MI-SP-SP-19 (MI-RUN)

Bytekód, typické instrukce, struktura, implementace interpretru bytekódu, Just-In-Time překlad

- instrukční sada navržená pro rychlé zpracování interpretrem
  - zdrojový kód moc složitý → rychlost
  - strojový kód závislý na architektuře → přenositelnost
- sekvence bytů následovaných případnými parametry

typické instrukce:
- ▷ práce se zásobníkem (push, pop)
- ▷ aritmetické a logické (add, negate, less-than, ...)
- ▷ volání metod a návraty (invoke, call)
- ▷ skoky a podmíněné skoky (jump, jump-if-true)
- ▷ vytváření objektů & konverze typů

→ zpracování interpretrem (Python, Perl)
- někdy kompiluje na bytekód sám interpretr, jindy před tím manuálně (Java)
  - případně se interpretuje přímo zdrojový kód

(+) rychlejší vývoj, kompatibilita, správa paměti (např. možnost přesouvat bloky paměti → řešení fragmentace)
(jednoduchost implementace interpretru vs kompilátoru)

JIT - dynamický překlad kódu za běhu programu (na strojový kód)
- přeložené části se ukládají do cache
- překládají se části často používané a náročné (nemusí celý program)
  - už mohou být známé některé proměnné, díky kterým budeme lépe optimalizovat