

Metodika převodu algoritmu do Turbo Pascalu

Hlavní prvky programu

- Program v Turbo Pascalu začíná **hlavičkou** s uvedením názvu programu. Název programu musí být odlišný od pojmenování ostatních prvků programu, především názvů proměnných.

```
program Název_Programu;
```

- Všechny **konstanty a proměnné**, které se vyskytují v algoritmu, je nutné uvést v deklarační části programu současně s uvedením jejich **datového typu** (typu hodnot, kterých může proměnná nabývat).

```
const název = hodnota;  
var název : datový_typ;
```

■ Základní datové typy:

■ integer	celé číslo v rozsahu -32768..32767
■ longint	celé číslo v rozsahu -2147483648..2147483647
■ byte	celé číslo v rozsahu 0..255
■ word	celé číslo v rozsahu 0..65535
■ real	reálné číslo v rozsahu 2.9E-39..1.7E38
■ char	jeden znak
■ string	řetězec max. 255 znaků
■ boolean	logická hodnota - true, false
■ dolní_mez .. horní_mez	interval hodnot

■ Datový typ pole

- **array [*index_od*..*index_do*] of *typ_položky***
- odkaz na jednotlivé položky pole prostřednictvím indexu: ***proměnná*[*index*]**

- **Hlavní program** tvoří příkazy uzavřené mezi klíčová slova **begin** a **end**. Za (posledním) endem v programu se píše **tečka**.

```
begin  
...  
end.
```

- Jednotlivé příkazy programu se **oddělují středníkem**.

- Přiřazení hodnoty proměnné se provádí pomocí **přiřazovacího příkazu**.

```
proměnná := výraz(hodnota)
```

- Běžné rovnítko (=) znamená v Turbo Pascalu vyhodnocení logického výrazu „rovnost“.

- Pro **vstup hodnot** (vložením z klávesnice) a **výstup hodnot** na obrazovku slouží příkazy:
 - `read(proměnná)` načtení hodnoty z klávesnice do proměnné
 - `readln(proměnná)` načtení hodnoty z klávesnice s přechodem na další řádek
 - `write(proměnná)` vypsání hodnoty proměnné na obrazovku
 - `writeln(proměnná)` vypsání hodnoty proměnné na obrazovku s přechodem na další řádek
 - `write('text')` výpis textového řetězce na obrazovku
- Pro ověření **syntaktické správnosti programu** lze použít volbu v prostředí Virtual Pascal:

Compile Alt+F9

- Spuštění programu** se provádí volbu v prostředí Virtual Pascal:

Run Ctrl+F9

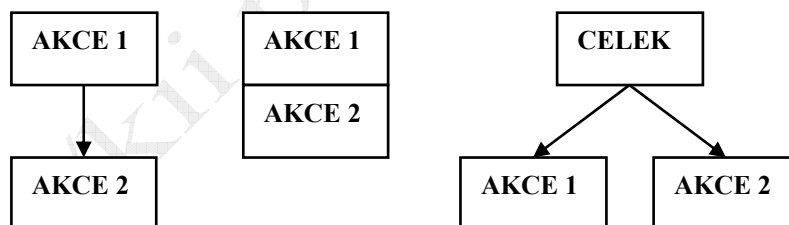
- Výstup programu je prováděn do samostatného okna, které se po ukončení programu uzavře. Pro pozdržení výstupu programu (čeká na stisk klávesy Enter) lze použít příkaz:

`readln`

uvedený na konci programu, před posledním endem.

Realizace základních struktur

- Sekvence** – v pořadí: *vývojový diagram, plošný strukturogram, strukturogram*

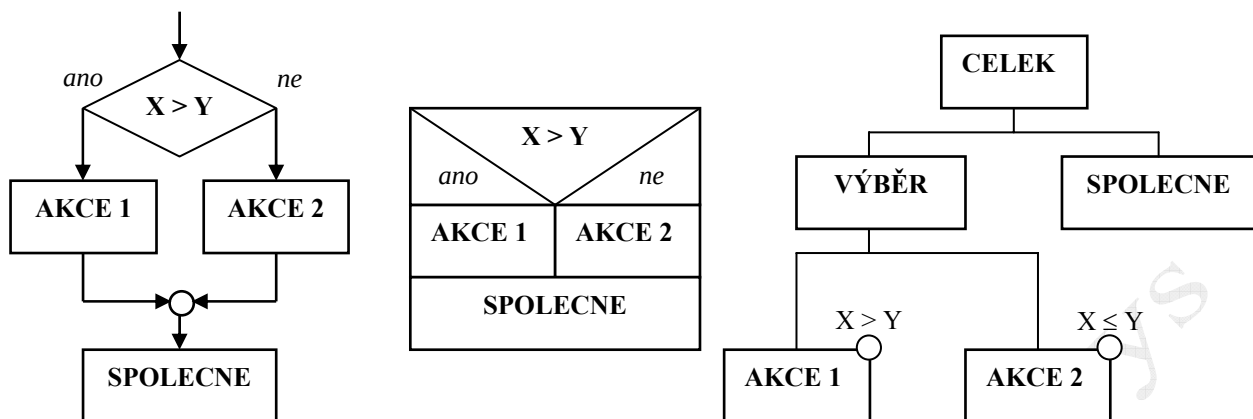


```

begin
  AKCE_1;
  AKCE_2;
  ...
end
  
```

- Příkazy zapsané mezi klíčová slova **begin** a **end** se chápou jako **jeden (složený) příkaz**

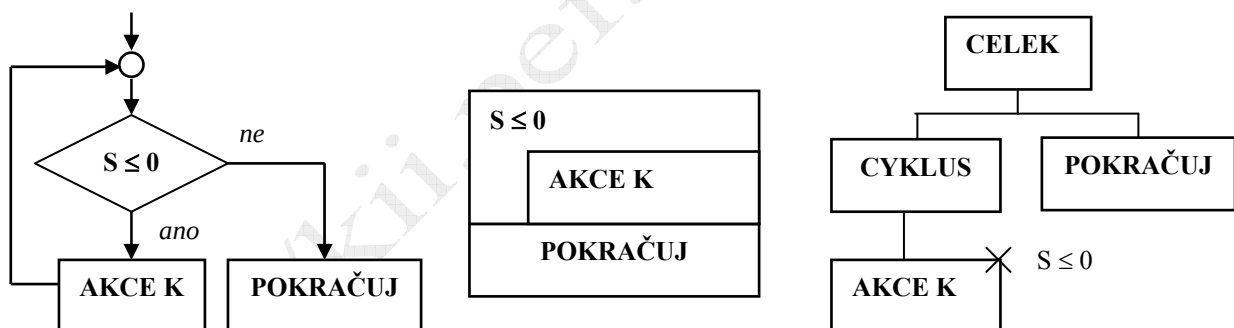
■ **Selekce** – v pořadí: vývojový diagram, plošný strukturogram, strukturogram



```
if X > Y then AKCE_1 else AKCE_2;
Spolecne
```

- Pokud je **AKCE_1** nebo **AKCE_2** složena z více příkazů (sekvence), musí být tyto příkazy uvozeny klíčovými slovy **begin** a **end** – složený příkaz.

■ **Iterace s podmínkou na začátku** – v pořadí: vývojový diagram, plošný strukturogram, strukturogram

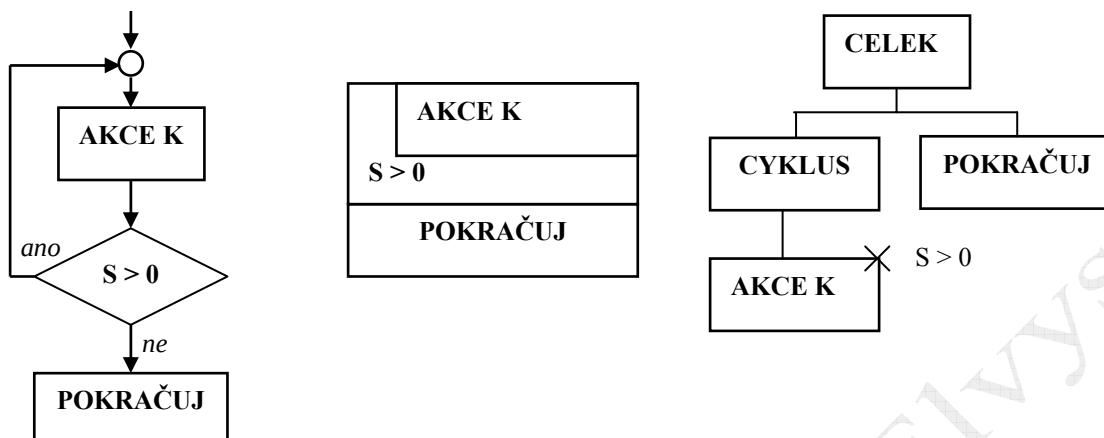


```
while S ≤ 0 do AKCE_K;
Pokracuj
```

- **Akce_K** může být opět složený příkaz uzavřený mezi **begin** a **end**.
- Speciálním případem cyklu s podmínkou na začátku je cyklus s pevným počtem opakováním:

```
for proměnná := počáteční to koncová do příkaz;
for proměnná := počáteční downto koncová do příkaz;
```

- **Iterace s podmínkou na konci** – v pořadí: vývojový diagram, plošný strukturogram, strukturogram



```
repeat AKCE_K until S <= 0;
Pokracuj
```

- Za klíčovým slovem **until** („dokud ne“) je třeba uvést **negaci podmínky** uvedené ve vývojovém diagramu.
- **Akce_K** může být opět složený příkaz uzavřený mezi **begin** a **end**.