

# **Algoritmy**

opakování

# Algoritmus

- algoritmus
  - postup (předpis) pro řešení nějakého problému, formálně (strukturovaně) zapsaný
  - jde o posloupnost nějakých **elementárních kroků**
  - všichni je znáte:
    - postup pro výpočet kořenů kvadratické rovnice
    - kuchařka: recepty
  - ten, kdo vykonává (provádí) algoritmus, je **procesor** , určuje, jaké budou el. kroky

- příklad:
  - recept na bábovku
    - procesorem je *kuchař*, elementární kroky jsou instrukce typu např. odvaž 0,5 kg hladké mouky
  - algoritmus pro počítač pro výpočet kořenů kvadratické rovnice
    - procesorem je *procesor počítače*, elementární kroky jsou instrukce daného procesoru
- napsat program znamená vymyslet algoritmus (tj. postup, jak problém řešit) a zapsat jej v nějakém programovacím jazyce

Příklad:

## Algoritmus pro výpočet kořenů kv. rovnice

1. je-li  $a=0$ , krok 8
2.  $D = b^2 - 4ac$
3. je-li  $D < 0$  krok 6
4. Rovnice má dva reálné kořeny ...
5. Konec
6. Rovnice má dva komplexní kořeny ...
7. Konec
8. atd.

# Vlastnosti algoritmů

- elementární
  - skládá se z konečného počtu jednoduchých kroků
- deterministický
  - po každém kroku lze rozhodnout, zda proces skončil
- konečný
  - počet kroků algoritmu je konečný
- hromadný
  - lze jej použít pro řešení více podobných úloh

## *Základní komponenty algoritmu:*

- algoritmus „tvoříme jen ze tří komponent“:
  1. posloupnost (sekvence) příkazů
    - kroky v daném pořadí
  2. větvení (podmínka)
    - výběr dalších prováděných kroků závisí na splnění/nesplnění nějaké podmínky
  3. cyklus
    - opakované provádění kroků
      - s pevným počtem opakování
      - s podmínkou (na konci/na počátku) - provádění kroků se opakuje, dokud je/není splněna podmínka

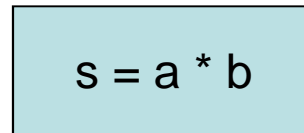
# Zápis algoritmu

- **vývojové diagramy (grafický zápis)**
  - jednotlivé kroky jsou vyjádřeny graf. symboly dle typu operace,
  - posloupnost vykonávání kroků je vyjádřena orientovanou hranou
  - de facto *orientovaný graf*

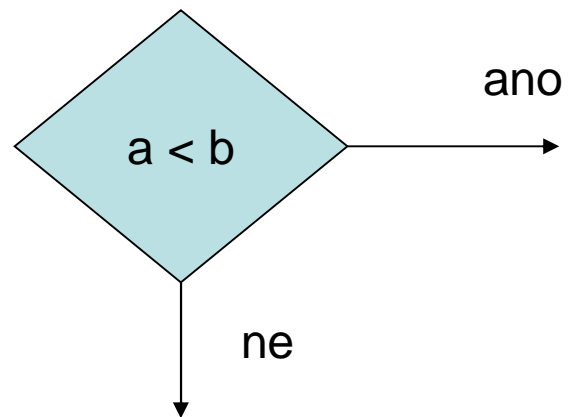
- začátek/konec



- akce (příkaz)



- podmínka





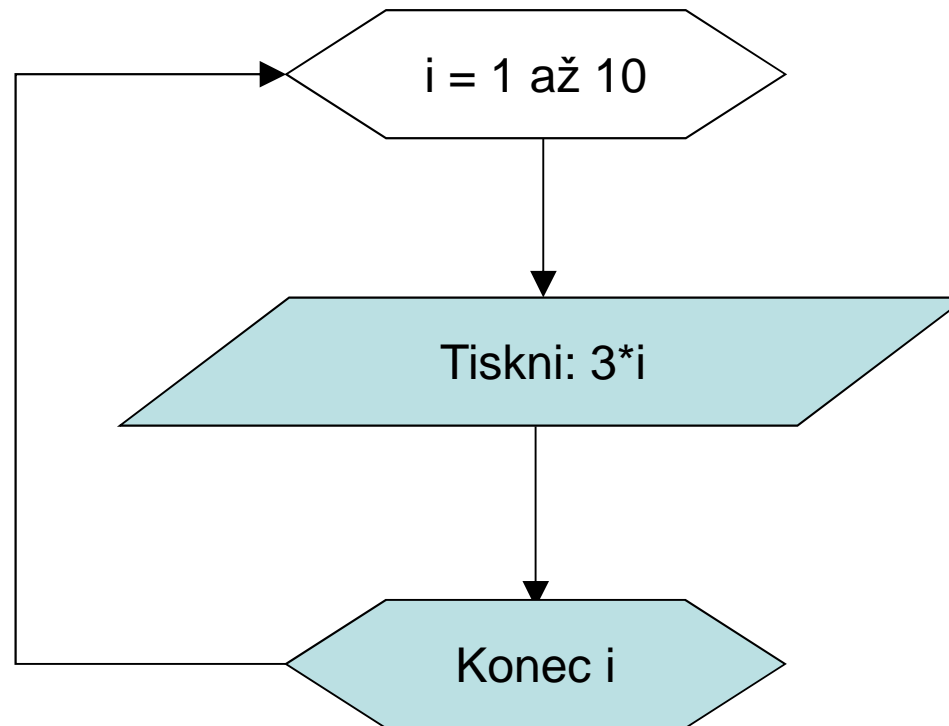
Existují i speciální značky pro cyklus s pevným počtem opakování, pro vstup/výstup dat (výpis textů, výsledků), práci se soubory

- vstup/výstup

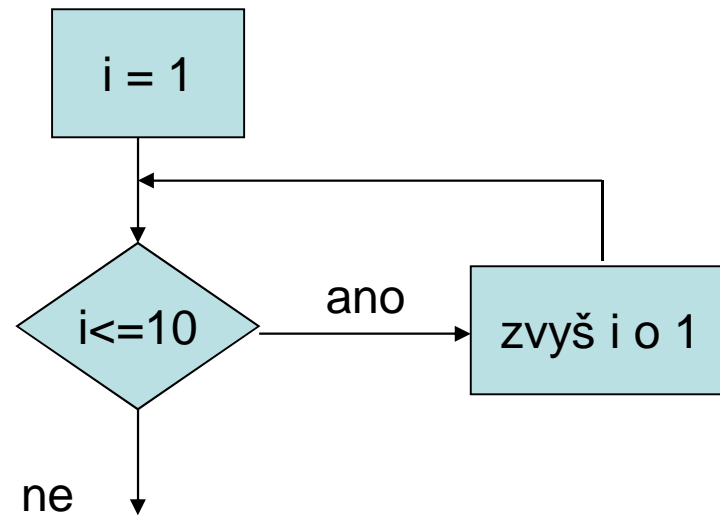


Tiskni: Obvod kruhu je  $r$

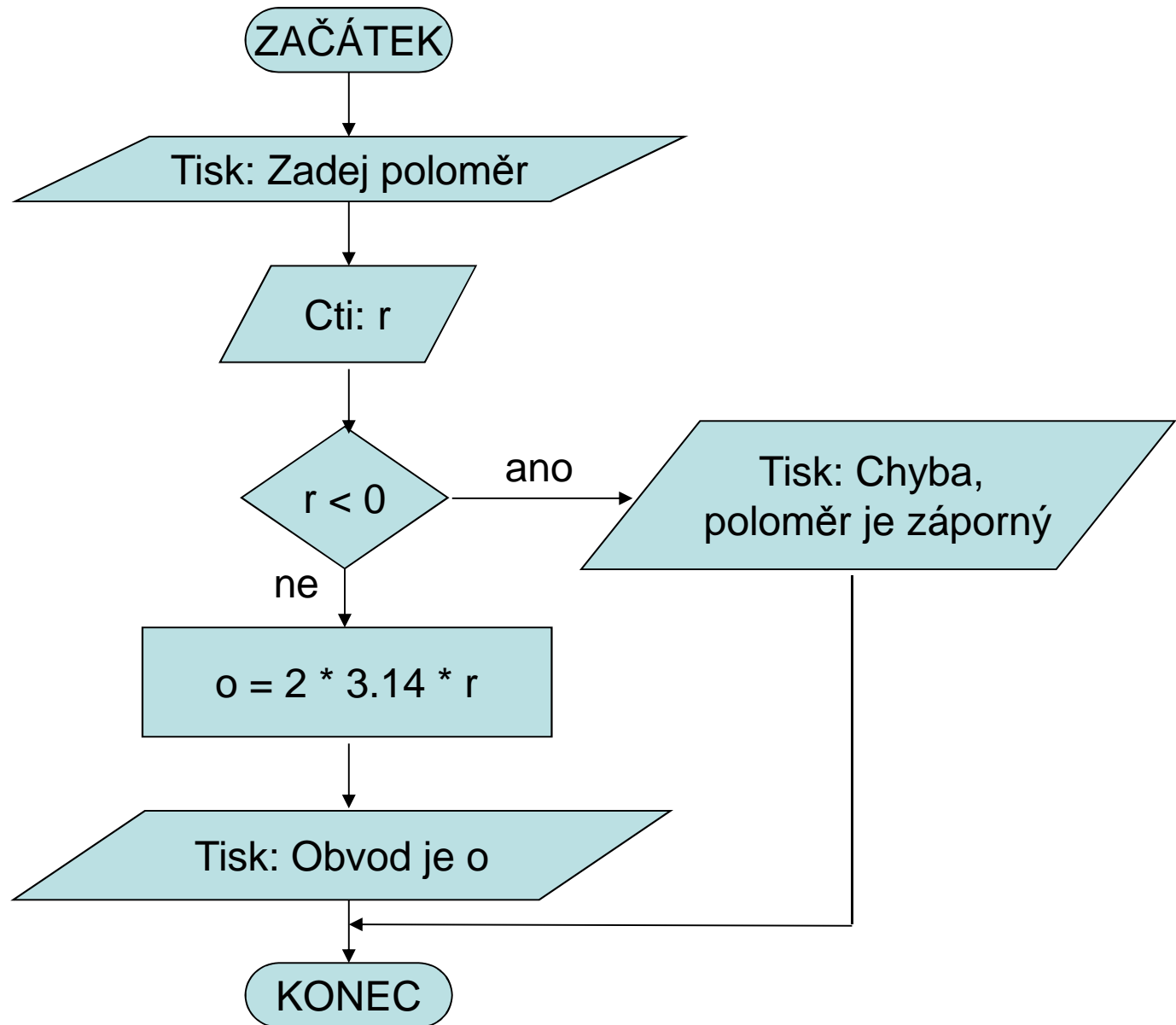
- cyklus s pevným počtem opakování  
– pomocí značky „příprava“



- my budeme pro jednoduchost i pro vstup a výstup využívat značek pro příkaz
- cyklus s podmínkou na konci/na začátku budeme zapisovat pomocí podmínky a příkazu



# Příklad: algoritmus pro výpočet obvodu kruhu



- **strukturovaný přirozený jazyk**

1. Tiskni: Zadej poloměr
2. Cti:  $r$
3. jestliže je  $r < 0$ , krok 7
4.  $o = 2 * 3,14 * r$
5. Tiskni: Obvod je  $o$
6. Konec
7. Tisk: Chyba: Polomer je záporný
8. Konec

- **pseudojazyk**
  - využíváme konstruktů nějakého programovacího jazyka, ale příkazy píšeme česky, často „volnou“ formou (zde Pascal)

Tiskni: Zadej poloměr

Cti: r

**if** r < 0 **then** Tiskni: Chyba: Poloměr je záporný

**else**

**begin**

o =  $2 * 3,14 * r$

Tiskni: Obvod je o

**end**

# Jak bude vypadat algoritmus zapsaný v jazyce C?

```
printf("Zadej polomer: ");  
scanf("%f",&r);  
if (r<0)  
    printf("Chyba: Polomer je zaporny");  
else  
{  
    o = 2*3.14*r;  
    printf("Obvod je %f",o);  
}
```

# Úkol

- Napište algoritmus vaření čaje.
- Napište algoritmus pro výpočet kořenu lineární rovnice  $ax + b = c$

*Domácí úkol:*

Dokončete algoritmus pro výpočet kořenů kvadratické rovnice (i s podmínkou, je-li  $a=0$ )