**Programovací jazyky - vlastnosti, srovnání, popis způsobu tvorby i běhu programů**

**Co jsou to programovací jazyky?**

Jazyky pro tvorbu počítačových programů

* Sada instrukcí pro počítač pro vyřešení problému

Jazyky se liší syntaxí, paradigma, abstrakcí, oblasti použití a způsobem překladů

**Co znamená umět programovat?**

* Schopnost přemýšlet abstraktně a logicky
* Schopnost spolupráce v týmu
* Schopnost dohledání problému
* Schopnost nacházet postupy za účelem řešení problému
* Schopnost formulovat postupy v daném programovacím jazyce
* Znalost syntaxí vůbec nedělá dokonalého programátora

**Jakým jazykem mluví počítač?**

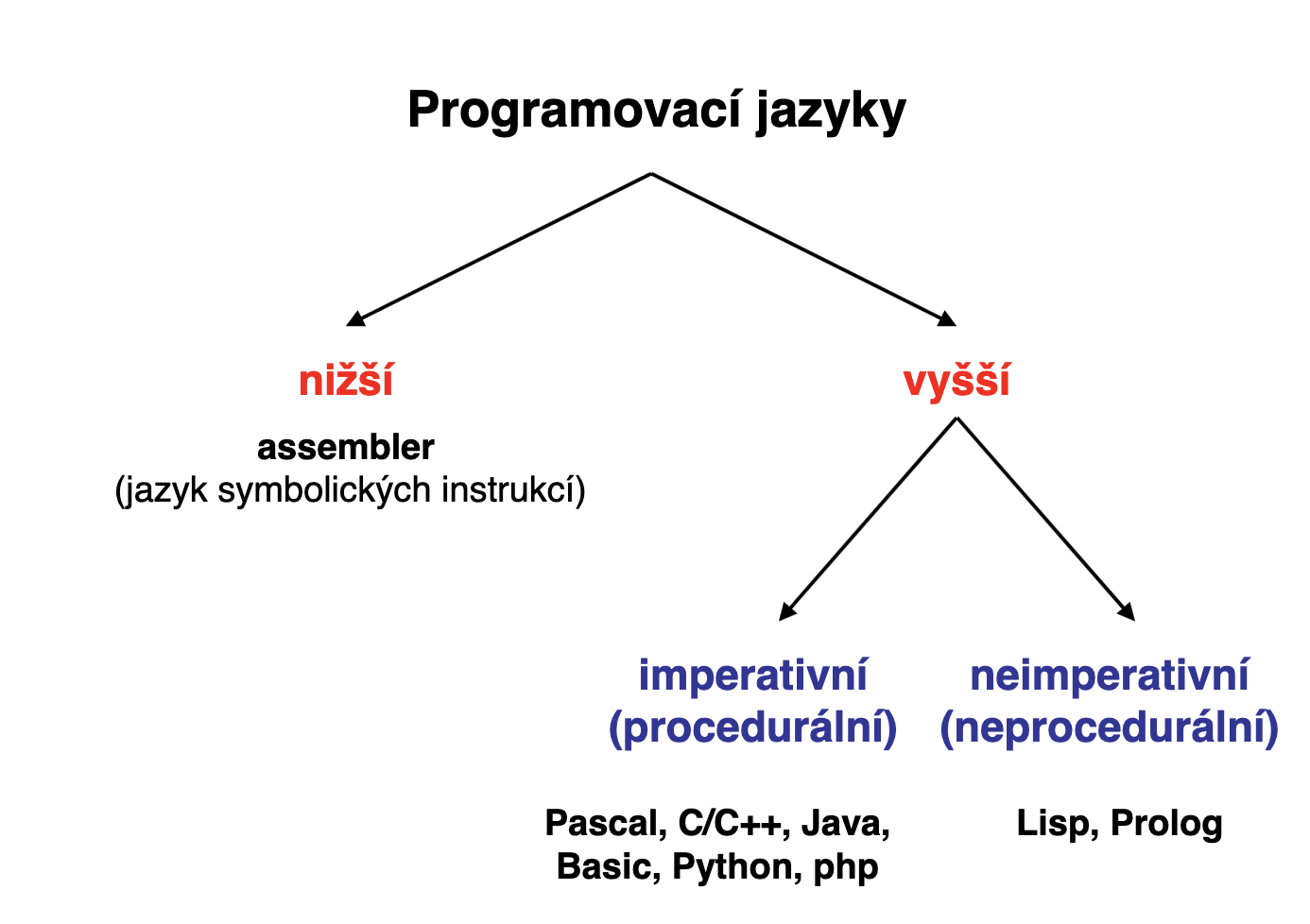
* **Strojový kód** – posloupnost čísel
* V programování se většinou kód píše v jazyce, který je pro počítač nesrozumitelný
  + Proto je potřeba provést překlad zdrojového kódu programu do strojového kódu procesoru (**KOMPILACE**)
    - pomocí překladače nebo interpretru
    - Pro každý programovací jazyk je potřeba jiný překladač/interpretr.

**Dělení podle kompilace:**

* **Kompilátor**
  + Vytvoří soubor obsahující program ve strojovém kódu
  + Překládá celý zdrojový kód najednou
    - Tím vytvoří výsledný soubor se strojovým kódem
    - Strojový kód, pokud kompilace proběhla úspěšně, je hned přímo spouštěn počítačem (CPU)
  + Výsledný zkompilovaný program je často rychlejší, než kompilovaný za běhu ale vyžaduje více času na zapnutí
  + **Při vyskytnutí chyby**
    - vyhodí chybu při kompilaci programu – je to hned vidět
  + **PŘ. Překladač**
    - Překladač dostane text na přeložení
    - Text přeloží a uvede výsledek překladu
  + C, C++, COBOL
  + Komplexní kód může zabrat hodně času zkompilovat
  + Musí být přeložen pro každou platformu zvlášť
  + Vyžaduje méně paměti
    - program je UŽ uložen
* **Interpret**
  + Překládá kód postupně za běhu programu
    - Nevytváří výsledný spustitelný soubor
    - Pracuje s jednotlivými příkazy v reálném čase
  + Zapnutí programu je rychlejší ale program je většinou pomalejší
  + **Při vyskytnutí chyby**
    - vyhodí chybu až za běhu programu – je to vidět až za běhu programu a v části, kde se ta chyba vyskytla
    - Program spadne
  + **PŘ. Tlumočník**
    - Člověk řekne větu
    - Tlumočník přeloží
    - Opakování akce do kola, dokud dialog neskončí
  + PHP, JavaScript, Python, BASIC
  + Je snadněji přenositelný
  + Potřebuje více paměti
    - Potřebuje paměť pro běh INTERPRETU

**Dělení podle abstrakce:**

* **Vyšší programovací jazyky** (většina)
  + Zápis se podobá myšlení člověka (anglicky, matematicky)
  + Tento zápis se **kompiluje do strojového kódu** (HEX)
    - Počítač nerozumí původnímu zápisu
  + **Procedurální (IMPERATIVNÍ)** (C, C++, Java, Python, PHP)
    - Zápis je posloupnost příkazů, cyklů a podmínek
    - Využití proměnných
    - **Objektově orientované**
      * Základní jednotkou programu je objekt
    - **Strukturované**
      * Např. C
      * Programování pomocí
        + Cyklů
        + Větvení (podmínky)
        + Struktury PODFUNKCÍ
  + **Neprocedurální** **(NEIMPERATIVNÍ)** (Haskell)
    - **Místo tvoření přesného postupu tvoříme pouze jeho cíl, čeho má dosáhnout**
    - **Funkcionální**
      * Program sestaven z matematických funkcí
      * Hodně blízké k matematice
    - **Logické**
      * Program sestaven z matematické logice
* **Nižší programovací jazyky**
  + **Strojově orientované**
  + **Příkazy jazyka = instrukce procesoru**
  + Dělí se na generace
  + Nesrozumitelný/těžko srozumitelný pro člověka zápis (srozumitelný pro počítače)
  + Příklad
    - jazyk symbolických adres (**ASSEMBLER**)
  + Pracuje v BIN
  + Můžeme manipulovat s HEXADECIMÁLNÍ částí (stejně nesrozumitelná)
  + Assembler – zjednodušení – kompiluje do HEX



**Dělení podle typování**

**Statický**

* Dat. typ určen při deklaraci
* V průběhu programu se dat. Typ nemůže měnit
* Výhoda
  + Při špatném dat. typu – vyhodí error
* Nevýhoda
  + Komplikace psaní kódu
  + Obcházení změn
* C#, JAVA

**Dynamický**

* Lze dynamicky měnit datový typ
* Dat. Typ určen hodnotou
* Měněn za běhu programu
* Nelze kontrolovat při kompilaci
  + Až v průběhu se zjistí

**Jazyky s virtuálním strojem**

**Co je virtuální stroj?**

SW, který simuluje počítačový HW

**Proč?**

Aby mohl být programovací jazyk spuštěn na různých platformách bez KOMPILACE PRO KAŽDOU PLATFORMU

* Je to kompilátor
  + Kompiluje do MEZIKÓDU (je binární)
  + Daný mezikód je spustitelný na jakékoliv platformě na virtuálním stroji
* JAVA, C#, Python, Ruby, JavaScript

**Neexistuje jazyk, který by byl optimální pro VŠE**

**Jazyky jsou většinou zaměřeny na oblasti**

**Python**

* Práce s daty
* Finance
* Analytika
* Umělá inteligence
* Jednoduché syntaxe
* Flexibilita
* Facebook, Netflix

**Java**

* SW v bankovnictví
* Průmysl

**Assembler**

* Nejblíže k HW
* Programy přímo řídící HW
* Kernel
* C (mezi nižším a vyšším)