

# PROMĚNY PARADIGMATU PROGRAMOVÁNÍ

**Branislav Lacko**

[branislav.lacko@seznam.cz](mailto:branislav.lacko@seznam.cz)

## **ABSTRAKT:**

Příspěvek popisuje vývoj a změny paradigmatu programování v druhé polovině 20. století. V závěru příspěvek charakterizuje pravděpodobný vývoj na začátku třetího tisíciletí. Odborný termín „paradigma“ příspěvek chápe ve smyslu světa idejí platónovské filosofie jako stylový přístup k řešení určitého problému. Příspěvek navazuje na odborný blok příspěvků, který se na konferenci Tvorba softwaru 2009 zbyval minulým a budoucím vývojem programování

## **KLÍČOVÁ SLOVA:**

Programování, paradigma, druhá gramotnost

## **1 Úvod**

Příspěvek navazuje na diskuze o minulosti, současnosti a očekávaném vývoji programování, které proběhly na jubilejním 35. ročníku konference Tvorba software 2009 [viz příspěvky na konferenci 16,17,18 a diskuze k nim].

Diskutována byla zejména změny programová v plynulých letech, postavení programován dnes, očekávané trendy vývoje v programování, probematika výky programován ve školách apod.

Cílem tohoto příspěvku je přispět prostřednictvím analýzy vývoje paradigmatu programování k možnosti predikce změny přístupu k programování ve střednědobém horizontu deseti až dvaceti let.

## **2 Postupné změny paradigmatu programování**

### **2.1 Programování jako umění**

Monografie o programování Donalda Knutha [1] vyšla roce 1968, tedy v době, kdy se tranzistorové počítače v neočekávaném množství rozšířily do praxe. Tato skutečnost nastolila tehdy aktuální problém, jak nalézat a vytvářet složité algoritmy různých úloh, které bylo následně nutno popsat prostřednictvím programovacích jazyků do takové formy, aby mohly být vloženy do samočinných počítačů a řídily jejich práci. Byl to problém zcela nový, před touto dobou v takovém rozsahu a s takovým rozsahem neřešený. Proto není divu, že první kroky lidstva v oblasti programování byly spojeny s potřebou velké myšlenkové kreativity, invence a originálního přístupu k řešení problémů, které byly spojeny s programováním počítačů. Logickým důsledkem bylo přirovnání této nové tvůrčí oblasti k umění. Poznamenejme, že základní algoritmy v knize popsané, jsou aktuální ještě i dnes a proto brněnské nakladatelství Computer Press vydalo český překlad [9] publikace D. Knuta v roce 2008.

### **2.2 Programování jako disciplína**

Úspěšný nástup tisíců počítačů III. generace, využívajících vynálezu elektronických integrovaných obvodů, vytvořil počátkem sedmdesátých let tlak na zvýšení počtu programátorů a zvýšení produktivity jejich práce, aby mohly být vypracovány rozsáhlé a komplikované operační systémy, zajišťující jejich efektivní provoz, a aby mohla být uspokojena poptávka po aplikačních programech všeho druhu. Programování již nemohlo být

výsadou malé hrstky originálně myslících jedinců. Právě v těchto letech se objevil termín „softwarová krize“, jako charakteristika skutečnosti, že tvorba programů v té době začala zaostávat ve své produktivitě za produktivitou hromadné výroby počítačů elektronického průmyslu. Bylo potřeba vychovat desítky tisíc programátorů, schopných profesionálně produkovat požadovaný software. Nejprve N. Wirth odhalil roušku tajemna slova „program“ tím, že popsal program jako vzájemnou symbiózu exaktních algoritmů a numerických datových struktur [2] v roce 1973. Následně E. Dijkstra představil odborné i laické veřejnosti programování jako novou profesní disciplínu [3].

### 2.3 Programování jako systematický a strukturovaný postup

Praxe programování stále složitějších a stále rozsáhlejších programů ukázala, že takové programy nelze úspěšně a správně sestavit, pokud nejsou navrženy na základě systémového přístupu a nevyužívá se progresivních racionalizačních technik při jejich tvorbě. Již v roce 1972 zveřejnila trojice průkopníků strukturovaného přístupu programování O. J. Dahl, E. W. Dijkstra, C. A. R. Hoare publikaci [7], která v přelomu sedmdesátých a osmdesátých let zahájila období strukturovaného programování. V důsledku tohoto přístupu se následně rozšířily metody normovaného programování, logického programování a funkčního programování, které odrážely technologický přístup k podstatě tvorby programů. Tyto přístupy iniciovaly vznik řady nových programovacích jazyků např. Pascal, LISP, PROLOG, Ada a dalších. Principy a myšlenky strukturovaného programování rozpracovali další propagátoři tohoto přístupu Yourdon, Jackson a další. V této době se již projevila potřeba realizace důkladné strukturované analýzy a návrhu struktury programu před vlastním programováním. [14,15] Kromě publikací, které popisovaly principy a obecné zásady strukturovaného přístupu k analýze, začaly pod jejich vlivem vznikat velmi důkladné a konkrétní metody pro analýzu a návrh SSADM – Structured System Analysis and Design Method (GB), SDM – Structured Development Method (NL) a další.

### 2.4 Programování jako druhá gramotnost

Pokroky v oblasti polovodičové techniky umožnily vyrábět na začátku osmdesátých let hromadně levné a výkonné mikroprocesory. Konstrukcí osobního počítače s využitím šestnáctibitového mikroprocesoru Intel 8086 v roce 1981 odstartovala firma IBM éru hromadné výroby a hromadného používání počítačů typu PC. Ale již předtím pronikly osmibitové domácí počítače až dětem. Mikropočítač se začal stávat nezbytným pomocníkem při všech lidských činnostech. To způsobilo velký zájem široké veřejnosti o jejich programování. Televizní společnosti zařazovaly pořady o programování v jazyku BASIC pro osmibitové počítače mezi své pořady, řada časopisů zařadila seriály o programování do svých čísel, velkým nákladem vycházely desítky populárních počítačových časopisů a v knihkupectví nabízela populární publikace o mikropočítačích a programování, vznikaly kluby uživatelů mikropočítačů různých značek ZX Spectrum, Amstrad, Commodore, Amiga a dalších [16]. Bylo zřejmé, že obsluha PC a porozumění jejich programování se brzy stane nezbytnou součástí znalostí každého vzdělaného člověka. Na to upozornil ve svém příspěvku A. Ješov [5] na světové konferenci o počítačovém vzdělávání ve švýcarském městě Lausanne. Jím použitý slovní obrat, že programování se stane „second literacy“ (vedle schopnosti čtení a psaní) moderního člověka se okamžitě stalo symbolickým pro tehdejší chápání přístupu k programování.

### 2.5 Programování jako specializovaná a specifická profese založená, na vědeckých poznatcích

Koncem osmdesátých let již bylo nutné vypracovat pro programování teoretické základy a zahájit výchovu specializovaných pracovníků, kteří budou cílevědomě připravováni a vychováni pro vykonávání programátorské profese. Kniha D. Griese [4] formulovala

základy vědy o programování a vzápětí začaly vznikat specializovaný pracoviště označovaná názvy computer science, software engineering, informatics apod. Současně řada vysokých škol technického i univerzitního typu zavedla odpovídající studijní programy a ustavila ústavy a fakulty, zaměřené na vzdělávání v tomto směru. Předměty, ve kterých studenti získávali základní znalosti o programování a o počítačích, zařadily vysoké školy do svých prvních ročníků pro všechny studenty jak technických, tak humanitních oborů. Studium bylo dále stále rozšiřováno o moderní informační a komunikační technologie, které byly výsledkem intenzivního celosvětového výzkumu a vývoje.

## 2.6 Objektově orientované paradigma

Problémy, které byly způsobovány potřebou izolovaně analyzovat a specifikovat funkce, data a události při strukturované analýze, která musela předcházet strukturovanému programování, vedly nakonec ke vzniku objektově orientovanému paradigmatu [8]. Jeho vznik se datuje již začátkem sedmdesátých letech v souvislosti se vznikem takových programovacích jazyků jako SIMULA a SMALLTALK, ale převažujícím paradigmatem se stal až koncem devadesátých let. Je zajímavé, že zde bylo přímo použito označení object-oriented paradigma.

Je potěšitelné, že objektově orientovaný přístup je u nás dobře podchycen českou původní a kvalitní odbornou literaturou, kterou mohou zájemci použít pro seznámení s touto problematikou. Jedná se o publikace R.Pecinovského [19] a M.Viriuse [20] (Oba autoři napsali řadu knih o problematice objektově orientovaného programování včetně společné dvoudílné publikace [21]. Zde jsou vevedeny jejich publikace, které stály při nástupu OOP u nás). Problematika objektově orientovaného paradigmatu je u nás diskutována na pravidelných konferencích Objekty (viz např. stránky 15. ročníku této konference na Ostravské univerzitě <http://konference.osu.cz/objekty2010> ).

Využívání objektově orientovaného programování v současných programovacích jazycích jako Java, Visual Basic a dalších vyžaduje i přechod na nové metody práce v týmech, které vytvářejí objektově orientované produkty. Jedná se především o využívání agilních metodik při organizaci práce vývojových týmů software informačních systémů [22], ale také o postup modelování reality prostřednictvím objektového modelu [12] při objektově orientované analýze a objektově orientovaném návrhu aplikace [11] a o uspořádání jednotlivých fází celého projektu tvorby objektově orientované aplikace např. metoda BORM [13] jako česká alternativa metodiky RUP firmy IBM-Rational.

## 3 Závěr

Programování počítačů již zůstane zřejmě nerozlučně spjata ještě po dlouhou dobu s existencí lidské společnosti. Z hlediska pohledu softwarového inženýrství lze a vývoje paradigmatu programování je však možno předpokládat následující trendy (poznámka pod čarou – bez ohledu na důležitost a časovou následnost):

- Počet profesionálních systémových programátorů a specialistů pro aplikace vědeckotechnických výpočtů a aplikace hromadného zpracování, kteří ke konci XX. století převažovali, se bude relativně zmenšovat, zatímco poroste počet programátorů, kteří budou programovat specializované automatizované soustavy v real-time aplikacích v důsledku velkého počtu zabudovaných mikroprocesorů různých výrobků všeho druhu.
- Budou přibývat další programátoři, kteří budou programovat specializované systémy např. typu obráběcích strojů CNC nebo různých programovatelných automatů typu PLC. Počet takových zařízení neustále roste. Je zajímavé, že zatímco se v osmdesátých letech XX. století pořádaly soutěže v programování počítačů, po roce

2000 už k takovým soutěžím nedochází, ale na druhé straně v rámci „Akademie CNC obrábění“ na stránkách magazínu „Technický týdeník“ se od roku 2009 pořádají soutěže programátorů CNC obráběcích strojů ([www.techtyden.cz](http://www.techtyden.cz)) a v rámci akce „Automatizace PANASONIC do škol“ soutěží žáci v řešení různých úloh pro PLC (viz [www.panasonic-electric-works.cz](http://www.panasonic-electric-works.cz)).

- V nejbližší době, zřejmě v horizontu let 2015 až 2020 let, se dostane do popředí problematika programování inteligentních automatů, zejména jak průmyslových, tak humanoidních robotů, a jiných automatizovaných soustav s umělou inteligencí.

Zejména programování prostředků umělé inteligence přinese podstatnou změnu paradigmatu programování. Tento očekávaný přístup k programování bychom mohli snad nejlépe charakterizovat jako programování cílů a jejich dosahování soustav s inteligentním chováním – tedy ovlivňování stojů s umělou inteligencí. Tedy nikoliv počítačů s von Neumanovou koncepcí. Tomuto paradigmatu budou zřejmě nejlépe vyhovovat problémově orientované, deklarativní, dialogové jazyky, využívající mluveného slova a prostorových vizualizačních prostředků 3D grafiky ve spojení s teach-in postupy.

### Seznam literatury:

- 1 Knuth, E. D.: The art of computer programming (vol. 1 & 2). Addison-Wesley, Reading Mass., 1968
- 2 Wirth, N.: Algorithm + Data Structures = Programs. Prentice Hall 1973 Englewood Cliffs (překlad - Wirth N.: Algoritmy a struktury údajov, Alfa 1988)
- 3 Dijkstra W. E.: Discipline of programming. Prentice Hall 1976 Englewood Cliffs
- 4 Gries, D.: The science of programming. Springer-Verlag 1981
- 5 Ershov, A.: Programming, the second literacy. In: Proceedings of 3rd IFIP World Conference on Computers in Education (WCCE81), Lausanne, 1981
- 6 Virius, M.: Programování dnes a zítra. Softwarové noviny, roč.IX.,(1998),č.5, str.28
- 7 O.-J. Dahl, E. W. Dijkstra, C. A. R. Hoare: Structured Programming. Academic Press, London, 1972
- 8 Abadi, M. - Cardelli, L.: A Theory of Objects. Springer-Verlag New York 1996
- 9 Donald E. Knuth: Umění programování. Computer Press 2008 Brno
- 10 peci 8 bit
- 11 Duben, J.: Objektové modely podniku. Grada Publishing Praha 1996
- 12 Merunka, V.: Objektové modelování Nakladatelství ALFA 2008 Praha
- 13 Polák, J.-Merunka,V.-Carda,A.: Umění systémového návrhu. Grada 2003 Praha
- 14 Langefors,B.: Theoretical Analysis of Information Systems, Studentlitteratur AB 1973
- 15 Chandor,A.-Graham,J.-Williamson,R.: PracticalSystems Analysis. Hart-Davis Educational Publishing 1972 London
- 16 Pecinovský, R.: Historie mimoškolní výuky programování u nás. Sborník konference Tvorba softwaru 2009. VŠB-TU Ostrava 2009 Ostrava, str. 129 – 133
- 17 Virius, M.: Výuka informačních technologií na vysokých školách u nás. Sborník z konference Tvorba softwaru 2009. VŠB- TU Ostrava 2009 Ostrava, str. 156 - 161
- 18 Lacko, B.: Programování a informační gramotnost ve znalostní společnosti. Sborník Konference Tvorba softwaru 2009. VŠB-TU Ostrava 2009 Ostrava, str. 58 - 66
- 19 Pecinovský, R.: OOP – Naučte se myslet a programovat objektově. Computer Press
- 20 M. Virius: Pasti a propasti jazyka C++, Grada, Praha 1997
- 21 R. Pecinovský, M. Virius: Objektové programování (I. a II.), Grada Praha 1996
- 22 Buchalcevoá, A.: Metodiky vývoje a údržby informačních systémů. Grada 2005 Praha

Přílohy: Obsahy výše uvedených publikací přibližují jejich zaměření.

## Příloha 1

Knuth, E. D.: The art of computer programming (vol. 1 & 2).

Addison-Wesley, Reading Mass., 1968

Chapter outline of published and unpublished volumes

([http://en.wikipedia.org/wiki/The\\_Art\\_of\\_Computer\\_Programming](http://en.wikipedia.org/wiki/The_Art_of_Computer_Programming)  
on line -February2011 ):

- Volume 1 - Fundamental Algorithms
  - Chapter 1 - Basic concepts
  - Chapter 2 - Information structures
- Volume 2 - Seminumerical Algorithms
  - Chapter 3 - Random numbers
  - Chapter 4 - Arithmetic
- Volume 3 - Sorting and Searching
  - Chapter 5 - [Sorting](#)
  - Chapter 6 - [Searching](#)
- Volume 4 - [Combinatorial](#) Algorithms
  - Volume 4A - [Enumeration](#) and [Backtracking](#)
    - Chapter 7 - Combinatorial searching
  - Volume 4B - [Graph](#) and Network Algorithms, in preparation
    - Chapter 7 *continued*
  - Volume 4C and possibly 4D - [Optimization](#) and Recursion, in preparation
    - Chapter 7 *continued*
    - Chapter 8 - [Recursion](#)
- Volume 5 - Syntactic Algorithms, planned (as of 2011, *estimated* in 2020).
  - Chapter 9 - [Lexical scanning](#)
  - Chapter 10 - [Parsing](#) techniques (includes also string search and data compression)
- Volume 6 - Theory of [Context-Free Languages](#), planned.
- Volume 7 - Compiler Techniques, planned

## Příloha 2

Wirth, N.: Algorithm + Data Structures = Programs. Prentice Hall 1973 Englewood Cliffs  
(on line Wikipedia - [Niklaus Wirth](#))

**Chapter outline:**

- Chapter 1 - Fundamental [Data Structures](#)
- Chapter 2 - Sorting
- Chapter 3 - Recursive Algorithms
- Chapter 4 - Dynamic Information Structures
- Chapter 5 - Language Structures and [Compilers](#)

Publikováno:

Lacko,B.: Transformation of programming paradigm.

In: Preceedings 37th National Conference with Interantional Participation  
SOFTWARE DEVELOPMENT 2011, TU of Ostrava 2011, p.59 – 63