

# ALGORITMIZACE



Výukový materiál pro osmiletá  
gymnázia

# CYKLY



**CYKLUS JE JEDNÍM Z NEJSILNĚJŠÍCH NÁSTROJŮ  
ALGORITMŮ.**

**JEJICH PODSTATOU JE:**

**OPAKOVÁNÍ URČITÉ ČÁSTI ALGORITMU**

**BUĎ SE STEJNÝMI,**

**NEBO POKAŽDÉ S JINÝMI DATY.**

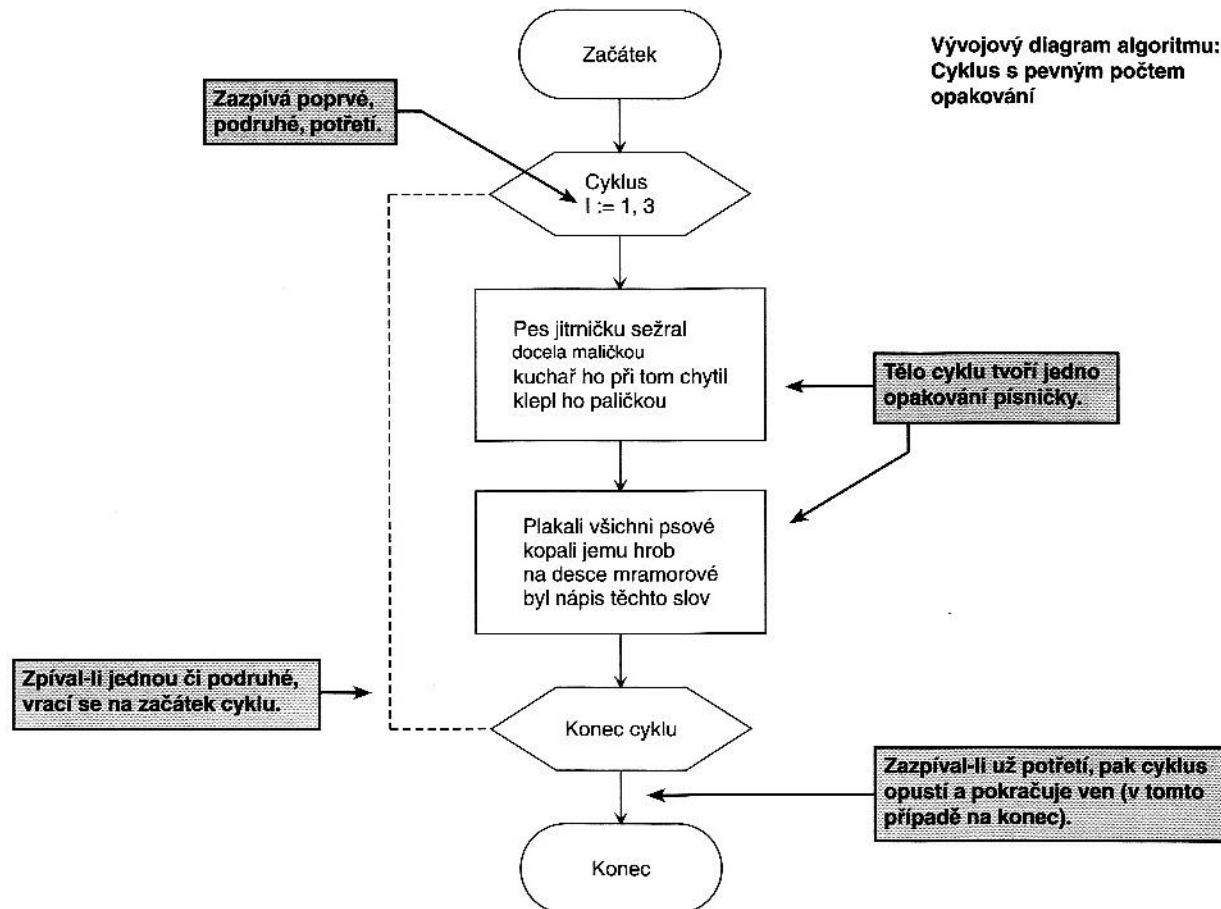
# CYKLY PODLE ZPŮSOBU ŘÍZENÍ



- 1) **CYKLY S PEVNÝM POČTEM OPAKOVÁNÍ**
- 2) **CYKLY ŘÍZENÉ PODMÍNKOU S PODMÍNKOU NA ZAČÁTKU CYKLU**
- 3) **CYKLY ŘÍZENÉ PODMÍNKOU S PODMÍNKOU NA KONCI CYKLU**

# CYKLY S PEVNÝM POČTEM OPAKOVÁNÍ

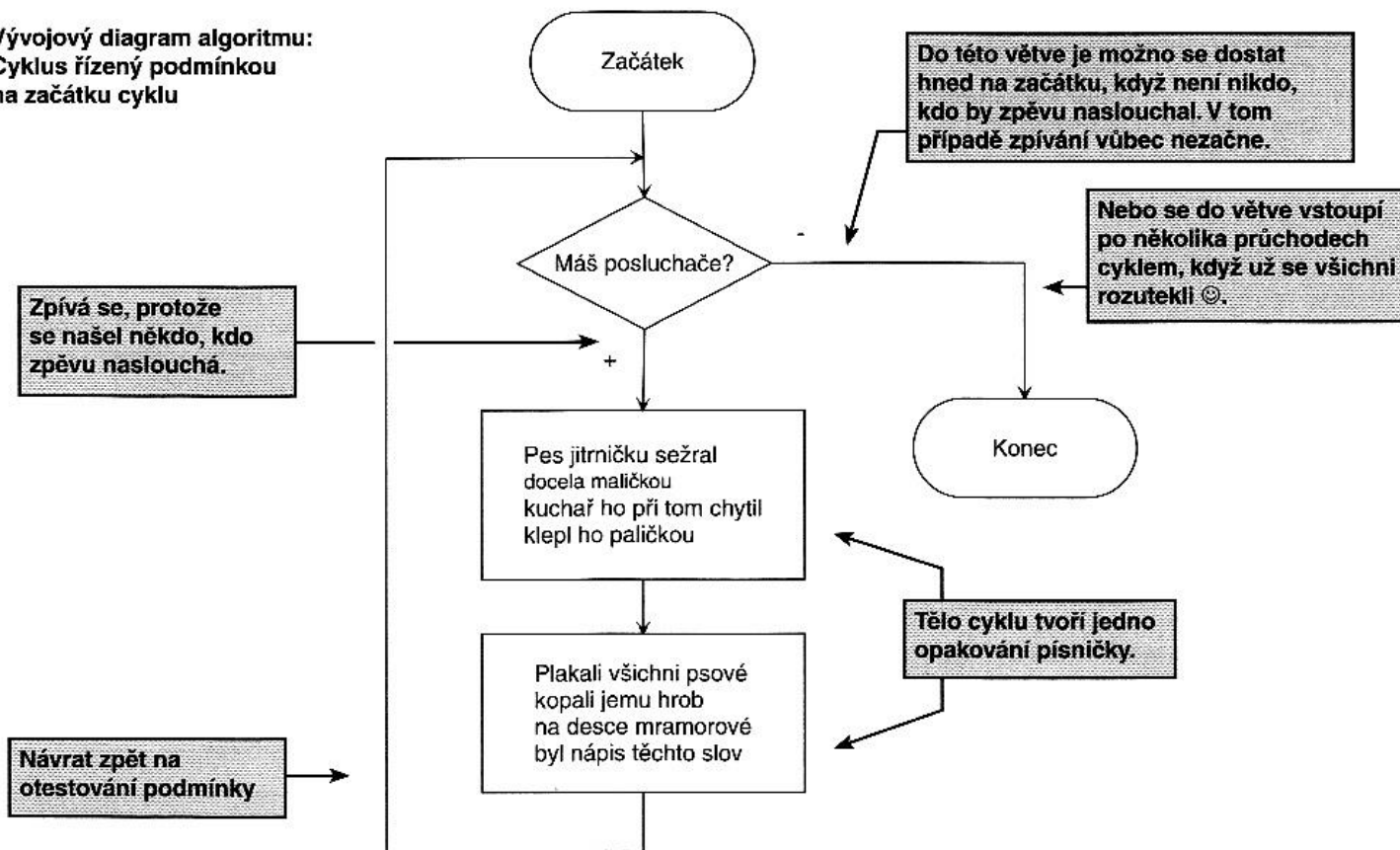
Takto vypadá správně zapsaný algoritmus „*Pes jitričku sežral*“, pokud si řeknete, že ji zazpíváte právě třikrát.



# CYKLY S PODMÍNKOU NA ZAČÁTKU

Takto bude vypadat správně zapsaný algoritmus „Pes jitričku sežral“, pokud si řeknete, že budete zpívat pouze v případě, když budete mít nějaké posluchače.

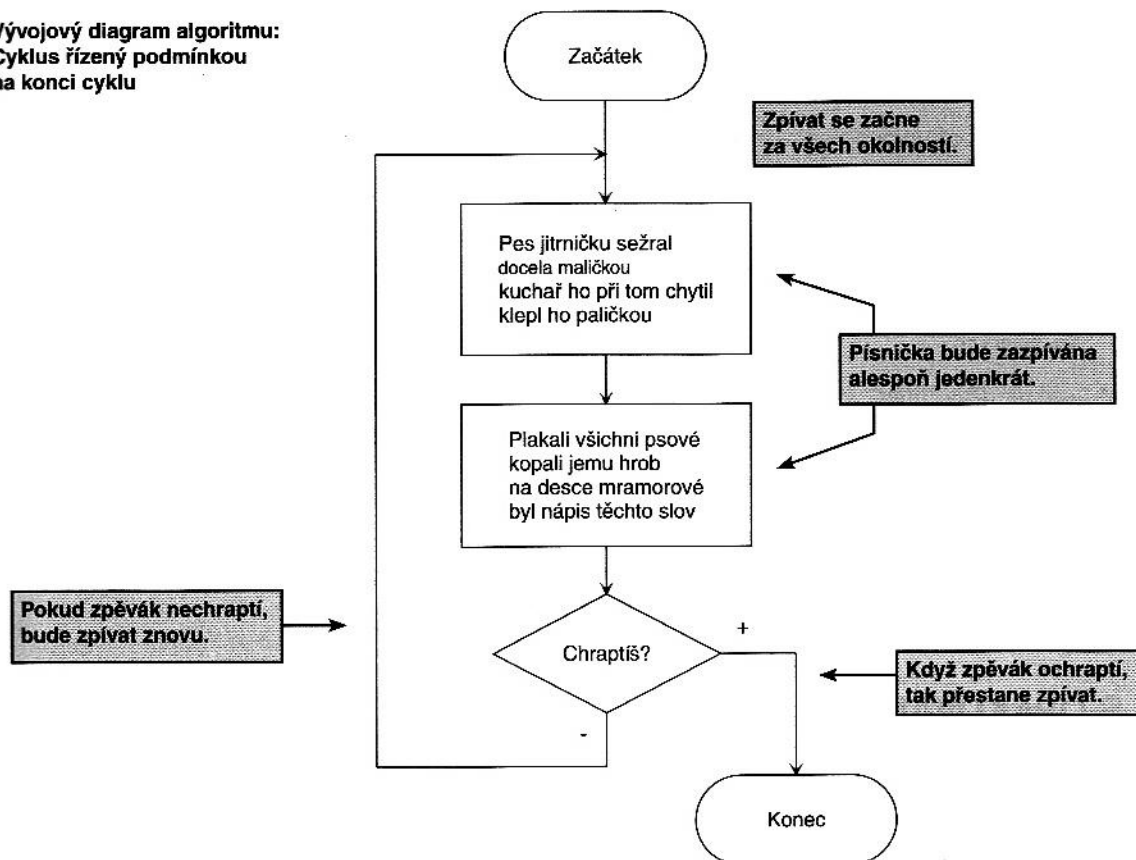
Vývojový diagram algoritmu:  
Cyklus řízený podmínkou  
na začátku cyklu



# CYKLY S PODMÍNKOU NA KONCI

Algoritmus zpívání písničky „Pes jitričku sežral“ nebude tentokrát záviset na posluchačích. Začnete zpívat a budete pokračovat tak dlouho, dokud neochrptíte.

Vývojový diagram algoritmu:  
Cyklus řízený podmínkou  
na konci cyklu



# TYPICKÉ ALGORITMY S CYKLY



- **úlohy s opakováním**
- **sumy, hledání maxima a minima, prohledávání řad**
- **oddechové úlohy – písničky, hry, hádanky**
- **číselné soustavy a převod mezi nimi**
- **řady – aritmetické, geometrické, mocninné**
- **operace s vektory a maticemi**
- **třídící algoritmy**

# ROZDÍL CYKLŮ S PODMÍNKOU



## 1) Podmínka na začátku:

do cyklu vstoupíte, pokud je podmínka splněna, pokud podmínka přestane platit, pak z cyklu vystoupíte nesplnění podmínky znamená nevstoupit do cyklu vůbec

## 2) Podmínka na konci:

do cyklu vstoupíte vždy, cyklus proběhne nejméně jednou nesplnění podmínky na konci znamená opakování cyklu splnění podmínky znamená opuštění cyklu a pokračování



# ČEKACÍ SMYČKA



ALGORITMY, KDE PODMÍNKA NA KONCI JE ŽÁDOUCÍ, SE NAZÝVAJÍ **ČEKACÍ SMYČKA**.

Používají se v software, který pracuje v reálném čase, např. operační systémy, technologické procesy...

Během cyklu se nic neděje, jen se testuje podmínka, na jejíž splnění se čeká.

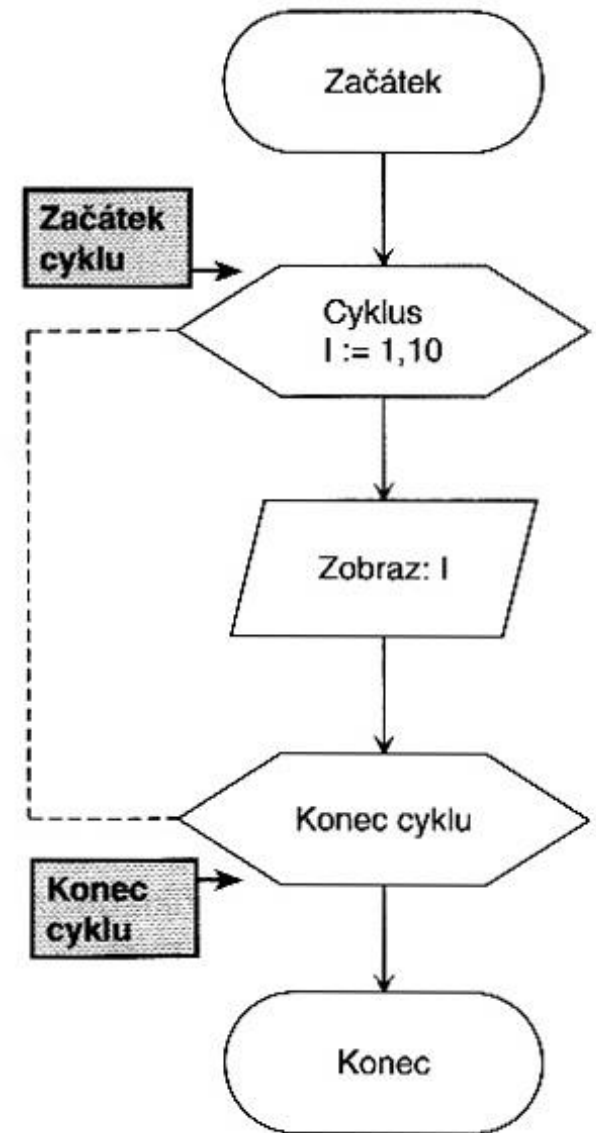
Pokud událost nastane, cyklus se opustí a dojde ke spuštění algoritmu, který navazuje.

# ČEKACÍ SMYČKA



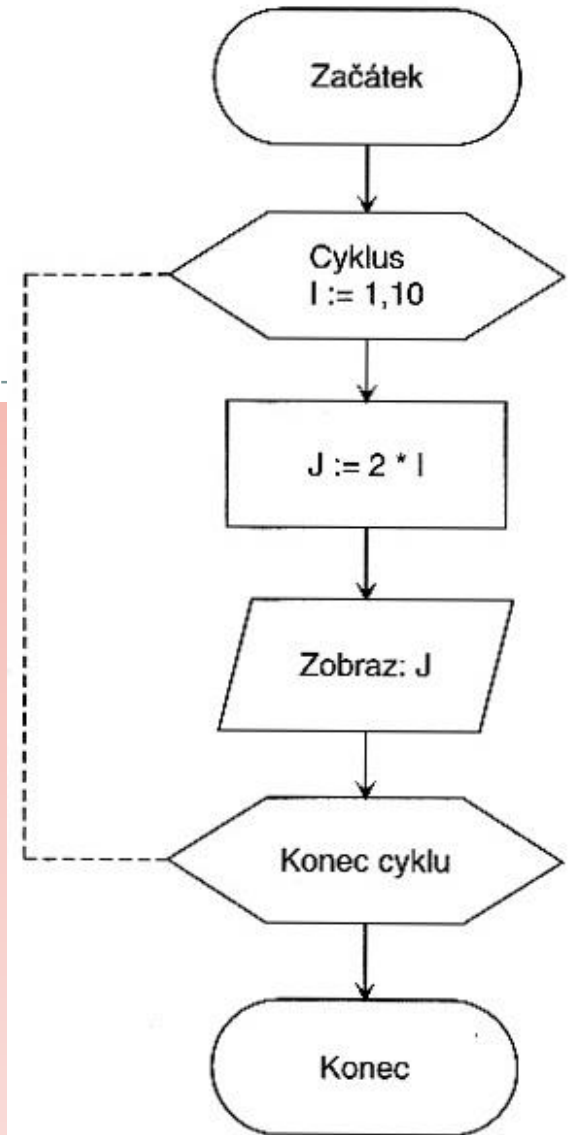
# PŘÍKLAD

NAKRESLI DIAGRAM ALGORITMU,  
KTERÝ ZOBRAZÍ ČÍSLA OD 1 DO 10.



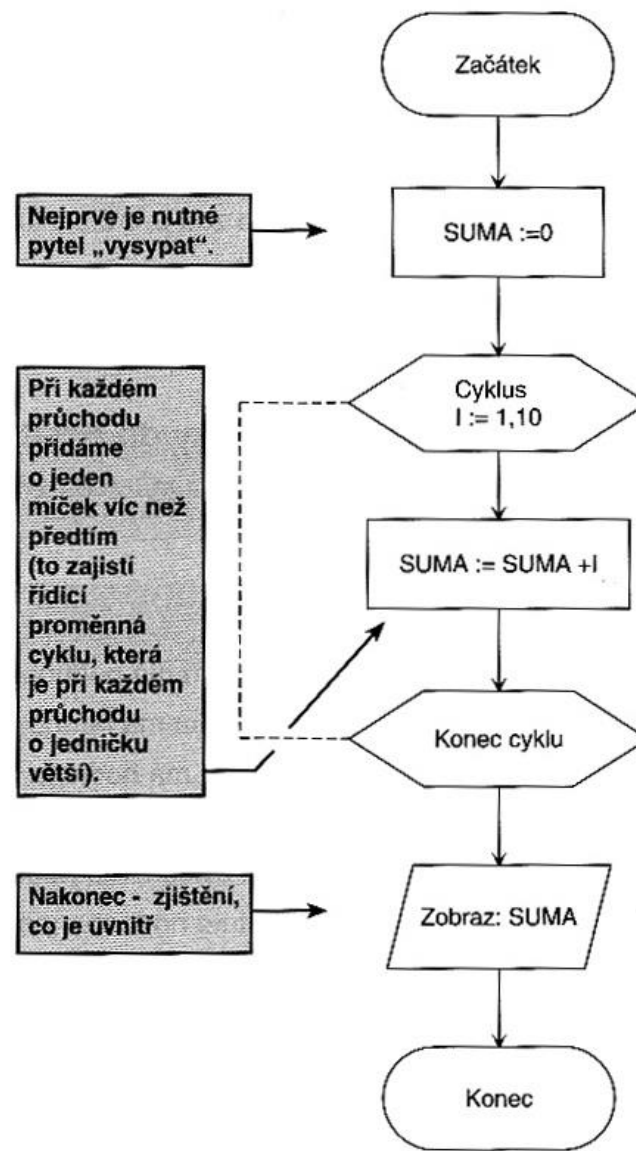
# PŘÍKLAD

NAKRESLI DIAGRAM ALGORITMU,  
KTERÝ ZOBRAZÍ  
SUDÁ ČÍSLA DO 20.



# PŘÍKLAD

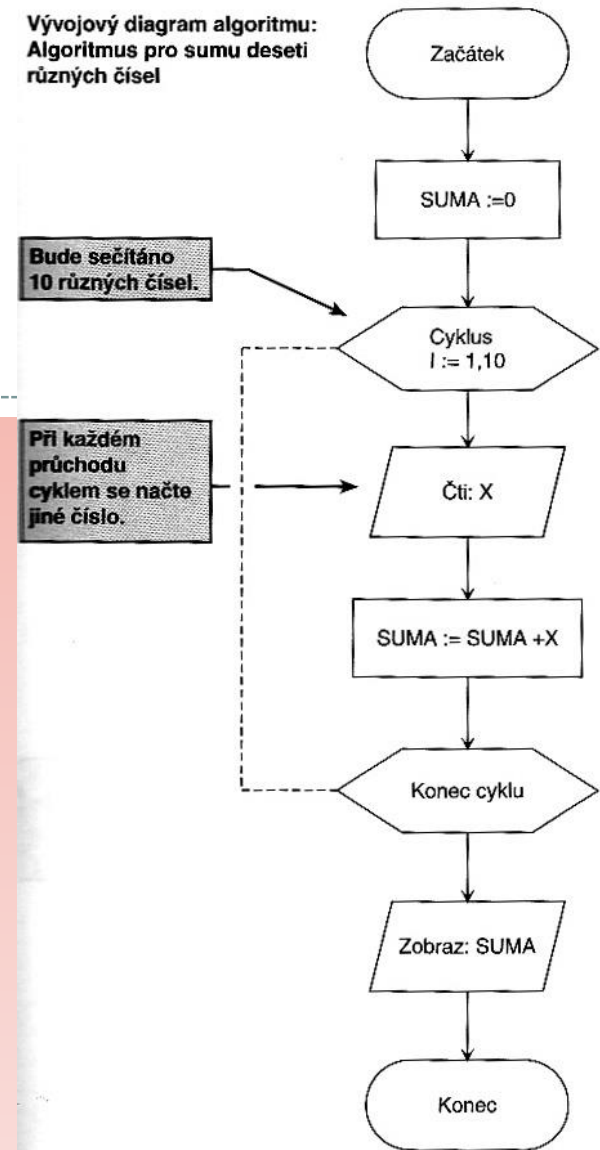
NAKRESLI DIAGRAM ALGORITMU,  
KTERÝ SEČTE ČÍSLA OD 1 DO 10.



# PŘÍKLAD

NAKRESLI DIAGRAM ALGORITMU,  
KTERÝ SEČTE 10 RŮZNÝCH ČÍSEL.

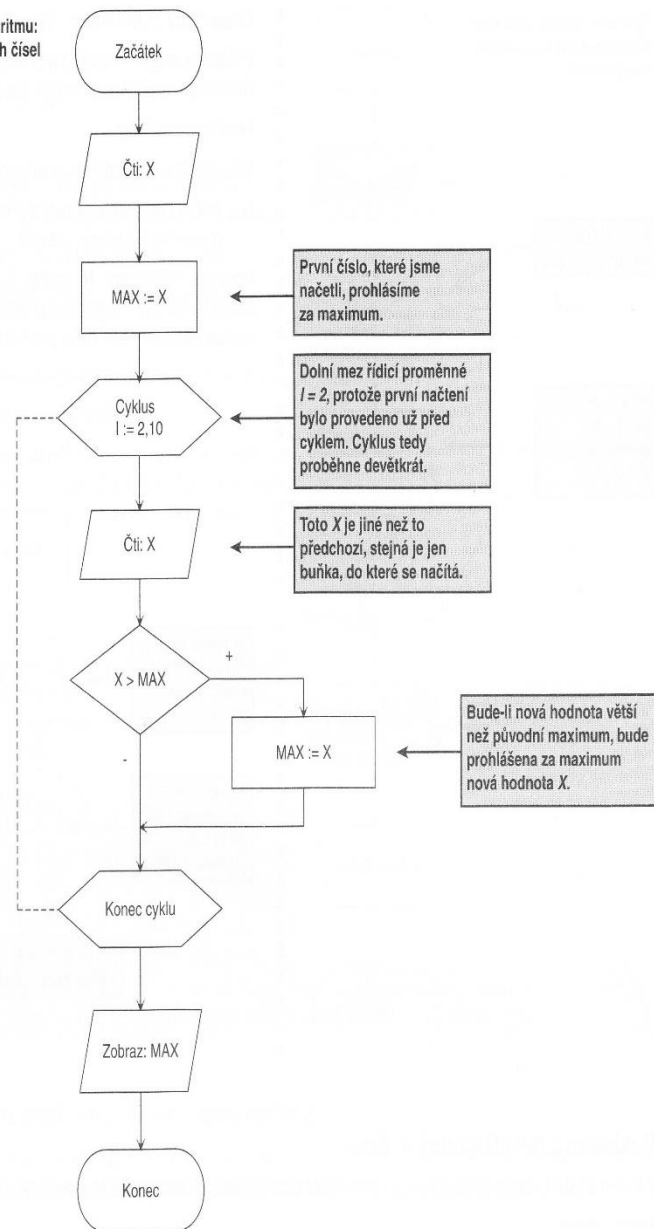
Vývojový diagram algoritmu:  
Algoritmus pro sumu deseti  
různých čísel



# PŘÍKLAD

NAKRESLI DIAGRAM ALGORITMU,  
KTERÝ NAJDE MAXIMUM  
Z 10 Kladných čísel.

Vývojový diagram algoritmu:  
Maximum z 10 kladných čísel



# CYKLY



**MLUVILI JSME O CYKLECH ŘÍZENÝCH PODMÍNKOU.  
PODMÍNKA SE MŮŽE VYSKYTOVAT NA TŘECH MÍSTECH:**

- 1) NA ZAČÁTKU CYKLU**
- 2) UPROSTŘED CYKLU**
- 3) NA KONCI CYKLU**

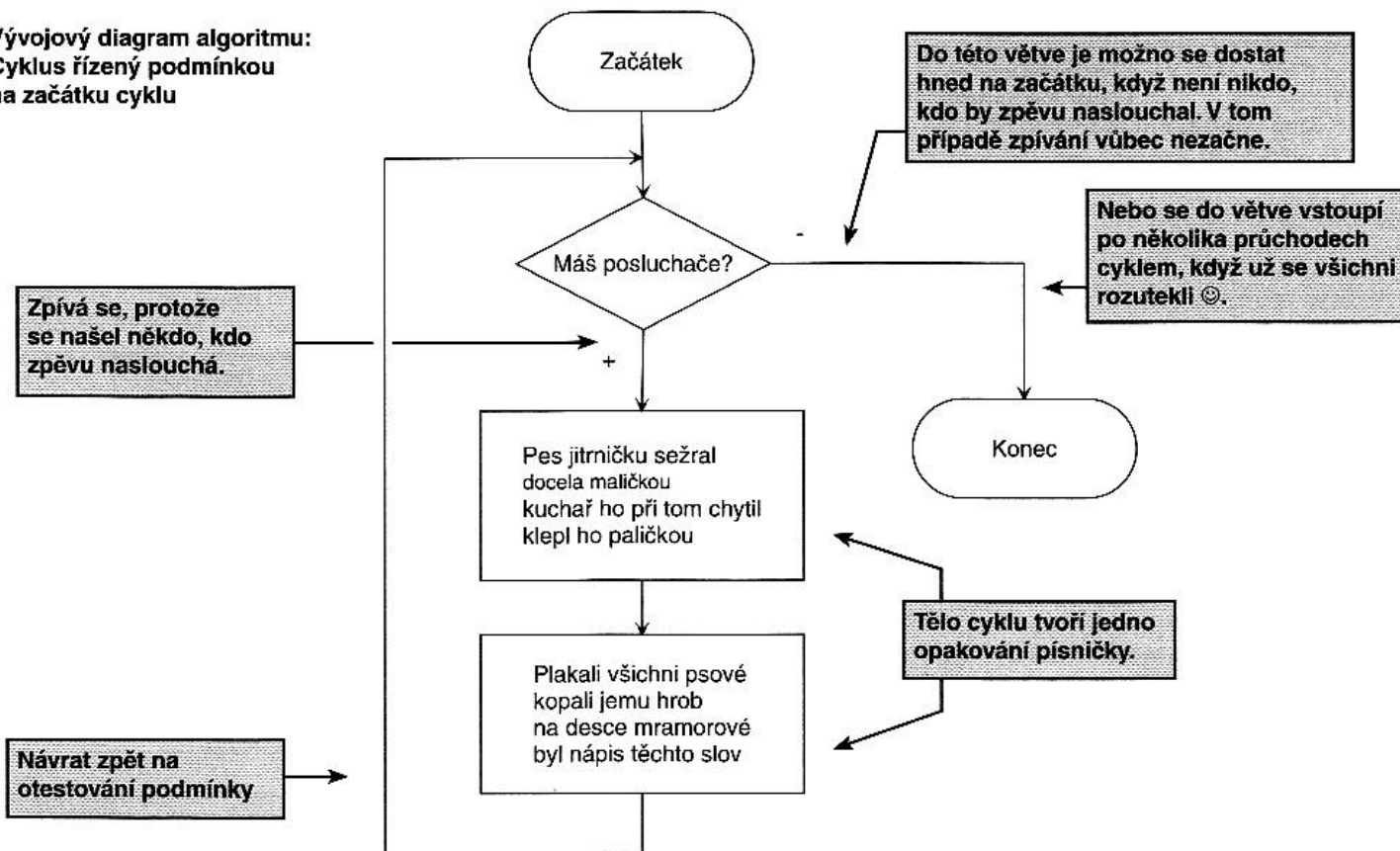
**VZPOMEŇTE NA PŘEDCHOZÍ HODINU CYKLŮ ...**



# CYKLY S PODMÍNKOU NA ZAČÁTKU

Takto bude vypadat správně zapsaný algoritmus „Pes jitričku sežral“, pokud si řeknete, že budete zpívat pouze v případě, když budete mít nějaké posluchače.

Vývojový diagram algoritmu:  
Cyklus řízený podmínkou  
na začátku cyklu

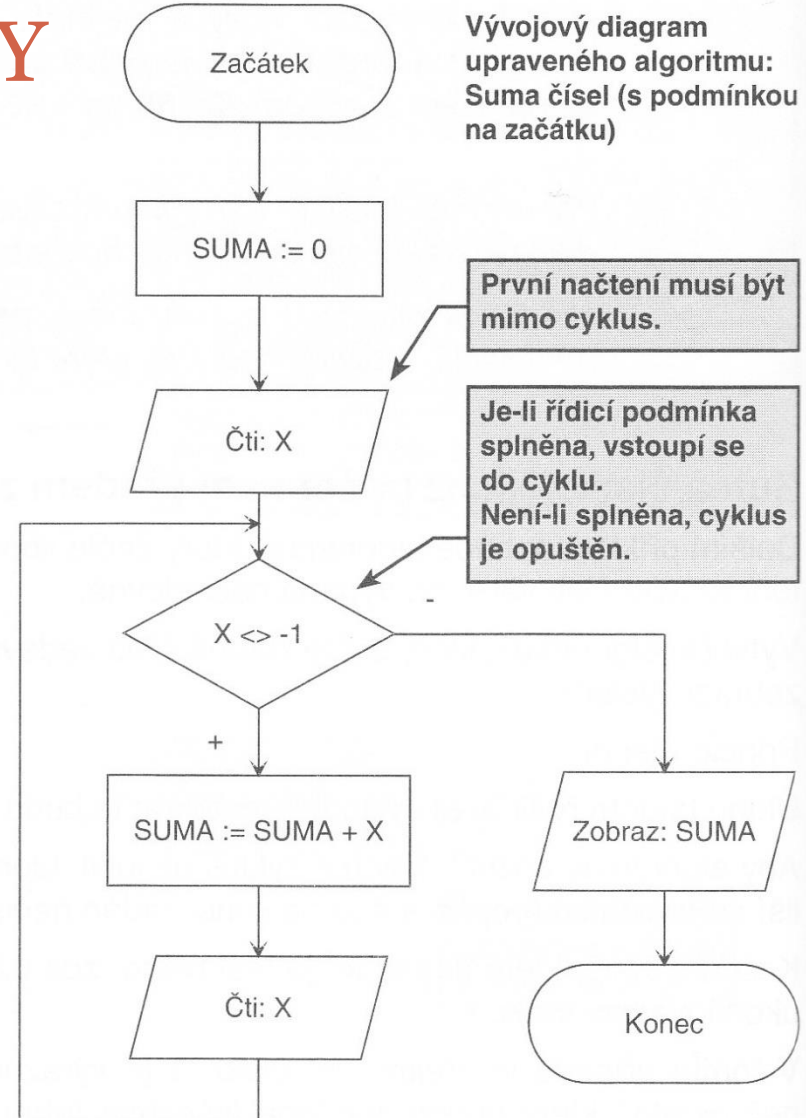


# CYKLY

VYTVOŘTE ALGORITMUS, KTERÝ SEČTE NEZNÁMÝ POČET Kladných čísel zadávaných zvenčí.

Zadáte-li číslo **-1**, vykreslí výsledek.

1) Podmínka na začátku cyklu





# CYKLY

VYTVOŘTE ALGORITMUS, KTERÝ SEČTE NEZNÁMÝ POČET KLADNÝCH ČÍSEL ZADÁVANÝCH ZVENČÍ.

ZADÁTE-LI ČÍSLO **-1**, VYKRESLÍ VÝSLEDEK.

3) Podmínka na konci cyklu ...

