

# Programovanie 10

Podmienky, vetvenia, cykly

# Podmienky v jazyku C

- Logický dátový typ neexistuje:
  - int 0 reprezentuje nepravdivú hodnotu
  - akýkoľvek iný údaj: pravdivá hodnota
- Ako podmienka môže byť použitý akýkoľvek výraz (ktorý sa dá skonvertovať na int):
  - $x == y$  : porovná, výsledok je 0, ak sa nerovnajú
  - $x = y$  : priradí, výsledok je 0, ak y bolo 0
- Porovnanie a logické operátory sa dajú použiť v aritmetike:
  - $x = 2 + (y > 0)$

# Operátory porovnania a logické operátory

- Porovnanie:

== > < <= >=

- Logické operátory:

! && || (negácia, a zároveň, alebo)

- zrýchlené vyhodnocovanie:

```
0 && printf("nikdy sa nevypise")
```

```
1 || printf("nikdy sa nevypise")
```

# Ternárny operátor

- Na vyjadrenie vetvenia pri výpočte bez potreby písania postupnosti príkazov:

PODMIENKA ? VYRAZ\_T : VYRAZ\_F

- definuje výraz s hodnotou VYRAZ\_T, ak je podmienka nenulová (pravdivá), inak VYRAZ\_F

- Príklad:

```
int abs(int x)
{ return x >= 0 ? x : -x; }
```

# Vetvenie if/else

```
if (PODMIENKA)
    BLOK_PRIKAZOV_T
else
    BLOK_PRIKAZOV_F
```

- Ak PODMIENKA je splnená (nenulová), vykoná sa BLOK\_PRIKAZOV\_T, inak BLOK\_PRIKAZOV\_F
  - Blok príkazov: viaceré príkazy uzavreté v { }
  - alebo: ak je len 1 príkaz, } sa môžu vynechať
  - else časť sa celá môže vynechať

# if / else if / else

```
if (PODMIENKA)
    BLOK_PRIKAZOV_T
else if (P2)
    BLOK_PRIKAZOV_T2
else if (P3)
    BLOK_PRIKAZOV_T3
    /* lubovolne vela podmienok */
else
    BLOK_PRIKAZOV_ELSE
```

# if / else if / else

- Je to vlastne len zjednodušený zápis zloženého vetvenia:
  - v prvom vetvení if všetko ostatné je súčasťou else bloku
  - aby sa vykonal BLOK\_i, musia byť všetky predošlé podmienky nepravdivé, a podmienka P<sub>i</sub> pravdivá
  - aby sa vykonalo finálne else, musia byť všetky podmienky nesplnené
  - po skončení BLOK\_i program ide za celý if/else

# switch/case

```
switch (VYRAZ) {  
case C1:  
    POSTUPNOST_PRIKAZOV_1  
    break;  
case C2:  
    POSTUPNOST_PRIKAZOV_2  
    break;  
default:  
    POSTUPNOST_PRIKAZOV_D  
    break;  
}
```



# switch/case

- VYRAZ sa vyhodnotí, výsledok musí byť číslo
- case návestia sú konštanty
  - intervaly, zložitejšie podmienky a pod: musí sa použiť if / else if /else
- program pokračuje za switch návestím, kde  $C_i == VYRAZ$ 
  - ak žiadne také nie je, pokračuje sa na default
- príkazy sa vykonávajú až po príkaz break
  - ak sa vynechá, pokračujú aj cez iné návestia

# while

```
while (PODMIENKA)  
    BLOK_PRIKAZOV_T
```

- ak je podmienka splnená, blok príkazov sa vykoná a opäť sa vráti riadenie na podmienku
- blok príkazov:
  - viac príkazov: nutné dať do { },
  - jeden príkaz, { } sa môžu vynechať (ale neodporúčam)

# do / while

do

```
BLOK_PRIKAZOV_T
```

```
while (PODMIENKA) ;
```

- blok príkazov sa vykoná, a ak je podmienka splnená, vrátenie sa vráti naspäť na blok príkazov
- blok príkazov:
  - viac príkazov: nutné dať do { },
  - jeden príkaz, { } sa môžu vynechať (ale neodporúčam)
- Záverečná ; ukončuje príkaz do/while
  - na rozdiel od iných vetvení a cyklov

# for

```
for (INIT; PODMIENKA; POST)
    BLOK_PRIKAZOV_T
```

- Sémantika:
  - vyhodnotí sa blok INIT
  - ak je podmienka splnená, vykoná sa blok príkazov
  - po vykonaní bloku príkazov sa vyhodnotí POST
  - riadenie sa vráti na podmienku
- Bloky vnútri for sú oddelené ;
  - môžu byť aj prázdne
  - môžu obsahovať ľubovoľné výrazy (aj s volaniami funkcií, priradeniami, ...)

# for

- Typické použitie for:

```
int i;  
for (i = 0; i < n; i++)  
{  
    /* nieco, co sa n-krat opakuje */  
}  
//v tomto bode i ma hodnotu n
```

# for

- Novší štandard (C99 a vyššie, ako aj C++) umožňuje aj lokálne i:

```
for (int i = 0; i < n; i++)  
{  
    /* nieco, co sa n-krat opakuje */  
}  
//POZOR: tu uz i neexistuje
```

# break/continue

- V cykloch je možné použiť príkazy:
  - `break`: vyskočí z cyklu, riadenie sa presunie na prvý príkaz za cyklom
  - `continue`: preskočí zvyšok cyklu, riadenie sa presunie na podmienkovú časť (v príkaze `for` na `POST` časť)
- Pri vnorených cykloch sa `break/continue` týka len najvnútornejšieho cyklu
  - je vhodné použiť podmienkové premenné a logické spojky na úplné opustenie vnorených cyklov