Programovací jazyky

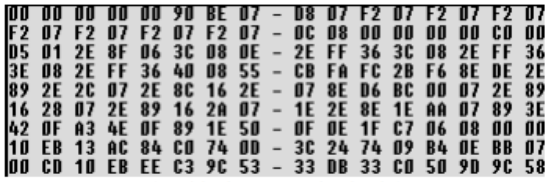
- co to je program, zdrojový kód
- co to je algoritmus, způsoby zápisu algoritmů, vlastnosti algoritmů
- rozdělení prg. jazyků: imperativní (procedurální) a deklarativní (neprocedurální)
- rozdělení prg. jazyků: kompilující a interpretující
- JAVA platforma
- událostmi řízené programování
- syntaxe a sémantika
- vysvětlit předložený zdrojový text v C#

Program

- **Je v informatice postup operací, který popisuje realizaci dané úlohy**
- Zápis algoritmů pomocí příkazů určitého programovacího jazyka
- **Počítačový program** (též jen program, obecně pak software) je v informatice **posloupnost instrukcí (ne nutně strojových instrukcí), která popisuje realizaci dané úlohy počítačem.** Aby počítač mohl vykonávat nějakou činnost, potřebuje mít ve své operační paměti alespoň jeden program

Zdrojový kód

- Zdrojový kód nebo zdrojový text je v informatice **označení zápisu textu počítačového programu v některém programovacím jazyce, který je uložen v jednom nebo více textových souborech.**
- Zdrojový kód obvykle programátor zapisuje pomocí textového editoru, ale může být též generován specializovaným programem.
- Textový editor může být součástí integrovaného vývojového prostředí (IDE), které programátorovi tvorbu zdrojového kódu usnadňuje a poskytuje mu další podporu:
  - zvýraznění syntaxe
  - vyznačení syntaktických chyb
  - nápověda
  - seznamy funkcí
  - příklady
  - generování zdrojového kódu
  - přímý přístup k navazujícím nástrojům (vyvolání kompilátoru, možnost krokování a sledování průběhu programu pomocí debuggeru, vytváření souborů pro řízení překladu - Makefile, zpracování dokumentace a podobně).....
- **Strojový kód** je v informatice posloupnost strojových instrukcí prováděných procesorem počítače. Strojové instrukce jsou zapsány formou číselných kódů.
  - Obecně řečeno strojový kód jsou příkazy napsané tak aby jim rozuměl procesor a mohl požadovanou akci vykonat. Strojový kod je ovšem pro člověka velmi nepřehledný, matoucí a složitý, neboť je to jen posloupnost číslic.



(Příklad strojového kódu)

Z tohoto důvodu se strojový kód k programování používá jen zřídka.

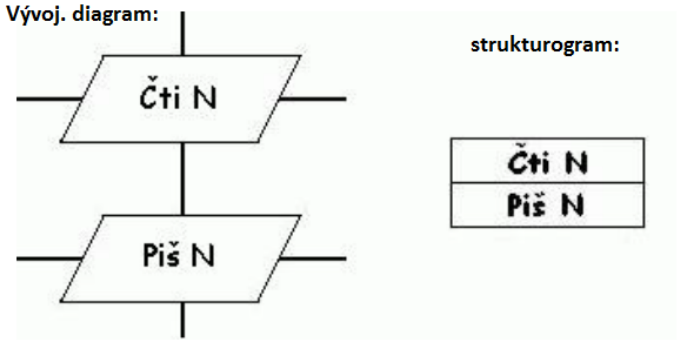
Instrukce zobrazené jako čísla v šestnáckové soustavě.

Algoritmus

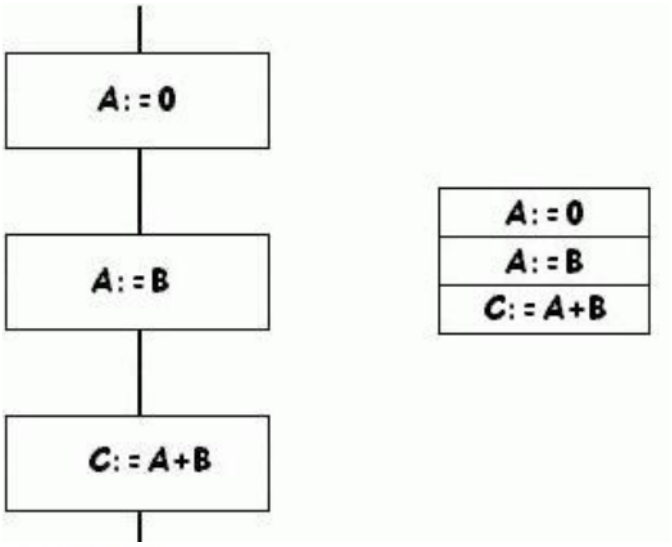
- je přesný návod či postup, kterým lze vyřešit daný typ úlohy. - jinak řečeno **je to posloupnost operací, která řeší daný úkol.**
- Myslí se jím teoretický princip řešení problému, oproti přesnému zápisu v konkrétním programovacím jazyce. Obecně se ale algoritmus může objevit v jakémkoli jiném odvětví. Jako jistý druh algoritmu se může chápat i např. kuchařský recept.
- **Vlastnosti algoritmu**
  - **konečnost - každý algoritmus musí skončit v konečném počtu kroků.** Tento počet kroků může být libovolně velký (podle rozsahu a hodnot vstupních údajů), ale **pro každý jednotlivý vstup musí být konečný.**
  - **obecnost (hromadnost, masovost, univerzálnost)** - algoritmus **neřeší jeden konkrétní problém, musí řešit danou úlohu pro různé vstupní hodnoty**
  - **determinovanost** - každý krok algoritmu musí být **jednoznačně a přesně definován**. V každé situaci musí být naprosto zřejmé, co a jak se má provést, jak má provádění algoritmu pokračovat.
  - **opakovatelnost** - pro stejné vstupy dostaneme pokaždé stejné výsledky.

- **Způsoby zápisu algoritmu**
  - **slovní vyjádření**
  - **grafické vyjádření**
    - ->**Strukturogram** je algoritmus přepsaný do tabulkového způsobu seřazení příkazů pod sebou v návaznosti tak, aby se po přepsání do programovacího jazyka dosáhnul požadovaný výsledek.
    - ->**vývojový diagram** je postup řešení určité úlohy lze zapsat pomocí vývojového diagramu. Ten se skládá ze značek, do kterých se zapisují jednotlivé příkazy při postupu řešení určitého úkolu. Tento postup řešení se nazývá algoritmus.

1. Vstup - výstup dat

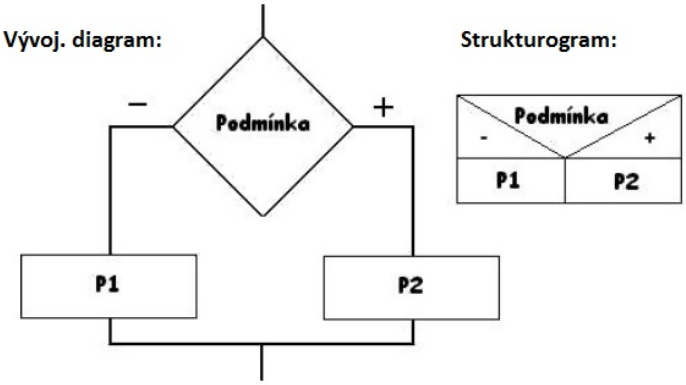


2. Přiřazení

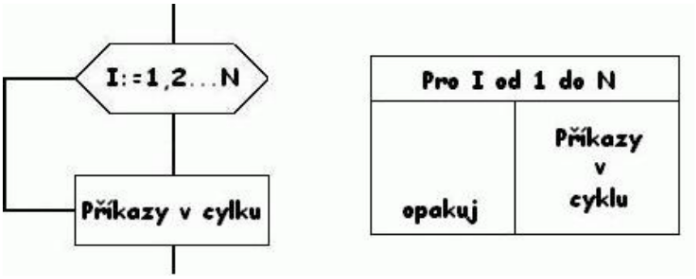


3. Rozhodování

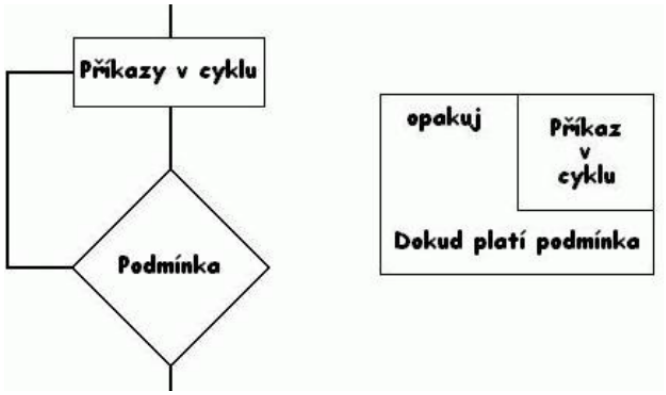
Vývoj. diagram: **Strukturogram:**



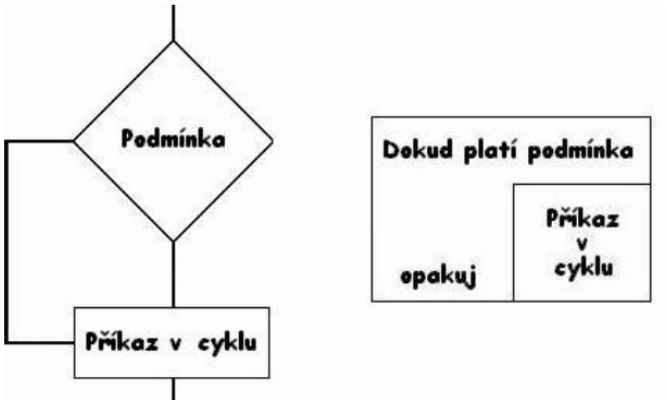
4. Cyklus řízený indexem



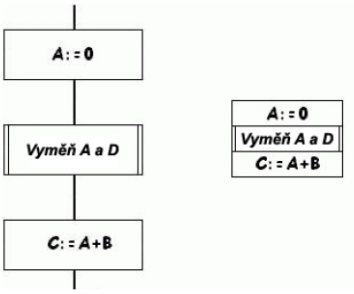
5. Cyklus s podmínkou na konci



6. Cyklus s podmínkou na začátku



7. Procedura (podprogram)
   Příklad: zavolání externí části algoritmu, která zajistí výměnu obsahu dvou proměnných A a D)



Rozdělení programovacích jazyků

Jazyky se dělí na:

- **Imperativní (procedurální)**
  - Program se zapisuje v podobě programových struktur, používá proměnné a datové struktury. Program je složen z příkazů, které krok po kroku vycházejí z algoritmu řešení.
    - Příklady: Pascal Delphi Visual Basic, C++, C# Object C, Java, Visual Basic for Aplication - VBA, PHP, JavaScript, Python
- **Deklarativní (neprocedurální)**
  - Program neříká jak se problém krok po kroku řeší, ale pouze definuje požadavek na výsledek.
    - Příklady: SQL (jazyk nad databází), XML (definiční jazyk pro výměnu datových struktur)

dále na

- **Kompilované**
  - Výsledkem je zpravidla strojový kód. Vyjímku tvoří např. jazyk Java - ten se sice kompiluje, ale pouze do podoby tzv. byte-code (posloupnost instrukcí pro jiný program (interpret), nejedná se o instrukce pro procesor). Byte-code je poté dále interpretován pomocí interpretu (v případě Javy v JVM - Java Virtual Machine). Zdrojová data (tj. zdrojový text, reference na knihovny, definice objektů UI apod.) jsou nejprve převedena do strojového kódu kompilátorem, potom se až spouští přeložená aplikace, nebo knihovna.
    - Příklady: Pascal Delphi (objektový Pascal), Visual Basic, C++, C# (všechny tři jazyky patří do skupiny jazyků .NET, ale kompilátory C++ existují samozřejmě i v jiných podobách i pro jiné platformy, než je MS Windows), Object C, Java (kompiluje do ByteCode – bajtový mezikód, který je spouštěn – interpretován v prostředí virtuálního stroje Javy, v tzv. JVM)
- **Interpretované**

- Program je překládán až v rámci vlastního spuštění, přičemž uložen je výhradně v podobě zdrojového textu. K provozu vyžaduje interpreter příslušného jazyka.
  - Příklady: Visual Basic for Aplication – VBA (makrojazyk v MS Office a dalších aplikacích), PHP (pro tvorbu webových aplikací), JavaScript (klientský skriptovací jazyk, pro tvorbu dynamických webových stránek), Python

a na konec ještě na

- **Vyšší**
  - Prakticky všechny dostupné nestrojově orientované jazyky. Programuje se na základě textových zápisů příkazů a vět.
- **Nižší**
  - Strojově orientované programy. Velmi primitivní jazyky, u nichž klíčová slova přímo zastupují instrukce procesoru.
    - Příklad: různé typy assembler

Java platforma

- Platforma Java je počítačová platforma (pracovní prostředí) zastřešující různé varianty použití programovacího jazyka Java pro vývoj a provoz různých typů aplikací.
- **Proč JAVA?**
  - Tisíce hotových komponent a zjednodušení, které je možno použít nebo je možno se jimi inspirovat
  - Dostupnost zdarma, velká komunita vývojářů, stabilní podpora
  - Osvědčená platforma prověřená velkými firmami
  - Vysoce výkonná platforma pro realizaci a běh výkonných řešení

Událostmi řízené programování

- Takto naprogramované programy nepracují tak, že *postupně* vyžadují nějaká data a nakonec vypíšou nějaký výsledek. Po spuštění (např. textového editoru) nabídnou své funkce a *čekají na události* - na které tlačítko klikneme, kterou nabídku vybereme, jakou klávesu či klávesovou zkratku stiskneme.
- Při definování způsobu chování programované aplikace používáme tzv. událostmi řízené programování, když zobrazíme formulář, provede se událost Load formuláře, když klikneme na tlačítko, provede se událost Click tlačítka apod. definujeme tedy, co se má provést když nastane nějaká událost
  - Např.: JAVA, Visual studio

Sémantika, syntaxe

sémantika jazyka = význam jednotlivých symbolů

- Z nepochopení významu příkazů jazyka vznikají chyby sémantické (logické), které se často projeví až při vlastním běhu programu. Zdrojový text v mnoha případech jde přeložit (zkompilovat), ale výsledný program dělá něco jiného než má. Odladění tohoto typu chyb se prování díky tzv. debuggeru (součást vývojových prostředí).

- Příklad chyb:
  - Dělení nulou
  - Nekonečný cyklus
  - Chyba formátu vstupu (program předpokládá číselný vstup a uživatel zadá nečíselný formát)

syntaxe jazyka= pravidla jejich spojování

- Při zapsání nepovoleného spojení příkazů, případně při uvedení neznámého či nepovoleného symbolu, vzniká tzv. syntaktické chyba. Tato chyba je detekována při vlastní kompilaci zdrojového textu. Zdrojový text obsahující syntaktické chyby nejde přeložit do spustitelného kódu.

Etapy programátorské práce

- **Nápad** , nadšení, velké plány, představení problému
- **Analýza problému** - se ukazuje jako složitější, vystřízlivění, podrobení problému důkladné analýze, vypracování základního algoritmu řešení, vybrání programovacího jazyka
- **Programování** - programátoři zapisují algoritmy v programovacím jazyce
- **Ladění** - nalezení a oprava chyb v programu
- **Používání** - vlastní využívání programu
- **Modifikace, aktualizace** - úprava, vylepšení a rozšíření verze programu

--Trnka.vaclav 24. 4. 2012, 21:44 (CEST)