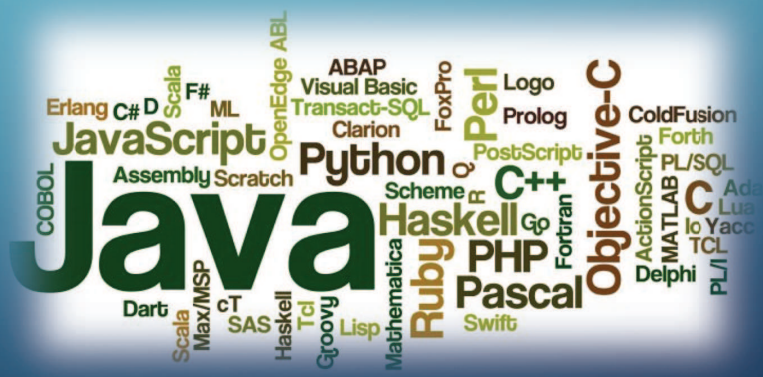
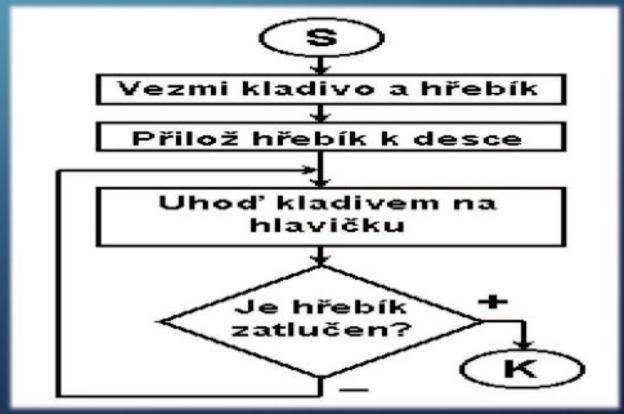
ALGORITMIZACE A PROGRAMOVÁNÍ

TERCIE



ALGORITMUS

je přesný návod či postup, kterým lze vyřešit daný typ úlohy. Nejčastěji se vyskytuje při programování, ale může se objevit v jiném vědeckém i nevědeckém odvětví (recept v kuchyni).



VLASTNOSTI ALGORITMŮ

1. **KONEČNOST** – algoritmus musí skončit v konečném počtu kroků

2. **OBECNOST (univerzálnost)** - algoritmus neřeší jeden konkrétní problém (např. „jak spočítat $\frac{3}{7} + \frac{1}{2}$ “, ale obecnou třídu problémů (např. „jak spočítat součet dvou zlomků $\frac{a}{b} + \frac{c}{d}$“)

3. **DETERMINOVANOST** – každý krok algoritmu musí být jednoznačně a přesně definován

4. **VÝSTUP (resultativnost)** – ze vstupních údajů algoritmus tvoří správný výstup

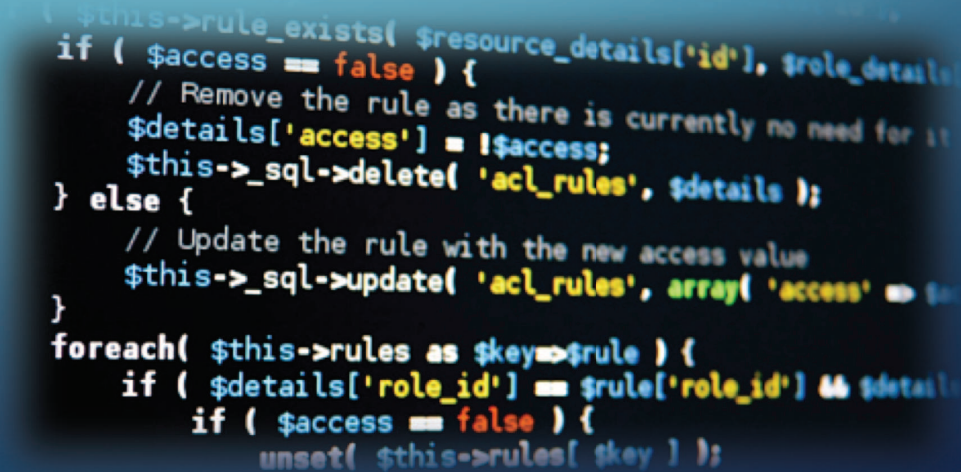
PROGRAMOVACÍ JAZYK

je prostředek pro zápis algoritmů, každý příkaz v programovacím jazyku má jasně definovaný význam (viz 3. vlastnost algoritmů).

Příklady: C, C++, C#, Java, Swift, Basic, PHP, Pascal, Fortran, Python, Karel…

PROGRAM

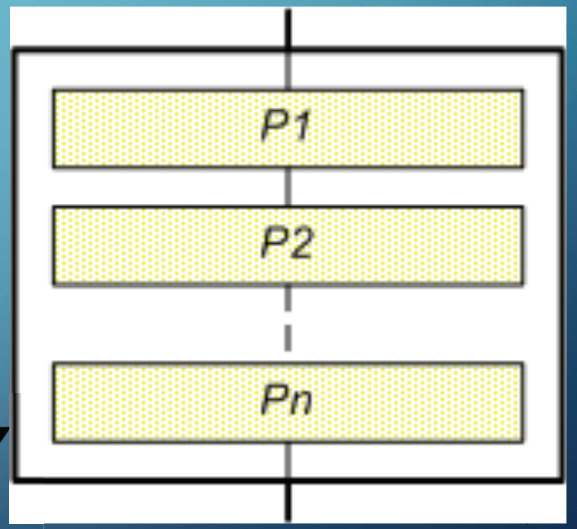
je zápis algoritmu v nějakém programovacím jazyce; je vytvořen programátorem.



ZÁKLADNÍ STRUKTURY V PROGRAMOVACÍCH JAZYCÍCH

1. POSLOUPNOST PŘÍKAZŮ

Příkazy P_1, P_2, …, P_n se provádí postupně jeden po druhém (po sobě)

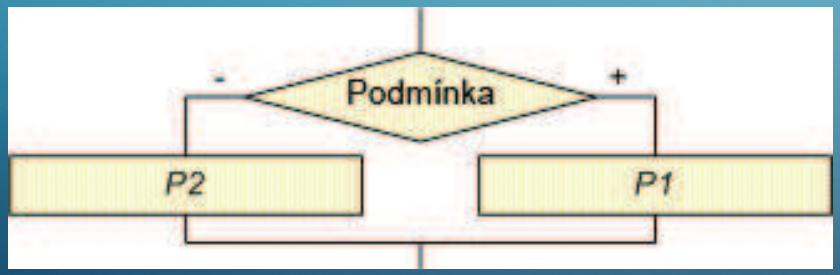


Celý blok posloupnosti příkazů, budeme dále označovat P (jeden nebo více příkazů)

2. PODMÍNKA (VĚTVENÍ)

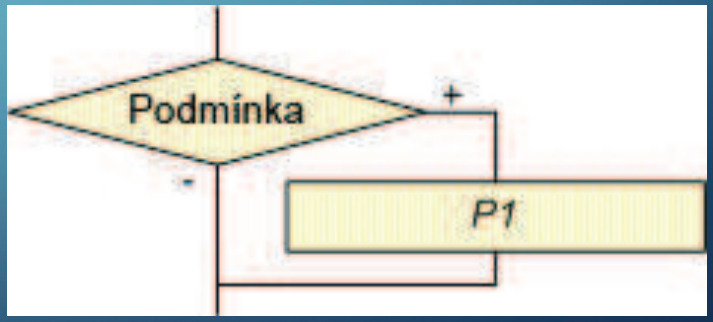
a) ÚPLNÁ PODMÍNKA

Vyhodnotí se podmínka (např. x>2), pokud je splněna (ANO, +), provede se příkaz P_1, pokud není splněna (NE, –) provede se příkaz P_2



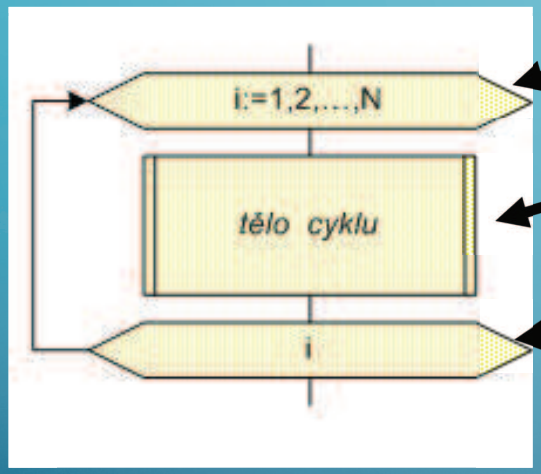
b) NEÚPLNÁ PODMÍNKA

Vyhodnotí se podmínka (např. a<b), pokud je splněna (ANO, +), provede se příkaz P_1, pokud není splněna (NE, –) neprovede se nic a pokračuje se dále v běhu programu



3. CYKLUS (OPAKOVÁNÍ)

a) S předem známým počtem opakování (cyklus „for“)

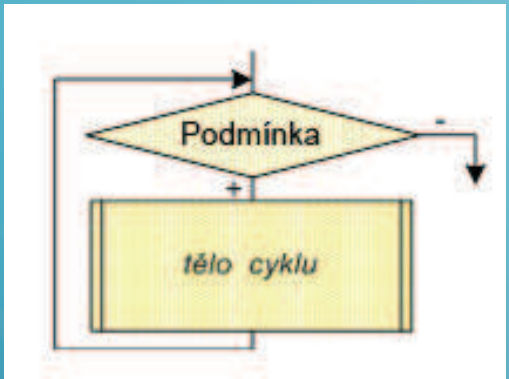


Parametr i nabývá postupně hodnot 1, 2, 3, … n

Tělo cyklu se tedy provede n-krát

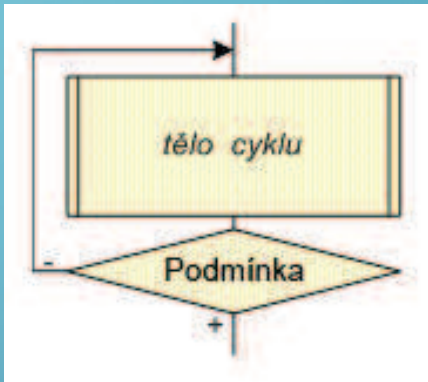
Na tomto místě se parametr i „zvedne“ o 1

b) Cyklus s podmínkou na začátku (cyklus “while“)



Vyhodnotí se podmínka, pokud je splněna (+), provede se příkaz (nebo blok příkazů). Poté se vrátí opět na vyhodnocení podmínky. Toto se stále opakuje a cyklus se u končí ve chvíli, kdy podmínka není (-) splněna.

c) Cyklus s podmínkou na konci (cyklus “repeat“)



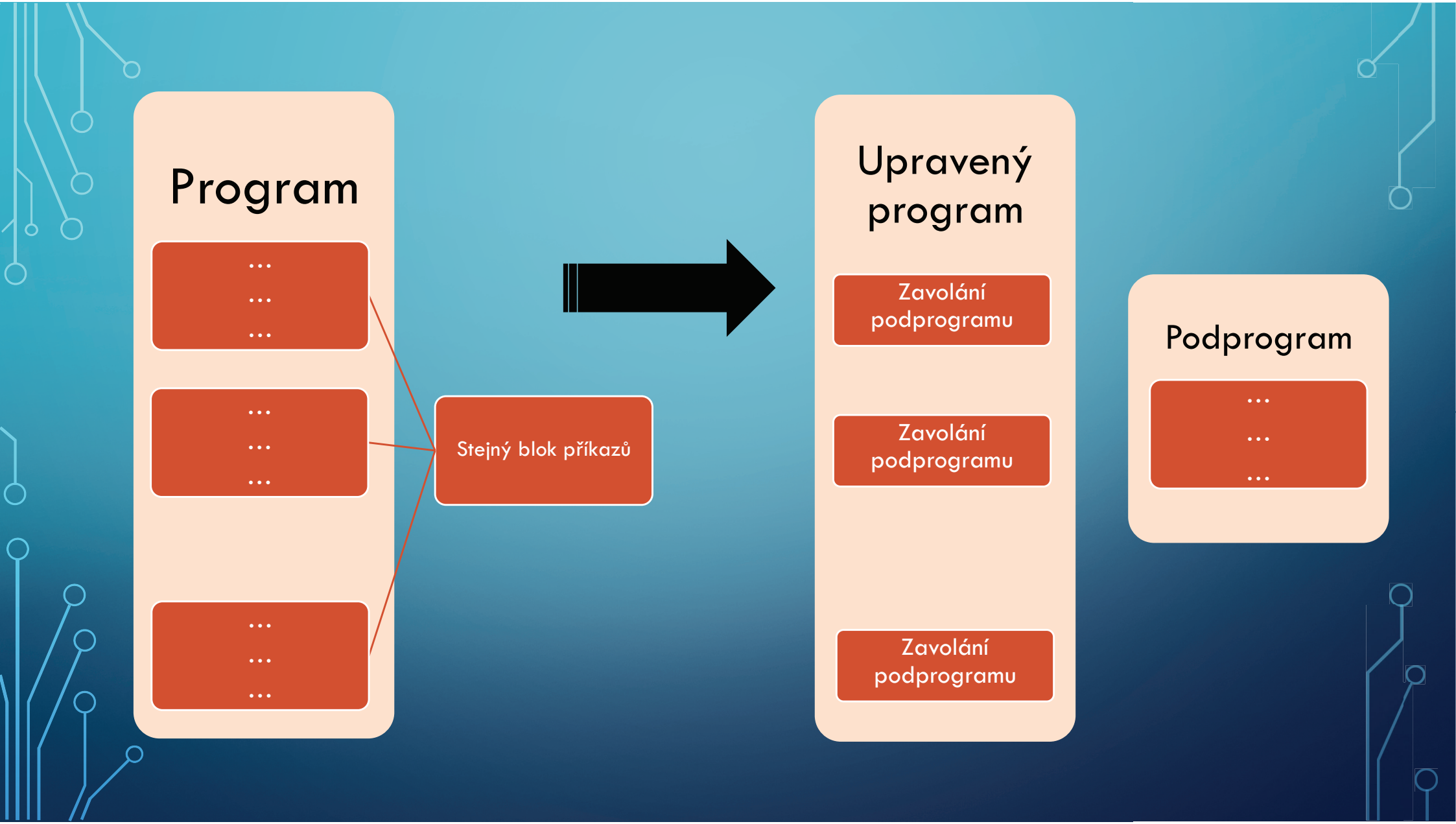
Nejdříve se provede tělo cyklu (cyklus se minimálně jednou provede), poté se vyhodnotí podmínka, pokud je splněna (+), tak se cyklus ukončí. Pokud není splněna podmínka, tak se opět provede tělo cyklu.

4. PODPROGRAMY

Program je výhodné rozdělit na menší části (menší celky), tzv. podprogramy. Podprogram má svůj název a pokud ho chceme v hlavním programu použít, tak ho "zavoláme" tímto názvem.

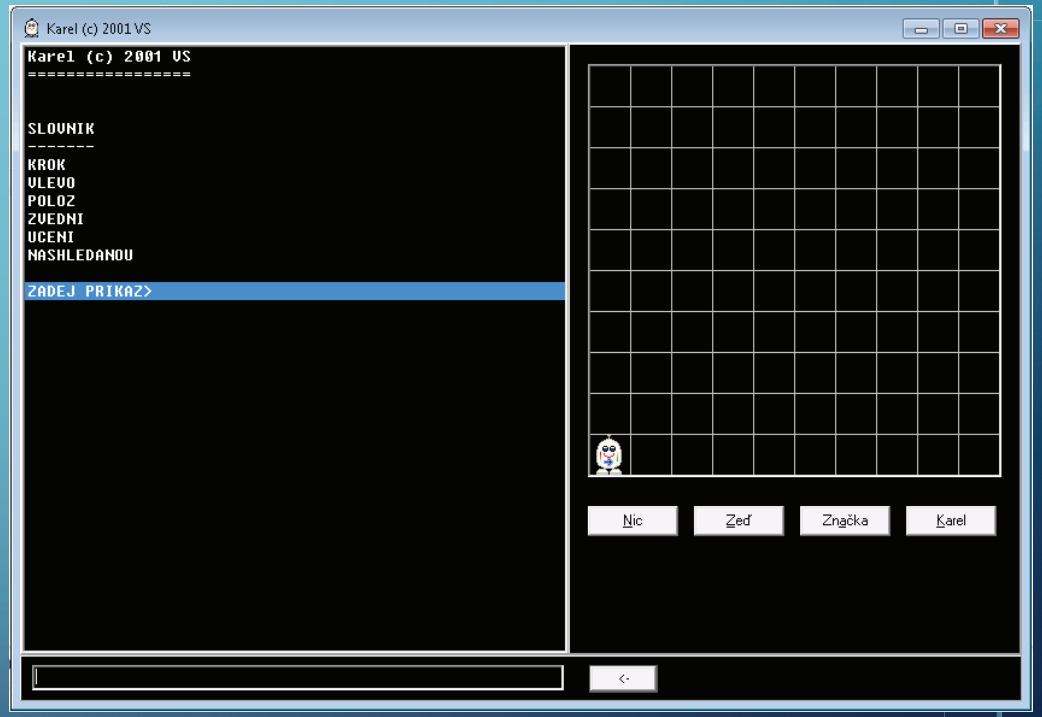
Výhody:

- Přehlednost hlavního programu
- Pokud se provádí nějaká činnost v programu na více místech – zkrátí se tím kód programu
- Lépe se hledají chyby
- Možnosti týmové spolupráce na vývoji programu


Program
...
...
...
...
...
...
...
...
...
Stejný blok příkazů
Upravený program
Zavolání podprogramu
Zavolání podprogramu
Zavolání podprogramu
Podprogram
...
...
...

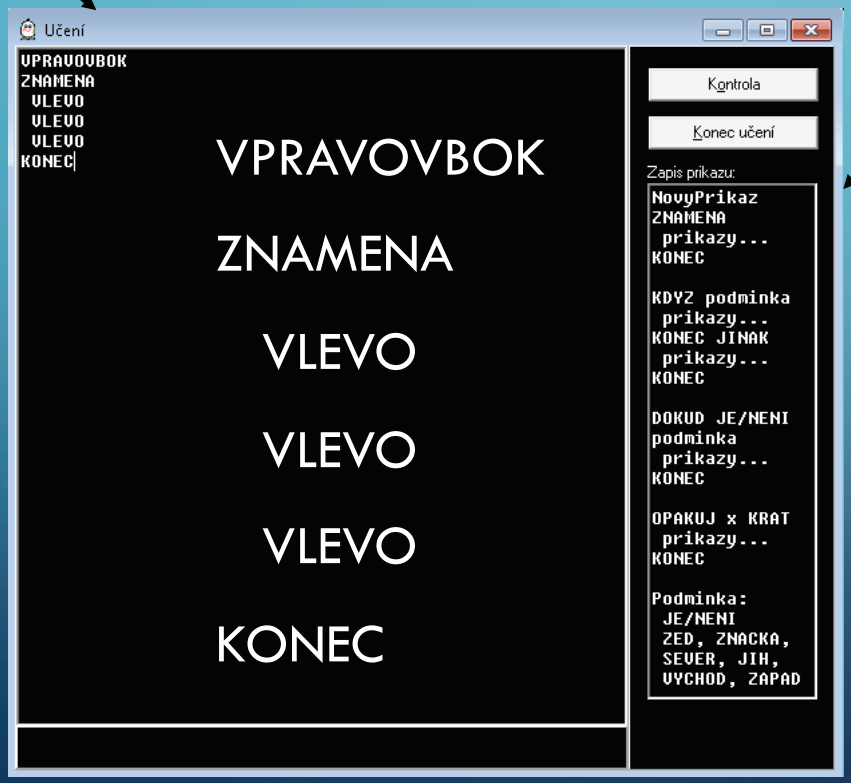
PROGRAMOVACÍ JAZYK KAREL

- Řídíme (programujeme) pohyb robota Karla
- Karel zná základní příkazy (má je „od výroby“) – KROK, VLEVO, POLOZ, ZVEDNI, UCENI, NASHLEDANOU
- Vše ostatní ho „učíme“ pomocí těchto základních příkazů – vytváříme podprogramy



TVORBA PODPROGRAMU V JAZYKU KAREL

V módu učení



Nápověda – jak zadat nový příkaz, podmínku, cyklus