

Větvení programu

Základy programování 1
Martin Kauer

Z minula

- ▶ K řešení úkolů se snažte **používat jen to co jsme už probrali**, nic víc totiž nepotřebujete.
- ▶ Nezapomeňte, že do proměnné musíte nastavit nějakou hodnotu před jejím prvním použitím.
- ▶ V jazyku C je zvykem (dokonce vyžadováno dle starších standardů), aby **všechny deklarace proměnných byly na začátku bloku**.
- ▶ Ke cvičení je většinou přiložena i ukázka běhu programu, použijte ji pro kontrolu správnosti vašeho programu.
 - ▶ Je tam jednak vidět co máte načítat.
 - ▶ Navíc je možné zkontrolovat pro dané vstupy korektní výstupy.

Konstrukce if

- ▶ Příkazy se provádějí pouze při splnění dané podmínky
- ▶ Podmínka = jakýkoli logický výraz = cokoli celočíselného
- ▶ Volitelně i příkazy prováděné při nesplnění podmínky (za `else`)
- ▶ Typický zápis:

```
if (podmínka)
{
    příkazy 1
    ...
}
else if (podmínka 2)
{
    příkazy 2
    ...
}
else
{
    příkazy 3
    ...
}
```

Příklady if

```
if (hodnota < 0)
{
    hodnota = -hodnota;
    zmeneno = 1;
}
else
{
    zmeneno = 0;
}
```

```
if (hodnota < 0)
{
    hodnota = -hodnota;
    zmeneno = 1;
}
else
    zmeneno = 0;
```

```
if (hodnota < 0)
    hodnota = -hodnota;
```

If v jiném if

- ▶ Relativně běžné

- ▶ Příklad:

```
if (hodnota < 0)
{
    printf("Záporné");
}
else
{
    if (hodnota > 0)
    {
        printf("Kladné");
    }
    else
    {
        printf("Nula");
    }
}
```

```
if (hodnota < 0)
    printf("Záporné");
else
    if (hodnota > 0)
        printf("Kladné");
    else
        printf("Nula");
```

Konstrukce switch

- ▶ Větvení podle hodnoty celočíselného výrazu
- ▶ Příkazy pro určitou hodnotu výrazu (tj. pouze podmínky ve tvaru rovnosti)
- ▶ Každá větev je zakončena příkazem `break`
- ▶ Default větev se provádí, pokud nebyla žádná podmínka splněna

▶ Typický zápis:

```
switch (výraz)
{
    case hodnota1:
        příkazy 1
        ...
        break;
    case hodnota2:
        příkazy 2
        ...
        break;
    ...
    default:
        příkazy
        ...
        break;
}
```

Příklady switch

```
int cislo;
char *text;
...
switch (cislo)
{
    case 1:
        text = "Jedna";
        break;
    case 2:
        text = "Dva";
        break;
    default:
        text = "Nevim";
        break;
}
printf(text);
```

```
char znak, posunout, novy_znak;
int posun = 2;
...
switch (posunout){
    case 'Y':
        novy_znak = znak + posun;
        posunout = 'N';
        break;
    case 'N':
        novy_znak = znak;
        break;
    default:
        printf("Chyba!\n");
}
```

Switch - větev pro více hodnot

- ▶ Není úplně typické, ale může se hodit
- ▶ Několik návěstí case za sebou
- ▶ Větev společná pro více uvedených hodnot
- ▶ Příklad:

```
switch (cislo)
{
    case 1:
    case 2:
        text = "Jedna nebo Dva";
        break;
    default:
        text = "Nevim";
        break;
}
```


Příklady složitějšího větvení

- ▶ Switch v if:

```
if (slovne){
    switch (cislo){
        case 1:
            printf("Jedna");
            break;
        case 2:
            printf("Dva");
            break;
        case 3:
            printf("Tri");
            break;
        default:
            printf("Neumim");
            stop = 1;
            break;
    }
} else {
    printf("%d\n", cislo);
}
```

Příklady složitějšího větvení

- ▶ If ve switch:

```
switch (akce) {
    case 'n':
        vysledek = vstup1 * vstup2;
        break;
    case 'd':
        if (vstup2 == 0) {
            printf("Nelze delit nulou.\n");
            break;
        } else {
            vysledek = vstup1 / vstup2;
            break;
        }
    default:
        printf("Neznama akce.\n");
        break;
}
```

Poznámky

- ▶ Zejména u složitějšího a většího větvení dbejte na správné formátování a odsazování kódu.
- ▶ Pokud větev ifu obsahuje pouze jeden příkaz, složené závorky je dobré vynechat.
- ▶ Pokud máte nějaký kód ve všech větvích stejný, dejte jej před / za větvení.

```
if (cislo < 0)
{
    vysledek = -1;
    printf("%i", vysledek);
}
else if (cislo == 0)
{
    vysledek = 0;
    printf("%i", vysledek);
}
else if (cislo > 0)
{
    vysledek = 1;
    printf("%i", vysledek);
}
```



```
int vysledek = 1;

if (cislo < 0)
    vysledek = -1;
else if (cislo == 0)
    vysledek = 0;

printf("%i", vysledek);
```

Poznámky

► Rozdíl mezi:

```
if (podminka1) {}  
if (podminka2) {}  
if (podminka3) {}
```

```
if (podminka1) {}  
else if (podminka2) {}  
else if (podminka3) {}
```

► Časté nesrovnalosti:

```
if (podminka);  
{  
    prikazy  
}
```

```
if (promenna = 0)  
{  
    prikazy  
}
```



```
if (promenna == 0)  
{  
    prikazy  
}
```

```
if (podminka)  
(  
    prikaz1,  
    prikaz2;  
)
```

```
if (podminka);  
else  
{  
    prikazy  
}
```



```
if (!podminka)  
{  
    prikazy  
}
```

Poznámky

- ▶ Nedělejte redundantní kontroly podmínek, větvení si promyslete a pořádně zkontrolujte.

```
if (vstup == 1 && pocitadlo >= 0 && pocitadlo < 10)
{...}
else if (vstup == 2 && pocitadlo >= 0 && pocitadlo < 10)
{...}
...
```



```
if (pocitadlo >= 0 && pocitadlo <= 10)
{
    if (vstup == 1) {...}
    else if (vstup == 2) {...}
    ...
}
```



```
if (pocitadlo >= 0 && pocitadlo <= 10)
{
    switch (vstup)
    {
        case 1: ...
            break;
        case 2: ...
            break;
        ...
    }
}
```

Cvičení 1

Napište program, který pro určitý znak slovy vypíše, o jaký znak se jedná.

- ▶ Pro malá písmena program vypíše „samohlaska“ / „souhlaska“ a zadaný znak.
- ▶ Pro velká písmena program vypíše „velke pismeno“ a zadaný znak.
Pro číslice program vypíše „cislice“, zadaný znak a zda je daná číslice dělitelná dvěma.
- ▶ Dále pro znaky ‘!’, ‘?’, ‘*’, ‘@’ a ‘#’ vypíše jim odpovídající text: "vykricnik", "otaznik", "hvezdicka", "zavinac" a "krizek".
- ▶ Pokud se jedná o jiný než výše uvedený znak, vypíše program text "jiny znak".

Zadejte znak: a
mala samohlaska 'a'

Zadejte znak: \$
jiny znak

Zadejte znak: 4
cislice '4', cislo je delitelne dvema

Zadejte znak: T
velke pismeno 'T'

Zadejte znak: @
zavinac

Zadejte znak: 5
cislice '5', cislo neni delitelne dvema

Cvičení 2

Od uživatele načtete datum ve formátu den.mesic.rok a rozhodnete, zda dané datum je validní (zda existuje v kalendáři). Rozhodnutí vypíšete uživateli na obrazovku.

Příklady běhu programu:

```
Zadejte datum: 1.1.1970  
Datum 1.1.1970 je validní.
```

```
Zadejte datum: 31.4.2018  
Datum 31.4.2018 není validní.
```

```
Zadejte datum: 31.1.2018  
Datum 31.1.2018 je validní.
```

```
Zadejte datum: 29.2.2000  
Datum 29.2.2000 je validní.
```