Cvičenie 3 - Podmienený príkaz a cyklus

Podmienený príkaz

Základný príkaz má rovnako ako v ostatných programovacích jazykoch blokovú štruktúru. Jednotlivé bloky sú oddelené čiarkou.

V najjednoduchšej verzii má príkaz **If** štruktúru tvorenú dvomi blokmi. Bloky sú v tomto prípade: testovacia podmienka, blok True.

```
If [ podmienka,
    čo robiť ak je podmienka splnená]
```

V základnej verzii má príkaz **If** štruktúru tvorenú tromi blokmi. Bloky sú v tomto prípade: testovacia podmienka, blok True, blok False

```
If [ podmienka,
    čo robiť ak podmienka je splnená ,
    čo robiť ak podmienka nie je splnená]
```

Mathematica na rozdiel od ostatných programovacích jazykov, umožňuje aj tzv. rozšírenú syntax podmieneného príkazu **If** - teda programovú štruktúru, ktorá je tvorená štyrmi blokmi.

```
If [ podmienka,
    čo robiť ak podmienka je splnená ,
    čo robiť ak podmienka nie je splnená ,
    čo robiť ako nedokážem podmienku vyhodnotiť]
```

Ak nevie rozhodnúť, či je podmienka splnená, tak poskytne ako výsledok tretí blok.

Príklad

Zostavte program, ktorý porovná dve čísla *a, b.* Čísla *a* a *b* sú zadané programátorom na začiatku programu. Ak je číslo $a < b$ na obrazovku sa vypíše správa "číslo *a* je menšie ako číslo *b*".
Funkčnosť programu otestujte tak, že overíte funkčnosť všetkých jeho vetiev.

```
Clear[a, b]
a = 2;
b = 5;
If[a < b,
 Print["Číslo ", a, " je menšie ako číslo ", b]
  ]
```

Príklad

Zostavte program, ktorý porovná dve čísla *a, b.* Čísla *a* a *b* sú zadané programátorom na začiatku programu - predpokladáme zadanie rôznych čísel. Ak je číslo $a < b$ na obrazovku sa vypíše správa "číslo *a* je menšie ako číslo *b*". V opačnom prípade sa na obrazovku vypíše správa "číslo *b* je menšie ako číslo *a*".
Funkčnosť programu otestujte tak, že overíte funkčnosť všetkých jeho vetiev.

```
Clear[a, b]
a = 2;
b = 5;
If[a < b,
 Print["Číslo ", a, " je menšie ako číslo ", b],
 Print["Číslo ", b, " je menšie ako číslo ", a]
  ]
```

```
Clear[a, b]
a = 12;
b = 5;
If[a < b,
 Print["Číslo ", a, " je menšie ako číslo ", b],
 Print["Číslo ", b, " je menšie ako číslo ", a]
  ]
```

Príklad

Zostavte program, ktorý porovná dve čísla *a, b.* Čísla *a* a *b* sú zadané programátorom na začiatku programu. Ak je číslo $a < b$ na obrazovku sa vypíše správa "číslo *a* je menšie ako číslo *b*". V opačnom prípade sa na obrazovku vypíše správa "číslo *b* je menšie ako číslo *a*". V prípade, že program nevie rozhodnúť, ktoré z týchto dvoch tvrdení je pravdivé, vypíše o tom správu na obrazovku.
Funkčnosť programu otestujte tak, že overíte funkčnosť všetkých jeho vetiev.

```
Clear[a, b]
a = 2;
b = 5;
If[a < b,
 Print["Číslo ", a, " je menšie ako číslo ", b],
 Print["Číslo ", b, " je menšie ako číslo ", a],
 Print["Neviem rozhodnúť o pravdivosti tvrdenia"]
  ]
```

```
Clear[a, b]
a = 12;
b = 5;
If[a < b,
 Print["Číslo ", a, " je menšie ako číslo ", b],
 Print["Číslo ", b, " je menšie ako číslo ", a],
 Print["Neviem rozhodnúť o pravdivosti tvrdenia"]
  ]
```

```
Clear[a, b]
a = 12 + 2 I;
b = 5;
If[a < b,
 Print["Číslo ", a, " je menšie ako číslo ", b],
 Print["Číslo ", b, " je menšie ako číslo ", a],
 Print["Neviem rozhodnúť o pravdivosti tvrdenia"]
  ]
```

```
" Neviem rozhodnúť o pravdivosti tvrdenia "
```

Príklad

Zostavte program, ktorý testuje, či ľubovolné reálne číslo c patrí do intervalu (a, b), kde a, b sú ľubovolné reálne čísla také, že interval (a, b) má zmysel. Čísla a, b a c sú zadané programátorom na začiatku programu. Ak je číslo $c \in (a, b)$ na obrazovku sa vypíše správa "číslo c je z intervalu (a, b)". V opačnom prípade sa vypíše správa "číslo c nie je z intervalu (a, b)".
Funkčnosť programu otestujte tak, že overíte funkčnosť všetkých jeho vetiev.

```
Clear[a, b, c]
a = 4;
b = 6;
c = 5;
If[a < c < b,
     Print["číslo ", c, " je z intervalu (", a, ",", b, ")"],
     Print["číslo ", c, " nie je z intervalu (", a, ",", b, ")"]
    ]
```

```
číslo 5 je z intervalu (4,6)
```

```
Clear[a, b, c]
a = 4;
b = 6;
c = 15;
If[a < c < b,
     Print["číslo ", c, " je z intervalu (", a, ",", b, ")"],
     Print["číslo ", c, " nie je z intervalu (", a, ",", b, ")"]
    ]
```

```
číslo 15 nie je z intervalu (4,6)
```

Príklad

Zostavte program, ktorý testuje, či ľubovolné reálne číslo c patrí do intervalu (a, b), kde a, b sú ľubovolné reálne čísla také, že interval (a, b) má zmysel. Čísla a, b a c sú zadané programátorom na začiatku programu. Ak je číslo $c \in (a, b)$ na obrazovku sa vypíše správa "číslo c je z intervalu (a, b)". V opačnom prípade sa vypíše správa "číslo c nie je z intervalu (a, b)".
Program bude zároveň testovať, či je interval správne zadaný.
Funkčnosť programu otestujte tak, že overíte funkčnosť všetkých jeho vetiev.

```
Clear[a, b, c]
a = 7;
b = 6;
c = 5;
If[a < b,

 Print["Interval je zadaný správne."];
 If[a < c < b,
      Print[c, " je z intervalu (", a, ",", b, ")"],
      Print[c, " nie je z intervalu (", a, ",", b, ")"]
      ],

 Print["b < a, Interval nie je zadaný správne."]

   ]
```

```
Clear[a, b, c]
a = 3;
b = 6;
c = 5;
If[a < b,

 Print["Interval je zadaný správne."];
 If[a < c < b,
      Print[c, " je z intervalu (", a, ",", b, ")"],
      Print[c, " nie je z intervalu (", a, ",", b, ")"]
      ],

 Print["b < a, Interval nie je zadaný správne."]

   ]
```

Cyklus - Do, s pevným aj s variabilným počtom opakovaní

Základná štruktúra príkazu cyklu má tvar:

Do[blok príkazov, {List}]

alebo

```
Do[
    príkaz1;
    príkaz2;
```

príkaz3, {List}]

Blok príkazov rovnako ako v prípade podmieneného príkazu If môže byť tvorený jedným príkazom, alebo niekoľkými príkazmi oddelených bodkočiarkami. Za posledným príkazom v bloku bude nasledovať čiarka, ktorá hovorí, že teraz je koniec vykonávacej časti cyklu a nasleduje časť, v ktorej hovoríme koľko krát a ako sa má cyklus zopakovať. Táto časť môže mať tvar

- {n} - krát sa zopakujú príkazy v tele cyklu,
- {i, n} - riadiaca premenná i nadobúda hodnoty od 1 do a podľa toho sa aj cyklus opakuje,
- {i, štart, koniec} - riadiaca premenná i nadobúda hodnoty od štart hodnoty do koniec hodnoty a podľa toho sa aj cyklus opakuje,
- {i, štart, koniec, krok} - riadiaca premenná i nadobúda hodnoty **od** štartovacej hodnoty **do** koniec hodnoty s krokom krok a podľa toho sa aj cyklus opakuje

Príklad

Pomocou cyklu vygeneruj 4 náhodné reálne čísla z intervalu [0, 1].
Pomocou cyklu vygeneruj 4 náhodné reálne čísla z intervalu [5, 10].
Pomocou cyklu vygeneruj 4 náhodné celé čísla z intervalu [–2, 10].

```
Do[Print[Random[]], {4}]
```

```
Do[Print[Random[Real, {5, 10}]], {4}]
```

```
Do[Print[Random[Integer, {-2, 10}]], {4}]
```

Príklad

Lodné motory so vstrekovaním Common-Rail majú menovitý výkon P_m v rozsahu 12 000 až 70 000 kW. Počas plavby motor pracuje na výkon P rovný 75 % menovitého výkonu. Špecifická spotreba paliva je $m_p = 170\ g/\text{kWh}$. Zostavte program, ktorý vypíše na monitor tabuľku vyjadrujúcu závislosť spotreby paliva S_p [kg/h] od výkonu motora P . Počítajte s krokom $\Delta P = 4350$ kW.

NÁVOD :
Ak dosadíme špecifickú spotrebu v g/kWh a výkon v kW , spotrebu paliva v kg/h vypočítame podľa vzorca $S_p = \frac{m_p P}{1000}$. Hranice pre výkon P vypočítame zo vzťahu $P = 0,75 . P_m$

Nápad 1 - najzákladnejšie procedurálne programovanie - primitívny spôsob vytvorenia tabuľky

```
mp = 170;
Pmin = 12000;
Pmax = 70000;
krok = 4350;

Print["P[kW]     Sp[kg/h]"]
Print["___________________"]
Do[   P = 0.75 * Pm;
      Sp = mp * P / 1000;
      Print[P, "        ", Sp],
   {Pm, Pmin, Pmax, krok}
  ]
```

Nápad 2 - naprogramované pomocou definovania funkcie

```
Clear[mp, Pmin, Pmax, krok, Sp]
mp = 170;
Pmin = 12000;
Pmax = 70000;
krok = 4350;

Sp[P_] := (mp * 0.75 * P / 1000);
vysledok = Table[{0.75 * P, Sp[P]}, {P, Pmin, Pmax, krok}]
TableForm[vysledok,
 TableHeadings → {None, {"P [kW]", "Sp [kg/h]"}}, TableSpacing → {1, 4}]
```

Nápad 2A - naprogramované pomocou definovania funkcie

V úlohe je zbytočné definovať samostatne premenné Pmin, Pmax a krok . Stačí potrebné hodnoty dosadiť priamo do výpočtu

```
Clear[ Sp]

Sp[P_] := 170 * 0.75 * P / 1000;
vysledok = Table[{0.75 * P, Sp[P]}, {P, 12000, 70000, 4350}]
TableForm[vysledok,
 TableHeadings → {None, {"P [kW]", "Sp [kg/h]"}}, TableSpacing → {1, 4}]
```

Príklad

Pomocou cyklu vytvor tabuľku 10-tich dvojíc v tvare: n- poradie a x_n, ak $x_{n+1} = x_n + i$, kde i je náhodné číslo, $i \in [1, 5]$. Ako počiatočnú hodnotu zvoľte $x_0 = 0$.

```
x[0] = 0;
Do[x[n + 1] = x[n] + Random[Real, {1, 5}], {n, 0, 10}]

TableForm[Table[{n, x[n]}, {n, 0, 10}],
 TableHeadings → {None, {"n", "xn"}}, TableAlignments → Center]
```

Príklad

Vytvorte množinu *A* tvorenú 12-timi náhodných reálnych čísel z intervalu [1, 10]. Zistite

1. počet čísel z danej množiny, ktorých hodnota je menšia ako 5,
2. súčet čísel z danej množiny, ktorých hodnota je menšia ako 5 .
3. počet a súčet čísel z danej množiny, ktorých hodnota je menšia ako 5 .

Vytvoríme množinu *A* s požadovanými vlastnosťami. Ukážeme si, ako sa dajú z množiny A vybrať jednotlivé prvky.

```
A = Table[Random[Real, {1, 10}], {i, 1, 12}]
```

Toto je nová vec - preberá sa až na prednáške venovanej poliam. Prvok z poľa vyberieme pomocou konvencie o dvoch zátvorkách. Vyberieme prvý prvok, piaty prvok a deviaty (štvrtý od konca) prvok

```
A[[1]]
```

```
A[[5]]
```

```
A[[-4]]
```

1. Zistite počet čísel z danej množiny, ktorých hodnota je menšia ako 5.

Nápad 1

```
Clear[pocet]
pocet = 0;
Do[
     If[A[[i]] < 5, pocet = pocet + 1],
 {i, 1, 12}]
Print["Pocet cisel mensich ako 5 je ", pocet]
```

Nápad 2 - používame konvenciu z programovacieho jazyka C

```
Clear[pocet]
pocet = 0;
Do[
     If[A[[i]] < 5, pocet++],
 {i, 1, 12}]
Print["Pocet cisel mensich ako 5 je ", pocet]
```

2. Zistite súčet čísel z danej množiny, ktorých hodnota je menšia ako 5.

Nápad 1

```
Clear[sucet]
sucet = 0;
Do[
     If[A[[i]] < 5, sucet = sucet + A[[i]]], {i, 1, 12}]
Print["Sucet cisel mensich ako 5 je ", sucet]
```

Nápad 2 - používame konvenciu z programovacieho jazyka C

```
Clear[sucet]
sucet = 0;
Do[
     If[A[[i]] < 5, sucet += A[[i]]], {i, 1, 12}]
Print["Sucet cisel mensich ako 5 je ", sucet]
```

3. Zistite počet a súčet čísel z danej množiny, ktorých hodnota je menšia ako 5.

Nápad 1

```
Clear[pocet]
pocet = 0;
sucet = 0;
Do[
     If[A[[i]] < 5, pocet = pocet + 1;
                    sucet = sucet + A[[i]]
       ],
 {i, 1, 12}]
Print["Pocet cisel mensich ako 5 je ", pocet]
Print["Sucet cisel mensich ako 5 je ", sucet]
```

Nápad 2 - používame konvenciu z programovacieho jazyka C

```
Clear[pocet]
pocet = 0;
sucet = 0;
Do[
     If[A[[i]] < 5, pocet++;
                    sucet += A[[i]]
       ],
 {i, 1, 12}]
Print["Pocet cisel mensich ako 5 je ", pocet]
Print["Sucet cisel mensich ako 5 je ", sucet]
```