Vstup

char c = getchar(); ← vstup jednoho znaku a uložení do prom. **c**

char s[100]; ← vytvoření prom **s** pro až 100 znaků, tedy textu (string).

gets(s); ← vstup řetězce do proměnné **s**

int x; ← vytvoření celočíselné proměnné **x**

scanf("%d", &x); ← načtení celočís. hodnoty ze vstupu a uložení do **x**

**C**

programovací jazyk

Bubílek 2021

Některé vestavěné funkce

stdio.h: printf, scanf, puts…

stdlib.h: abs, rand, srand…

math.h: sin, asin, exp, pow, sqrt, floor…

Vlastní funkce

návratová-hodnota název-funkce(vstupní-parametry){příkazy}

void hello(){puts(”Hello students.”);}

int sum(int x, int y){return x + y;}

Logické operace

< > ← menší, větší

<= >= ! ← ≤, ≥, negace

== != ← rovno, nerovno

&& ← AND, logický součin

|| ← OR, logický součet

Aritmetické operace

+ ← sčítání, ++ inkrementace, += přičítání

- ← odčítání, -- dekrementace, -= odečtení

\* ← násobení, \*= přinásobení

/ ← dělení, /= přidělení, výsledek dělení celým číslem je celé číslo => (float)dělitel

% ← zbytek po dělení, %=

Priorita: \* / % + -, možné závorkovat ()

Přetypování: int x = (int) 3.14;

Escape sekvence

\n ← nová řádka

\t ← tabulátor

\’ ← apostrof

\” ← uvozovky

%% ← procento

Proměnné a konstanty

int x; ←deklarace proměnné x, 4B v paměti

↑ C negarantuje nulovou hodnotu po deklaraci

x = 42; ←uložení hodnoty 42 do proměnné x

float x, y; ←vícečetná deklarace

double x = 3.14; ←deklarace s inicializací

char c = ’A’; ←deklarace s inicializací znaku

C je case-sensitive – rozlišuje se velikost písmen, A≠a

const double PI = 3.14; ←konstanta

#define PI 3.14 ←konstanta v hlavičce prg

% - printf(), scanf()

%[-][velikost].[přesnost]typ

d, i ← decimal, integer

ld ← velká čísla, long decimal

c ← znak

s ← textový řetězec, string

f ← float, double

e ← vědecký formát čísla

x ← šestnáctkový zápis

Datové typy

int ←integer, celé číslo, 42, -5, 4B

float ←desetinné číslo, 3.14, 4B

double ← desetinné číslo, 3.14, 8B

char ←znak, ‘A’, 1B

sizeof()←vrací velikost parametru v Bytes

C nemá boolean, int místo toho.

Hello World

#include <stdio.h> ←hlavičkový soubor

int main() ←vstupní funkce programu

{ ←začátek těla (zdrojového kódu) funkce

↓výpis textového řetězce na stdout

printf("Hello, World!\n");

↓ukončení běhu programu bez chyby

return 0; ←středník ukončuje příkaz

} ←konec těla funkce

Výstup

putchar(’A’); ← výstup jednoho znaku

puts(”Hello”); ← výstup text. řetězce

printf(”%i-%i”, x, y); ← vypíše int hodnotu **x,** pomlčku a int hodnotou **y**

Podmínky

if(logický výraz){příkazy}else{jiné příkazy}

↑ Když je **výraz** true, dělej **příkazy**, jinak dělej **jiné příkazy**.

Podmínky lze vnořovat: **příkazy** mohou být opět podmínky.

if(výraz1){A}else if(výraz2){B}else{C}

↑ Podmínky lze zřetězovat

x = (x > 0) ? (x) : (-1\*x); ←podm. výraz

↑ Když je **x** kladné, tak do **x** ulož **x**, jinak do **x** ulož -1\*x.

switch(výraz){case val: příkazy; break;}

↑ vícevětvá podmínka, možnost kaskády, break vyskakuje

Pole

Množina prvků stejného typu. Prvek má klíč (od nuly) a hodnotu.

int pole[5]; ← deklarace pole pro pět celočíselných prvků

int p[3] = {4,5}; ← deklarace s inicializací, třetí je nula.

p[0] = 42; ← První prvek (klíč nula) má nyní hodnotu 42.

printf(”%i”,p[1]); ← Vypíše hodnotu druhého prvku

Cykly

for(start, podmínka, krok){příkazy}

↑ Od **startu** opakuj **příkazy** krok po **kroku** dokud platí **podmínka**

for(int i=0; i<10; i++){printf(”%d”,i);} ← příklad

while(podmínka){příkazy}

↑ Dokud platí podmínka, dělej příkazy. První se ptej, pak dělej.

do{příkazy}while(podmínka)

↑ Dělej příkazy, dokud platí podmínka. První dělej, pak se ptej.

break; ← Vyskočí z cyklu a pokračuje dál v programu za cyklem.

continue; ← Ukončí aktivní průchod a skočí na další průchod cyklu.

Komentáře

//řádk. kom.

/\*

blokový

komentář

\*/