

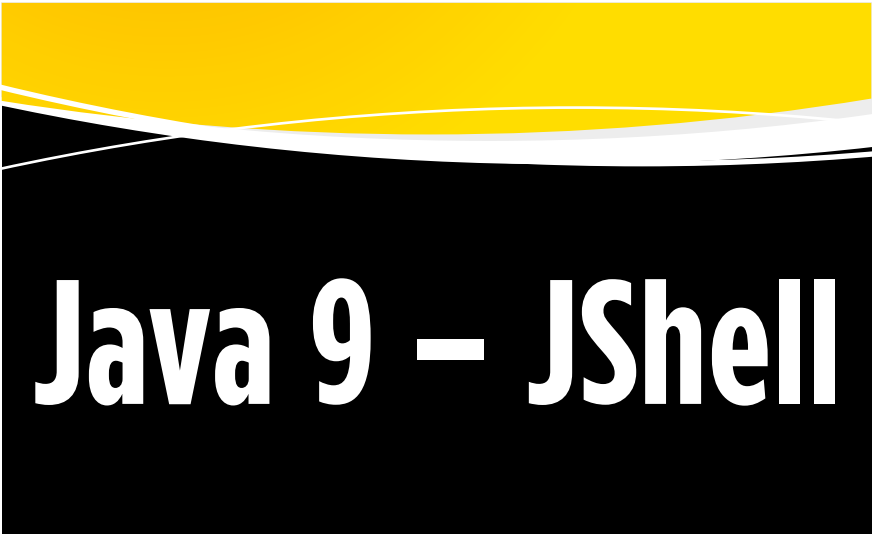
Rudolf Pecinovský

# Java 9 – JShell

- Poskytuje informace v předstihu
- Nezatěžuje výkladem známých konstrukcí
- Probírá nástroj pro výuku a rychlé experimenty
- Ukazuje, jak zabudovat JShell do svého programu

edice  
**PRO NETRPĚLIVÉ**

 GRADA®



# Java 9 – JShell

Grada Publishing

**Ing. Rudolf Pecinovský, CSc.** je absolventem *Fakulty Elektrotechnické ČVUT* z roku 1979. Titul CSc. získal v Ústavu teorie informace a automatizace ČSAV v roce 1983. Od počátku 80. let učí a publikuje, přičemž svůj výzkum soustředí především na oblast vstupních kurzů moderního programování pro naprosté začátečníky. V současné době učí na *Fakultě informatiky a statistiky Vysoké školy ekonomické v Praze*, na *Fakultě jaderné a fyzikálně inženýrské ČVUT* a na *Vysoké škole podnikání a práva*. Doposud mu vyšlo přes 50 knih, které byly přeloženy do pěti jazyků. Většina jeho knih je zaměřena na výuku moderního programování a na umění návrhu objektově orientované architektury.



*Všem, kteří se chtějí něco naučiti*

---

**Upozornění pro čtenáře a uživatele této knihy**

Všechna práva vyhrazena. Žádná část této tištěné či elektronické knihy nesmí být reprodukována a šířena v papírové, elektronické či jiné podobě bez předchozího písemného souhlasu nakladatele. Neoprávněné užití této knihy bude **trestně stíháno**.

**Rudolf Pecinovský**

**Java 9 – JShell**  
**Edice pro netrpělivé**

Vydala Grada Publishing a.s.  
U Průhonu 22, Praha 7

Redakce: Jaroslava Palasová  
Návrh vnitřního layoutu: Rudolf Pecinovský  
Zlom: Rudolf Pecinovský  
Počet stran 163  
První vydání, Praha 2017

*V knize použité názvy mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.*

Copyright © Grada Publishing, a.s., 2017

ISBN 978-80-271-9786-6 (epub)  
ISBN 978-80-271-9785-9 (pdf)

---

# Stručný obsah

Stručný obsah .....	5
Podrobný obsah.....	6
Seznam programů.....	10
Seznam obrázků.....	12
Úvod.....	13
1 Seznamte se: JShell.....	19
2 Začínáme programovat.....	26
3 Rekapitulační příkazy.....	34
4 Ukládání a načítání skriptů.....	51
5 Nápověda.....	65
6 Vylučování úryvků.....	77
7 Konfigurace uživatelského prostředí.....	87
8 Úpravy úryvků.....	104
9 Použití platformy <i>JShell</i> z programu.....	124
A. Příprava prostředí a spuštění programu v systému <i>Windows</i> .....	141
B. Příprava prostředí a spuštění programu v systémech <i>Mac OS</i> a <i>Linux</i> .....	152
Literatura.....	155

---

# Podrobný obsah

Stručný obsah .....	5
Podrobný obsah .....	6
Seznam programů .....	10
Seznam obrázků .....	12
Úvod .....	13
Komu je kniha určena .....	13
Terminologie .....	14
Objekt × Instance .....	14
Rozhraní × interfejs .....	14
Textový podpis .....	14
Syntaktické definice a diagramy .....	15
Potřebné vybavení .....	15
Doprovodné programy .....	16
Použité typografické konvence .....	16
Zpětná vazba .....	18
1 Seznamte se: JShell .....	19
1.1 Prostředí typu REPL .....	19
1.2 Příprava testovací složky (adresáře) .....	21
1.3 Příprava programu <i>JShell</i> a první spuštění .....	21
První spuštění .....	21
Zveřejnění cesty k programu .....	21
1.4 Spuštění a ukončení programu .....	22
1.5 Úryvky (snippets) a příkazy (commands) .....	23
1.6 První úryvky .....	23
1.7 Pomocné proměnné (scratch variables) .....	25
1.8 Soubory pro opakování .....	25
2 Začínáme programovat .....	26
2.1 Definice proměnné .....	26
2.2 Definice proměnné pro rozepsaný výraz .....	27
2.3 Neinicializované proměnné .....	28
2.4 Definice konstant .....	29
2.5 Zadávání kódu ve více řádcích .....	29
2.6 Definice metod .....	30
2.7 Reakce na syntaktické chyby .....	31
2.8 Dopředné odkazy .....	32
2.9 Soubory pro opakování .....	33
3 Rekapitulační příkazy .....	34
3.1 Úryvky (snippets) a příkazy (commands) .....	34
3.2 Listování historií pomocí šipek .....	34

3.3	<b>Příkaz /list</b> .....	35
	Zobrazení vybraného prvku .....	37
	Parametr -start .....	38
	Parametr -all .....	38
	Parametr -history .....	41
	Parametry <name> a <ID> .....	41
	Syntaktický diagram .....	42
3.4	<b>Výpis definovaných proměnných – /vars</b> .....	42
	Syntaktický diagram .....	43
3.5	<b>Výpis definovaných metod – /methods</b> .....	43
	Syntaktický diagram .....	44
3.6	<b>Výpis definovaných datových typů – /types</b> .....	44
	Syntaktický diagram .....	45
3.7	<b>Výpis definovaných importů – /imports</b> .....	46
3.8	<b>Přidání dalšího importu</b> .....	46
	Využití <SHIFT+TAB><I> .....	46
	Přímé zadání importu .....	47
3.9	Soubory pro opakování .....	50
4	<b>Ukládání a načítání skriptů</b> .....	51
4.1	Uložení aktuálního stavu – /save .....	51
4.2	<b>Načtení skriptu – /open</b> .....	55
	Interní zabudované skripty .....	56
4.3	Restart seance – /reset .....	58
4.4	Znovuzavedení – /reload .....	58
4.5	<b>Pokračování z minula – /reload -restore</b> .....	61
4.6	Parametr -quiet .....	63
4.7	Syntaktický diagram .....	63
4.8	Soubory pro opakování .....	64
5	<b>Nápověda</b> .....	65
5.1	<b>Získání nápovědy</b> .....	65
	Syntaktický diagram .....	66
5.2	<b>Zrychlená nápověda</b> .....	67
5.3	<b>Automatické doplňování</b> .....	69
5.4	<b>Zkrácené příkazy</b> .....	70
5.5	<b>Doplňování parametrů příkazů</b> .....	71
5.6	<b>Doplňování dat a kódu</b> .....	73
5.7	<b>Vyhledávání předchozích zadání</b> .....	74
5.8	<b>Vyhledávání následujících zadání</b> .....	76
5.9	Soubory pro opakování .....	76
6	<b>Vylučování úryvků</b> .....	77
6.1	<b>Spuštění úryvku zadáním jeho ID</b> .....	77
	Uložení záznamu o takto spuštěném úryvku do historie .....	78
	„Spuštění“ úryvku s příkazem, definicí metody či datového typu .....	78
6.2	<b>Odebrání (vyloučení) zadaného úryvku – /drop</b> .....	79
	Vylučování úryvků zadaných názvem .....	80
	Vyloučené úryvky se pouze deaktivují .....	81
	Reaktivace úryvků .....	82
	Více úryvků se shodným názvem .....	84
	Vyloučení již vyloučeného úryvku .....	85
	Syntaktický diagram .....	86
6.3	Soubory pro opakování .....	86

7	<b>Konfigurace uživatelského prostředí</b> .....	87
7.1	<b>Zjištění aktuálního nastavení</b> .....	87
7.2	<b>Nastavení zpětnovazebního režimu</b> .....	88
	Syntaktický diagram.....	88
	Podrobný režim (verbose mode).....	89
	Standardní režim (normal mode).....	90
	<b>Stručný režim (concise mode)</b> .....	91
	Tichý režim (silent mode).....	92
	Parametr <code>-retain</code> .....	92
	Trvalé nastavení aktuálního režimu.....	94
	Uživatelský režim.....	94
	<b>Nastavení zpětnovazebního režimu v příkazovém řádku</b> .....	94
7.3	<b>Nastavení startovního skriptu</b> .....	96
	<b>Nastavení startovního skriptu v příkazovém řádku</b> .....	100
7.4	<b>Nastavení proměnné <code>classpath</code></b> .....	100
7.5	<b>Soubory pro opakování</b> .....	103
8	<b>Úpravy úryvků</b> .....	104
8.1	<b>Editační klávesové zkratky</b> .....	104
8.2	<b>Použití zabudovaného editoru</b> .....	105
8.3	<b>Reakce prostředí na úpravy v editoru</b> .....	107
8.4	<b>Rizika některých úprav</b> .....	109
	<b>Přípravné operace</b> .....	110
	<b>Redefinice třídy Demo</b> .....	111
	Problémy po redefinici.....	111
	<b>Analýza aktuálního stavu prostředí</b> .....	113
	<b>Experimenty s instancemi původní a redefinované třídy</b> .....	114
	<b>Závěr</b> .....	116
8.5	<b>Nastavení vlastního editoru</b> .....	116
	<b>Zadání příkazu bez parametrů</b> .....	117
	Parametr <code>-retain</code> .....	117
	Parametr <code>-default</code> .....	118
	Parametr <code>-delete</code> .....	118
	Parametr <code>-wait</code> .....	118
	<b>Nastavení požadovaného editoru</b> .....	118
	Specifika <i>Windows</i> .....	119
	<b>Komunikace s některými známými editory</b> .....	119
	Programy pro editaci nevhodné.....	119
	Programy pro editaci nešikovné.....	119
	Programy, které je možno spustit a opět zavřít.....	120
	Programy simulující své okamžité zavření.....	122
	Programy pracující v multiinstančním režimu.....	122
	Další možnosti.....	123
8.6	<b>Soubory pro opakování</b> .....	123
9	<b>Použití platformy <i>JShell</i> z programu</b> .....	124
9.1	<b>Nejdůležitější součásti platformy <i>JShell</i></b> .....	124
	<b>Balíček <code>jdk.jshell</code></b> .....	125
	<b>Třída <code>JShell</code></b> .....	125
	Vláknová bezpečnost.....	126
	<b>Třída <code>SnippetEvent</code></b> .....	126
	Vláknová bezpečnost.....	127
	<b>Třída <code>Snippet</code></b> .....	127
	Vláknová bezpečnost.....	127
	<b>Výčtový typ <code>Snippet.Status</code></b> .....	127
	<b>Třída <code>SourceCodeAnalysis</code></b> .....	128
9.2	<b>Ukázka využití v kódu</b> .....	128



---

	Spouštěcí metoda .....	129
	Metoda processSnippetEvent(SnippetEvent).....	130
	Metoda evaluateStatus(SnippetEvent, StringBuilder) .....	132
9.3	<b>Test vytvořeného kódu</b> .....	133
	Jednoduché výrazy a příkazy .....	134
	Definice metod a dopředné odkazy.....	136
	Definice metod s problematickou signaturou.....	136
	Úprava definice datového typu.....	137
	Současná definice více úryvků.....	138
	Závěrečné informace .....	139
9.4	Soubory pro opakování.....	140
A.	<b>Příprava prostředí a spuštění programu v systému Windows</b> .....	141
A.1	<b>Úprava proměnné Path</b> .....	141
	Windows 10 .....	144
	Windows 7 .....	145
A.2	<b>Spuštění programu</b> .....	146
	Zástupce na ploše, resp. v zadané složce.....	146
	Úprava vlastností zástupce.....	147
	Vytvoření vlastní dávky .....	148
	Přímé spuštění příkazového řádku .....	149
	Total Commander .....	149
	Průzkumník ve Windows 10.....	149
	Dodatečné nastavení aktuální složky .....	150
B.	<b>Příprava prostředí a spuštění programu v systémech Mac OS a Linux</b> .....	152
B.1	Omluva .....	152
B.2	<b>Úprava proměnné PATH</b> .....	153
B.3	<b>Spuštění programu</b> .....	154
	Literatura.....	155

---

# Seznam programů

Výpis 1.1:	Odezva prostředí JShell po spuštění.....	22
Výpis 1.2:	Reakce prostředí na zadání výrazu.....	23
Výpis 1.3:	Odezva prostředí na zadání dalších výrazů.....	24
Výpis 2.1:	Definice proměnné pozdrav a její použití.....	27
Výpis 2.2:	Definice proměnné pozdrav2 a její použití.....	28
Výpis 2.3:	Deklarace neinicializovaných proměnných.....	28
Výpis 2.4:	Pokus o deklaraci konstanty.....	29
Výpis 2.5:	Definice výrazu rozepsaná na více řádcích.....	30
Výpis 2.6:	Definice metody currentFile() vracející aktuální složku (adresář).....	30
Výpis 2.7:	Demonstrace zpracování syntaktických chyb.....	31
Výpis 2.8:	Demonstrace zpracování dopředného odkazu.....	32
Výpis 3.1:	Odpověď programu po zadání příkazu /list.....	36
Výpis 3.2:	Použití příkazu /list k zobrazení vybraných výpisků.....	37
Výpis 3.3:	Použití příkazu /list k zobrazení startovní sady úryvků.....	38
Výpis 3.4:	Odpověď programu po zadání příkazu /list -all.....	38
Výpis 3.5:	Odpověď programu po zadání příkazu /list -history.....	40
Výpis 3.6:	Vypsání zadaných úryvků.....	41
Výpis 3.7:	Odpověď programu po zadání příkazu /vars.....	42
Výpis 3.8:	Odpověď programu po zadání příkazu /methods.....	43
Výpis 3.9:	Odpověď programu po zadání příkazu /types.....	44
Výpis 3.10:	Definice datového typu a jeho použití.....	45
Výpis 3.11:	Odpověď programu po zadání příkazu /imports.....	46
Výpis 3.12:	Definice potřebného importu při vkládání úryvku.....	47
Výpis 3.13:	Dodatečná definice potřebného importu.....	48
Výpis 3.14:	Seznam aktivních úryvků na konci kapitoly.....	49
Výpis 3.15:	Výsledný seznam zadaných importů.....	49
Výpis 4.1:	Obsah souboru uloženého v tomto okamžiku příkazem /save -history 04.1_Save_Hist.jsh.....	53
Výpis 4.2:	Obsah interního skriptu DEFAULT.....	56
Výpis 4.3:	Obsah interního skriptu PRINTING.....	57
Výpis 4.4:	Seance, jež uloží svoji historii do souboru 04.3_Reset_Hist.jsh.....	59
Výpis 4.5:	Obsah souboru 04.3_Reset_Hist.jsh uloženého ve výpisu 4.4.....	59
Výpis 4.6:	Demonstrace chování příkazů /reset a /reload.....	60
Výpis 4.7:	Obsah souboru 04.4_Reload_Hist.jsh uloženého na konci výpisu 4.6.....	61
Výpis 4.8:	Výpis, který je pokračováním výpisu 4.6, demonstruje chování programu po zadání příkazu /reload -restore.....	62
Výpis 4.9:	Obsah souboru 04.5_Reload_Restore_Hist.jsh ukládaného na konci výpisu 4.8.....	63
Výpis 5.1:	Nápověda, jak používat nápovědu.....	66

Výpis 5.2:	Reakce prostředí na další tři po sobě jdoucí stisky klávesy <TAB> .....	68
Výpis 5.3:	Postupné doplňování zadávaného příkazu /list s10 s2.....	69
Výpis 5.4:	Postupné doplňování zadávaného příkazu /list s10 s2.....	72
Výpis 5.5:	Doplňování dat a zobrazení programové dokumentace .....	74
Výpis 5.6:	Akce předcházející ukázce chování prostředí při zpětném hledání.....	75
Výpis 5.7:	Průběh zpětného hledání.....	75
Výpis 6.1:	Opětne spuštění prvních dvou úryvků .....	78
Výpis 6.2:	Začátek seznamu úryvků po vyloučení proměnné \$1.....	79
Výpis 6.3:	Reakce na použití neexistující proměnné.....	80
Výpis 6.4:	Odpovědi programu po zadání příkazu /drop start() a následném spuštění úryvku 29 .....	81
Výpis 6.5:	Vypsání podoby odebraných, neaktivních úryvků.....	82
Výpis 6.6:	Používání aktivních i vyloučených úryvků .....	83
Výpis 6.7:	Odpovědi programu po zadání příkazu /drop.....	84
Výpis 6.8:	Vyloučení již vyloučeného úryvku.....	85
Výpis 7.1:	Zjištění aktuálního nastavení.....	88
Výpis 7.2:	Posloupnost příkazů zadávaná pro test reakcí v jednotlivých zpětnovazebních režimech a uložená v souboru 07.2_Test.jsh.....	89
Výpis 7.3:	Reakce na příkazy z výpisu 7.2 v podrobném režimu .....	90
Výpis 7.4:	Reakce na příkazy z výpisu 7.2 ve standardním režimu.....	91
Výpis 7.5:	Reakce na příkazy z výpisu 7.2 ve stručném režimu.....	91
Výpis 7.6:	Reakce na příkazy z výpisu 7.2 v tichém režimu.....	92
Výpis 7.7:	Vliv parametru -retain na nastavení zpětnovazebního režimu .....	92
Výpis 7.8:	Vliv parametru -retain na nastavení zpětnovazebního režimu .....	95
Výpis 7.9:	Nastavování startovního skriptu .....	97
Výpis 7.10:	Nastavování proměnné classpath .....	101
Výpis 8.1:	Reakce prostředí na úpravy v zabudovaném editoru .....	108
Výpis 8.2:	Pokračování výpisu 8.1 zobrazující přípravné operace .....	110
Výpis 8.3:	Upravená definice (druhá verze) třídy Demo.....	111
Výpis 8.4:	Pokračování výpisu 8.2 zobrazující problémy po redefinici třídy Demo .....	112
Výpis 8.5:	Pokračování výpisu 8.4 demonstrující mé analytické kroky.....	114
Výpis 8.6:	Pokračování výpisu 8.5 demonstrující experimenty s instancemi původní i opravené verze třídy Demo .....	114
Výpis 8.7:	Použití editoru WordPad.....	120
Výpis 8.8:	Použití editoru jEdit.....	121
Výpis 8.9:	Použití editoru PSPad.....	122
Výpis 9.1:	Definice metody main ve třídě ExampleJShell .....	129
Výpis 9.2:	Definice metody processSnippetEvent(SnippetEvent) ve třídě ExampleJShell.....	131
Výpis 9.3:	Definice metody evaluateStatus(SnippetEvent, StringBuilder) ve třídě ExampleJShell.....	132
Výpis 9.4:	Obsah souboru 09.3_Test.jsh se zadávanými testovacími texty .....	133
Výpis 9.5:	Úvodní část seance demonstrující zpracování jednoduchých výrazů .....	134
Výpis 9.6:	Část demonstrující definici metod s dopřednými odkazy.....	135
Výpis 9.7:	Část demonstrující definici metod s dopřednými odkazy v signatuře .....	137
Výpis 9.8:	Část demonstrující změnu definice používaného datového typu.....	138
Výpis 9.9:	Část demonstrující změnu definice používaného datového typu.....	139
Výpis 9.10:	Závěrečná část se souhrnným výpisem definovaných úryvků.....	140

# Seznam obrázků

Obrázek 3.1: Syntaktický diagram příkazu <code>/list</code> .....	42
Obrázek 3.2: Syntaktický diagram příkazu <code>/vars</code> .....	43
Obrázek 3.3: Syntaktický diagram příkazu <code>/methods</code> .....	44
Obrázek 3.4: Syntaktický diagram příkazu <code>/types</code> .....	45
Obrázek 3.5: Dialogové okno otevřené úryvkem .....	48
Obrázek 4.1: Syntaktický diagram příkazu <code>/save</code> .....	52
Obrázek 4.2: Syntaktický diagram příkazu <code>/open</code> .....	55
Obrázek 4.3: Syntaktický diagram příkazu <code>/reload</code> .....	64
Obrázek 5.1: Syntaktický diagram příkazu pro získání nápovědy .....	67
Obrázek 6.1: Syntaktický diagram příkazu <code>/drop</code> .....	86
Obrázek 7.1: Syntaktický diagram příkazu <code>/set feedback</code> pro nastavení zpětnovazebního režimu .....	89
Obrázek 7.2: Syntaktický diagram příkazu <code>/set start</code> pro nastavení startovního skriptu .....	96
Obrázek 8.1: Syntaktický diagram příkazu <code>/edit</code> .....	106
Obrázek 8.2: Okno zabudovaného editoru otevřené po zadání příkazu <code>/edit</code> .....	106
Obrázek 8.3: Definice třídy <code>Demo</code> v editačním okně .....	108
Obrázek 8.4: Syntaktický diagram příkazu <code>/set editor</code> .....	117
Obrázek 9.1: Závislosti mezi hlavními třídami balíčku <code>jdk.jshell</code> .....	124
Obrázek A.1 Okno otevřené ve Windows 10 po žádosti o vlastnosti složky <code>Tento počítač</code> .....	142
Obrázek A.2 Okno <b>Vlastnosti systému</b> .....	142
Obrázek A.3 Okno <b>Proměnné prostředí</b> ve Windows 10 .....	143
Obrázek A.4 Okno <b>Proměnné prostředí</b> ve Windows 7 .....	144
Obrázek A.5 Okno <b>Upravit proměnné prostředí</b> ve Windows 10 .....	145
Obrázek A.6 Okno <b>Úpravy systémové prostředí</b> ve Windows 7 .....	145
Obrázek A.7: Místní nabídka po puštění souboru přesouvaného <b>pravým</b> tlačítkem myši .....	146
Obrázek A.8: Dialogové okno pro nastavení vlastností zadaného zástupce .....	147
Obrázek A.9: Místní nabídka po puštění souboru přesouvaného <b>pravým</b> tlačítkem myši .....	150

# Úvod

Otevíráte první ze série knih, které se vás pokusí seznámit s nejnvýraznějšími novinkami nové verze jazyka *Java*, aniž by vás rozptylovaly výkladem základních programových konstrukcí.

Tato kniha se soustředí na výklad nového nástroje, kterým je program *JShell*. Tento program je reakcí na volání programátorů a učitelů programování, kterým v základní sadě nástrojů platformy *Java* chyběl program určený pro práci v režimu označovaném zkratkou REPL (*Read-Evaluate-Print-Loop*).

Příští díl se bude věnovat zavedení modulárního systému označovaného zkratkou JPMS (*Java Platform Modular System*) a z toho vyplývající úpravy a rozšíření základní sady nástrojů. V případě zájmu přibude i díl probírající ostatní novinky určené většinou zkušenějším programátorům.

## Komu je kniha určena

Nástroj *JShell* je zaměřen na dvě skupiny uživatelů:

- Na programátory, kteří si potřebují rychle a jednoduše vyzkoušet nějaký nový nápad.
- Na učitele programování, kteří chtějí jednoduše demonstrovat studentům přednášenou látku a vlastnosti probírané programové konstrukce.

Tomu odpovídá i zaměření knihy. U obou skupin lze předpokládat, že znají jak programovací jazyk *Java*, tak stejnojmennou platformu a chtějí se seznámit s hlavními novinkami deváté verze jazyka a platformy *Java*. V knize proto nebudu vysvětlovat všeobecně známé konstrukce a knihovny, ale soustředím se na co nejlepší výklad práce s probíraným nástrojem. Na druhou stranu však nebudu předpokládat žádné hluboké znalosti programování ani hlubší znalosti práce s operačním systémem, takže by knihu měl být schopen číst i začínající programátor – absolvent základního kurzu programování.

Dohodněme se, že budu předpokládat programátorské znalosti na úrovni své knihy *Java 8 – Úvod do objektové architektury pro mírně pokročilé* ([Pecinovský, 2014](#)).

---

## Terminologie

Nové termíny vysvětlím, až na ně narazím v textu. Tady bych chtěl pouze upozornit na termíny běžně používané, které se však v různých kurzech vysvětlují různě, takže při jejich interpretaci může občas dojít k nedorozumění.

### Objekt × Instance

Prvním z nich je obecně známý termín **objekt**. Objektově orientované programování (OOP) říká, že „*objekt je vše, co můžeme označit podstatným jménem*“. Objektem jsou tedy i datové typy, metody a další součásti programu. Řada programátorů ale omezuje své chápání tohoto termínu na objekty, které jsou instancemi nějaké třídy. Toto chápání je důsledek kompromisů, které provedli autoři některých programovacích jazyků pro zvýšení efektivity vytvořených programů.

V čistém OOP implementovaném např. v jazyku *Smalltalk* je však objektem cokoli. V tomto významu budu tento termín dále používat. Budu-li chtít hovořit o objektech, které jsou instancemi nějaké třídy, budu vždy hovořit o instancích.

O programu *JShell*, kterému je věnována tato kniha, bychom mohli říci, že ke všem objektům přistupuje z jistého hlediska stejně a je mu jedno, zda se právě jedná o instanci nějakého typu, příkaz, metodu nebo datový typ. Připravte se proto na to, že budu-li v textu hovořit o objektech, budu mít na mysli libovolný druh objektu.

### Rozhraní × interfejs

Termínem *rozhraní* (přesněji rozhraní objektu) označuji souhrn vlastností a schopností, které daný objekt zveřejňuje a jeho okolí je může využívat při komunikaci s daným objektem.

Termínem *interfejs* pak budu označovat programovou konstrukci, která měla původně reprezentovat rozhraní datových typů, ale v osmé verzi *Javy* přibrala i omezené možnosti definice implementace.

### Textový podpis

Budu-li někde hovořit o textovém podpisu instance (případně jenom o podpisu instance), budu tím mít na mysli textový řetězec obdrženy po zavolání metody `toString()`.

## Syntaktické definice a diagramy

Pro definice možných způsobů zápisu nějakého příkazu se používají syntaktické definice a/nebo syntaktické diagramy. Syntaktické definice jsou výhodnější pro strojové zpracování, ale syntaktické diagramy jsou pochopitelnější pro člověka. V učebnicích se sice syntaktické diagramy příliš neobjevují, ale to je většinou pouze důsledek nechuti autorů k tvorbě obrázků. V této knize jim ale budu dávat přednost.

Syntaktický diagram ukazuje, jak je možno zobrazovanou konstrukci zapsat. Pojedete-li po čarách, tak jakýkoliv průjezd generuje syntakticky správnou konstrukci. Toho, kdo syntaktické diagramy ještě nezná a chtěl by rychle některý vidět, bych odkázal např. na diagram na obrázku [3.1](#) na straně [42](#).

## Potřebné vybavení

Nechcete-li si knížkou pouze listovat, ale chcete-li ji použít jako příručku či dokonce učebnici, budete si jistě chtít vše vyzkoušet. K tomu potřebujete vhodný vývojový nástroj. Pro studium této knihy bude s výjimkou poslední kapitoly stačit pouhé JDK. Pro experimentování s programem probíraným v poslední kapitole je vhodné instalovat IDE podporující *Javu 9*.

## Vývojová sada JDK 9

Pro úspěšné studium této knihy budete potřebovat mít instalovanou *Javu 9*. Tu stáhnete ze stránky <http://www.oracle.com/technetwork/java/javase/downloads/>.

Patříte-li mezi ty opravdu netrpělivé, kteří si chtějí vše vyzkoušet ještě před tím, než bude uvolněna oficiální verze (firma *Oracle* plánuje její uvedení na 21. září 2017), musíte si stáhnout předběžnou verzi označovanou jako *Early Access*. Odkaz na ni najdete na téže stránce, jenom musíte popojet kousek níž a stisknout tlačítko **Download** v sekci *Early Access Releases*.

## Vývojové prostředí

Teoreticky už nic dalšího nepotřebujete, nicméně většina programátorů používá nějaký profesionální vývojový nástroj. Je zcela na vás, jaký nástroj zvolíte. Já jsem při vývoji výukových nástrojů používal vývojové prostředí *NetBeans*, protože ta ostatní nenabízejí potřebnou nestandardní funkcionalitu. Jsou totiž optimalizována pro profesionální programátory, kteří tyto nestandardní funkce nepotřebují.

---

Opět si ale budete muset zkontrolovat, že daná verze podporuje *Javu* 9. Očekávám, že ostré verze podporující *Javu* 9 se objeví až na přelomu roku 2017 a 2018. Budete-li si proto chtít vyzkoušet vše dříve, budete muset použít vývojovou verzi daného prostředí. Podrobnosti se dozvíte na stránce výrobce.

Pro tři nejrozšířenější vývojová prostředí, tj. pro prostředí *NetBeans*, *IntelliJ IDEA* a *Eclipse* se pokusím sledovat poslední změny a informovat o nich na stránce knihy na adrese [http://knihy.pecinovsky.cz/53\\_java9\\_jshell](http://knihy.pecinovsky.cz/53_java9_jshell). Psát o nich zde nepovažuji vzhledem k častým změnám za vhodné.

## Doprovodné programy

Všechny doprovodné programy zmiňované a používané v textu najdete na stránce knihy na adrese [http://knihy.pecinovsky.cz/53\\_java9\\_jshell](http://knihy.pecinovsky.cz/53_java9_jshell). Měli byste zde najít soubory `53_Java9_JShell_WWW.zip`, což je archiv obsahující dvě virtuální složky:

- Složku `JShell` se záznamy seancí spolu se soubory ukládanými v průběhu výkladu. Na tyto soubory se bude v každé kapitole odkazovat její závěrečná podkapitola nazvaná *Soubory pro opakování*.
- Složku `Lib` obsahující dvě knihovny, které začneme používat v podkapitole [7.4 Nastavení proměnné classpath](#) na straně [100](#). V této složce jsou dva soubory:
  - Soubor `js.Win_Class.jar` obsahující jednoduchou demonstrační knihovnu s třídou `js.Win`.
  - Soubor `N56-Lib-en_Framework.jar` s jednoduchým frameworkem, který používám ve svých začátečnických učebnicích. Zájemci o její zdrojový kód najdou příslušný odkaz na stránce knihy.

## Použité typografické konvence

K tomu, abyste se v textu lépe vyznali a také abyste si vykládanou látku lépe zapamatovali, používám několik prostředků pro odlišení a zvýraznění textu.

**Termíny** První výskyt nějakého termínu a další texty, které chci zvýraznit, vysazuji **tučně**.

**Název** Názvy firem a jejich produktů vysazuji *kurzivou*. Kurzivou vysazuji také názvy kapitol, podkapitol a oddílů, na které se v textu odkazují.

**Citace** Texty, které si můžete přečíst na displeji, např. názvy polí v dialogových oknech či názvy příkazů v nabídkách, vysazuji **tučným bezpatkovým písmem**.

---



- Program** Identifikátory a další části programů zmíněné v běžném textu vysazují **neproporcionálním písmem**, které je v elektronických verzích pro zvýraznění tmavě červené.
- metoda(?)** Při odkazech na metody budu v závorkách za názvem metody vždy uvádět seznam typů jejich parametrů – např. `equals(Object)`. Nebude-li v danou chvíli jasné, jaké má zmiňovaná metoda parametry, budu do závorek psát otazník – např. `metoda(?)`.
- úryvek** Tučně červeně bude ve výpisech konverzace se systémem vypisován text zadaný uživatelem. Text zapsaný systémem bude vypsán černě.
- <KLÁVESAS>** Modře, kapitálkami a ve špičatých závorkách budou vysazovány klávesové zkratky. Ve výpisech budou označovat místa, kde uživatel danou klávesovou zkratku stiskl. Má-li se klávesa stisknout při stisknutí přepřařovači, bude tento přepřařovač uveden v závorkách spolu s klávesou – např. `<SHIFT+TAB>`.

Kromě výše zmíněných částí textu najdete v knize ještě řadu doplňujících poznámek a vysvětlivek. Všechny budou v jednotném rámečku, který bude označen ikonou charakterizující druh informace, kterou vám chce poznámka či vysvětlivka předat.



Symbol jing-jang bude uvozovat poznámky, s nimiž se setkáte na počátku každé kapitoly. Zde vám vždy prozradím, co se v dané kapitole naučíte.



Otevřená schránka s dopisem označuje informace o projektu, s nímž budeme v dalším textu pracovat, nebo v něm najdete vzorové řešení aplikující probranou látku. Příslušný projekt získáte pomocí generátoru projektů popsaného výše.



Obrázek knihy označuje poznámku týkající se používané terminologie. Tato poznámka většinou upozorňuje na další používané termíny označující stejnou skutečnost nebo na konvence, které se k probírané problematice vztahují.



Píšící ruka označuje obyčejnou poznámku, která pouze doplňuje informace z hlavního proudu výkladu o nějakou zajímavost.



Ruka s hrozícím prstem upozorňuje na věci, které byste měli určitě vědět a na které byste si měli dát pozor, protože jejich zanedbání vás většinou dostane do problémů.



Usměváček vás bude upozorňovat na různé tipy, kterými můžete vylepšit svůj program nebo zefektivnit svoji práci.



Mračoun vás naopak bude upozorňovat na různá úskalí programovacího jazyka nebo programů, s nimiž budeme pracovat. Bude vám radit, jak se těmto nástrahám vyhnout či jak to zařídit, aby vám alespoň pokud možno nevadily.



Brýle označují tzv. „poznámky pro šouraly“, ve kterých se vás snažím seznámit s některými zajímavými vlastnostmi probírané konstrukce nebo upozorňuji na některé souvislosti. Jejich znalost však není k pochopení látky nezbytná.

## Zpětná vazba

Kniha vznikla na základě přání řady mých studentů a programátorů z mých kurzů. Vznikala poměrně narychlo, takže lze očekávat, že se v ní může vyskytnout větší množství chyb, protože ve zrychleném režimu procházela i redakcí. Objevíte-li proto v knize nějakou chybu nebo budete mít návrh na nějaké její vylepšení, neostýchejte se napsat na adresu [rudolf@pecinovsky.cz](mailto:rudolf@pecinovsky.cz). Pokusím se co nejdříve zanést na stránku knihy příslušná errata s opravou, kterou pak zapracujeme do případného dalšího vydání.

Pokud vám bude někde připadat text nepříliš srozumitelný nebo budete mít nějaký dotaz, ať už k vykládané látce či použitému vývojovému prostředí, pošlete [mail s předmětem ADVENT DOTAZ](mailto:mail@pecinovsky.cz). Bude-li se dotaz týkat něčeho obecnějšího, zveřejním na stránce knihy [http://knihy.pecinovsky.cz/53\\_java9\\_jshell](http://knihy.pecinovsky.cz/53_java9_jshell) odpověď i pro ostatní, které by mohl obdobný dotaz napadnout za pár dní, anebo jsou natolik ostýchaví, že se netroufnou sami zeptat.

# Kapitola 1

## Seznamte se: *JShell*



### Co se v kapitole naučíte

Tato kapitola vám vysvětlí, co to jsou prostředí typu REPL, a seznámí vás s programem a prostředím *JShell*, které je od 9. verze integrální součástí *Javy*. Naučí vás, jak upravit konfiguraci svého systému tak, aby bylo možno program *JShell* snadno spustit, a na příkladu několika jednoduchých příkazů vám ukáže, jak s ním pracovat.



*JShell* bude v textu knihy střídavě vystupovat jako program a prostředí. Dokud o něm budu hovořit z hlediska operačního systému, bude to program. Jakmile ale tento program spustím, stane se pro mne prostředím, v němž mohu zadávat své požadavky a očekávat odpovědi.

Program *JShell* zařazujeme do kategorie nástrojů typu REPL, což je zkratka z anglického *read-evaluate-print-loop* (česky: *cyklus přečti-vyhodnoť-vytiskni*). Tyto nástroje se používají jako poskytovatelé prostředí (platformy) pro rychlou interakci s podkladovým prostředím, kterým může být jak operační systém, tak nějaký program. Uživatel platformy může komunikovat s podkladovým prostředím v programovacím jazyce dané platformy bez potřeby překladu a samostatného spuštění.

## 1.1 Prostředí typu REPL

Prostředí typu REPL jsou jednoduchá programovací prostředí, která vyhodnocují uživatelské příkazy ihned po jejich zadání a netrvají na dodržení některých pravidel použitého jazyka.

Ve výše zmíněném cyklu se opakovaně vyvolávají jeho jednotlivé funkce, které jsou charakterizovány následovně:

- Funkce **čtení (read)** přijímá zadání od uživatele a zadaný výraz, příkaz či definici uloží v nějaké vnitřní podobě do své paměti. Uživatel může např. zadat

```
(3 + 5) * 7
```

Prostředí výraz předzpracuje a vrátí jej převedený do vnitřního tvaru, v kterém jej lze lépe zpracovat.

- Funkce **vyhodnocení** (správně by mělo být *evaluate*, ale většinou se používá pouze zkrácené **eval**) převezme uživatelské zadání převedené do vnitřního tvaru a vyhodnotí je. V našem případě sečte čísla 3 a 5, součet vynásobí číslem 7 a uloží vyhodnocený výsledek.
- Funkce **tisk (print)** převezme obdrženy výsledek a ve vhodném tvaru jej vytiskne. Většinou ho ještě doplní o další informace.

Prostředí typu REPL se používají buď jako příkazová prostředí anglicky označovaná zkratkou *shell*, která jsou spouštěna jako programy z operačního systému. Používají se také jako nástroje spouštěné z jiného programu a umožňující rozšiřovat schopnosti tohoto programu. Výrazně však převažuje první z uvedených použití a stejně tomu bude i v této knize.

Prostředí spouštěná jako programy přímo ze systému umožňují svým uživatelům přístup k důležitým funkcím operačního systému a spolu s nimi i přístup k různým programovacím funkcím. Využívají se však nejenom ke komunikaci se systémem, ale také pro rychlé prototypování připravovaných programů. Nezanebatelné je i jejich použití při výuce programování, protože poskytují nováčkům rychlou zpětnou vazbu.

Prostředí typu REPL byla vyvinuta pro řadu rozšířených programovacích jazyků, mezi nimi i pro *Java*. *Java* však až do verze 9 nenabízela žádný program realizující prostředí typu REPL, který by byl integrální součástí platformy. Od verze 9 se tato situace mění – integrální součástí platformy *Java* stává program *JShell*.

Program *JShell* však slouží nejenom jako interaktivní prostředí, ale je koncipován tak, aby mohly jeho služeb (jeho prostředí) využívat i jiné programy. O tom si však povíme až v kapitole [9 Použití platformy JShell z programu](#) na straně [124](#).