Dokumentace NewSpeaku

Ployer David

Obsah

[Spuštění programu 2](#_Toc66388174)

[Proměnné 2](#_Toc66388175)

[Typy proměnných 2](#_Toc66388176)

[Jak vytvořit proměnnou 2](#_Toc66388177)

[Další informace pro proměnné 3](#_Toc66388178)

[Ukázka kódu: 3](#_Toc66388179)

[Vstupy a výstupy 3](#_Toc66388180)

[Výpis proměnných 3](#_Toc66388181)

[Čtení proměnných 3](#_Toc66388182)

[Ukázka kódu 4](#_Toc66388183)

[Matematické operace 4](#_Toc66388184)

[Matematické operace pro Int 4](#_Toc66388185)

[Matematické operace pro boolean 4](#_Toc66388186)

[Matematické operace pro string 4](#_Toc66388187)

[Ukázka kódu 4](#_Toc66388188)

[Podmínky 5](#_Toc66388189)

[Ukázka kódu 5](#_Toc66388190)

[Ukázka kódu 2 5](#_Toc66388191)

[Smyčky 5](#_Toc66388192)

[Ukázka kódu 5](#_Toc66388193)

[Ukázka kódu 2 6](#_Toc66388194)

[Ukázka kódu 3 6](#_Toc66388195)

[Metody 6](#_Toc66388196)

## Spuštění programu

Pro spuštění vašeho programu naprogramovaného v jazyce NewSpeak stačí \*.txt soubor umístit do stejné složky, jako je .exe soubor interpretera jazyka. Poté co je tam soubor umístěn stačí interpretor s pustit a napsat do něj název souboru, který chcete spustit.

## Proměnné

### Typy proměnných

Jazyk NewSpeak má hned několik typů proměnných. Těmito typy jsou:

* String: řetězec jakýchkoliv znaků. (text) například: “string”, “Ahoj, jak se máš?”, “String 123”
* Int: celé číslo -2 147 483 648 až 2 147 483 648. Například: 10, -851, 55612
* Boolean: Pravda/Nepravda ve výrokové logice. Například: true, false

### Jak vytvořit proměnnou

Proměnné se vytvářejí jednoduchým způsobem, že napíšete jméno, které chcete pro proměnnou, za to napíšete znaménko rovná se a hodnotu, které chcete, aby so proměnná rovnala.

Jména proměnných může být jakákoliv kombinace znaků, která se v sobě nemá mezery a nezačíná číslem, nebo speciálním znakem:

**Povoleno:** Promena1, hodnota, int1, bool5

**Nepovoleno**: 8Promeněna, $hodnota, promena jedna

Pro přidělení hodnoty proměnné si musíme ukázat, jak hodnoty zapsat:

* String se zapisuje jako text s hvězdičkou jako první znak, například:   
   \*string  
   \*Dobrý den  
   \*165484
* Int zapisujeme jako číslo, bez jakéhokoliv jiného označení, například: 15, 25, 1587, -88
* Boolean zapisujeme jako hodnotu true, nebo false

Ukázky vytváření proměnných:

* Promenna1 = \*Proměnná
* Int0 = 15
* Ahoj = true
* String1 = \*Pokusný string
* Boolean = 3

### Další informace pro proměnné

Pokud chcete změnit hodnotu proměnné stačí napsat jméno již vytvořené proměnné, za to napsat znaménko rovná se, a nakonec novou hodnotu, kterou tomu chcete přidělit.

### Ukázka kódu:

promenna1 = \*červená  
promenna2 = \*modrá  
promenna1 = \*zelená  
promenna2 = promena1

Po ukončení kódu je hodnota v proměnné “promenna2” rovna hodnotě v proměnné “promenna1” a obě mají hodnotu “zelená”.

## Vstupy a výstupy

Když chceme, aby se nám některá hodnota, nebo proměnná vypsala, nebo načetla, stačí napsat klíčové slovo IN, nebo OUT a před, nebo za napíšeme hodnotu, kterou chceme vypsat, nebo načíst.

### Výpis proměnných

Pokud chceme vypsat proměnnou. Uděláme to takto

OUT = promenna

Tímto řádkem kódu se vypíše na jeden řádek hodnota uložená v proměnné “promenna”   
(v c# Console.WriteLine(promenna);)

Také můžeme vypisovat pomocí klíčového slova OUTPART. Toto nám vypíše proměnnou, ale nezalomí řádek na konci, tudíž tím můžeme vypsat více proměnných na jeden řádek. (v c# Console.Write())

Promena1 = \*12  
Promena2 = \*34  
OUTPART = Promena1  
OUTPART = Promena2

Výstupem tohoto kódu by bylo 1234.

Také můžeme vypisovat rovnou hodnoty, bez toho, aby byly uložené v proměnných, například:

* OUT = \*Ahoj
* OUTPART = 1984
* OUT = true
* OUT = \*Výpis
* OUTPART =\*Výpis2

### Čtení proměnných

Pro čtení proměnných použijeme klíčové slovo IN, před kterým musí být jméno proměnné, do které chceme načtenou proměnnou ukládat a znaménko rovná se.

Promena1 = IN

Tímto ukázkovým kódem se nám uloží uživatelský vstup, co proměnné “Promena1”.

### Ukázka kódu

hodnota1 = IN  
hodnota2 = IN  
OUTPART = \*první zadaná hodnota byla:   
OUT = hodnota1  
OUTPART = \*druhá zadaná hodnota byla:  
OUT = hodnota2

Tento kód hezky ukazuje, jak využít OUT a OUTPART

## Matematické operace

Matematické operace používáme, pokud chceme například sečíst dvě čísla, čí porovnat jejich hodnoty

### Matematické operace pro Int

Pro int máme několik operací, které navracejí zpět číselnou hodnotu:

* +: pokud chceme sečíst dvě čísla (15 + 5)
* -: pokud chceme odečíst dvě čísla (17 - 7)
* \*: pokud chceme vynásobit dvě čísla (3 \* 5)
* /: pokud chceme vydělit dvě čísla (15 / 5)
* %: pokud chceme získat zbytek po dělení (15 % 7)

Pro int máme také několik operací které navracejí zpět booleanovou hodnotu:

* <: navrací true pokud je číslo vlevo menší než číslo vpravo
* >: navrací true pokud je číslo vpravo menší, než číslo vlevo
* ^: navrací true pokud se hodnota vpravo rovná hodnotě vlevo

### Matematické operace pro boolean

Pro hodnoty boolean máme logické operace, které vracejí zpět booleanovou hodnotu:

* |: navrací true pokud alespoň jedna z hodnot je true (logická operace OR)
* &: navrací true pokud jsou obě hodnoty true (logická operace AND)
* !: navrací negaci hodnoty, která stojí před vykřičníkem (logická operace negace)

### Matematické operace pro string

String má pouze jednu operaci :

* ^: navrací true pokud se hodnota vpravo rovná hodnotě vlevo

### Ukázka kódu

int1 = IN  
int2 = IN  
OUT = int1 < int2

Tento kód vypíše true pokud je první zadaná hodnota menší než druhá zadaná hodnota. (POZOR: program spadne, pokud jedna z hodnot bude string)

## Podmínky

Podmínky se v tomto programovacím jazyce zapisují pomocí klíčového slova IF, které je následováno závorkami, v nichž je booleanová hodnota.

Pokud je vložená booleanová hodnota true, tak se spustí řádek hned pod IF a přeskočí následující řádek. Pokud je ale vložená hodnota false, tak se přeskočí řádek pod IF a spustí o řádek “ob jedna” od IF

### Ukázka kódu

OUT = \*zadejte své jméno  
jmeno = IN  
IF(jmeno ^ \*Petr)  
OUT = \*Vaše jméno je Petr  
OUT = \*=Vaše jméno není Petr

Tento krátký ukázkový kód pozná, zda je vaše jméno Petr či nikoliv (pokud porovnáváte proměnnou a stringovou hodnotu, jako v ukázce dejte stringovou hodnotu na druhé místo.)

### Ukázka kódu 2

OUT = \*zadejte číslo  
cislo = IN  
zbytek = cislo % 5  
IF(zbytek ^ 0)  
OUT = \*vaše číslo je dělitelné 3  
  
zbytek = cislo % 3  
IF(zbytek ^ 0)  
  
OUT =\*Vaše číslo není dělitelné 3

Tento ukázkový kód vypíše pokud je zadané číslo dělitelné 5 a vypíše pokud číslo není dělitelné 3

## Smyčky

Smyčky se v tomto programovacím jazyce zapisují pomocí klíčového slova LOOP. Za slovem LOOP v závorkách se musí napsat počet, kolikrát se má smyčka opakovat a poté nepovinně se může zadat jméno nové proměnné, do které se ukládá počet kolikrát bylo už smyčkou projeto. To kolikrát se má smyčka opakovat může být zadáno již existující proměnnou.

Řádek pod slovem LOOP se bude opakovat určeně mnohokrát. Pokud je ve smyčce potřeba udělat více než jednu věc, použijte metodu (vysvětlení, jak zapsat metody níže)

### Ukázka kódu

LOOP(10)   
OUT = \*Výpis

Tento kód vypíše desetkrát po sobě slovo “Výpis”

### Ukázka kódu 2

LOOP(10 i)   
OUT = 5 \* i

Tento kód vypíše násobilku pěti

### Ukázka kódu 3

Kolik = IN   
LOOP(Kolik)   
OUT = \*A

Tento kód vypíše tolikrát písmeno A kolikrát zadáme.

## Metody

Metody se vytvářejí tak, že ve stejné složce jako máme .exe soubor NewSpeaku a našeho programu vytvoříme nový .txt soubor a pojmenujeme ho tak jak chceme, aby se metoda jmenovala. Poté otevřeme-li ji tak na první řádek napíšeme vstupní parametry metody. Vstupní parametry píšeme pouze tak, že napíšeme jména parametru oddělené mezerou. Pokud nechceme žádné vstupní parametry, necháme první řádek volný, ale stejně začneme psát až na řádek druhý.

V metodě poté můžeme psát kód stejně jako mimo ni, ale vidíme pouze proměnné, které se nám uložily do vstupních argumentů.

Pokud chceme navrátit nějakou hodnotu, na úplném konci metody napíšeme klíčové slovo return, a za něj napíšeme hodnotu, nebo proměnnou, kterou bychom chtěli navrátit.

Metody se volají tak, že kde chceme metodu zavolat pouze napíšeme její jméno se závorkami. Pokud má metoda vstupní parametry, tak do závorek napíšeme jména proměnných, které bychom tam chtěli poslat oddělené mezerou. Pokud metoda něco navrací a my to chceme někam uložit, pouze před volání metody napíšeme jméno proměnné, kam to chceme uložit a znaménko rovná se.

### 

### Ukázka kódu

 Hlavní metoda:

LOOP(5)   
vypsat()

Metoda vypsat.txt:

(volný řádek)

OUT = \*výpis

Tento kód vypíše pětkrát slovo výpis.

### Ukázka kódu 2

Hlavní metoda:

PrvniC = IN   
DruheC = IN   
Soucet = soucet(PrvniC DruheC)   
OUT = Soucet

Metoda soucet.txt

A B   
soucet = A + B

Return soucet

Tento kód ukazuje jednoduchou metodu s parametry, která sčítá dvě čísla a navrací výsledek