

Výukový kurs programovacího jazyka Lua

Tutorial for Programming Language Lua

Tomáš Hála

Abstrakt: Příspěvek představuje důvody vzniku nového předmětu o programovacím jazyce Lua a seznamuje s prvními poznatky z pilotního kursu vybraných částí. Návrh syllabu tohoto předmětu je rovněž připojen.

Tvorba nového kursu je řešena jako součást projektu *Innovative Open Source Courses for Computer Science*.

Klíčová slova: programovací jazyk Lua, kurs programování, open source software, výuka na vysoké škole

Abstract: This paper presents the reasons for the creation of a new course on programming language Lua and introduces the first findings from the pilot course of selected parts. A draft syllabus for this course is also attached.

The creation of the new course is being addressed as part of the *Innovative Open Source Courses for Computer Science*.

Key words: programming language Lua, programming course, open source software, university education

1 Úvod

Programování patří mezi stálé a neodmyslitelné součásti přípravy studentů informatických oborů. Volba programovacích jazyků, které jsou využívány ve výuce algoritmizace a programování na vysokých školách, vychází z historických tradic, z možností školy i z profilů absolventů a jejich předpokládaného pracovního zařazení, avšak souhrnně lze říci, že se většinou jedná o jazyky vhodné spíše pro produkční programování než o jazyky vhodné k výuce, a to i za cenu, že jsou zatíženy mnoha úskalími (syntaktický zápis, množství knihoven), které komplikují přímočarou cestu k pochopení algoritmu.

Upřednostňování „okamžitého“ výstupu v podobě znalosti produkčního jazyka může být jedním z důvodů, proč méně známé jazyky jsou zastoupeny ve výuce programování jen zřídka. V nabídce předmětů vysokých škol lze nalézt pak jen předměty zabývající se konkrétním programovacím jazykem, avšak programovací jazyk Lua není dosud v České republice vyučován ani touto formou, což je jedním z důvodů, proč byl kurs progra-

mování v jazyce Lua zařazen do projektu *Innovative Open Source Courses for Computer Science*.

Potřebu samostatného předmětu pro výuku programovacího jazyka Lua násobí i fakt, že na Provozně ekonomické fakultě Mendelovy univerzity v Brně vzniklo v minulých letech několik bakalářských a diplomových prací, které využily programovací jazyk Lua pro řešení zadaného tématu, například pro syntaktickou kontrolu textů v ConTeXtu (Hanuš, 2015; Makeš, 2015) nebo pro výpočty ve vexilografickém modulu (Trefák; 2016) či dvojice Kocurová (2020) a Kašparová (2020), které využily programovací jazyk Lua pro výpočty a řízení vykreslování statistických grafů. Všichni studenti byli nuceni se jazyk naučit sami, neorganizovaně a jen s podporou vedoucího práce, což by nebylo po zavedení takového předmětu nutné a přípravu studentů pro práci s jazykem Lua v rámci závěrečných prací výrazně zkvalitnilo.

2 Programovací jazyk Lua

Programovací jazyk Lua, ač méně známý, nepatří mezi tzv. „výkřiky poslední módy“ – jedná se o programovací jazyk, jenž vznikl v 90. letech minulého století. Jeho první verze byla vytvořena již v roce 1993.

Byl navržen jako jazyk skriptovací a řadíme jej do kategorie vysokoúrovňových jazyků, multiparadigmatických, imperativní a procedurální programovací jazyků. Je charakterizován jako jazyk odlehčený. Jeho významnou vlastností je též reflexivita.

Srovnáme-li programovací jazyk Lua s ostatními programovacími jazyky, zjistíme, že stojí někde „na půl cesty“ mezi jazyky vhodnými pro výuku (např. Pascal) a jazyky produkčními (např. C, C++, Java). Syntakticky se podobá v řadě rysů srozumitelnému a původně pouze pro účely výuky konstruovanému programovacímu jazyku Pascal (např. Wirth, 1981), což umožňuje velmi rychle osvojení syntaktických konstrukcí, na druhou stranu neobsahuje některé užitečné rysy, například striktní typové kontroly, které velmi účinně brání uživateli zapisovat nesmyslné výrazy.

Již zmíněné snadné osvojení jazyka je – kromě souborů – podpořeno i jediným *typickým* strukturovaným datovým typem *tabulka* (*table*), sloužící jako indexované i asociativní pole, a to jak homogenní, tak heterogenní. Pomocí něho lze pak simulovat i další strukturované datové typy – množinu, multimnožinu, lineární seznam, stromové struktury atd.

Možností využití programovacího jazyka ve výuce se zabývala snad jen jediná práce (Vévoda, 2015), porovnávající možnosti dříve používaného jazyka Pascal a jazyka Lua. Po zhodnocení výhod a nevýhod využití jazyka Lua ve výuce nevylučuje, pokud se přijme

fakt absence striktní kontroly datových typů a dalších jevů souvisejících s odlehčením jazyka.

3 Obsah kursu

Tématické celky potřebné pro zvládnutí programovacího jazyka Lua se, jak je obvyklé, skládají z výkladové (levý sloupec) i z praktické části (pravý sloupec).

Programování v jazyce Lua

1. Úvod a popis jazyka Lua

- | | |
|--|---|
| <ul style="list-style-type: none">• historie• datové typy, hodnoty, výrazy• čísla, boolean, nil, řetězce | <ul style="list-style-type: none">• proměnné, přiřazení• konverze mezi datovými typy• vstup a výstup• matematické funkce |
|--|---|

2. Operátory

- | | |
|--|---|
| <ul style="list-style-type: none">• aritmetické operátory• bitové operátory• logické operátory• relační operátory | <ul style="list-style-type: none">• spojení řetězců• vyhodnocování výrazů• priorita operátorů |
|--|---|

3. Konstrukce programovacího jazyka Lua

- | | |
|--|---|
| <ul style="list-style-type: none">• podmínky a příkazy větvení• příkazy cyklu s podmínkou – příkazy while, repeat/until | <ul style="list-style-type: none">• příkaz cyklu for• zpracování dat |
|--|---|

4. Řetězce

- | | |
|--|--|
| <ul style="list-style-type: none">• operace s řetězci• kódování UTF-8• syntaktické zkratky | <ul style="list-style-type: none">• vzory pro hledání• captures |
|--|--|

5. Tabulky

- pole, asociativní pole
- přístup k prvkům asociativního pole
- řazení, modifikace řazení
- vytvoření tabulky
- zjištění počtu prvků
- operace s tabulkou

6. Funkce

- deklarace, volání, návratové hodnoty
- parametry, volitelné parametry
- rekurze
- rekurzivní algoritmy
- serializace, výpis struktury

7. Funkce

- funkce jako datový typ
- funkce předávané jako parametry
- iterátory, uzávěry
- využití vnějších podprogramů
- konstrukce vlastního iterátoru

8. Textové soubory

- popis a vlastnosti textových souborů
- operace s textovými soubory
- použití různých režimů zpracování textových souborů

9. Binární soubory

- popis a vlastnosti binárních souborů
- operace s binárními soubory
- konverze binárních dat na textová a opačně
- přímý přístup k datům

10. Moduly

- struktura a použití modulu
- diagram signatury
- návrh uživatelského abstraktního datového typu
- implementace abstraktního datového typu

11. Komunikace s OS

- příkazový řádek
- systémové proměnné
- volání příkazů
- datum a čas
- knihovna os
- zpracování konfiguračních souborů

12. Využití jazyka lua v aplikacích

- principy využití
- popis aplikací
- Lua a programovací jazyk C
- Lua v ConTeXtu
- využití při tvorbě her

4 Použité technologie

4.1 Verze programovacího jazyka

Verze programovací jazyka Lua není rozhodující pro zvládnutí základů tohoto jazyka. Lze však doporučit verze 5.3 a vyšší.

4.2 Vývojové prostředí

Před započítím výuky je potřeba rozhodnout o tom, které vývojové prostředí bude doporučeno, případně vyžadováno.

Volba vývojového prostředí je velmi často závislá na preferencích uživatele-programátora, ale i na operačním systému a možnostech instalace vývojových prostředí. Proto nelze dát jednoznačnou odpověď na to, který způsob práce je nejvhodnější. Omezíme se zde tedy jen na přehled možností a získané poznatky.

Jako zastánce využití volně dostupného programového vybavení v co nejširší míře budu jako první volbu vždy zkoumat, jaké možnosti použití nabízí **operační systém Linux**. Tento přístup naplňuje poslání projektu *Innovative Open Source Courses for Computer Science*, v němž syllabus kursu Programování v jazyce Lua vzniká, i v širším slova smyslu.

Pro použití pod OS Linux postačuje přítomnost libovolného textového editoru (joe, vim, emacs, nano, ...) a přítomnost interpretu jazyka Lua, obojí lze bez dalších komplikací spouštět z příkazového řádku, případně s možným využitím přesměrování standardního vstupu a výstupu.

Pro výuku lze též použít i některé **aplikace přístupné on-line** – ty však mají různé vlastnosti a ne vždy dané prostředí může vyhovovat dané situaci, ať již hovoříme opět o preferencích studentů, nebo o potřebách řešených úloh.

Zcela nekomfortním se projevil on-line překladač na stránkách www.lua.org. I přesto však bohatě postačuje pro demonstraci jednoduchých ukázek, například pro vyhodnocování výrazů. Nesporně jej lze doporučit také pro ověření funkčnosti kódu, neboť se

jedná o aplikaci z referenčních stránek programovacího jazyka Lua, na jejichž obsahu se podílejí autoři jazyka.

Pro operační systém **Windows** existuje několik integrovaných vývojových prostředí, které umožňují práci s řadou programovacích jazyků Lua. Ten však nebývá zařazen mezi výchozí instalované jazyky, a proto je potřeba jej instalovat dodatečně. I zde je potřeba rozvážit, které z integrovaných vývojových prostředí splňuje potřeby řešených úloh a poskytuje patřičné pohodlí, a podle toho přijmout rozhodnutí co doporučit studentům.

5 Diskuse a závěr

V kursu Programování v jazyce Lua, který je připravován v rámci mezinárodního projektu *Innovative Open Source Courses for Computer Science*, jsou využity zkušenosti z dlouholeté výuky programování v různých programovacích jazycích uskutečňované v rámci studia informatických předmětů na Provozně ekonomické fakultě Mendelovy univerzity v Brně. Navržený syllabus obsahuje nosná témata, která vedou studenta k pochopení programovací jazyka Lua, demonstrována na adekvátních sadách příkladů.

Omezená časová dotace pro běh kursu v rámci Letní školy neumožňuje se studenty probrat všechna připravená témata, i přesto tento první běh pracovní verze kursu umožnil získat ohlasy od studentů. Ač byly vesměs příznivé, řada studentů poskytla zajímavé postřehy a podněty, které budou po kritickém zhodnocení následně reflektovány jednak úpravou syllabu, jednak doplněním a úpravou připravovaných studijních materiálů.

Jedním z takových doplňků, který bude zařazen do náplně cvičení, se stane „vlastnoručně“ sestavená funkce pro vizualizaci datových struktur a v nich uložených hodnot, což se při pilotním běhu kursu ukázalo jako nezbytné pro zvýšení názornosti výkladu.

Návrh syllabu kursu předpokládá budoucí zařazení mezi vysokoškolské předměty vyučované pro informatické obory na Provozně ekonomické fakultě Mendelovy univerzity v Brně. Jeho první běh by se měl uskutečnit v letním semestru akademického roku 2022/2023.

Po zavedení kursu lze očekávat následující přínosy:

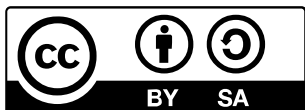
- využití nástroje, kterým lze efektivněji vysvětlit principy některých algoritmů;
- získání nástroje pro snadnou tvorbu jednoduchých programů pro zpracování dat, ať již pro vlastní potřebu, nebo pro více uživatelů;
- podporu pro studenty, kteří chtějí vyvíjet programové vybavení v jazyce Lua s využitím například v typografickém systému ConTeXt nebo pro vývoj her;
- zvýšení povědomí o oblasti open source a jejím využití v praxi.

Literatura

1. IERUSALIMSKY, R. *Programming in Lua*. Rio de Janeiro : Lua.org, 2016. 369 s. ISBN 978-85-903798-6-7.
2. HANUŠ, A. *Syntaktický analyzátor zdrojových textů ve formátu ConTeXt*. Diplomová práce. Brno : Mendelova univerzita v Brně, 2015. 83 s.
3. KAŠPAROVÁ, A. *Implementation of module drawing charts for ConTeXt* Bakalářská práce. Brno : Mendelova univerzita v Brně, 2020.
4. KOCUROVÁ, T. *Implementation of module drawing charts for ConTeXt* Bakalářská práce. Brno : Mendelova univerzita v Brně, 2020.
5. *Lua 5.4 Reference Manual [on-line]*. [Rio de Janeiro] : Lua.org, PUC-Rio, 2022. Dostupné na: <http://www.lua.org/manual/5.4/>.
6. MAKEŠ, D. *Detekce a korekce typografických jevů ve značkováném textu*. Diplomová práce. Brno : Mendelova univerzita v Brně, 2015. 81 s.
7. TREŠÁK, M. *Tvorba vexilografického modulu s využitím jazyků MetaPost a Lua*. Bakalářská práce. Brno : Mendelova univerzita v Brně, 2016. 50 s.
8. VÉVODA, P. *Programovací jazyk Lua a možnosti jeho využití v předmětu Programovací techniky*. Bakalářská práce. Brno : Mendelova univerzita v Brně, 2015. 70 s.
9. WIRTH, N. *Algoritmy a struktury údajov*. Bratislava : Alfa, 1988. 488 s.

Autor

RNDr. Tomáš Hála, Ph.D., Mendelova univerzita v Brně, Provozně ekonomická fakulta, ústav informatiky, Zemědělská 1, CZ 613 00 Brno, e-mail: thala@mendelu.cz



Open Access. This article is licensed under the terms of the Creative Commons Attribution-ShareAlike 4.0 International License, CC BY-SA 4.0 (<https://creativecommons.org/licenses/by-sa/4.0/>)