

Větvení

3. cvičení

Jiří Zacpal

KMI/ZP1 – Základy programování 1

Blok (složený příkaz)

- umožňuje zapsat více příkazů na místě jednoho
- syntaxe:

```
{  
    deklarace_1;  
    ...  
    deklarace_M;  
  
    příkaz_1;  
    ...  
    příkaz_N;  
}
```

Větvení programu

- konstrukce **if**
 - umožní rozvětvení programu; příkazy se provedou, pokud je splněna daná podmínka
 - pomocí klíčového slova `else` lze definovat větev, která se provede, pokud podmínka splněna nebude
 - příkaz `if` lze vnořit do jiného `if`
- konstrukce **switch**
 - podle hodnoty celočíselného výrazu umožní rozvětvení programu do libovolného množství větví
 - nelze vytvořit větev pro interval hodnot
 - pomocí klíčového slova `default` lze vytvořit větev, jejíž příkazy se budou provádět, pokud nebyla vybrána jiná větev

Konstrukce IF

- syntaxe jednoduchého if:

```
if ( podmínka ) příkaz
```

- příklad:

```
if (x==0) printf("Proměnná x je  
nula.\n");
```

- syntaxe s využití větve else:

```
if ( podmínka ) příkaz_1 else  
příkaz_2
```

- příklad:

```
if (x>0) printf("x je kladné\n");  
else printf("x není kladné\n");
```

Příklad 1

```
main()
{
    int c1,c2,c3, max;

    printf("Zadejte tri cisla:");
    scanf("%d%d%d", &c1, &c2, &c3);
    if(c1>=c2)
        if(c1>=c3)
            max=c1;
        else
            max=c3;
    else
        if(c2>=c3)
            max=c2;
        else
            max=c3;
    printf("Nejvetsi cislo je: %d\n", max);
}
```

Líné vyhodnocování

- V logických výrazech se vyhodnocuje pouze část výrazu (podle asociativity) nutná pro získání výsledku.

- příklady:

```
if ( (c>0) && (c<(i=m%n)) )  
{ ... }
```

```
if ( (c<0) || (c>(i=m%n)) )  
{ ... }
```

Příklad 2

```
main_2()
{
    int z,i,pbilych,postatnich;
    int pcislic[10];

    pbilych=postatnich=0;
    for (i=0;i<10;i++) pcislic[i]=0;

    while((z=getchar())!=EOF)
        if(z>='0'&&z<='9')
            ++pcislic[z-'0'];
        else if (z==' ' || z=='\n' || z=='\t')
            ++pbilych;
        else
            ++postatnich;

    printf("cisla= ");
    for (i=0;i<10;i++) printf(" %d ", pcislic[i]);
    printf(", bile znaky = %d, ostatni = %d\n", pbilych,
    postatnich);
}
```

Konstrukce SWITCH

- syntaxe:

```
switch (celočíselný_výraz) {  
    case konstanta_1: příkazy_1  
  
    break;  
    ...  
    case konstanta_N: příkazy_N  
  
    break;  
    default: příkazy_default_větve  
}
```


Příklad 3

```
main()
{
    int z,i,pbilych,postatnich;
    int pcislic[10];

    pbilych=postatnich=0;
    for (i=0;i<10;i++) pcislic[i]=0;

    while((z=getchar())!=EOF)
        switch(z){
            case '0':case '1':case '2':case '3':case '4':case '5':case '6':case '7':case
            '8':case '9':
                ++pcislic[z-'0'];
                break;
            case ' ':
            case '\n':
            case '\t':
                ++pbilych;
                break;
            default:
                ++postatnich;
        }
    printf("cisla= ");
    for (i=0;i<10;i++) printf(" %d ", pcislic[i]);
    printf(", bile znaky = %d, ostatni = %d\n", pbilych, postatnich);
}
```

Úkol

- Napište v jazyku C program, který pro zadaný znak **slovy** vypíše, o jaký znak se jedná...
 - pro malá písmena vypíše program text "*male pismo*" a daný znak,
 - pro velká písmena vypíše text "*velke pismo*" a daný znak,
 - pro číslice vypíše "*cislice*" a daný znak
 - dále pro znaky "!", "?", "*", "@", "#", "^" vypíše odpovídající text: "*vykricnik*", "*otaznik*", "*hvezdicka*", "*zavinac*", "*krizek*", "*striska*". Pokud se jedná o jiný než výše uvedený znak, vypíše program text "*jiny znak*".
- **Příklad výstupu:**
Zadejte znak: c
Zadany znak je: male pismo c
- **Povolené knihovny:** stdio.h, stdlib.h