

5GL: Pátá generace programovacích jazyků

5GL (Fifth-Generation Language) jsou jazyky založené na řešení problémů pomocí **omezení a logického odvozování**, nikoliv pomocí algoritmů napsaných programátorem. Zatímco **4GL** byly navrženy pro práci s daty, 5GL byly vytvořeny pro oblast **umělé inteligence**, expertních systémů a přirozeného jazyka.

Základní myšlenkou je, že počítač by měl být schopen vyřešit problém sám, pokud mu dodáme správná fakta a pravidla.

1. Klíčové vlastnosti 5GL

- **Deklarativní programování:** Programátor popisuje stav světa a pravidla, která v něm platí.
- **Inference (Odvozování):** Jazyk obsahuje „intelektuální motor“ (inference engine), který kombinuje fakta a pravidla k nalezení odpovědi.
- **Práce s vědomostmi:** Místo databází pracují 5GL se **znalostními bázemi**.
- **Přirozený jazyk:** Mnohé 5GL se snaží o to, aby komunikace s počítačem připomínala běžnou lidskou řeč.

2. Hlavní zástupci a koncepty

A. Prolog (Programming in Logic)

Nejznámější zástupce 5GL. Program se skládá z faktů a pravidel. * **Fakt:** `rodic(karel, jan).` (Karel je rodičem Jana) * **Pravidlo:** `predek(X, Y) :- rodic(X, Y).` (X je předkem Y, pokud X je rodičem Y) * **Dotaz:** `?- predek(karel, jan).` → Systém odpoví `Yes`.

B. LISP (List Processing)

I když je starší, jeho schopnost pracovat s kódem jako s daty a jeho využití v rané umělé inteligenci ho často řadí do diskuzí o 5GL konceptech.

C. OPS5 a Mercury

Jazyky používané pro tvorbu expertních systémů (programy, které simulují rozhodování lidského experta v určitém oboru, např. v medicíně).

3. Historický kontext: Projekt páté generace

V 80. letech 20. století zahájilo Japonsko masivní státní projekt „Fifth Generation Computer Systems“ (FGCS). Cílem bylo vytvořit počítače, které by se podobaly superpočítačům a komunikovaly by v

přirozeném jazyce pomocí logického programování. Projekt však narazil na tehdejší limity výkonu hardwaru a na fakt, že logické programování se ukázalo jako příliš rigidní pro reálný svět.

4. Srovnání generací (Shrnutí)

Generace	Typ	Přístup k programování
1GL / 2GL	Nízká úroveň	Přímé řízení hardwaru (binární kód, assembler).
3GL	Vyšší úroveň	Definice algoritmu krok za krokem (Jak to udělat).
4GL	Velmi vysoká úroveň	Definice datového dotazu (Co chci získat).
5GL	Inteligentní úroveň	Definice pravidel a logiky (Jaká jsou fakta).

5. Současnost a budoucnost 5GL

Dnešní vlna umělé inteligence ([Deep Learning](#)) se od klasického 5GL (logiky) odklonila směrem ke statistice a neuronovým sítím. Nicméně principy 5GL stále žijí v:

- **Sémantickém webu:** Organizace informací tak, aby jim stroje rozuměly.
- **Constraint Programming:** Řešení složitých logistických úloh (např. jízdní řády).
- **Moderních expertních systémech:** Diagnostika a právní analýzy.

Související články:

- [Čtvrtá generace: SQL a data](#)
- [Deep Learning a neuronové sítě](#)
- [Expertní systémy a báze znalostí](#)

Tagy: programming_languages 5GL prolog ai logic_programming history

From:
<https://serviceit.cz/> - **IT ENCYKLOPEDIE**

Permanent link:
<https://serviceit.cz/doku.php?id=5gl>

Last update: **2026/01/02 13:22**

