

## **Č aneb C s mäkčeňom aneb c\_s\_makcenom**

Tento jazyk je určený pre komplikovanie slovenského jazyka, minimálne jeho veľmi divnej podmnožiny. Svoju filozofiu zdieľa s Antonom Bernolákom, ktorý sa pri kódifikácii riadil heslom „Programuj ako počuješ“... alebo niečo také.

V jazyku bude niekedy kľúčové slovo mať viaceré variantné koncovky, ktoré bude môcť užívateľ použiť ľubovoľne, aby sedeli ku gramatickému rodu (nebude kontrolované jazykom, je to len pre krásu).

Statement sa povie v Ččku „výrok“ – začína sa veľkým písmenom a končí sa bodkou. Veľkosť písmen nie je v jazyku okrem tohto prípadu dôležitá.

Názvy premenných je odporúčané písat' v slovenčine, ideálne používajte slová, ktoré majú rovnaký nominatív a akuzatív (aby programy zneli prirodzenejšie).

Výrok zabalený do okrúhlych zátvoriek je komentár:  
(Poznámka pre seba, pridaj dokumentáciu.)

### **Typy:**

- Celé číslo = 32 bitový signed integer
- (Reálne číslo = double, možno nepridám)
- Pravdivosť = bool, môže mať hodnoty pravda/áno a lož/nepravda/nie
- Text = string
- Znak = znak (pravdepodobne veľkosti uint8, možno UTF-32 znak, ešte neviem)

Meno premennej musí byť jedno slovo, môže používať iba znaky slovenskej abecedy.

### **Priradenie:**

Názov premennej nasledovaný textom "bude" alebo "budú" a hodnotou alebo inou premenou:

*Počet bude 4.*

*Privítanie bude "Pozdravený bud', svet môj!" .*

*Používateelia budú 4.*

V prípade priradenia pravdivostí je možné použiť aj „platí keď“:

*Lacné platí keď cena je menšia ako 20.*

### **Deklarácia:**

Začína sa kľúčovým slovom „Majme“ nasledovaným názvom typu v akuzatíve a potom názvom premennej:

*Majme reálne číslo výška.*

*Majme podmienku hotové.*

V tom prípade bude mať východiskovú hodnotu typu. Ak chceme rovno priradiť hodnotu, napíšeme za ňu čiarku a „ktorý/á/é/í bude/ú“ nasledované hodnotou:

*Majme celé číslo počet, ktoré bude 3.*

*Majme text popis, ktorý bude "Vitaj".*

*Majme reálne číslo výška, ktorá bude 0 celých 5.*

Áno, desatinný oddeľovač pri reálnych číslach je skutočne slovo „celých“, „celá“ alebo „celé“.

**Výstup:**

Konštanty vypíšeme príkazom „Vypíš“:  
Vypíš „Ahoj!!!“.

Výpis nepridá nový riadok, iba medzeru. Na nový riadok je nutné použiť buď príkaz Odriadkuj . alebo po vypísaní napísať „a odriadkuj .“ takto:  
Vypíš číselko a odriadkuj .

**Vstup:**

Vstup získavame príkazom „Načítaj <vstupný typ> do premennej <názov existujúcej premennej>“:  
Načítaj celé číslo do premennej výška .

Vstupné typy sú:

- Celé číslo
- (Reálne číslo = akceptuje zápis s desatinnou bodkou, čiarkou aj slovom „celá“ atď... uvidím, či pridám)
- Pravdivosť = akceptuje hodnoty pravda/áno/ano/A/Y a lož/nie/N
- Riadok
- Znak
- Slovo = text ukončený medzerou, čiarkou, bodkočiarkou alebo bodkou

**Aritmetické operácie:**

plus = +

mínus = -

krát = \*

deleno = /

záporné/ý/á/í = unárne mínus

zátvorky sú bežné okrúhle ako v Českodních jazykoch.

Príklad:

*Obsah* bude výška krát šírka .

*Nasledovník* bude *nasledovník* plus 1 .

**Relačné operácie**

je menšie/a/í ako = <

je väčšie/a/í ako = >

je menšie/a/í alebo rovné/á/ý ako = <=

je väčšie/a/í alebo rovné/á/ý ako = >=

sa rovná ==

sa nerovná !=

Príklad:

Pravdivosť použiteľné platí ak výška je väčšia alebo rovná ako šírka.

### Logické operácie

X a Y = X AND Y

X či Y = X OR Y

X alebo Y = X XOR Y

nie X = opak X = NOT X

Som za to, že slovo „či“ lepšie vyjadruje inkluzívnosť než slovo „alebo“, ktoré ľudia berú veľmi exkluzívne.

### Polia (a texty)

Deklarácia pola začína slovom „Zoznam“ a názvom typu v genitíve množného čísla:

- Zoznam celých čísel
- Zoznam reálnych čísel
- Zoznam pravdivostí
- Zoznam textov
- Zoznam znakov

Zoznam znakov je to isté ako text, preto aj Č akceptuje viac možností prístupu k položkám. Obsah zoznamu vieme ľahko vymenovať tak, že vymenujeme čiarkami oddelené hodnoty v hranatých zátvorkách:

Majme zoznam celých čísel výšky, ktorý bude [1, 2, 3].

Prístup k prvkom poľa realizujeme slovnými spojeniami:

<názov premennej>-tý/ty prvak zoznamu <názov zoznamu>

<názov premennej>-tý/ty znak textu <názov textu>

prvak zoznamu <názov zoznamu> na pozícii (<názov premennej>) znak textu <názov textu> na pozícii (<názov premennej>)

Zátvorky sú nutné, lebo pri viacrozmerných poliach by to robilo problém.

Vieme takto čítať aj zapisovať z/do zoznamu:

Majme celé číslo hodnota, ktoré bude 3.

Majme celé číslo aktuálny, ktoré bude 8.

Hodnota bude aktuálny-ty prvak zoznamu výšky.

Aktuálny-ty prvak zoznamu výšky bude hodnota.

Hodnota bude prvak zoznamu výšky na pozícii (aktuálny).

Špeciálne kľúčové slová „prvý/á/é/í“ a „posledný/á/é/í“ umožňujú pristupovať (neprekvapivo) k prvému a poslednému prvku poľa:

Hodnota bude prvý prvak zoznamu výšky.

Písmenko bude prvý znak textu pozdrav.

Dĺžku zoznamu získame ako

dĺžka zoznamu <názov zoznamu>

Dĺžku textu získame ako

dĺžka textu <názov textu>

### Dynamická práca so zoznamom

Vieme pridať dynamicky do poľa hodnoty:  
Pridaj na začiatok zoznamu výšky prvok *hodnota*.  
Pridaj na koniec zoznamu výšky prvok *hodnota*.

A rovnako aj odoberať:  
Zmaž prvý prvok zoznamu výšky.  
Zmaž posledný prvok zoznamu výšky.

### Viacrozmerné polia

Pre viacrozmerné polia vložíme medzi „Zoznam“ a názov typu ďalšie slová „zoznamov“:  
Majme zoznam zoznamov zoznamov celých čísel kocky, ktorý bude  
[[[0, 1], [20]], [[5]]].

Na prístup k prvkom viacrozmerného poľa je nutné použiť iba konštrukciu  
prvok zoznamu <názov zoznamu> na pozícii (<názov premennej 1>,  
<názov premennej 2>, ... <názov premennej n>)  
Kocka bude prvok zoznamu *kocky* na pozícii (*výška*, *šírka*, *hĺbka*).

### Podmienky

Ak <podmienka s výsledkom pravda/lož> tak <príkazy>. Ak <podmienka s výsledkom pravda/lož> tak <príkazy>, inak <príkazy>.

Príklad:

Ak *cena* je menšia ako 10 tak vypíš „lacné!“, inak vypíš „drahé!“.

Pri postupnosti príkazov využijeme dvojbodku nasledovanú odsadenými riadkami.  
Odsadenie je možné vykonať buď medzerami alebo tabulátormi, ale komplilátor odmietne ich kombinovanie.

Ak *cena* je menšia ako 10 tak:  
    Vypíš „lacné!“.

Inak:  
    Vypíš „drahé!“.

V tomto prípade nie je pred Inak čiarka.

### Cykly

#### For cyklus:

Ak chceme iteračnú premennú:

Opakuj pre <novovytvorená celočíselná iteračná premenná> od <prvé>  
po <posledné> <príkazy>.

Ak nepotrebujeme iteračnú premennú:

Opakuj od <prvé> po <posledné> <príkazy>.

Príklad:

Opakuj pre aktuálne od 1 po dĺžka mínuš 1 vypíš akutálne.

Pre prípad potreby viacerých príkazov je tiež možné použiť dvojbodku a odsadené riadky:

Opakuj pre aktuálne od 1 po dĺžka mínuš 1:  
Vypíš aktuálne.

### **While cyklus:**

Kým <pravdivostná hodnota> tak <príkaz>.

Príklad:

Kým počet je väčší ako 0 tak počet bude počet mínuš 1.

Kým počet je väčší ako 0 tak:

Vypíš počet.

Počet bude počet mínuš 1.

### **Riadenie toku:**

- „Koniec.“ ukončí beh programu.
- „Dlabat.“ funguje ako break v cykle
- „Preskoč.“ funguje ako continue v cykle

### **Funkcie:**

Deklarácia funkcie:

Funkcia <názov funkcie> vracajúca <typ> berie <typ argumentu 1> <názov argumentu 1>, ... <typ argumentu n> <názov argumentu n> a robí: <príkazy>.

Odporúča sa nazývať funkcie ako slovesá v prikazovacom spôsobe druhej osoby singuláru.

Pre void funkcie píšeme iba Funkcia <názov funkcie> nevracajúca nič berie...

Na návrat hodnoty z funkcie používame Vráť <výraz>.

Ak nič nevraciame, používame Hotovo.

Príklad:

Funkcia sčítaj vracajúca celé číslo berie celé číslo jedno, celé číslo druhé a robí:

Vráť jedno plus druhé.

Použitie funkcie:

Malé bude 5.

Veľké bude 15.

Výsledok bude sčítaj (malé, veľké).

A pre funkcie čo nič nevracajú:  
Zakrič („Ahoj“, 5).