

UPR

Cvičení 2

Podmínky

- chceme at' platí víc podmínek zároveň

```
if ( podminka1 && podminka2 )  
{...
```

```
if ( podminka1 || podminka2 )  
{...
```

```
if ( podminka1 || podminka2 && podminka3 )  
{...
```

Podmínky

- Datový typ bool

```
bool podminka1 = true;  
bool podminka2 = false;  
bool podminka3 = 3 > 2;  
bool podminka4 = 1;
```

Podmínky

- Úkol č.1
 - Najděte maximum ze tří desetinných čísel

```
float a, b, c;  
printf("zadejte 3 cisla: ");  
scanf("%f", &a);  
scanf("%f", &b);  
scanf("%f", &c);  
printf("%.1f %.1f %.1f\n", a, b, c);
```

Cyklus for

- Minule jsme si ukázali cyklus while

```
while (<podminka>) { ...
```

```
int a = 0;
while (a <= 10)
{
    printf ("%d\n", a);
    a++;
}
```

Cyklus for

- Známe počet opakování?

- Použijme cyklus for

```
for (<inicializace>; <podminka>; <inkrement>)
```

```
for (int a = 0; a <= 10; a++)  
{  
    printf ("%d\n", a);  
}
```

Cyklus for

- Úkol č.2
 - Vypište sudá čísla od 0 po číslo x , které zadá uživatel

Funkce

- Prozatím jsme psali kód do hlavní funkce main()
 - Pokud je kódu hodně, zdroják je dlouhý a méně přehledný
 - Pokud opakujeme některé algoritmy vícekrát, musíme je psát znova
- Proto je vhodnější používat funkce

Funkce

- Formát funkce

`<navratovy_typ> nizev_funce(<parametry>) { ...`
známe

`int main()` - vrací celé číslo (proto `return 0`),
nemá žádné parametry

Funkce

- Funkce na součet dvou čísel:

```
int soucet(int a, int b)
{
    return a+b;
}
```

- Proměnné existují pouze v rámci funkce!

Funkce

- Úkol č.3
 - Vytvořte funkci pro načítání celých čísel
 - Ve funkci vyzvěte uživatele k zadání čísla a vraťte přečtené číslo

Pole

- Zatím jsme pracovali s dvěma až třemi proměnnými
- Co když jich bude více?
- Museli bychom mít:
`int a, b, c, ..., o, p, q;`
- Místo toho použijeme pole

Pole

- Pole celých čísel o 3 prvcích:

```
int pole[3];
```

```
pole[0] = 1;
```

```
pole[1] = 2;
```

```
pole[2] = 3;
```

nebo

```
int pole[4] = {1, 2, 3, 4};
```

```
int pole[] = {1, 2, 3, 4};
```

Pole

- Úkol č.4

Načtete čísla do pole a vytvořte funkci, která vrátí součet čísel v poli

- Počet čísel zadejte předem pomocí `const int`

- `int soucetPole(int* pole, int pocet)`