1. Tvorba programu – algoritmizace a programování

## **Algoritmus**

* Přesný návod či postup, kterým lze řešit daný typ úlohy
* Princip řešení problému

### **Vlastnosti algoritmů**

* **Elementárnost** – Algoritmus se skládá z konečného počtu co nejjednodušších jednotlivých kroků
* **Konečnost** – Každý algoritmus musí skončit po určitém počtu kroků / v reálném čase
* **Hromadnost (Obecnost)** – Neřeší jeden konkrétní problém ale množinu, využití proměnných
* **Determinovanost** – V každém kroku algoritmu víme co bude následovat, stejné vstupy stejný výsledek
* **Rezultativnost** – Algoritmus má vždy výsledek

**Vlastnosti algoritmů z praxe (Kuchařka – dort)**

* **Elementárnost** – jdeme krok za krokem, máme přesně daný postup kroků
* **Konečnost** – víme že po určitém počtu kroků dojdeme na konec receptu
* **Hromadnost** – Umíme změnit přísady (místo jahodové marmelády švestková) ale i tak vznikne dort
* **Determinovanost** – V každém kroku víme co nás čeká dále a co máme použít za přísady
* **Resultativnost** – Výsledkem bude vždy hotový dort

### **Algoritmu lze vyjádřit:**

* **Slovně** – sepsání postupu větami/slovy v mluveném jazyce (kuchařka)
* **Graficky** – za pomocí vývojových diagramů, grafické značky (návod na sestavení nábytku)
* **Matematicky** – soustavou rovnic, vztahy mezi veličinami
* **Programem** – kroky popsány funkcemi procesoru

### **Značky vývojového diagramu:**

Začátek / konec Cyklus s určitým počtem opakování 

Běžný příkaz Cyklus s podmínkou na začátku

Ruční vstup Cyklus s podmínkou na konci

Zobrazení výstupu Spojka

Podmínka Spojovací čára

Vlastní funkce

### **Efektivnost algoritmu:**

* **Časová** – úloha řešena co v nejkratším čase
* **Paměťová** – co nejnižší zátěž paměti
* **Přehlednost, srozumitelnost** – pro další vývoj a úpravy

## **Algoritmizace**

* Proces vytváření a sestavování algoritmů

### **Kroky algoritmizace**

1. **Formulace problému**
   * Přesná formulace požadavků
   * Určení výchozích hodnot a výsledků
2. **Analýza úlohy**
   * Ověření srozumitelnosti
   * První představy o řešení
   * Zamyšlení nad více řešeními
   * Výběr nejvhodnějšího řešení
3. **Vytvoření algoritmu**
   * Ověření řešitelnosti
   * Navrhnutí vývojového diagramu
4. **Sestavení programu**
   * Problém naprogramujeme
5. **Odladění programu**
   * Odstranění chyb z programu
   * Debug
   * Optimalizace

## **Programování**

* Řešení algoritmů pomocí výpočetních technik ke spuštění počítačového programu

### **Nižší programovací jazyky**

* Instrukce odpovídají příkazům procesoru
* Závislé na procesoru, nepřenositelné
* Strojově orientované
* Rychlé
* Složité programování, hůře srozumitelný zápis
* Jádra operačních systémů
* **Strojový kód** – soubor číslicových instrukcí (.exe)
* **Jazyk symbolických adres (Assembler)**

### **Vyšší programovací jazyky**

* Formální jazyky (definovány matematicky)
* Strukturovaný zápis
* Nezávislé na procesoru
* Dělení: imperativní, neimperativní

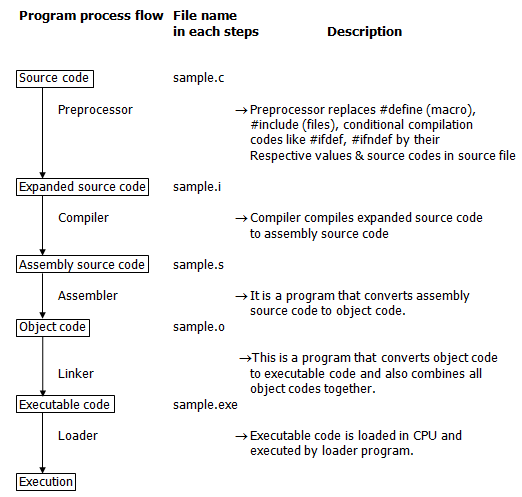
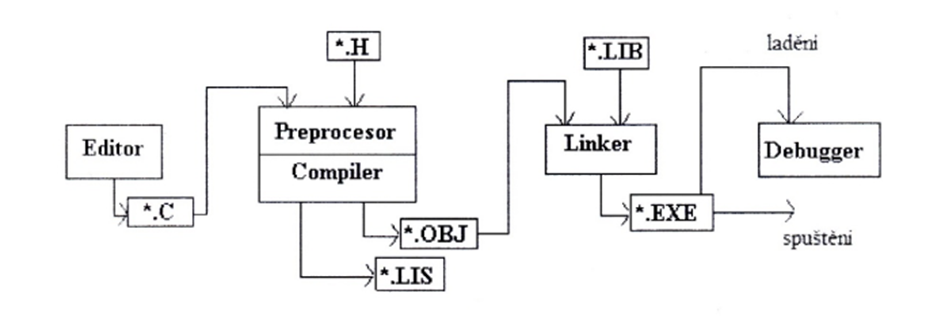
### **Syntaxe**

* Definuje pravidla zápisu programu

### **Sémantika**

* Určuje význam zápisu
* (cykly prováděny tak dlouho dokud podmínka platí)

## **Zpracování programu**



**Editor**

* tvorba zdrojového souboru (.c)

### **Preprocesor**

* součást překladače
* předzpracovává zdrojový soubor
  + vynechání komentářů
  + rozvoj maker
  + správné vložení hlavičkových souborů (.h)

### **Kompilátor (překladač)**

* překládá zdrojový soubor do relativního (objektového) kódu počítače - jazyka relativních adres (.o)
* Adresy proměnných a funkcí ještě nejsou vůbec známy
* Soubor (.lis) je na přání
* Protokol o překladu a obsahuje informace o chybách nalezených překladačem

### **Linker (sestavovací program)**

* přidělení relativních adresám absolutní adresy a provede všechny odkazy na dosud neznámé identifikátory
* připojí knihovny (.a, .o)
* výstupem je spustitelný (.exe) soubor

### **Debugger (odvšivovač)**

* ladící program
* slouží pro laděni, nalézání chyb, které nastanou při běhu programu
* po nalezení chyby se celý proces opakuje tak dlouho, až si myslíme, že je náš program bez chyb