

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

Programovací jazyky obecně

Programovací jazyky slouží pro tvorbu programů. Pod programem si můžeme představit sadu instrukcí pro počítač, které volá, aby vyřešil nějaký určitý úkol - algoritimizace problémů. Je to prostředek komunikace mezi programátorem a počítačem

Vlastnosti programovacích jazyků jak se liší

Syntaxe - způsob psaní kódu (gramatika)

Míra abstrakce - vzdálenost jazyka od HW

Paradigma - způsob organizace kódu

Způsob překladu - interpretovaný/kompilovaný/bytecode

Způsob užití

Rozdělení podle úrovně (míra abstrakce)

Nižší

Nižší jazyky pracují blízko HW, nemají komplexní syntax, pracují přímo s instrukcemi CPU, instrukce jsou ukládány jako sekvence bitů, mají výhodu v tom že jsou velmi rychlé a efektivní ale jsou poměrně náročné na psaní a vyžadují detailní znalost funkčnosti CPU, firmware, drivers....

Assembler

Vyšší

Srozumitelnější pro nás pro lidi, umožňují pracovat s komplexnějšími koncepty, potřebují překlad do strojových instrukcí, automatizované spravování paměti (garbage collector) a knihovny

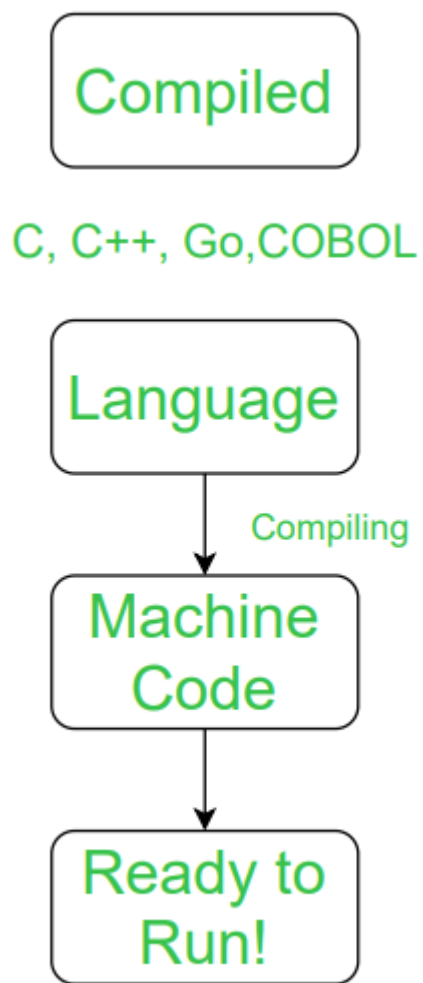
Python, Java, C#

Způsoby překladů

Kompilované jazyky

Zdrojový kód se překládá přes kompilátor do strojového kódu, program se přeloží celý před jeho spuštěním, výsledkem je spustitelný soubor, jsou poměrně rychlé a odhalují chyby při překladu, nevýhodou kompilování po každé změně a jsou závislé na platformě a typu procesoru nebo OS.

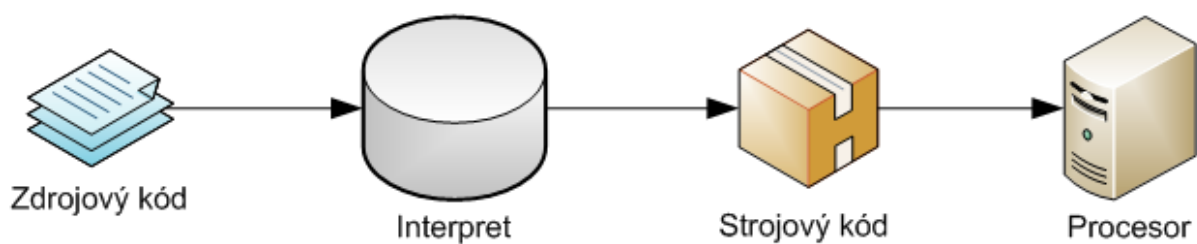
C, C++ atd..



Interpretované jazyky

Kompiluje jenom určité části a kompiluje je během běhu jak jsou potřeba, výhodou je že je přenositelný mezi platformami a nevyžaduje datové typy (mohou být za běhu programu měněny), nevýhodou je pomalý běh a to že chyby se objevují až za běhu

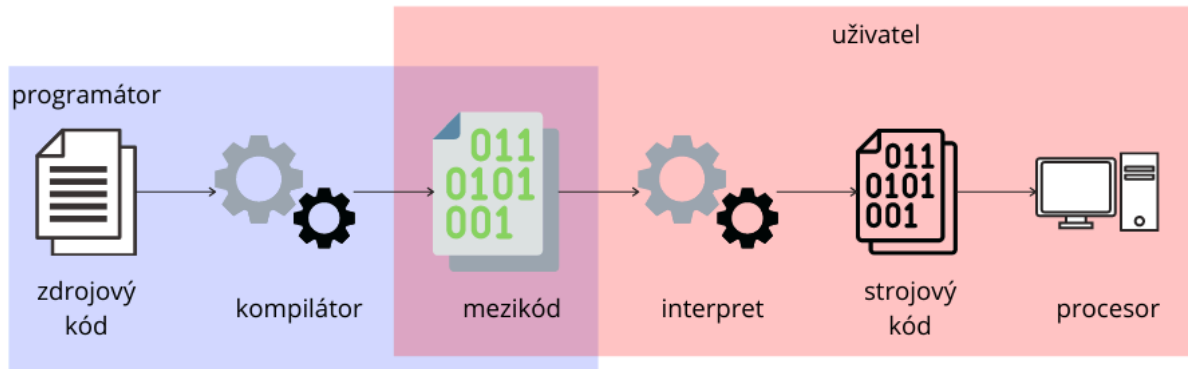
JavaScript, PHP atd...



Jazyky s virtuálním strojem (bytecode)

Kód se kompiluje do mezikódu (bytecode) běžící na VM, kombinuje oba dva způsoby, výhoda je taky přenositelnost a jednoduché odhalování chyb a používá knihovny pro vývoj, je méně výkonný

Java (JVM), C#(.NET)



Dělení podle paradigmat

Paradigmata jsou základní programovací styly - organizace a strukturovanost kódu, dělí se na procedurální a neprocedurální

Procedurální

kód je organizován do funkcí nebo procedur, přesný postup kroků, používá strukturované řízení toku programu (cykly, větvení...), OOP a strukturované programování, C, Python, Java....

Neprocedurální

popisují co chceme dosáhnout ne jak toho dosáhnout, zaměřují se na cíl, systém sám rozhoduje o způsobu provedení, funkcionální a logický, SQL, HTML, CSS,

funkcionální - program z matematických funkcí

logický - program z logických výroků

Dělení podle účelu

Každý programovací jazyk je na něco dobrý a na něco horší než ostatní, neexistuje nějaký jazyk tak univerzální aby byl dobrý v podstatě na vše

Python - AI, analytika, práce s daty, věda....

Java - bankovní systémy, databázové systémy, webové stránky

Assembler - programování hardwaru

...

Datové typy

Statické datové typování

datový typ je určen při deklaraci, v průběhu programu nelze změnit, špatný typ = spadne program

Java a C#

Dynamické datové typování

proměnná se může měnit, nevýhoda během kompilace - nevíme kde je chyba pořádně

Python

Syntaxe

bloky kódu - někde se používají závorky, někdy dvojtečky atd....