

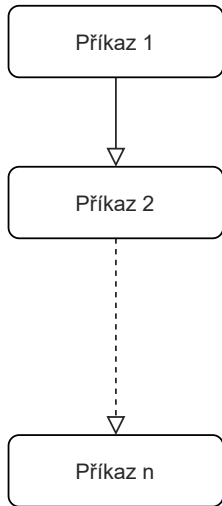
Jazyk C

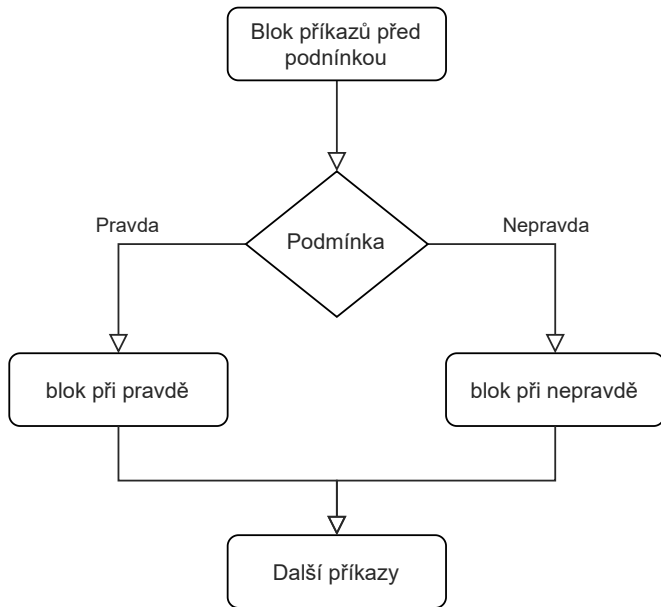
Řízení běhu programu

Mgr. Markéta Trnečková, Ph.D.



Palacký University, Olomouc





Větvení pomocí if a else



```
if (podminka)
    blok_pri_pravde
else
    blok_pri_npravde
```

Logické operátory

- `<`, `<=` menší, menší rovno
- `>`, `>=` větší, větší rovno
- `==` rovnost (pozor nezaměňovat s operátorem přiřazení `=`)
- `!=` nerovnost

Example (Co vypíše následující část kódu?)

```
int a = 0;

if (a = 0){
    printf("a je rovno 0");
}
else{
    printf("a není rovno 0");
}
```



Logické spojky

- `||` nebo
- `&&` a zároveň
- `!` negace

Líné vyhodnocování `||` a `&&`

```
if ((y != 0) && ((x / y) < z))
```



Example

```
if (a >= b){  
    printf("%d", a);  
}  
else{  
    printf("%d", b);  
}
```



Example (Jak by vypadala podmínka v kladném znění?)

```
if (!(c == '\0' || c == ' ' || c == '1'))
```


Example (Řešení)

```
if (c != '\\0' && c != ' ' && c != '1')
```

Example

```
int a = 5, b = 1, c = 3, foo = 10;

if (a > b){
    if (b > c)
        foo = b;
}
else foo = c;
```

Example

```
int a = 5, b = 1, c = 3, foo = 10;

if (a > b)
    if (b > c)
        foo = b;
    else foo = c;
```



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



Example

```
if (a == 1)
    printf("a je jedna");
else if (a == 2)
    printf("a je dva");
else if (a == 3)
    printf("a je tri");
else
    printf("a není 1, 2 ani 3");
```

Example

```
switch (a){
    case 1:
        printf("a je jedna");
        break;
    case 2:
        printf("a je dva");
        break;
    case 3:
        printf("a je tri");
        break;
    default:
        printf("a není 1, 2 ani 3");
}
```



```
podminka ? vyraz1 : vyraz2;
```

Example

```
if (a < b)
    x = a;
else
    x = b;
```

Example

```
x = (a < b) ? a : b;
```

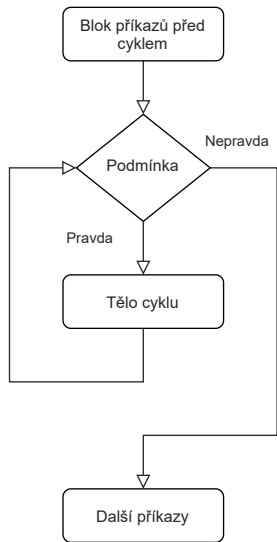


- cyklus `while`,
- cyklus `for`,
- cyklus `do while`.

Cyklus while



```
while (podminka)  
    telo_cyklu
```



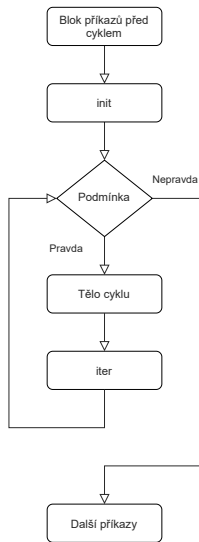


Example

```
int j;  
  
j = 0;  
while(j < 10){  
    printf("%d ", j);  
    j = j + 1;  
}
```



```
for (init; podminka; iter)  
    telo-cyklus
```





Example

```
int j;  
  
for(j = 0; j < 10; j = j + 1){  
    printf("%d ",j);  
}
```

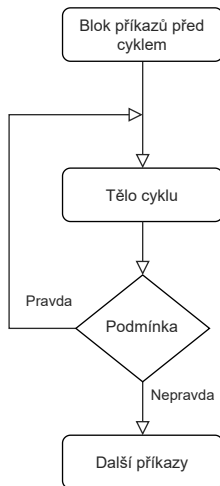
Example

```
int i = 0;
/* doporučene pouziti cyklu for */
for(i = 0; i < 10; i++){
    printf("%d ", i);
}
/* vyuziti inicializace i pri definici. */
for(; i < 10; i++){
    printf("%d ", i);
}
/* ridici promena se meni v tele cyklu */
for(i = 0; i < 10; ){
    printf("%d ", i++);
}
/* vyuziti operatoru carky */
for(i = 0; i < 10; printf("%d ", i), i++)
    ;
```

Cyklus do while



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





Example

```
int j;  
for(j = 1; j < 10; j = j + 1){  
    if((j % 3) == 0)  
        break;  
    printf("%d ", j);  
}
```

Example

```
int j;  
for(j = 1; j < 10; j = j + 1){  
    if((j % 3) == 0)  
        continue;  
    printf("%d ", j);  
}
```



```
goto navesti;
```

```
...
```

```
navesti:
```

```
...
```

Přepište následující kód bez použití příkazu goto.

Example

```
for (i = 0; i < 10; i++){
    for (j = 0; j < 10; j++){
        for (k = 0; k < 10; k++){
            /* x je pole a x[k] vraci jeho k-ty prvek */
            if (x[k] == 0)
                goto chyba;
            printf("%d", (x[i] + x[j]) / x[k]);
        }
    }
}
...
chyba:
printf("Nelze delit nulou\n");
```

Příkaz return



```
return 0;
```


- 1 Za použití podmínkového operátoru napište program, který pro zadané číslo vypíše jeho absolutní hodnotu.
- 2 Napište program, který rozhodne, zda zadaný rok je přestupný. (Definici přestupného roku naleznete například na wikipedii https://cs.wikipedia.org/wiki/P%C5%99estupn%C3%BD_rok).
- 3 Napište program, který rozhodne, zda je zadané písmeno malé nebo velké.
- 4 Pro zadané číslo od 1 do 10, vypíše sestupnou posloupnost čísel od zadaného čísla po 1. (pro 4 bude výstup 4 3 2 1). K řešení zkuste vhodně použít konstrukci `switch`. (Vhodnější je použít cyklus. Tento úkol je spíše k zamyšlení.)
- 5 Napište program, který načte celá čísla a a b a pak
 - 1 vypíše prvních a násobků čísla b
 - 2 spočítá a -tou mocninu čísla b
 - 3 určí kolik číslic má číslo a
 - 4 vypočítá a -té Fibonacciho číslo
 - 5 sečte všechna čísla větší než a a menší než b

6 Pro zadané n vykreslete do konzole následující obrázky

1 pro $n=3$

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

pro $n = 4$

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

2 pro $n=2$

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

pro $n = 3$

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

7 3 šachovnici o straně n pro $n = 4$

```
. * . *
* . * .
. * . *
* . * .
```

8 Napište program, který pro zadané číslo vrátí číslo zapsané pozpátku. (Pro 1234 vrátí číslo 4321)

Přiřaďte výstupy k blokům kódu po dosažení bloku do následujícího kódu.

Example

```
#include <stdio.h>

int main(){
    int x = 0;
    int y = 0;
    while (x < 5){
        /* sem
           vložte
           blok
           kódu */
        printf("%d%d ", x, y);
        x = x + 1;
    }
    return 0;
}
```

Bloky kódu:

```
y = x - y;
```

Možné výstupy:

22 46

```
y = y + x;
```

11 34 59

```
y = y + 2;
```

```
if (y > 4)
```

```
    y = y - 1;
```

02 14 26 38

02 14 36 48

```
x = x + 1;
```

```
y = y + x;
```

00 11 21 32 42

```
if (y < 5){
```

```
    x = x + 1;
```

```
    if (y < 3)
```

```
        x = x - 1;
```

```
}
```

```
y = y + 2;
```

11 21 32 42 53

00 11 23 36 410

02 14 25 36 47