticky; matches only from the index indicated by the lastIndex |



Regex Crossword

Welcome to the fantastic world of nerdy regex fun! Start playing by selecting one of the puzzle challenges below. There are a wide range of difficulties from beginner to expert.

[How to play »](#)

Mobile (NEW!)

Try our new mobile version! Optimized for phones and solving puzzles on the go.

[Play »](#)

Tutorial

A step by step tutorial, teaching you the different symbols and regex patterns.

[Play »](#)

Intermediate

So you've got skills eh? Let's see how you handle a tougher challenge...

[Play »](#)

Double Cross

Don't get cross-eyed, the clues are coming from all

Beginner

Cut your teeth on an easy set of crosswords, learning the basics of regular expressions.

[Play »](#)

Experienced

Now it's getting difficult. We are ramping up the size and complexity. Try to keep up!

[Play »](#)

Cities

Rome wasn't built in a day you know, but see if you can

Palindromeda

Bend your mind around these cubistic 2D palindrome puzzles.

[Play »](#)

Volapük

This is not a crossword, this is an exercise in pure regex



[Prev Class](#) [Next Class](#) [Frames](#) [No Frames](#) [All Classes](#)

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

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.

Summary of regular-expression constructs

[Prev Class](#) [Next Class](#) [Frames](#) [No Frames](#) [All Classes](#)

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

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.

Summary of regular-expression constructs

[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#) [ALL CLASSES](#)

SUMMARY: NESTED | FIELD | CONSTR | METHOD

DETAIL: FIELD | CONSTR | METHOD

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



IntelliJ IDEA

2021.3.2

Projects

Remote Development Beta

Customize

Plugins

Learn IntelliJ IDEA

Marketplace Installed

Search Results (31)

Sort By: Relevance

Regex

| Plugin | Downloads | Rating | Author | Install |
|--------------------------------|-----------|--------|------------------|-------------------------|
| RegexTester | ↓ 330,3K | ☆ 4.24 | Sergey Evdokimov | Install |
| Regex Rename Files | ↓ 3,5K | ☆ 4.85 | | Install |
| PHP RegExp Support | ↓ 40K | ☆ 2.59 | | Install |
| RegEx Tool <small>Paid</small> | ↓ 2,2K | ☆ 4.21 | | Install |
| Dukescript Java Regex Tester | ↓ 3K | ☆ 3.46 | | Install |
| Open Regex101 | ↓ 1,4K | | | Install |
| any-rule | ↓ 22,7K | ☆ 4.46 | | Install |
| Zoolytic - Zookeeper tool | ↓ 18K | ☆ 4.65 | | Install |

Marketplace Installed



RegexTester

↓ 330,3K ☆ 4.24 Sergey Evdokimov

Miscellaneous 1.0.8 jún 16, 2020

[Install](#)[Plugin homepage](#)

Regular Expression Tester for IntelliJ IDEA.

Allows you to experiment with Java regular expressions in a dynamic environment.

- Full highlighting of regular expression syntax for an easy visual clue (including bracket matching and error detecting)
- Tooltips with context help will be shown when the mouse is over a part of the regex.
- Synchronized selection of regular expression and text: Just select part of the regexp to see which part of the text is matched by this part.

Functionality of this plugin is available as java-applet at <http://myregexp.com/applet.html>

Size: 146,28 KB



Example of Regular Expressions

IP Address Regexp

```
\b(?:(?: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]?)\b
```

MAC Address Regexp

```
^([0-9a-fA-F][0-9a-fA-F]:){5}([0-9a-fA-F][0-9a-fA-F])$
```

Domain Name Regexp

```
^([a-zA-Z0-9]([a-zA-Z0-9\-\-]{0,61}[a-zA-Z0-9])?\.\+\[a-zA-Z]{2,6}$
```

Windows File Name Regexp

```
(?i)^(?!\^(PRN|AUX|CLOCK\$|NUL|CON|COM\d|LPT\d|\..*)\(\..\+)?\$)[^\\\./:\*\?\>\|][^\\\/\:\*\?\>\|]{0,254}\$
```

Float Number Regexp

```
[+-]?(?:\b[0-9]+(?:\.[0-9]*))?\.\.[0-9]+\b)(?:[eE][+-]?[0-9]+\b)?
```

Roman Number Regexp

```
^(?i:(?==[MDCLXVI])((M{0,3})((C[DM])|(D?C{0,3})))?((X[LC])|(L?XX{0,2})|L)?((I[VX])|(V?(II{0,2})))|V)?))$
```

Date in format yyyy-MM-dd

```
(19|20)\d\d([- /])(0[1-9]|1[012])\2(0[1-9]|12)[0-9]|3[01])
```

Čo sú regulárne výrazy?

Retázec popisujúci celú množinu retázcov,
konkrétny regulárny jazyk

```
s/^𓁃𓁃*𓁄-𓁅*(&𓁆??𓁃 )𓁂$/\${1}/mg
```

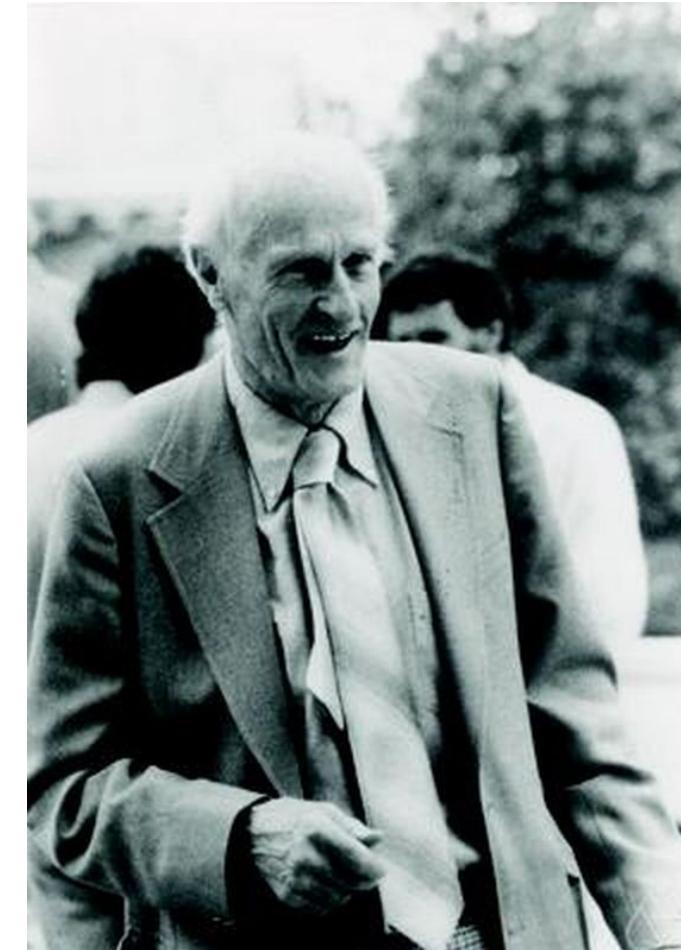
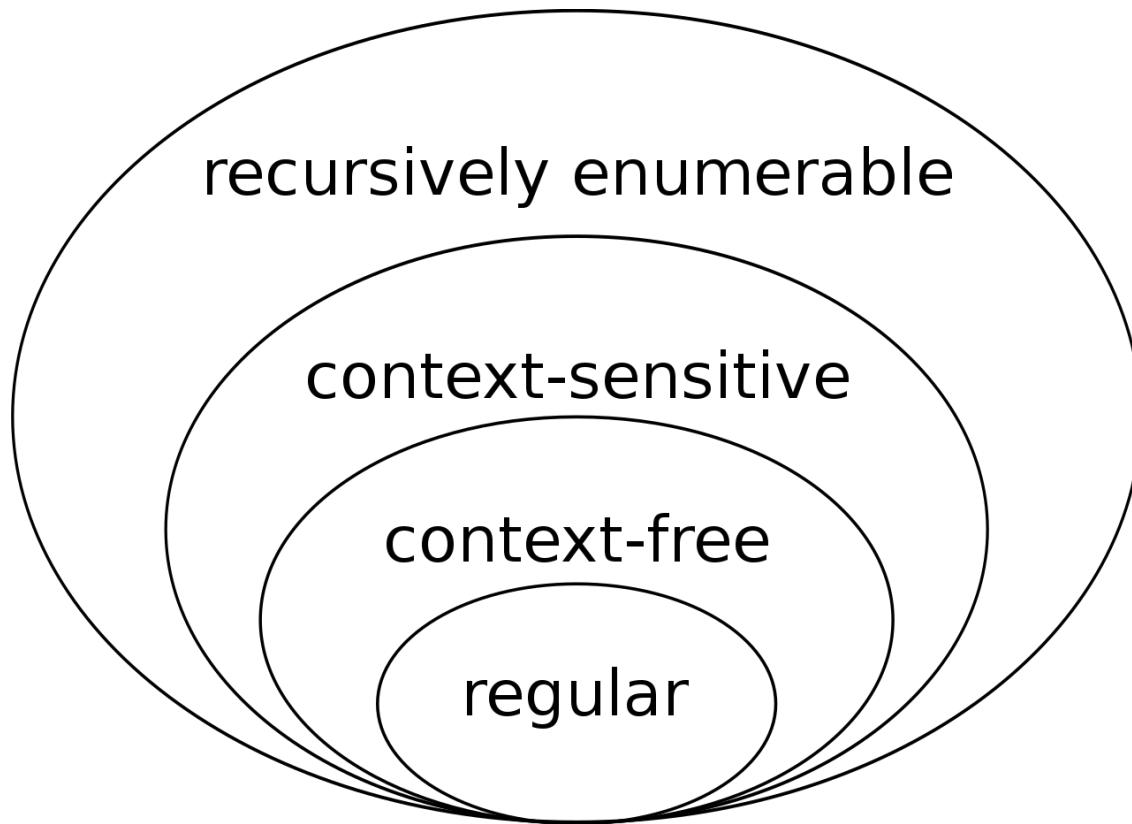


geek and poke

ANCIENT EGYPTIAN REGEXP

regexp | regex | re

Stephen Cole Kleene



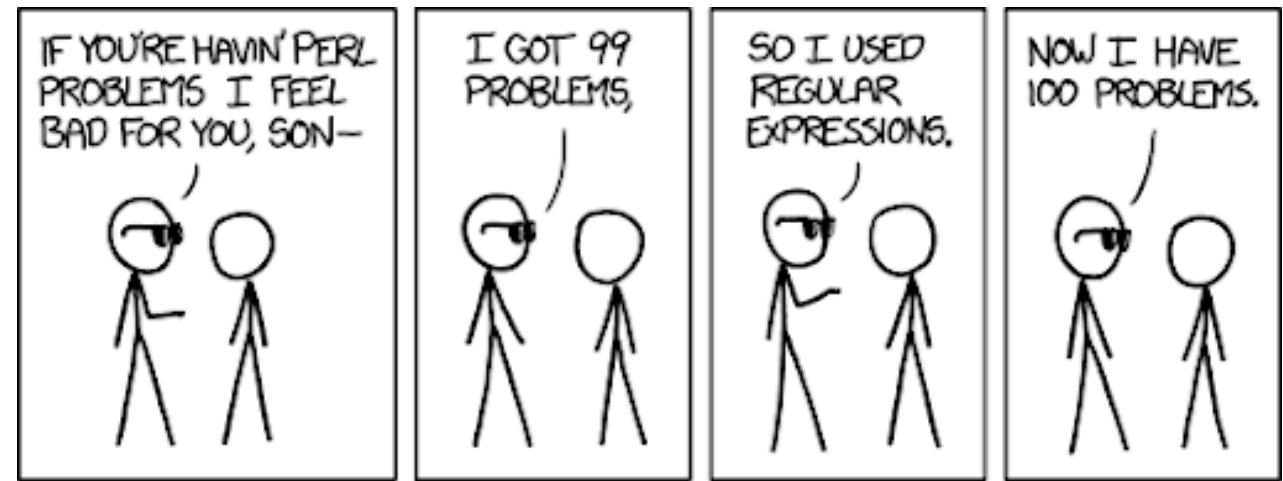
Načo je to dobré?

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 zadefinovaným 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 ASCII . medzera 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

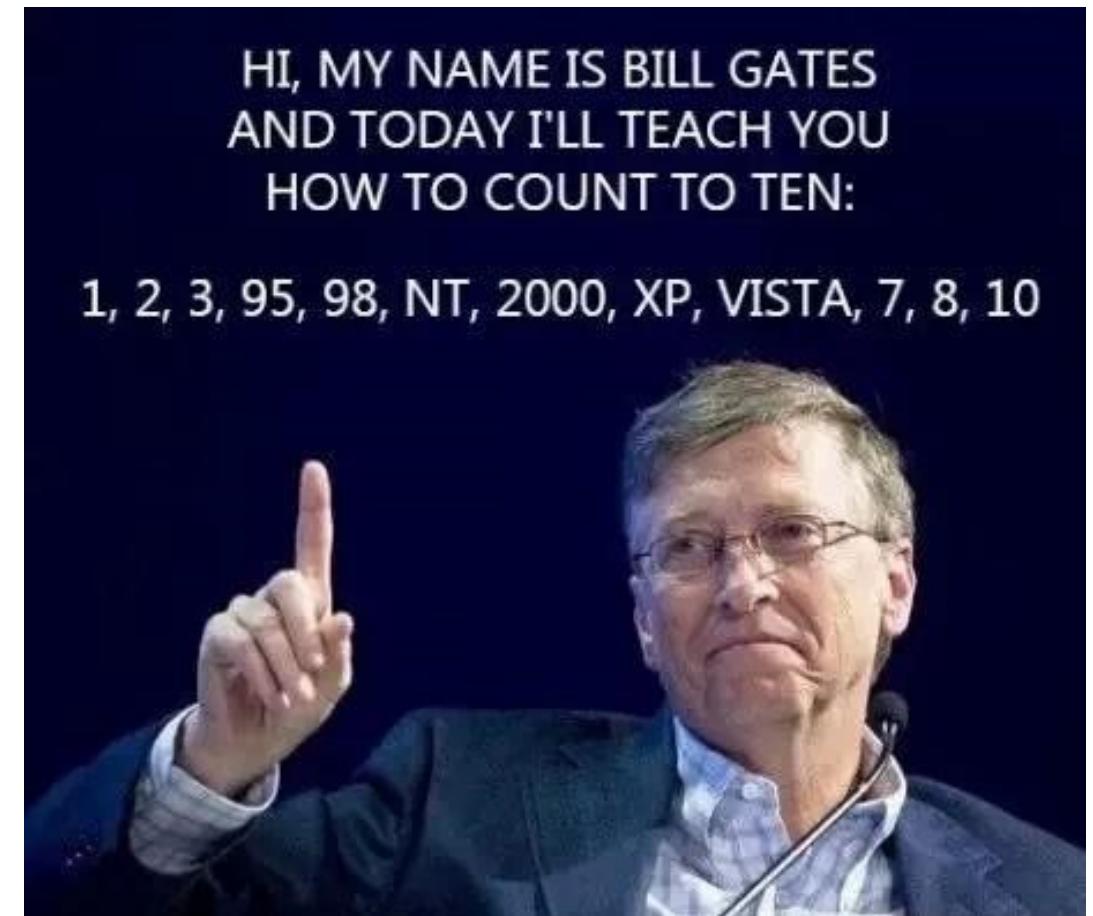
The diagram illustrates a regular expression boundary with various annotations:

- 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



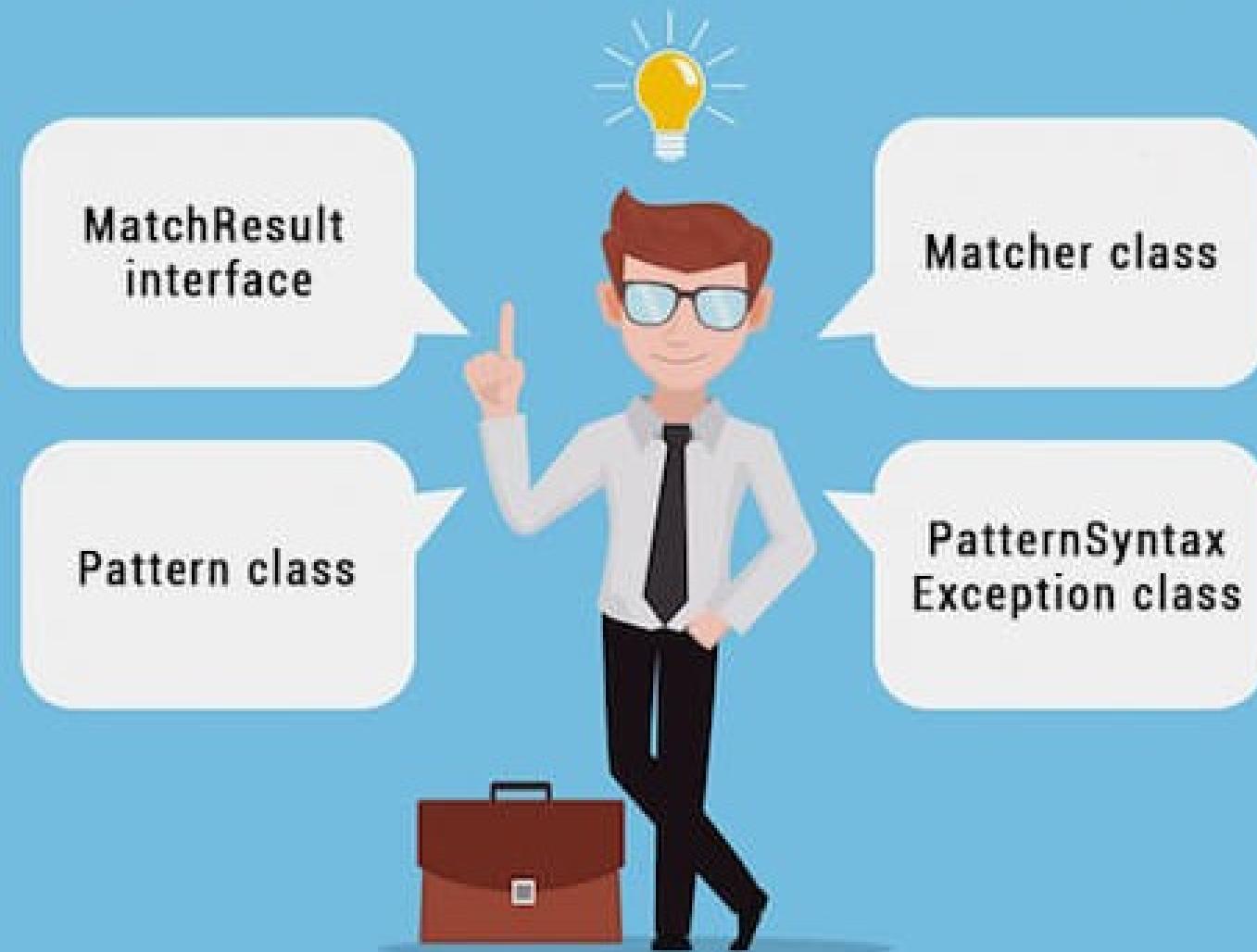
Príklady regulárnych výrazov

- **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 |
|-----------------------------|----------------------------------|
| MatchResult | The result of a match operation. |

Class Summary

| Class | Description |
|-------------------------|--|
| Matcher | An engine that performs match operations on a character sequence by interpreting a Pattern . |
| Pattern | A compiled representation of a regular expression. |

Exception Summary

| Exception | Description |
|--|--|
| PatternSyntaxException | 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

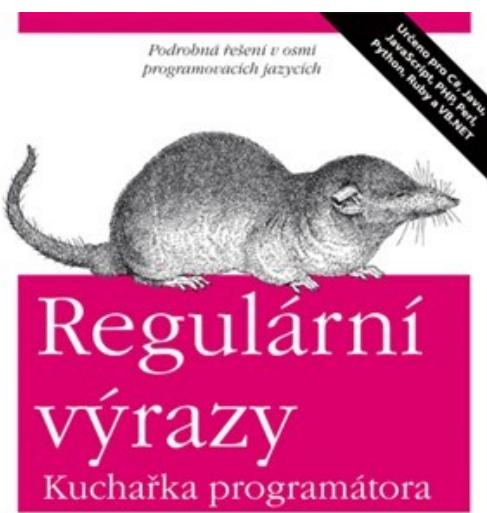
string
two" "strings"
multi
line
string
"string with \"quotes\""
multi
line
string
with
\"quotes\""

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 'RegExBuddy4.rbl' and 'Java 8; · Match; · Case sensitive; · Exact spacing; · Dot doesn't match line breaks; · \$ don't match at line breaks; · Default line breaks'. 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 a preview of the regex results on sample text: "string", "two"strings", "multi-line", "string", "string with \"quotes\"", "multi-line", "string", "with", "\"quotes\"".

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 ∞ | ab*c | “ac”, “abc”, “abbbc” |
| + | 1x to ∞ | 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 ∞ | 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)

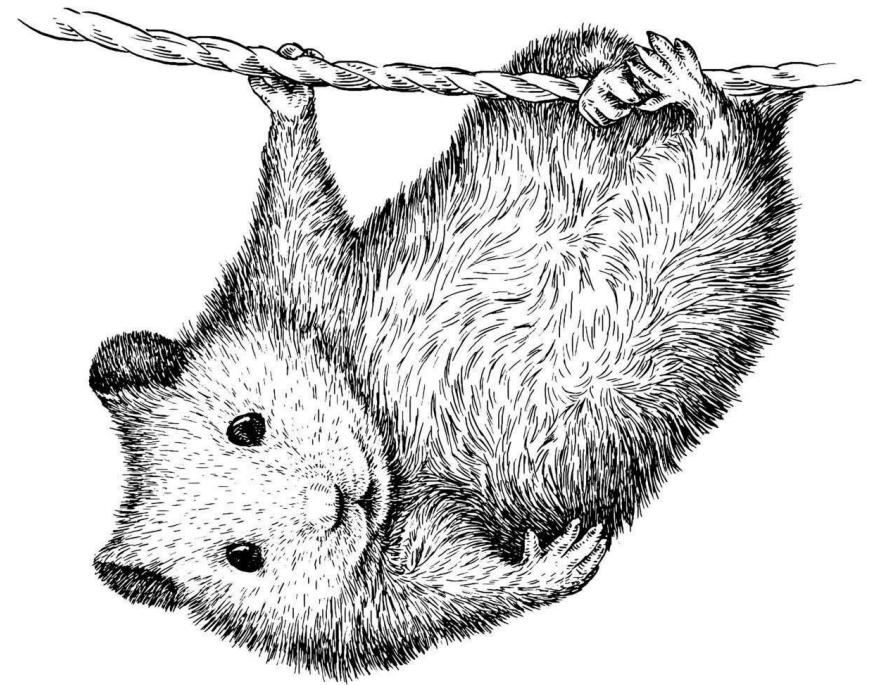
- ^#?([a-f0-9]{6}|[a-f0-9]{3})\$

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)))$`

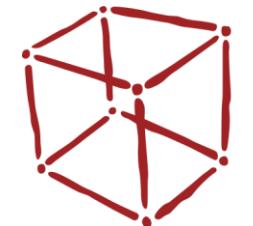
Formát dátumu v USA (mm / dd / rrrr)

- ^(((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}\$

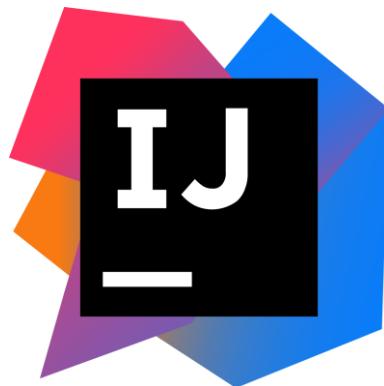
Hodiny a minúty, 24-hodinový formát (HH:MM)

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

Aké IDE mám použiť?



NetBeans

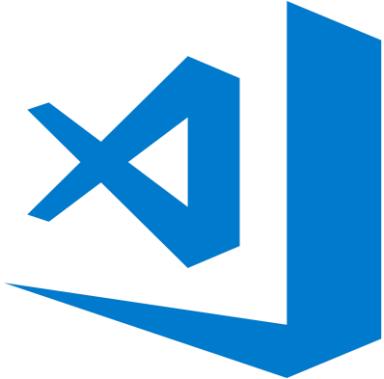
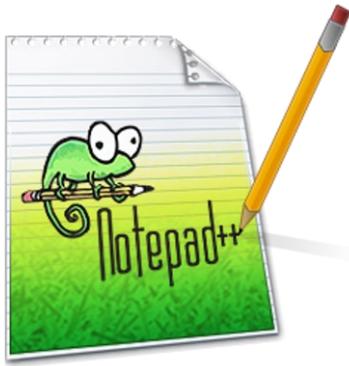


Visual Studio

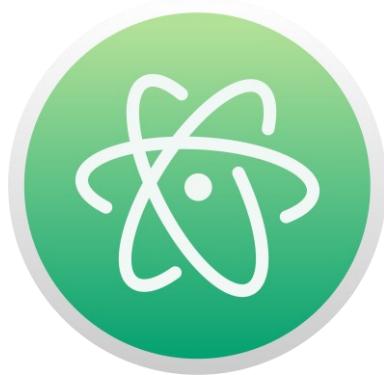


Integrated development environment

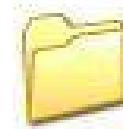
Aký editor mám použiť?



```
:::  
iLE88Dj. :jD88888Dj:  
.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

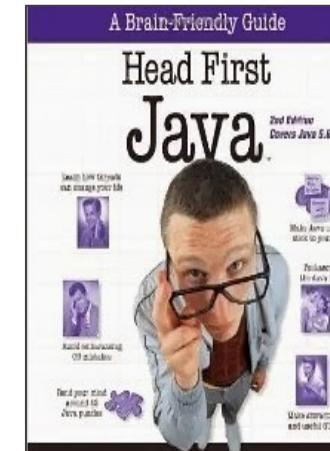
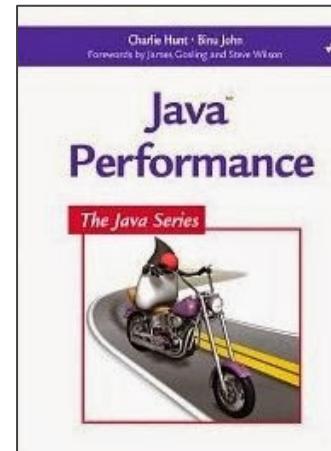
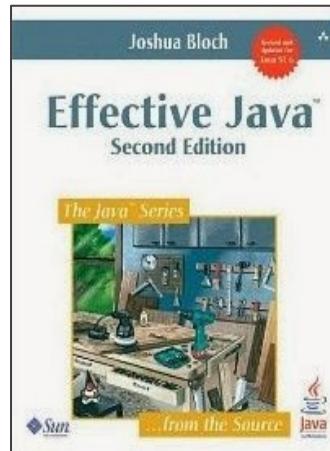
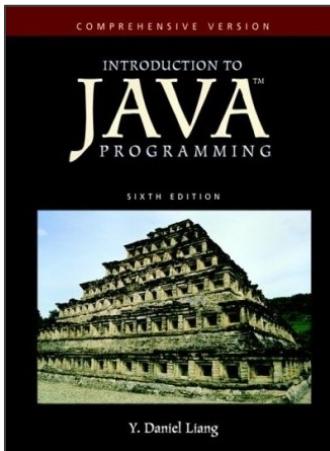
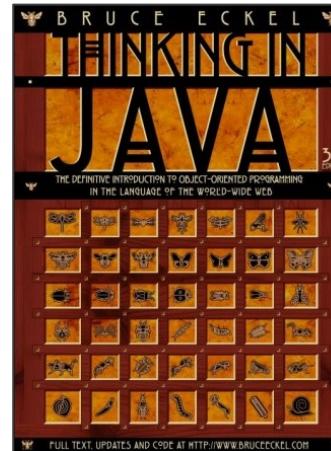
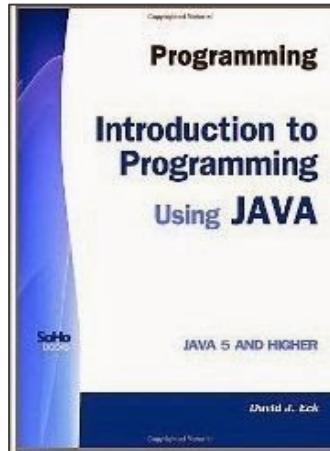
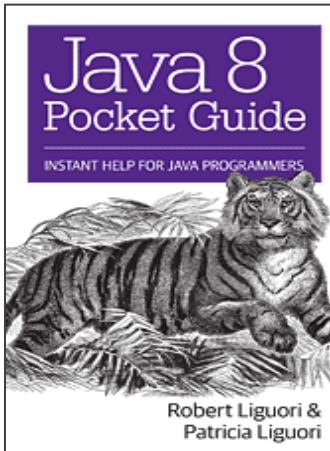
YouTube tutoriály

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...

...

Skratky

Podnikanie na Slove...

2

UK Manazment Externe...

Testovacia firma

VITA - Virtual It Academy

Startupisti

2

Rubyslava

2

Vývojári

▼ Zobrazit viac

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



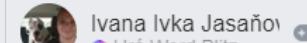
Jarmila Palenčárová



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

Stack Overflow

Tags

Users

FIND A JOB

Jobs

Companies

TEAMS

What's this?

Free 30 Day Trial

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](#)

tomcat

Popular

Name

New

tomcat

for questions about Apache Tomcat (or simply Tomcat, formerly also Jakarta Tomcat) which is an open source Servlet Container developed by the...

40936 questions 11 asked today, 47 this week

tomcat7

Version 7.x (June 2010) of the Apache Tomcat servlet container. Use only if your question is specifically related to features of this version.

5541 questions 10 asked this month, 143 this year

tomcat8

Version 8.x (August 2013 onwards) of the Apache Tomcat servlet container. Use only if your question is specifically related to features of this version.

2576 questions 6 asked this week, 20 this month

tomcat6

Version 6.x (December 2006) of the Apache Tomcat servlet container. Use only if your question is specifically related to features of this version.

1877 questions 11 asked this year

tomcat9

Version 9.x (August 2017 onwards) of the Apache Tomcat servlet container. Use only if your question is specifically related to features of this version.

784 questions 8 asked this week, 32 this month

tomcat5.5

Version 5.5.x (August 2004) of the Apache Tomcat servlet container. Use only if your question is specifically related to features of this version.

252 questions 4 asked this year

embedded-tomcat-8

Embedded Apache Tomcat 8

209 questions 21 asked this year

maven-tomcat-plugin

The Tomcat Maven Plugin provides goals to manipulate WAR projects within the Tomcat servlet container.

202 questions 2 asked this year

embedded-tomcat-7

Questions about running Apache Tomcat 7 as an embedded server in...

tomcat8.5

Version 8.5.x (June 2018 onwards) of the Apache Tomcat servlet container

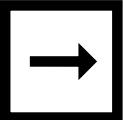
tomcat-valve

a type of component that can be inserted into Tomcat's...

tomcat-jdbc

about tomcat and jdbc working together

Mrkni na náš YouTube kanál a daj odber

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