

Programování v GIS 1

6 – grafický zápis algoritmu

Michal Kačmařík

A924, tel.: 5512

e-mail: michal.kacmarik@vsb.cz

<https://www.hgf.vsb.cz/548/cs/>

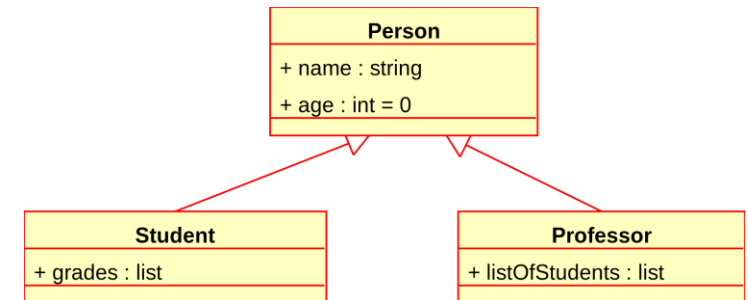
Zápis algoritmu

- Pomocí přirozeného jazyka (slovní popis),
- **pomocí grafického znázornění,**
- pomocí speciálního jazyka (pseudojazyk),
- pomocí programovacího jazyka.

- Algoritmus je popsán formalizovanou strukturou grafických objektů
- Výhody: přehlednost, názornost, poskytuje informace o postupu řešení
- Nevýhody: náročnost přípravy zejména v případě složitých a komplexních algoritmů, obtížná možnost editace při rozsáhlejších změnách

- strukturogram (kombinace grafického a textového popisu, tvořen obdélníkovou tabulkou)
- vývojový diagram (viz dále v prezentaci)
- jazyk UML

- UML = Unified Modeling Language
- univerzální jazyk pro vizuální návrh a popis programových systémů
- umožňuje vytvářet různé typy diagramů (diagram tříd, diagram objektů, diagram aktivit, diagram stavů, ...)
- z diagramů lze např. vygenerovat kostru programu
- open-source, standardizovaný



Ukázka diagramu tříd,

https://cs.wikipedia.org/wiki/Unified_Modeling_Language

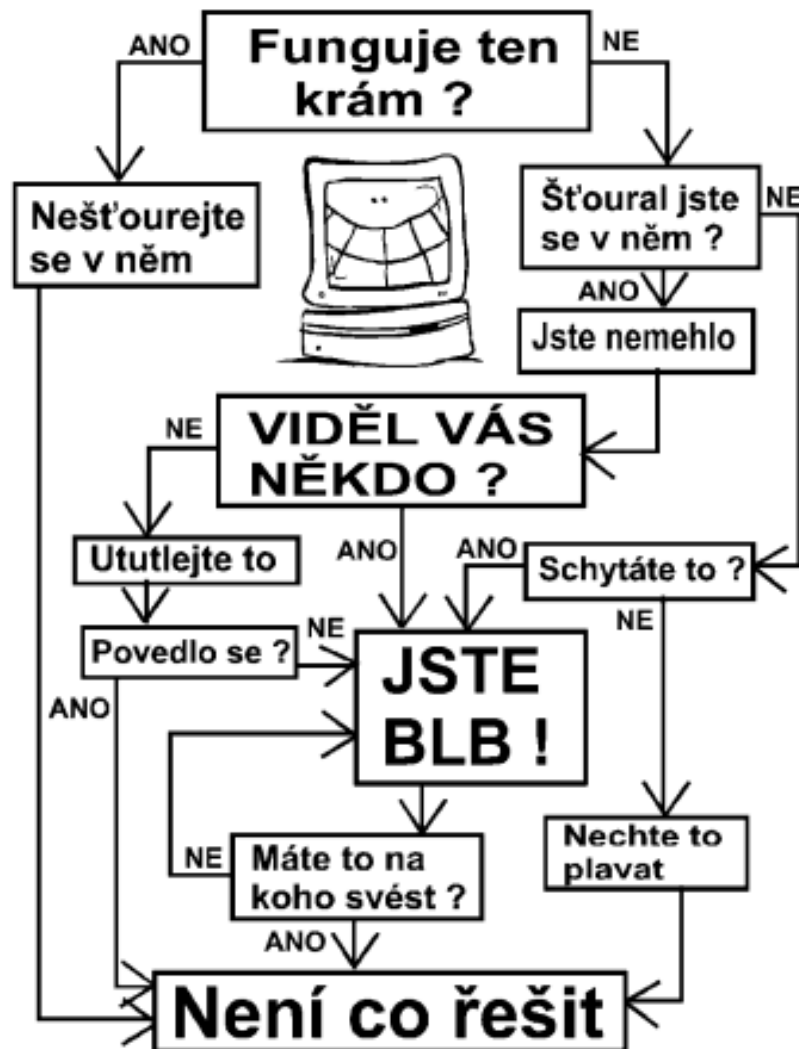
detailně v předmětu Objektově orientované technologie ve 3. ročníku

Vývojový diagram

- Vývojový diagram používá *symbols*, které jsou navzájem propojeny pomocí orientovaných hran.
- Symbols reprezentují jednotlivé kroky algoritmu, *šipky tok řízení*.

Vývojový diagram

TECHNOLOGICKÉ SCHÉMA ŘEŠENÍ PROBLÉMU

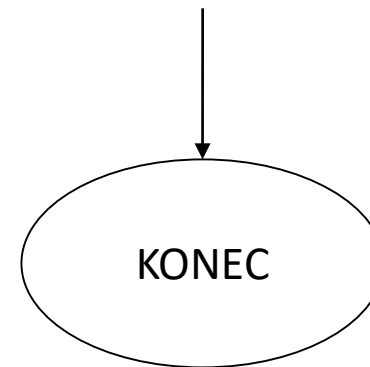
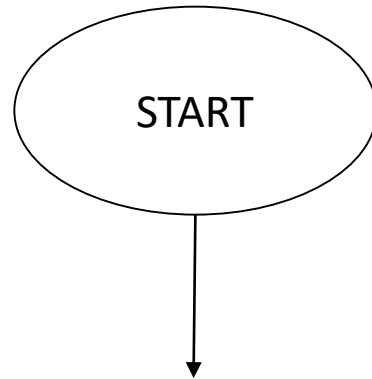


Symbols

- Význam symbolů představuje *sémantiku* jazyka.
- Pravidla pro jejich používání označujeme jako *syntaxi*.

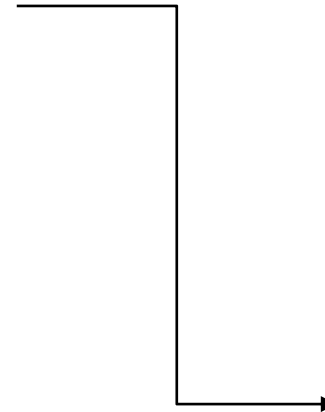
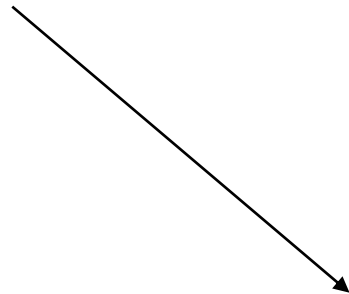
Startovací a ukončovací symboly

- Jsou znázorněny symbolem *oválem* nebo *obdélníkem se zaoblenými rohy*.
- Uvnitř je zapsán vysvětlující popis:
 - *Start, Příchod objednávky,*
 - *Konec, Objednávka stornována.*



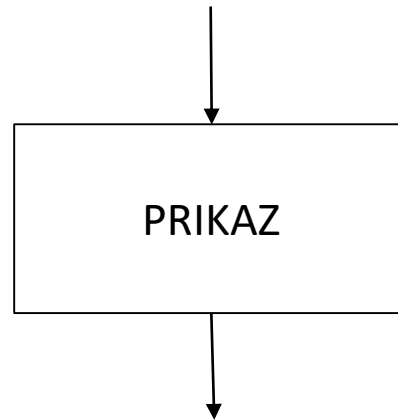
Hrany

- Zobrazují *tok řízení*.
- Hrana je opatřena šipkou, která směřuje od jednoho symbolu k symbolu následujícímu.



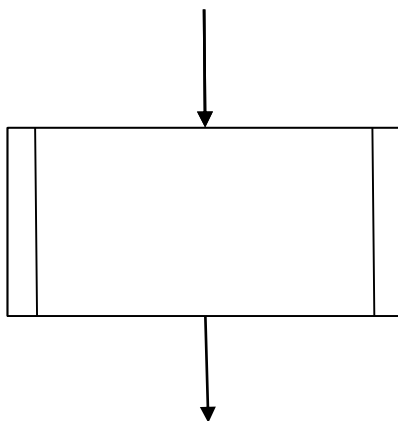
Dílčí krok algoritmu

- Značí se *obdélníkem*.
- Uvnitř je stručný zápis charakterizující činnost – krok algoritmu.
- Může jít o jednoduchý elementární příkaz přiřazení, nebo několik souvisejících příkazů.



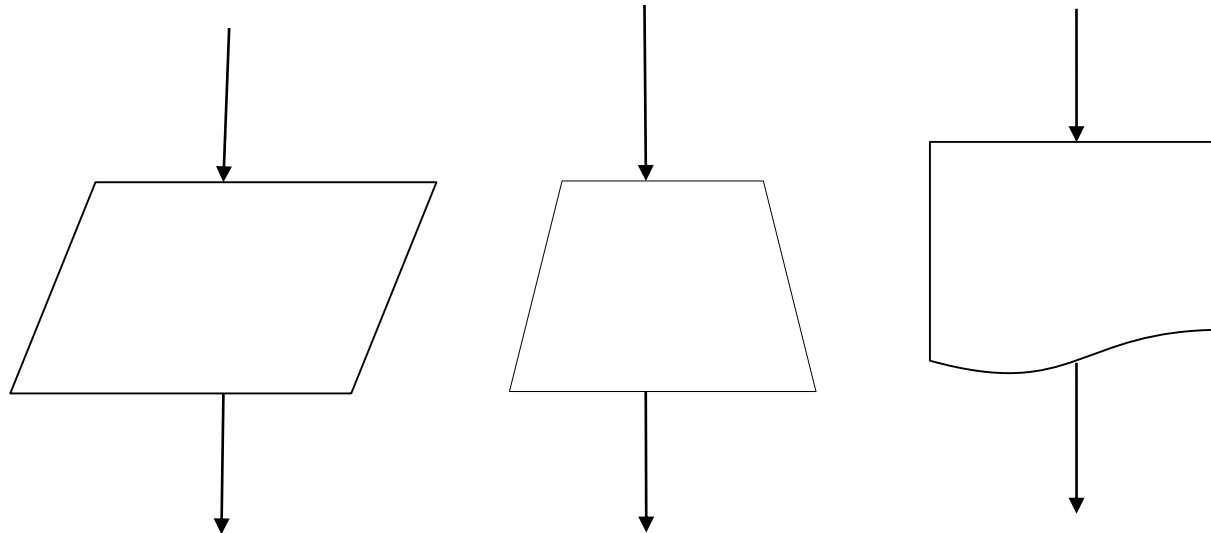
Podprogramy

- Symbolem *obdélníku se svislými čarami po stranách* je značen podprogram – skupina několika kroků algoritmu.
- Tato skupina kroků se může opakovat a může být rozepsána samostatným vývojovým diagramem.



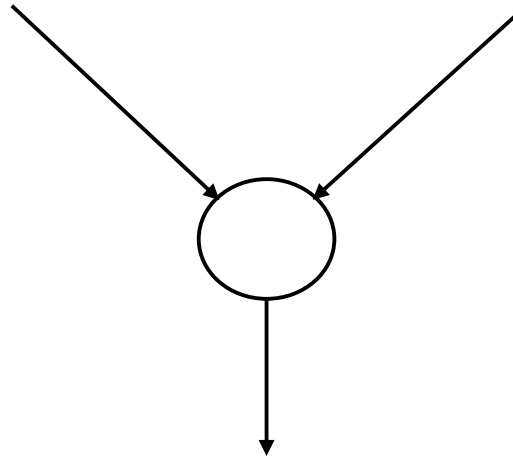
Vstup/Výstup

- *Rovnoběžník* symbolizuje *Vstup*.
- V případě ručně zadávaného vstupu používá *lichoběžník*.
- *Výstup* je obdélník s vlnkou místo spodní hrany.

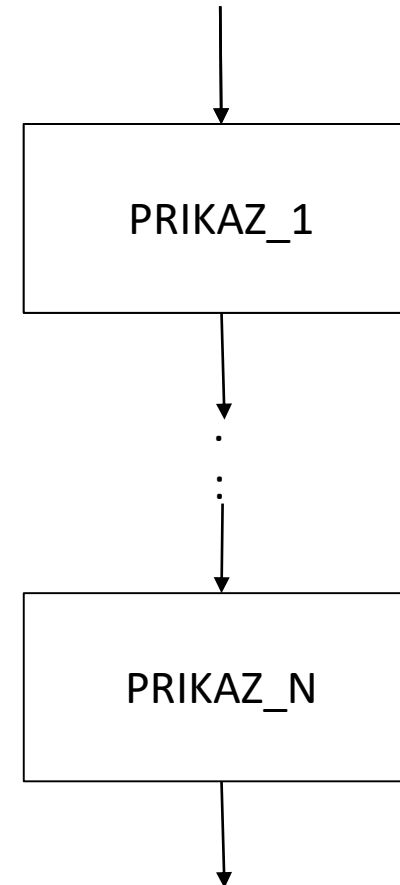


Spojovací značka

- *Kruh* představuje spojení více toků procesu do jednoho.



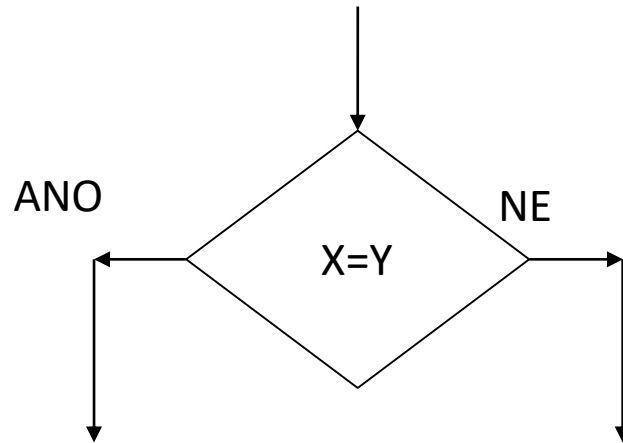
- *Sekvence* je tvořena posloupností obdélníků, ve kterých je popis kroků algoritmu.



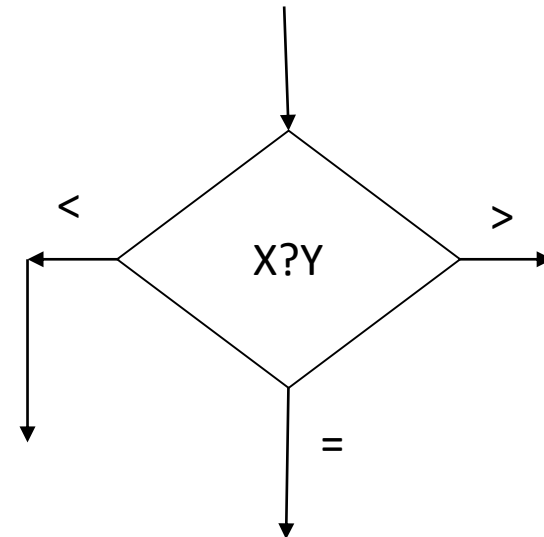
- *Kosočtverec* uvnitř s výrazem charakterizuje podmínku pro rozhodnutí. Vyhodnocení výrazu rozhoduje, která větev algoritmu se má dále vykonávat.
- Lze zapsat otázkou, kdy odpověď nabývá hodnoty *Boolean*.
- Výstupní hrany jsou pak nadepsány *Ano/Ne*.
- Z podmíněného výrazu může vycházet více hran než dvě.

Řídící struktury – Selekce

Podmínka – 2 větve

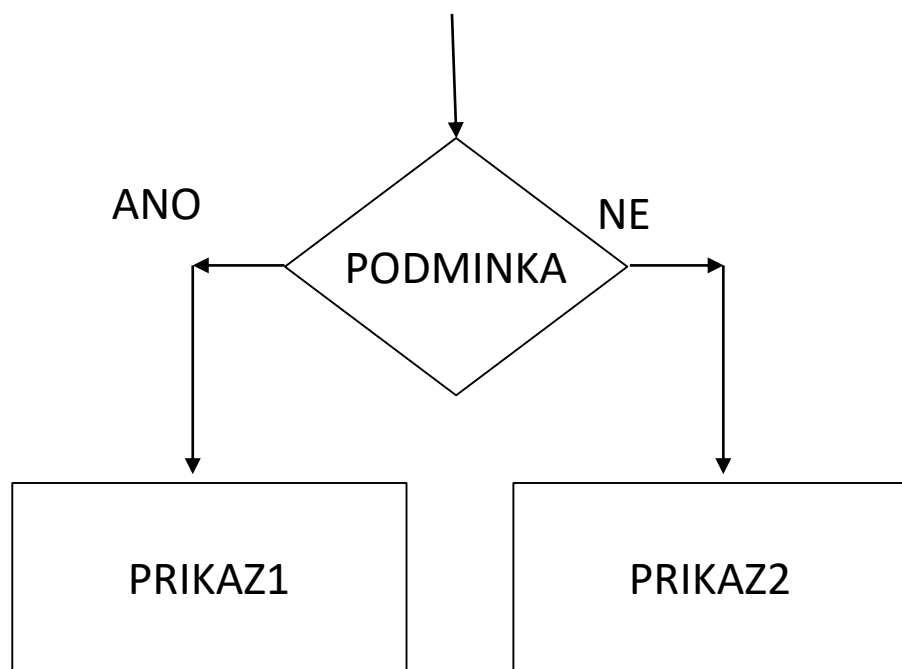


Podmínka – tři větve

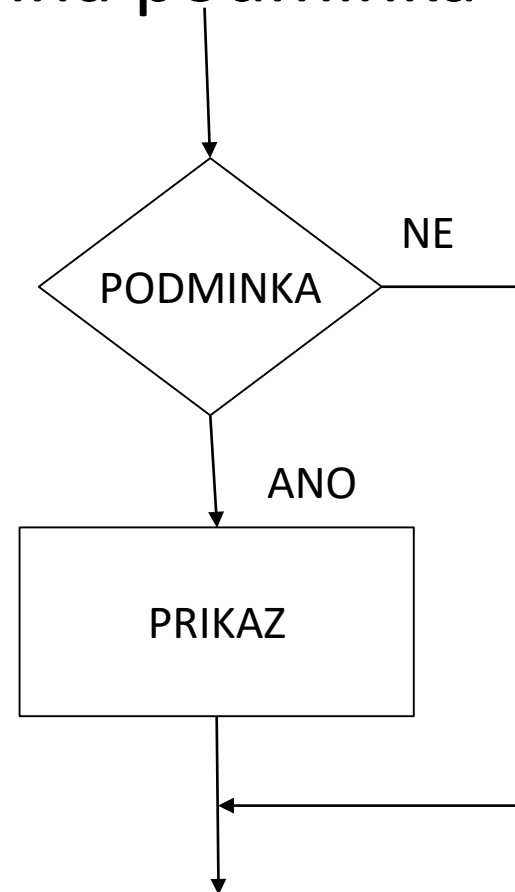


Řídící struktury – Selekce

Úplná podmínka

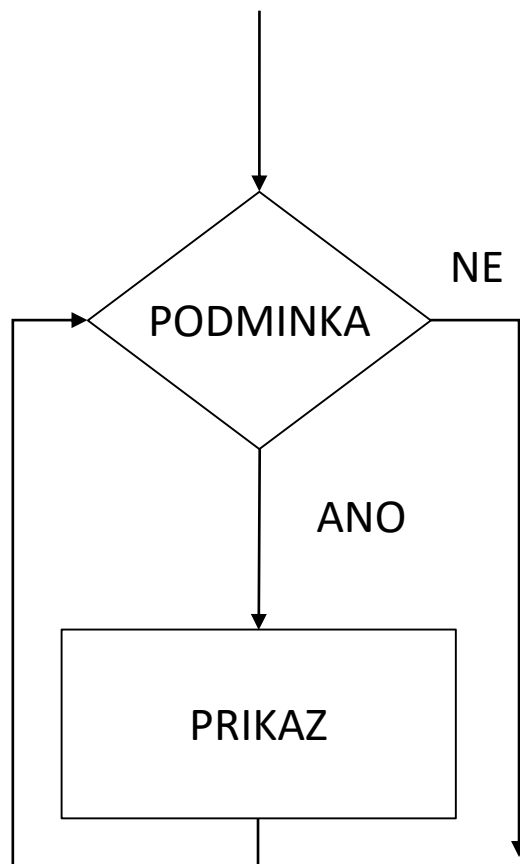


Neúplná podmínka



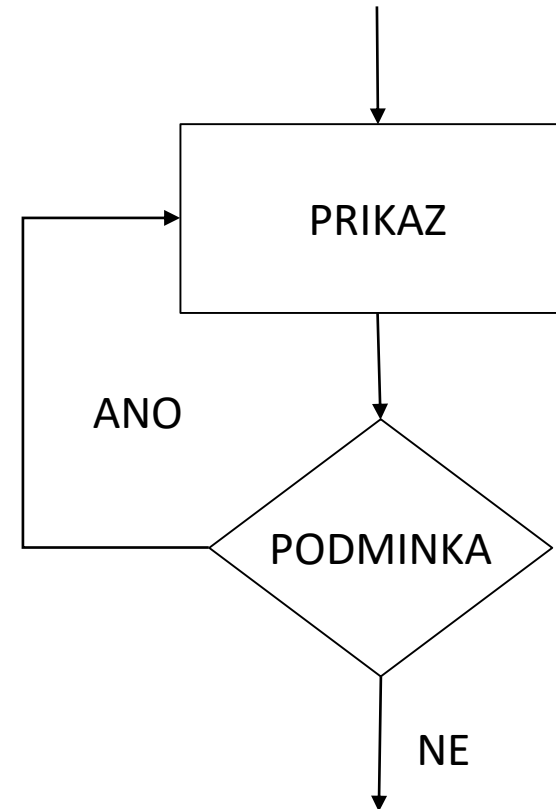
Řídící struktury – cyklus s podmínkou před vykonáním

- Průběh cyklu je podmíněn splněním podmínky před tělem cyklu.
- Není-li podmínka splněna, řízení přechází k dalšímu kroku algoritmu.



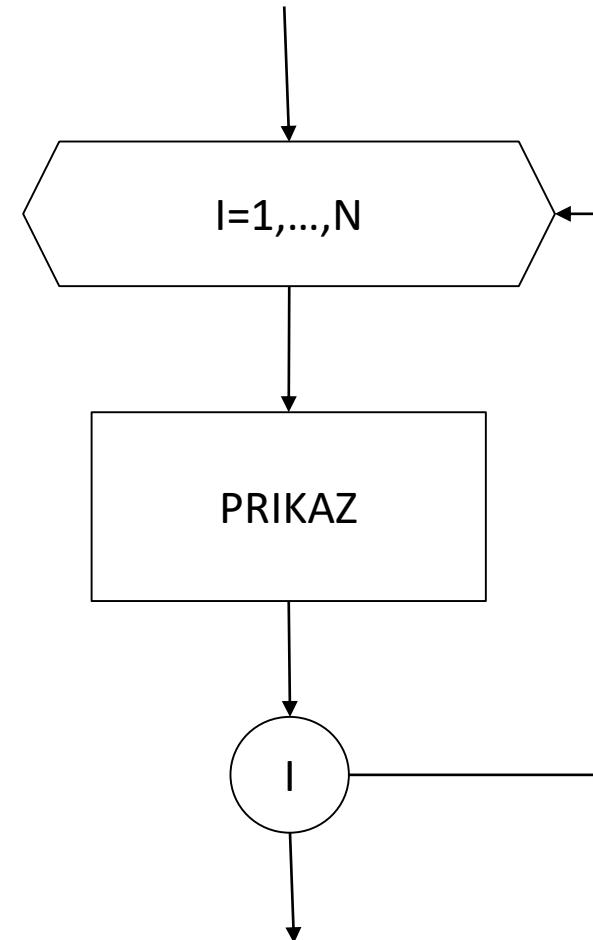
Řídící struktury – cyklus s podmínkou za vykonáním

- Průběh cyklu proběhne minimálně jednou, teprve pak se vyhodnotí podmínka.
- Je-li podmínka splněna, cyklus se opakuje, jinak se přejde k dalšímu kroku algoritmu.



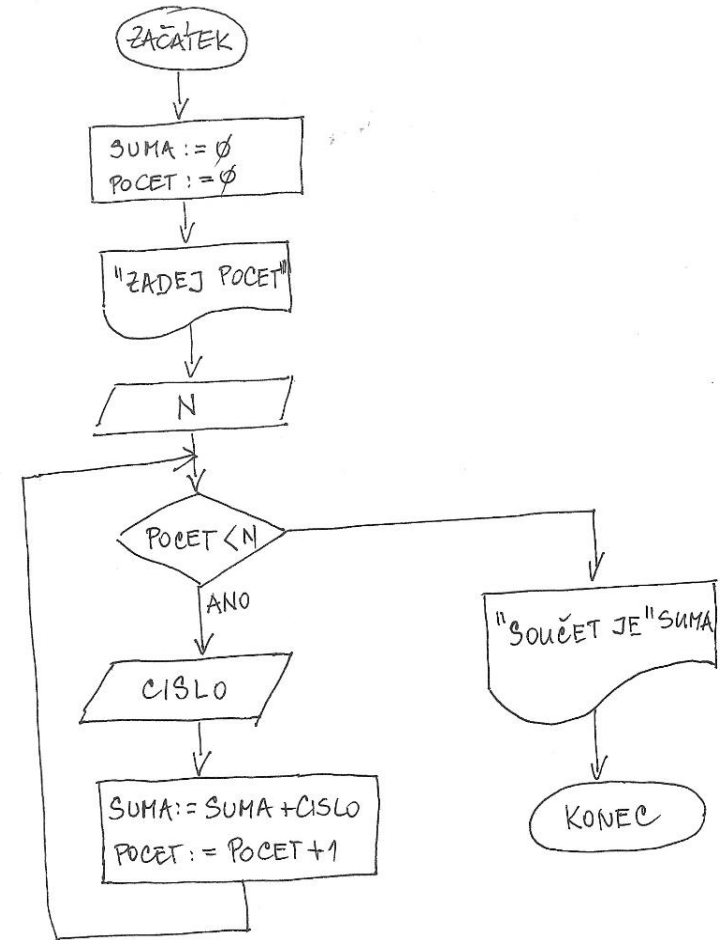
Řídící struktury – cyklus For

- Zaznamenává se *šestiúhelníkem*, za kterým následují příkazy těla cyklu, *řídící proměnnou* je možné uvést do kruhu.



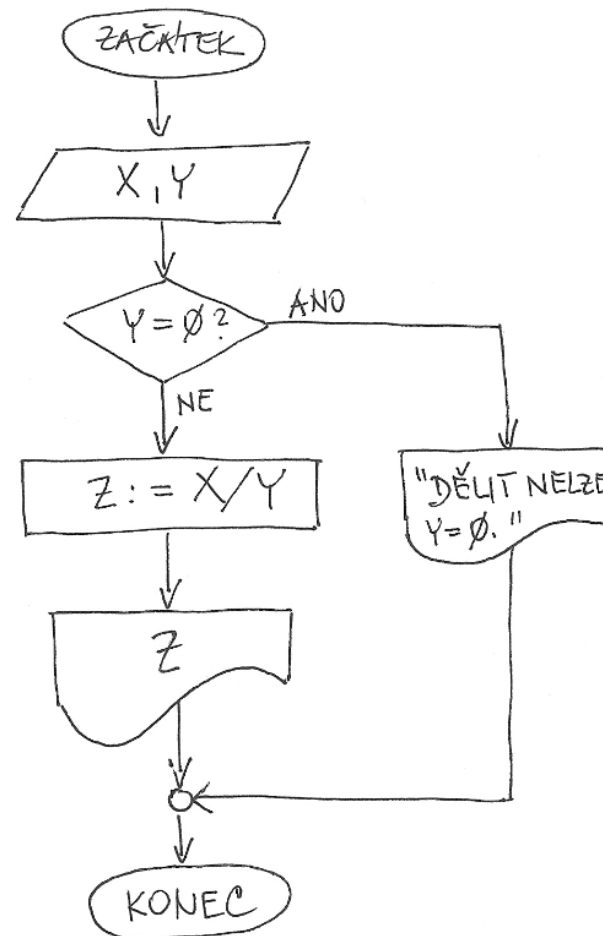
Příklad – Součet čísel

- Vstup: Počet čísel, která budou sčítána N, číselná hodnota zahrnutá do součtu CISLO.
- Výstup: součet zadaných čísel, SUMA.



Příklad – dělení dvou čísel

- Vstup: Čísla X, Y.
- Výstup: Podíl Z, nebo hlášení „Dělit nelze, Y=0.“



Příklad

Vytvořte cyklus, který bude na standardní výstup tisknout celá čísla od 1 do 10, a ke každému číslu bude na stejný řádek zároveň tisknout informaci o tom, zdali číslo je sudé, či liché.

Příklad

Na vstupu je dán seznam obsahující prvky různého datového typu. Vytvořte výstupní seznam a uložte do něj pouze prvky datového typu float a integer ze vstupního seznamu.

Příklad

S využitím cyklu vytvořte Fibonacciho posloupnost do čísla 100 a uložte ji do seznamu.

Děkuji za pozornost

Michal Kačmařík

michal.kacmarik@vsb.cz

www.vsb.cz