

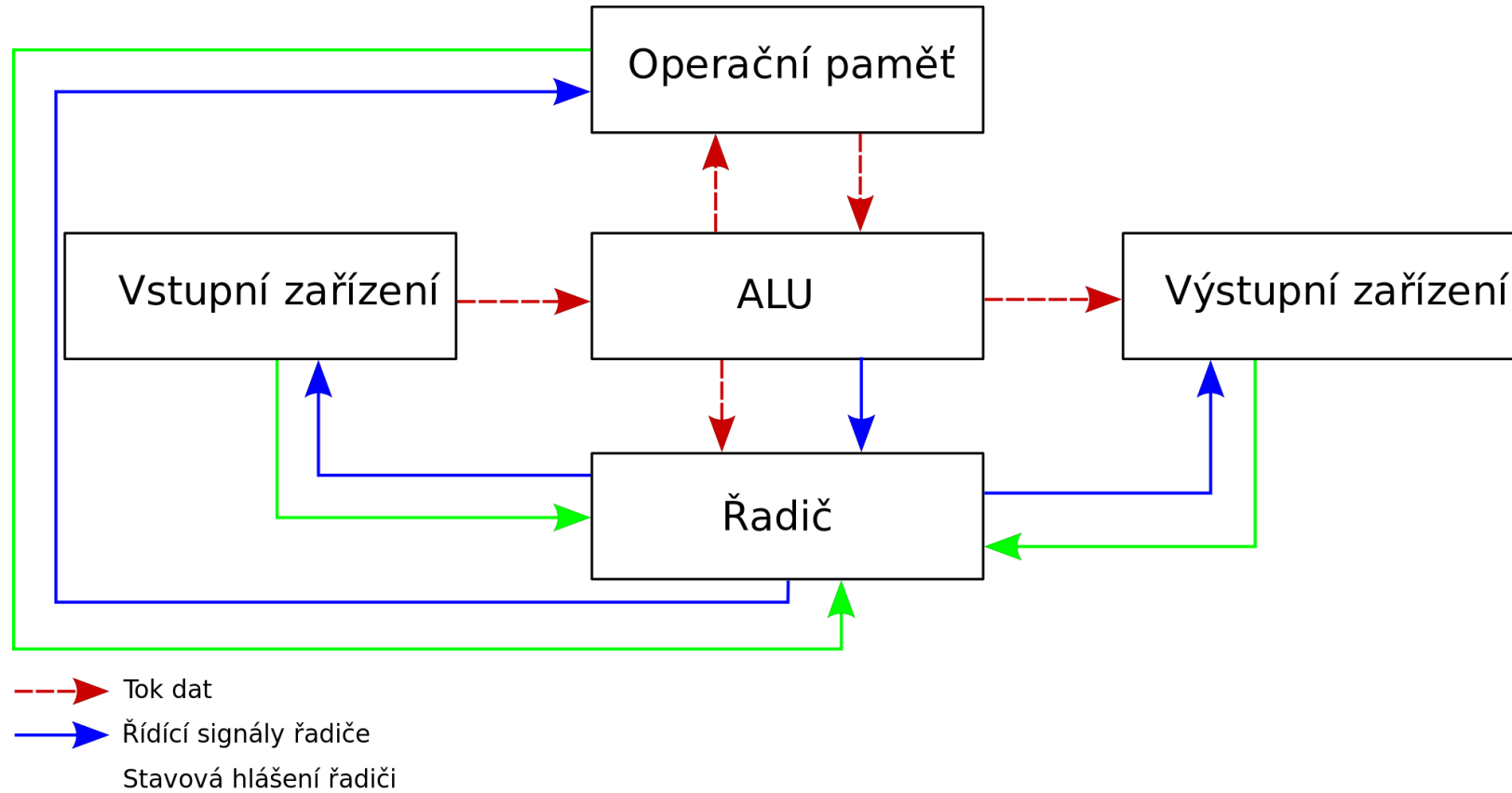
Úvod do programování (UPG)

značky vývojových diagramů

DAVID KUBÁT (213)

KUBAT@UZLABINA.CZ

Von Neumannovo schéma



Instrukce

Instrukce

Elementární operace procesoru, kterou je procesor schopen přímo vykonat.

Program

Program

- Program je **posloupnost instrukcí**, které řeší zadaný problém pomocí počítače.
- Program je obvykle realizován na základě **algoritmu**.

Algorithmus

Algoritmus

Algoritmus je přesný postup, který popisuje řešení daného problému a splňuje následující vlastnosti:

- Konečnost – Skončí po konečném počtu kroků
- **Determinovanost** – V každém kroku musí být přesně definován krok následující. Pro stejný vstup musí vždy vrátit stejný výsledek.
- (Hromadnost – Neřeší jednu instanci problému, ale celou třídu problémů)
- *Resultativnost* – *Musí mít alespoň jeden výstup (výsledek/chybové hlášení).*

Způsoby zápisu algoritmu

Slovně

Matematicky

Graficky

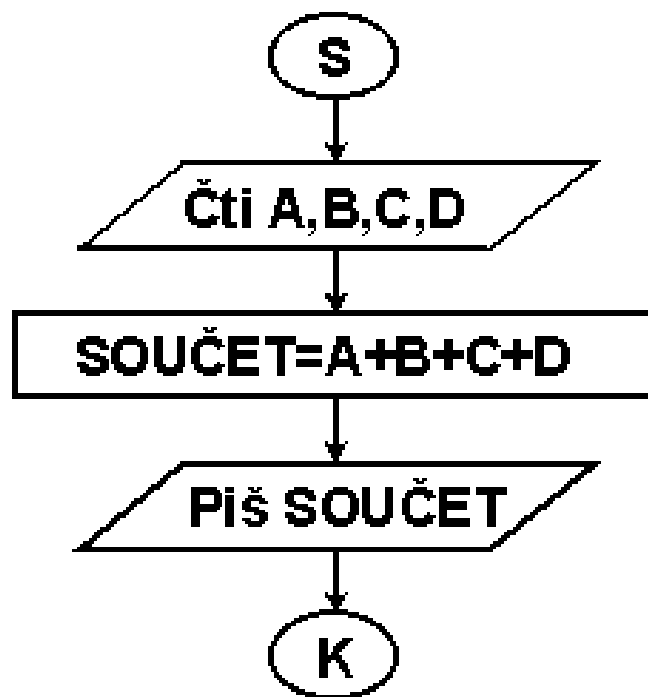
- Obrázkové návody (piktogramy, IKEA)
- Blokované diagramy
- Vývojové diagramy

Programovacím jazykem

Pseudokódem – Průnik běžné řeči a pravidel programovacích jazyků

Ukázka algoritmů

Sestavte algoritmus na součet čísel A,B,C,D a vytisknutí jejich součtu.

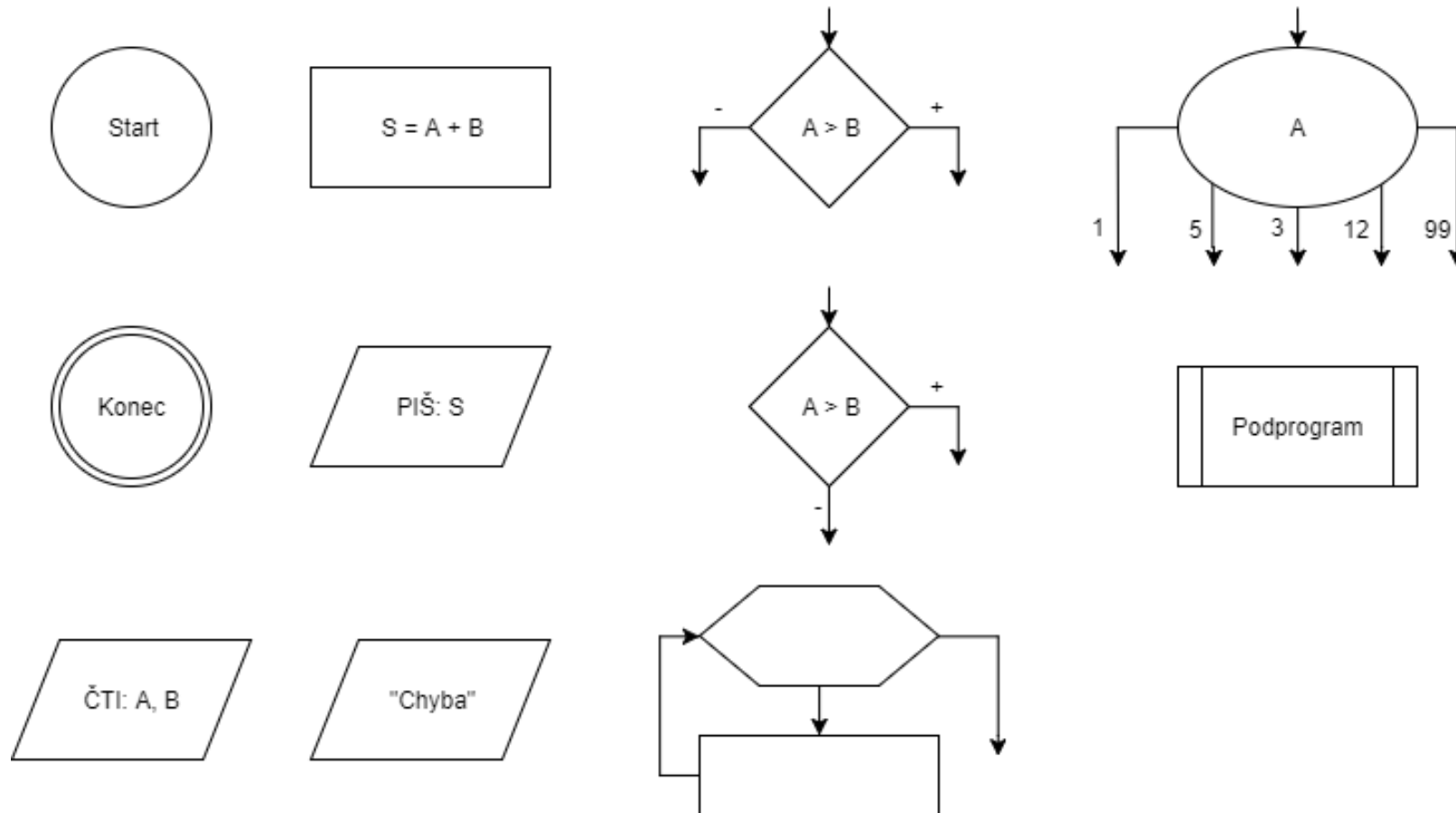


Proměnná

Proměnná

- Proměnná je **pojmenované místo v paměti**, které může v průběhu programu měnit svojí hodnotu.
- Rozsah a možnosti transformace hodnot jsou dány datovým typem proměnné (záleží na konkrétním programovacím jazyce).
- Slovní příklad: proměnná s názvem A v oboru celých čísel může v průběhu programu nabývat hodnot: -10, 20, 0, -42, 42, . . .

Značky vývojových diagramů

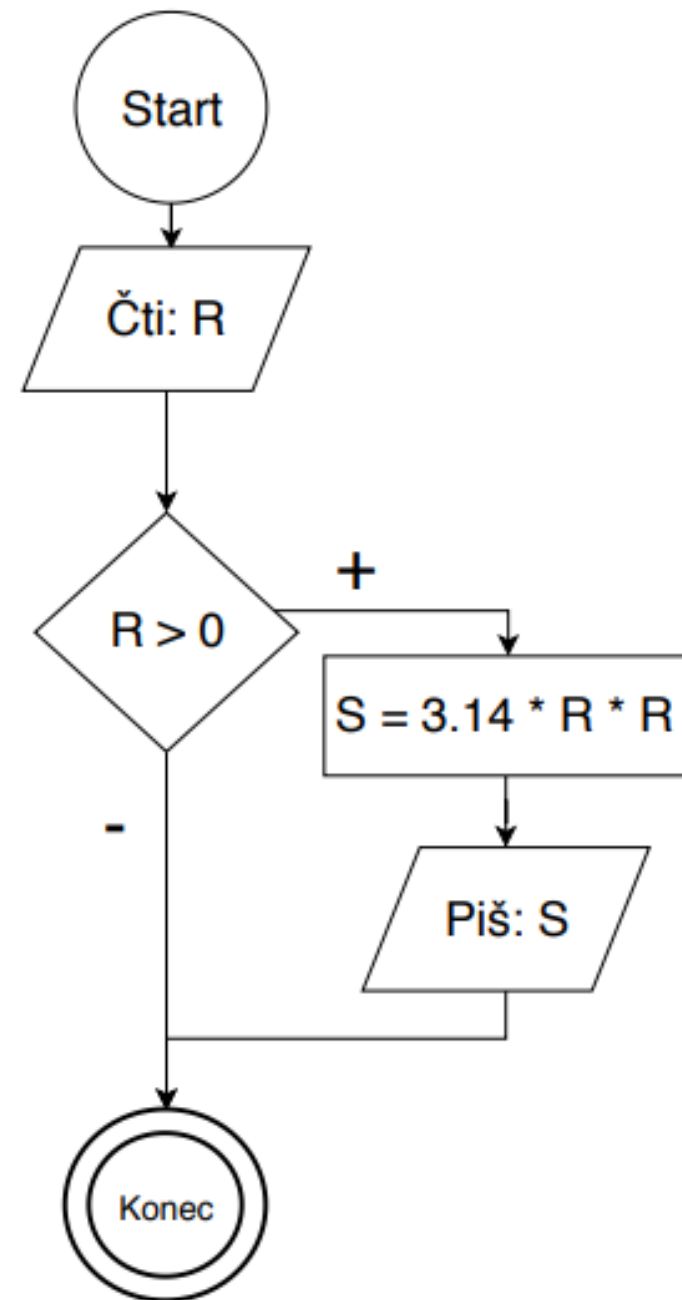


Příklady

Sestavte algoritmus pro výpočet obsahu kruhu.

Příklady

Sestavte algoritmus pro výpočet obsahu kruhu.



Příklady

Sestavte algoritmus, který:

1) přečte číslo do proměnné A

2) vypíše na výstup:

- „Kladné“, pokud bylo kladné
- „Záporné“, pokud bylo záporné.

Příklady

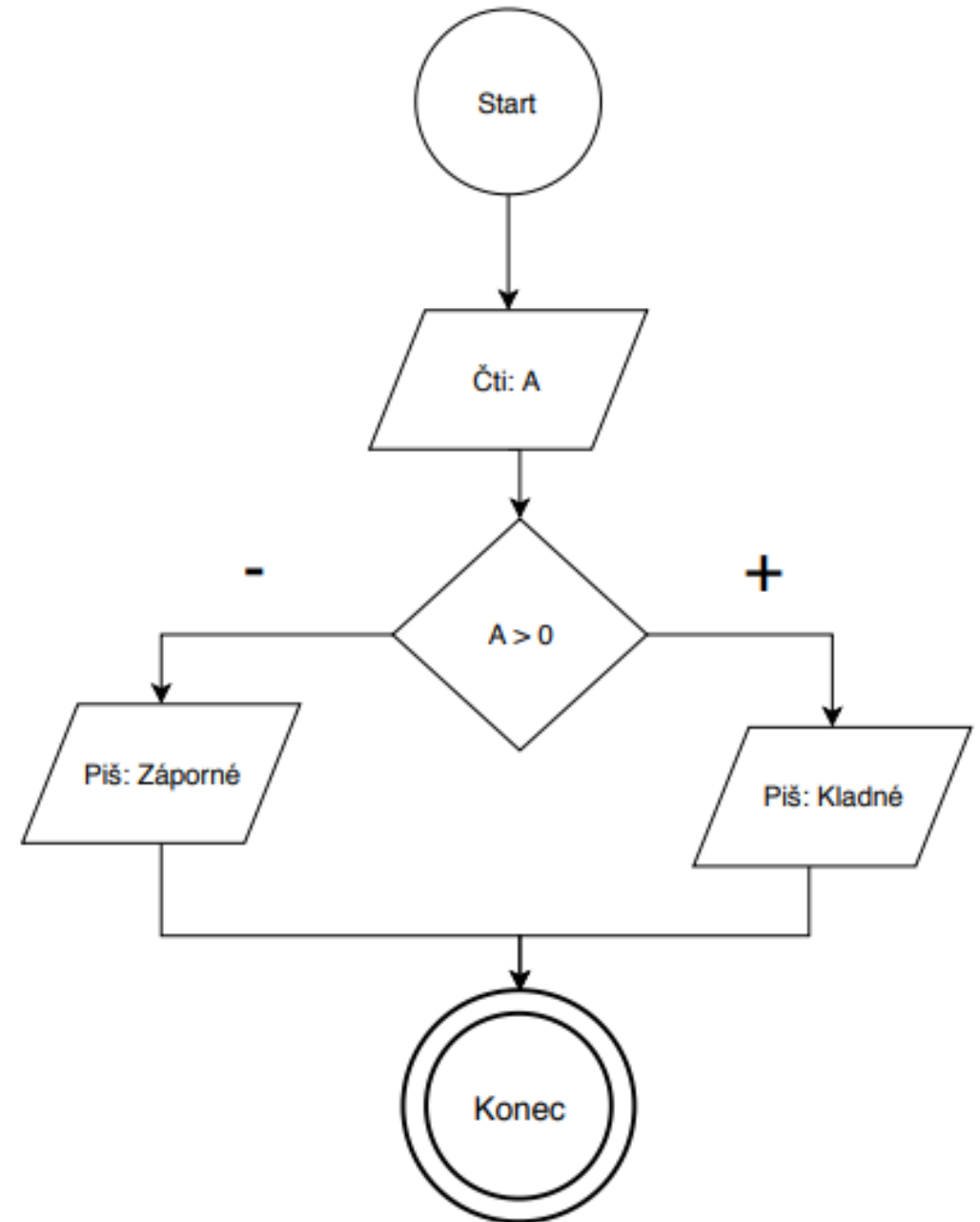
Sestavte algoritmus, který:

- 1) přečte číslo do proměnné A
- 2) vypíše na výstup:
 - „Kladné“, pokud bylo kladné
 - „Záporné“, pokud bylo záporné.

Například:

Vstup: $A = 2$

Výstup: Kladné



Příklady

Sestavte algoritmus, který:

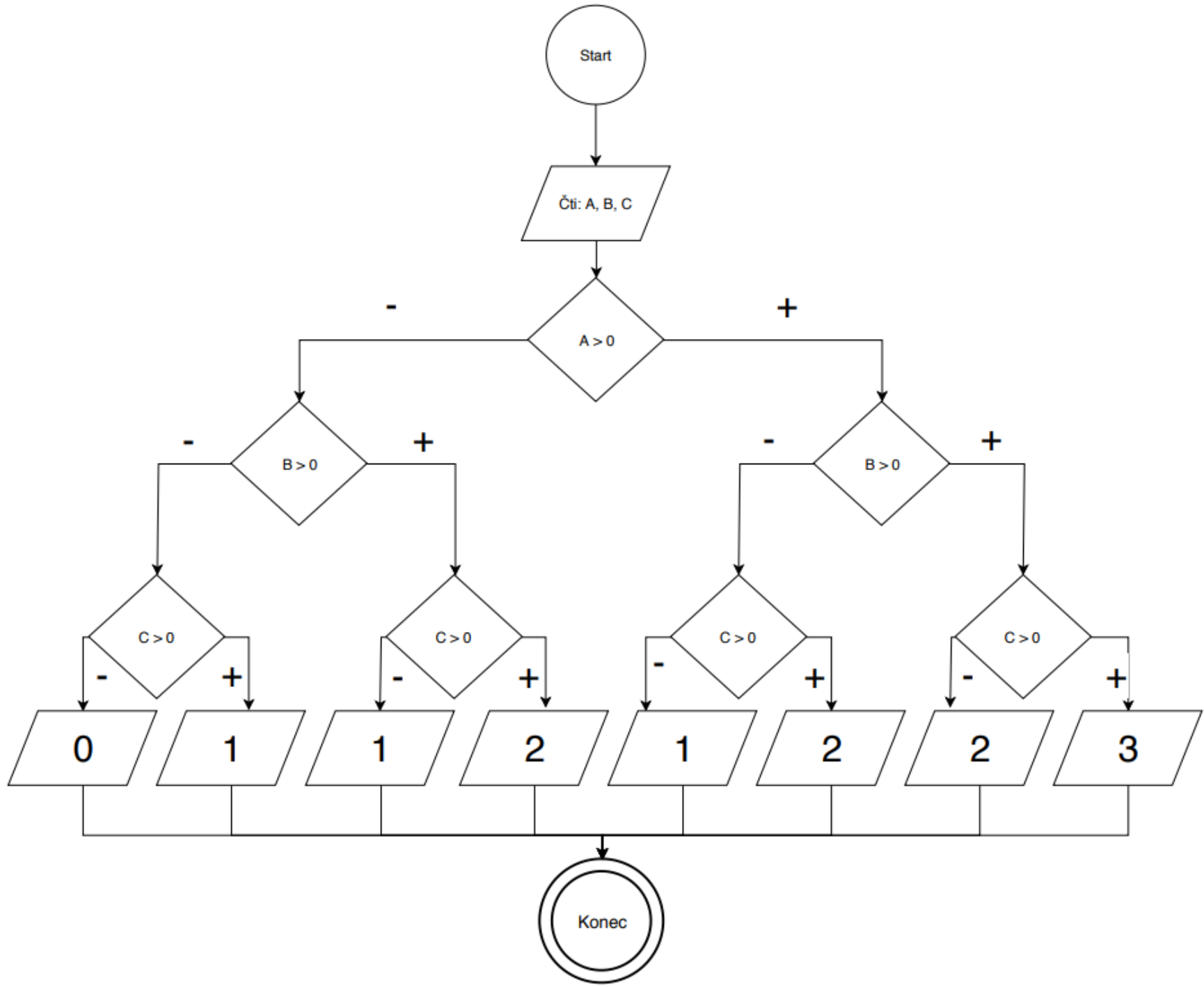
- 1) přečte **3 čísla** do proměnných **A, B, C**
- 2) vypíše na výstup počet kladných čísel
(*kolik ze zadaných čísel A, B, C bylo kladných*)

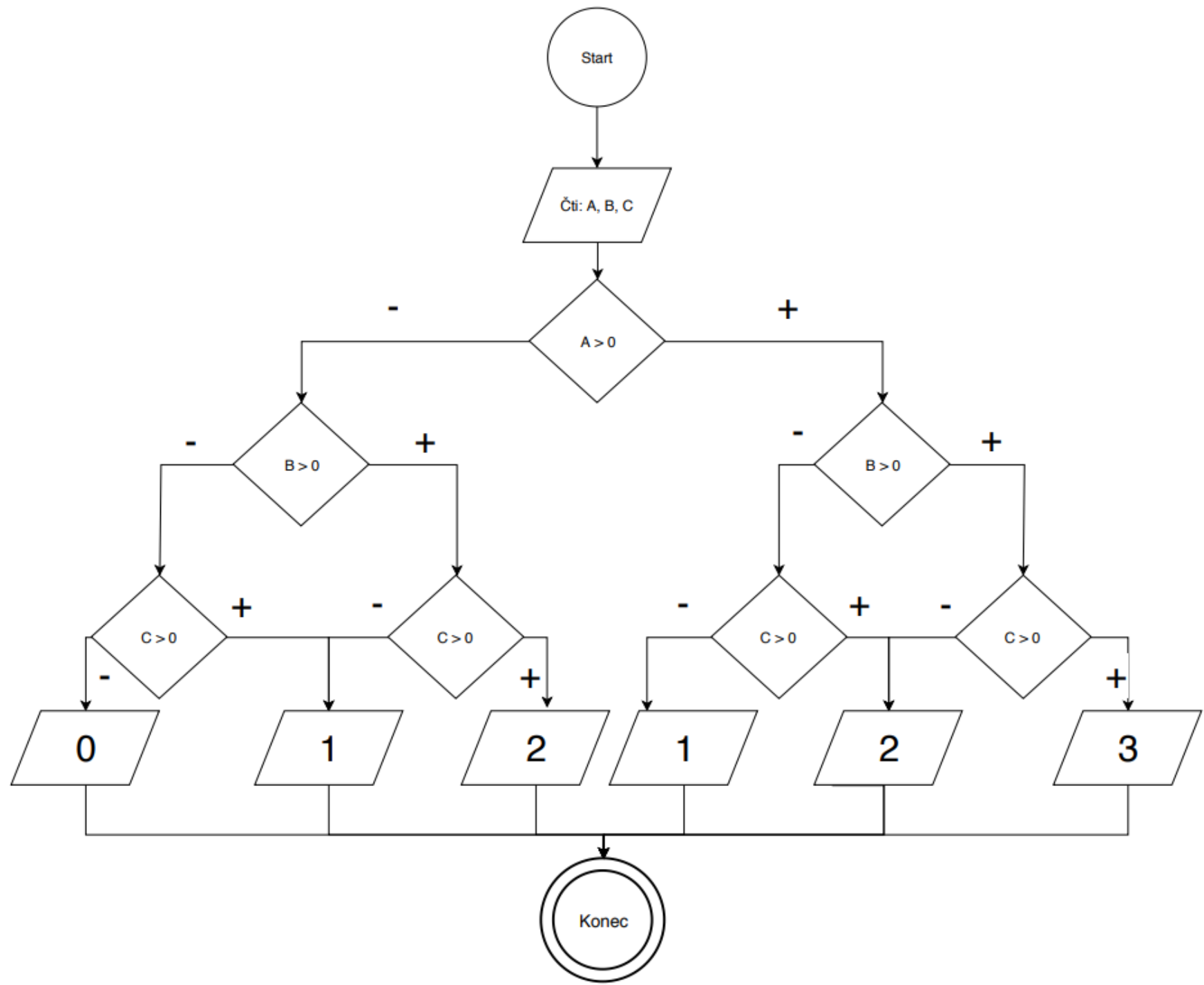
Například:

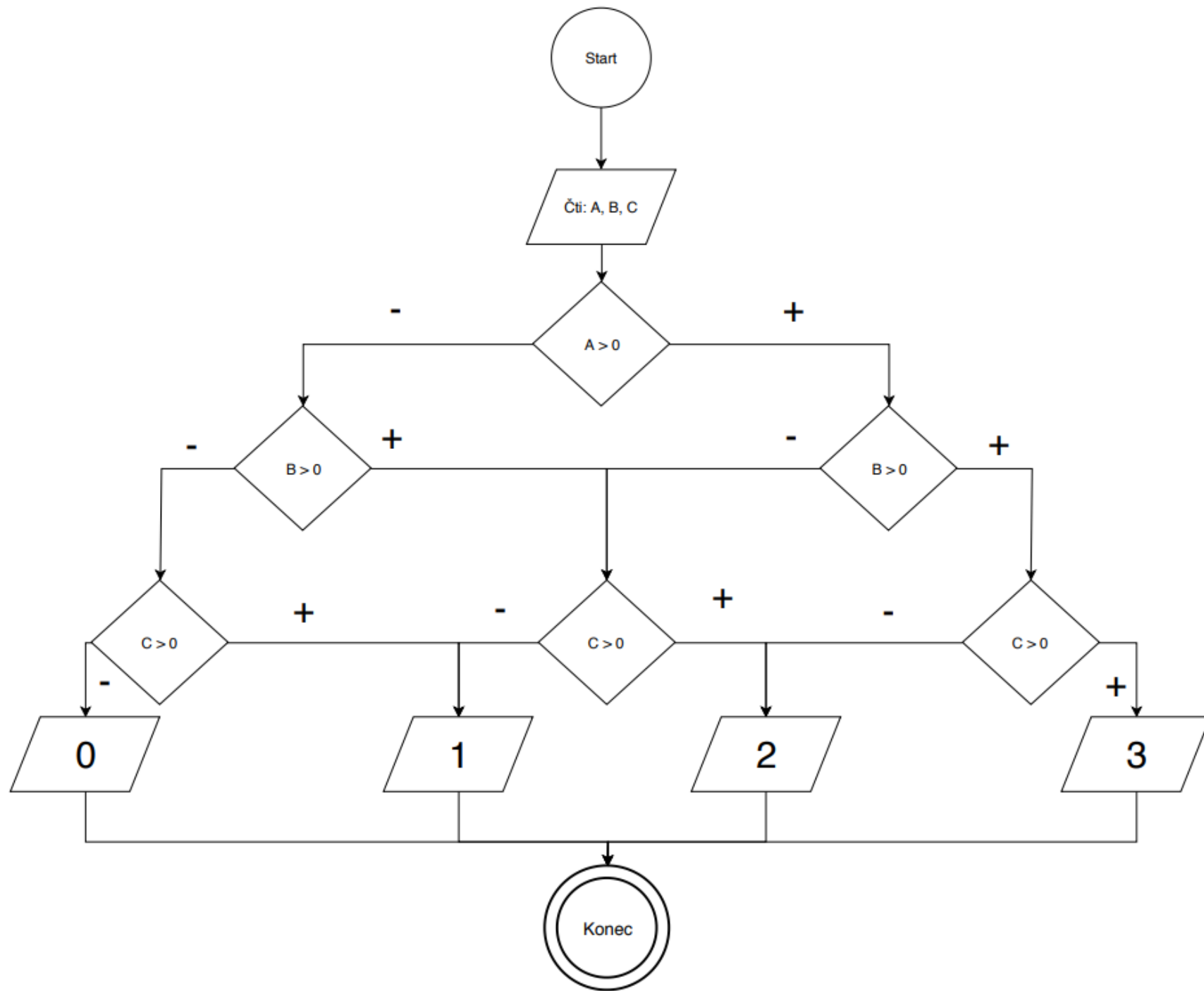
Vstup: A = 3, B = -8, C = 9

Výstup: 2

(Výstup je 2, protože jen proměnné A, C obsahují kladná čísla)







Příklady

Sestavte algoritmus, který:

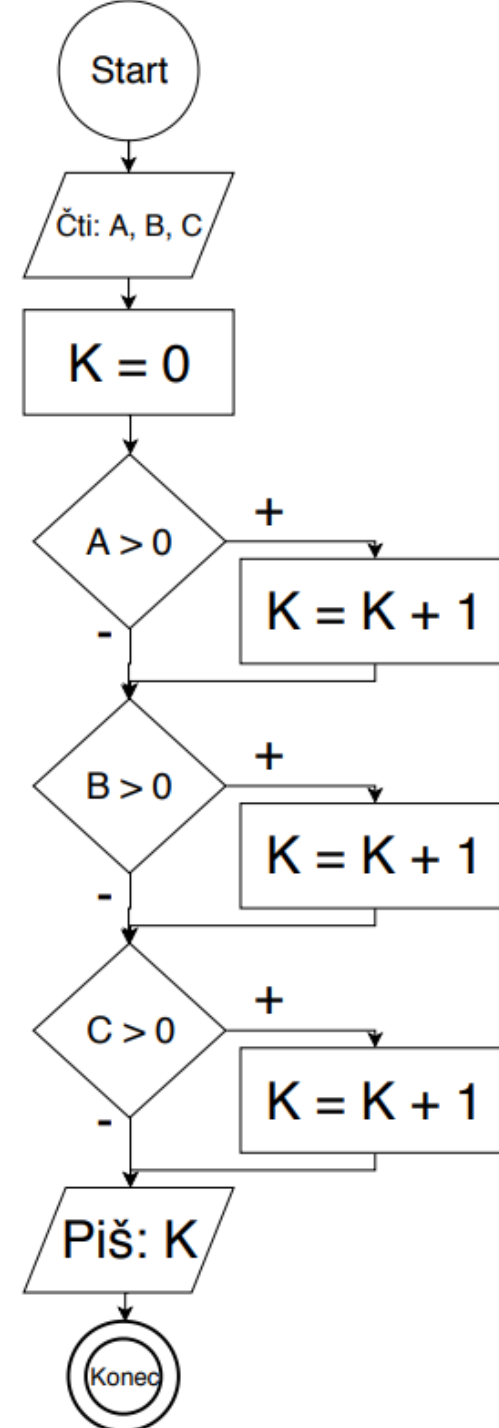
- 1) přečte **3 čísla** do proměnných **A, B, C**
- 2) vypíše na výstup počet kladných čísel
(kolik ze zadaných čísel A, B, C bylo kladných)

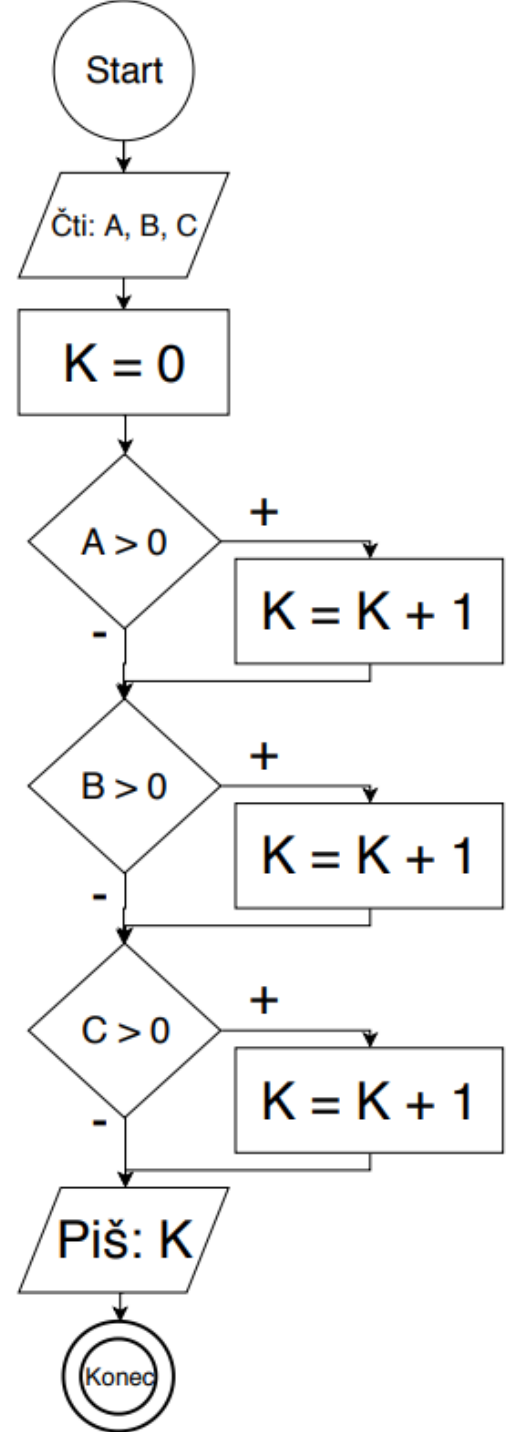
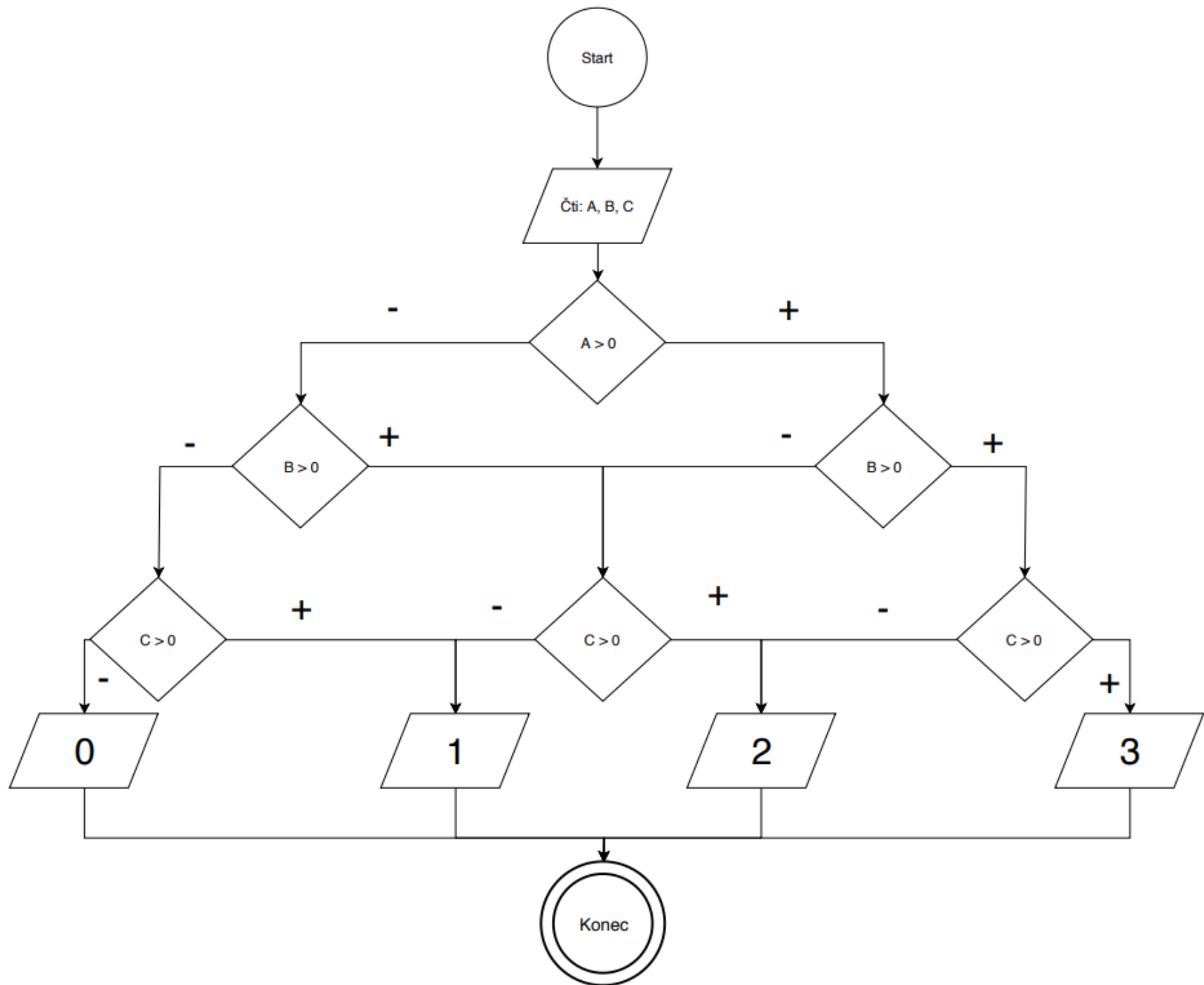
Například:

Vstup: $A = 3, B = -8, C = 9$

Výstup: 2

(Výstup je 2, protože jen proměnné A, B obsahují kladná čísla)





Příklady

Sestavte algoritmus, který:

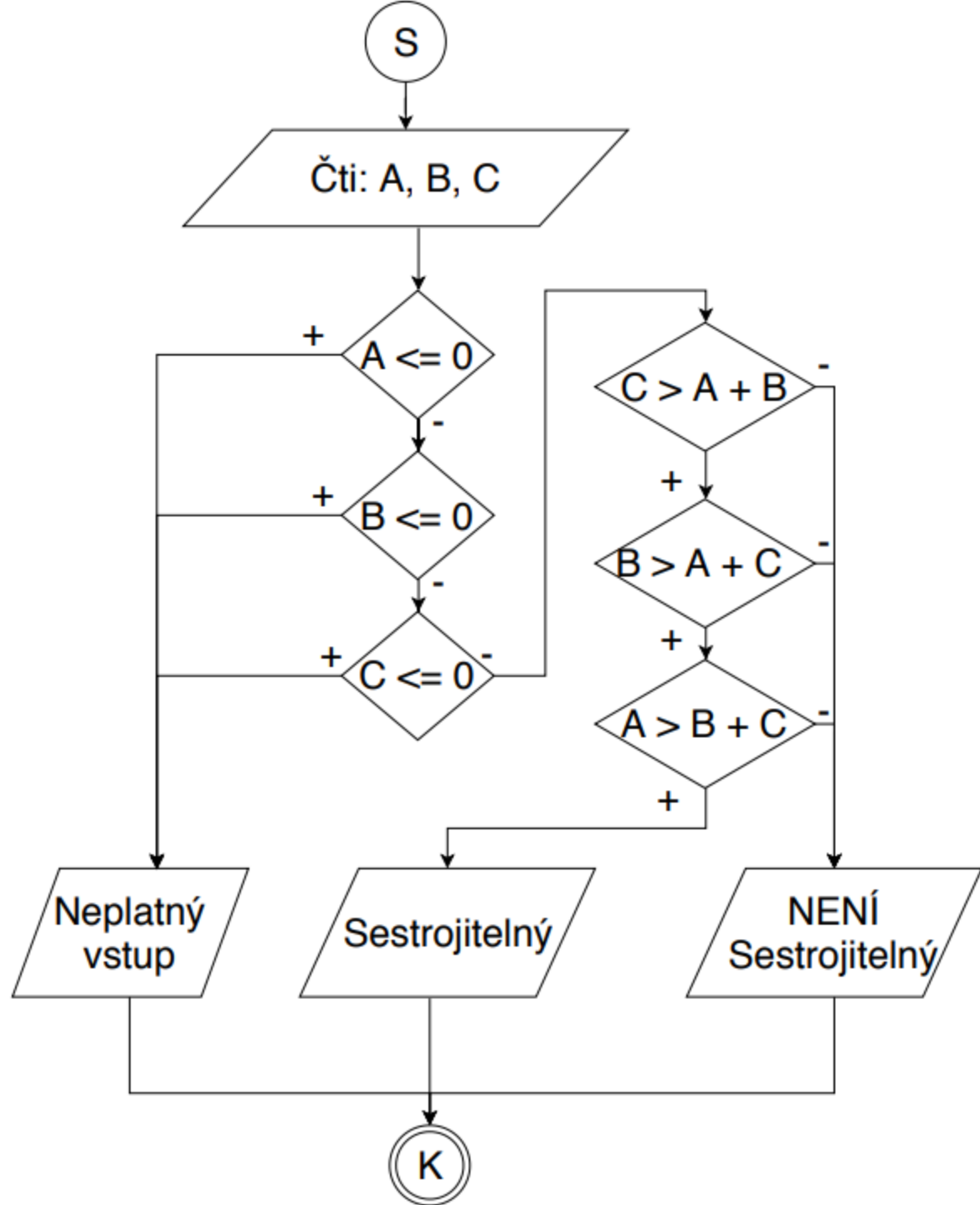
- 1) přečte **3 čísla** do proměnných **A, B, C**, které reprezentují strany trojúhelníku
- 2) vypíše na výstup:
 - „Sestrojitelný“, pokud lze trojúhelník sestrojít
 - „NENÍ sestrojitelný“, pokud NELZE trojúhelník sestrojít

Například:

Vstup: A = 1, B = 3, C = 5

Výstup: NENÍ sestrojitelný

Bonus: ošetřete pro záporná čísla na vstupu



Příště

- Trasovací tabulky
- Cykly

Příklady

Sestavte algoritmus, který vypočítá aritmetický průměr z posloupnosti celých kladných čísel končící nulou, která do posloupnosti nepatří.

a) předpokládejte, že posloupnost není prázdná

b) posloupnost může být prázdná (hned první přečtené číslo je 0).