Opakování

Základy programování 2

Mgr. Markéta Trnečková, Ph.D.

Palacký University, Olomouc

Program

- Funkce
- Proměnné
- Funkce `main()`
- Knihovny
- Argumenty funkcí

Proměnné

- Proměnná, Konstanta
- Typ
- Deklarace:
  - `typ jmeno;`
  - `typ jmeno = hodnota;`
- Inicializace
- `typ jmeno1 = hodnota1, jmeno2, jmeno3 = hodnota3;`

Datové typy

- `char`
- `int`
- `float`
- `double`
- navíc: `short`, `long`, `signed`, `unsigned`

Konstanty

- Celočíselná (123, 123L, 0..., 0x..., 0X, ...)
- Znaková ('a', '\n', '\000', ...)
- Konstantní výrazy

```
#define MAXIMUM 100
int cislo = MAXIMUM -10;
```

- Řetězcová konstanta `"ja jsem retezec"`
- Výčtová konstanta `enum boolean { NE, ANO };`

Operátory

- Operátor
- Arita — unární, binární, ternární, ...
- **Aritmetické operátory**: -, +, *, /, %
- **Operátor přiřazení**: `l-value = r-value;`
- **Operátor přiřazení s aritmetickým operátorem**: +=, -=, *=, /=, %=
- **Operátor ++, −−**

printf a scanf

- `printf(format, h1, h2, ...)`
- `printf("Konstanta a je rovna %i", a);`
- formátovací instrukce %...

- `scanf(format, h1, h2, ...)`
- `int i;`
- `scanf("%i", &i);`

Větvení programu

Podmínky: $<, <=, >, >=, ==, !=$
Spojování podmínek: $||, \&\&$

- `if-else`
- `switch`
- `jednoduché podmínky:` `podminka ? vyraz1 : vyraz2;`

Větvení programu

```
if (podminka)
    blok-pri-pravde
else
    blok-pri-nepravde
```

Větvení programu

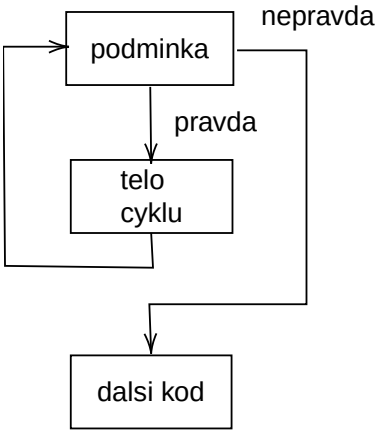
```
switch (vyraz) {
  case konstanta1:
    blok1
    break;
  case konstanta2:
    blok2
    break;
...
  default:
    blok
}
```

Cykly

- `while`
- `for`
- `do while`

While

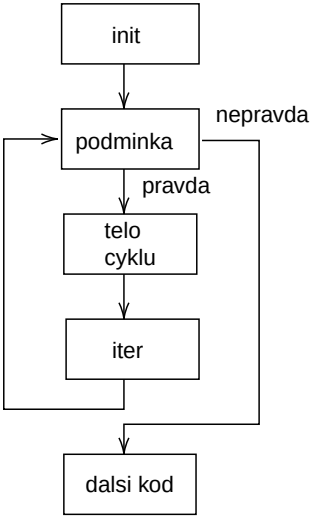
```
while (podminka)
  telo cyklu
```



For

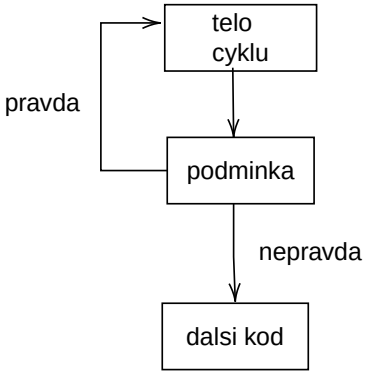
```
for (init; podminka; iter)
  telo cyklu
```

- krokovací proměnná



Do while

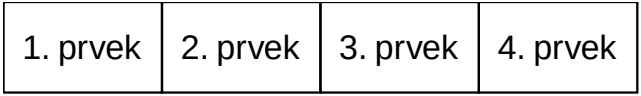
```
do
  telo cyklu
while (podminka)
```



Další příkazy

- `break;`
- `continue;`

Pole



- **index**: 0, 1, ...
- `typ jmeno[velikost];`
- `typ jmeno[velikost]={p1, p2, ..., pn};`
- `jmeno[index];`

Řetězce (pole char)

- `char retezec[11]= {'A','h','o','j',' ','s','v','e','t','e','\0'};`

'A'	'h'	'o'	'j'	' '	's'	'v'	'e'	't'	'e'	'\0'

- `char retezec[]= "Ahoj svete";`
- `printf("%s", retezec);`

Funkce

- **Hlavička funkce**: `typ jmeno(typ1 a1, typ2 a2, ..., typn an)`
- **Tělo funkce**
- `return hodnota;`
- **Deklarace**: `double mocnina(double, int);`
- `typ y = jmeno(a1, a2, ..., an);`

Rozsah platnosti proměnných

- **Globální proměnné**
- **Lokální proměnné**

Struktury

```
struct jmeno {
polozka1
polozka2
...
polozkan
} seznam promennych;
```

- `struct jmeno foo = {h1, ... hn};`
- `foo.polozka1`

Pojmenování typů

```
typedef deklarace promenne;
```

Chyby v programu

- **Syntaktické chyby**
- **Sémantické chyby**
  - Nesprávné použití jazyka
  - Chyba algoritmu

Cvičení

1. Naprogramujte funkci, která jako argument bere řetězec a vypisuje počet výskytů jednotlivých znaků ve vstupním řetězci.
2. Naprogramujte funkci, která jako vstup bude brát desetinné číslo a celé číslo určující počet desetinných míst. Funkce pak vypíše zadané číslo se zadanou přesností. Např. pro desetinné číslo 1.23456 a celé číslo 2 vrátí číslo 1.23
3. Napište funkci počítající progresivní daň. Pro účel této úlohy uvažujme progresivní zdanění ve výši 10 % pro příjem do 10000, 20 % pro příjem od 10000 do 20000 a 30 % pro příjem nad 20000. Například, pokud máme hrubou mzdu 24000, bude se prvních 10000 danit 10 % (tj. daň z této části mzdy je 1000), dalších 10000 se daní 20 % (daň z této části je 2000) a zbývající 4000 se daní 30 % (daň je 1200). Celkovou výši daně pak vypočítáme jako součet jednotlivých "částečných" daní (tj. celková daň 4200).
4. Napište funkci, která vykreslí pomocí znaku "*" na obrazovku čtverec zadané velikosti. Pro velikost 4 bude výsledek:

```
* * * *
*     *
*     *
* * * *
```

Cvičení

5. Pomocí hrubé síly prolomte Caesarovu šifru:
"mrwfvqbfcryivfiqborxqlfrmnqanmirpvpbfvcerwrfarqnxbhcvgibopubqr"
6. Naprogramujte funkci, která pro zadané n vrátí n-tý prvek posloupnosti, která je zadána rekurentním vztahem: $a_1 = 14688$, $a_{n+1} = \frac{1}{2}a_n + 1200$. (10. člen je roven 2424)
7. Naprogramujte funkci, která jako vstup bere 2 celočíselné kladné argumenty m a n větší rovny 2 a pracuje podle následujícího pseudokódu:
   i. Vypiš $n - 2$ mezer, pak řetězec "(\\o/)"
   ii. Opakuj m krát:
      - Na nový řádek vypiš n teček, velké X a n teček.
   iii. Na nový řádek vypiš 2*n + 1 krát X.
   iv. Opakuj krok ii.