

Třetí generace jazyků (3GL)

Třetí generace programovacích jazyků (3GL) jsou tzv. **vyšší programovací jazyky**. Jsou navrženy tak, aby byly snadno čitelné pro člověka pomocí anglických klíčových slov (jako ``if``, ``while``, ``for``, ``print``). Na rozdíl od **strojového kódu** a **assembleru** jsou tyto jazyky **hardwarově nezávislé**.

1. Hlavní přínosy 3GL

Před nástupem 3GL musel programátor přesně vědět, jak procesor manipuluje s registry a pamětí. 3GL přinesly revoluci v několika směrech:

- **Abstrakce:** Programátor pracuje s proměnnými a datovými typy místo přímých adres v paměti.
- **Přenositelnost (Portability):** Zdrojový kód napsaný v 3GL lze s minimálními změnami přeložit a spustit na různých architekturách (např. x86 i ARM).
- **Strukturované programování:** Zavedení logických bloků a funkcí, což umožnilo psát mnohem složitější a udržitelnější software.
- **Produktivita:** Jeden řádek v 3GL může odpovídat desítkám instrukcí v assembleru.

2. Překlad kódu: Kompilace vs. Interpretace

Počítač stále rozumí pouze nulám a jedničkám. Kód v 3GL proto musí být převeden do strojového kódu jedním ze dvou způsobů:

1. **Kompilace:** Celý zdrojový kód je najednou přeložen kompilátorem do spustitelného souboru (`.exe``). Příklad: **C, C++, Rust**.
2. **Interpretace:** Interpret čte kód řádek po řádku a okamžitě ho vykonává. Příklad: **Python, JavaScript**.
3. **Hybridní přístup:** Kód je přeložen do mezistupně (bytecode), který běží ve virtuálním stroji. Příklad: **Java, C#**.

3. Významní zástupci a jejich éry

Období	Jazyky	Hlavní přínos
50. - 60. léta	FORTRAN, COBOL, LISP	První vědecké výpočty, obchodní data a základy AI.
70. léta	C, Pascal, Ada	Systémové programování, strukturovaný návrh, efektivita.
80. - 90. léta	C++, Java, Python	Objektově orientované programování, internetové aplikace.
Současnost	Rust, Go, Swift	Bezpečnost paměti, moderní paralelizace a výkon.

4. Srovnání čitelnosti

Podívejte se, jak se zápis zjednodušil od 2GL k 3GL:

2GL (Assembler - specifické pro procesor):

```
MOV AL, 10
ADD AL, 20
```

3GL (Jazyk C - univerzální):

```
int vysledek = 10 + 20;
```

5. Co následovalo?

Zatímco 3GL definovaly **jak** má počítač úkol vykonat (procedurální přístup), další generace se posunuly dál:

- **4GL (Čtvrtá generace):** Jazyky zaměřené na to, **co** se má udělat (např. SQL pro databáze).
- **5GL (Pátá generace):** Jazyky pro umělou inteligenci založené na řešení omezení a logice (např. Prolog).

Související články:

- [Druhá generace: Jazyky symbolických adres](#)
- [Programování v C++](#)
- [Kompilátor vs. Interpret](#)

Tagy: *historie programming_languages 3GL abstraction software_development*

From:

<https://www.serviceit.cz/> - IT ENCYKLOPEDIE

Permanent link:

<https://www.serviceit.cz/doku.php?id=it:lang:3gl>

Last update: **2026/01/02 13:21**

