06. Podmínka úplná a neúplná

Mezi každodenní činnosti člověka patří rozhodování. *Když bude hezky, půjdeme ven. Pokud dostanu jedničku z matiky, půjdu na pivo.* V algoritmizaci je podmínka jednou z nejpoužívanějších konstrukcí vůbec. Je velmi jednoduchá a zároveň účinná.

Podmínka v algoritmizaci je vlastně klasická otázka, na kterou je možné odpovědět pouze ano x ne/ pravda x nepravda.

SYNTAXE:

Používá se příkazu **jestliže … pak … jinak … ;** (if … then … else … ;)

```
jestliže {podmínka} pak {větev pro ano}
                    jinak {větev pro ne};
```

Je dobrým zvykem psát větev pro ano a větev pro ne pod sebe. Z důvodu přehlednosti to vřele doporučuji.

Syntexe je trochu delší, ale jednoduchá. Za příkazem ***jestliže*** bude následovat samotná podmínka, obvykle vyjádřená matematicky (x>7; y=2 apod.); za příkazem ***pak*** následuje příkaz nebo blok příkazů, které se mají provést, pokud je podmínka splněna; za příkaz ***jinak*** se píše příkaz nebo blok příkazů, které se provedou, pokud podmínka splněna není.

Podmínce, která má definované obě větve (tj. alespoň jeden příkaz je napsán u obou větví) se říká *podmínka úplná*. Někdy však bude stačit pouze větev pro ano. Taková podmínka je *neúplná*.

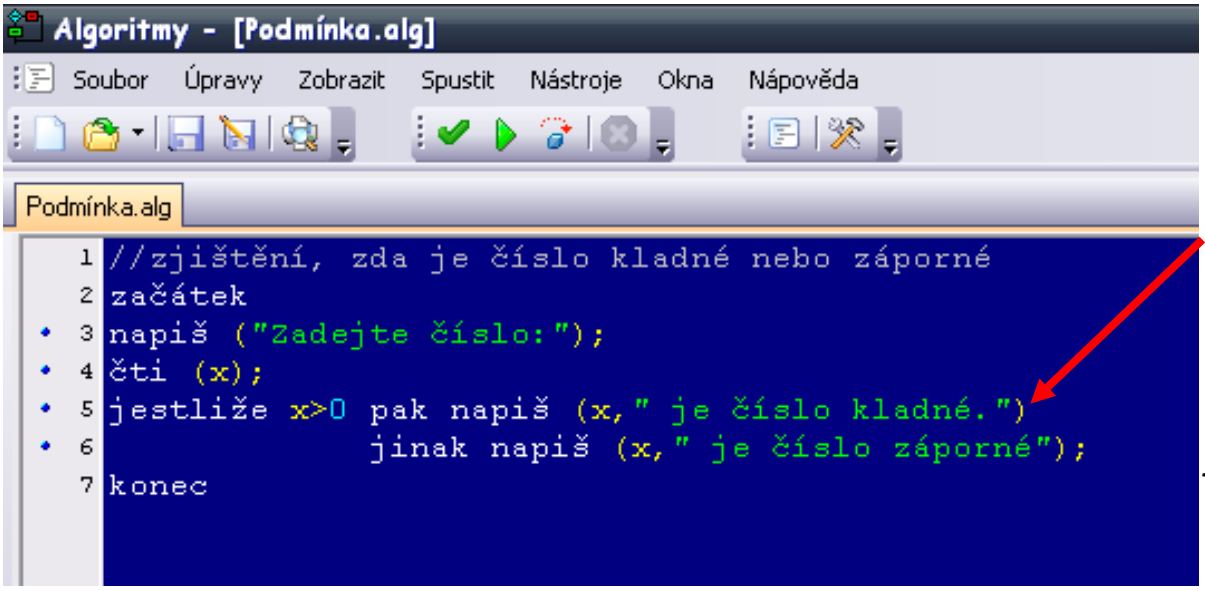
```
jestliže {podmínka} pak {větev pro ano};
```

Pokud je podmínka neúplná a program vyhodnotí podmínku jako nesplněnou (takže by v úplné podmínce přešel na větev pro ne), bude pokračovat dalším příkazem, jako by se nic nestalo.

Ve většině programovacích jazyků se nepoužívá matematické značení, ale nějaká jeho upravená verze. Následující tabulka platí pro program Algoritmy.

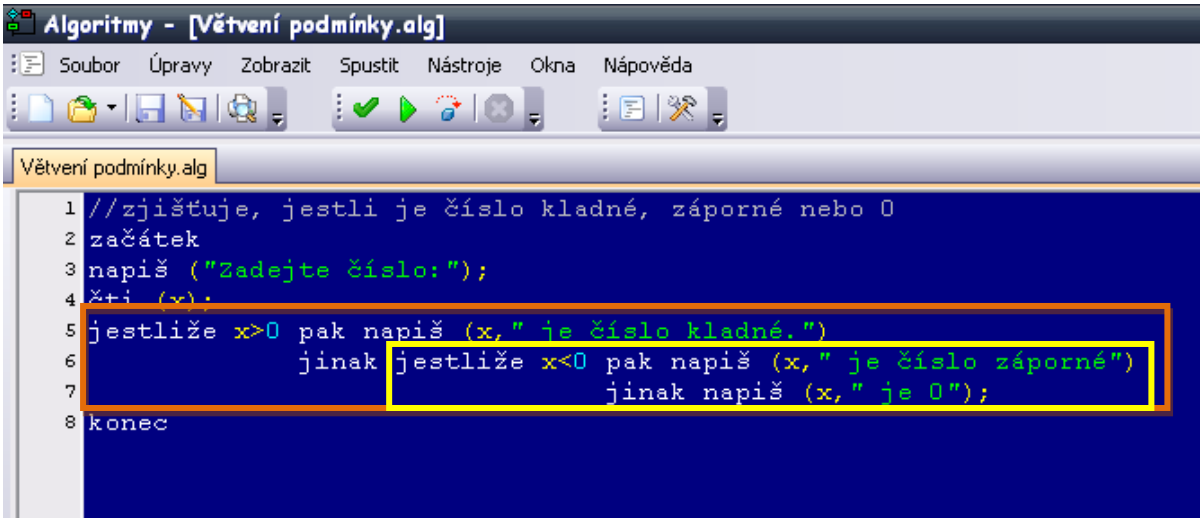
Algoritmy	Matematika	Význam
= (==)	=	rovná se
<> (!=)	≠	nerovná se
> >=	> ≥	větší / nebo rovno
< <=	< ≤	menší / nebo rovno
AND	∧	a zároveň
OR	∨	nebo
div	/	celočíselné dělení
mod	mod	zbytek po cel. dělení

Ukázkový program, který zjišťuje, jestli je číslo kladné nebo záporné:



Všimněte si, že zde není napsán středník. Kdyby byl, program by zahlásil chybu, protože příkaz ***jinak*** nemůže stát samostatně.

Program je to sice pěkný, ale nepočítá s možností, že zadané číslo by byla nula. Musíme ho tedy trochu rozšířit.



Dostali jsme se k zajímavé a velice praktické věci - vnořování podmínek. Klasická podmínka byla v tomto případě nedostačující, protože my potřebujeme tři možnosti reakce (kladné, záporné nebo nula). **Oranžový rámeček představuje první podmínku, žlutý druhou, vnořenou podmínku.** Je jedno, jestli bude podmínka vnořená do větve pro ano nebo do větve pro ne, výsledek bude stejný.

Úkoly:
1. Napište program, který určí, zda je celé číslo zadané uživatelem sudé nebo liché. (Budou se hodit funkce MOD a DIV viz výše)

2. Napište program, který určí větší ze dvou zadaných čísel (celých) a vypíše výsledek ve formátu *Číslo x je větší než číslo y.* nebo *Číslo x je menší než číslo y.*

3. Upravte program z úkolu 2. tak, aby rozpoznal i případ, kdy si jsou čísla rovna.

4. Napište program, který zjistí, kolik řešení má kvadratická rovnice ve tvaru $ax^2 + bx + c = 0$. Koeficienty a, b a c bude zadávat uživatel. Zajímá nás pouze počet řešení, ne řešení samotná.