

ZPRO 2. cvičení

Podmínky

Syntaxe (zjednodušená):

Neúplné if: **if** (*podmínka*) { *příkazy* }

Je-li podmínka splněna, provedou se *příkazy*, jinak se neprovede nic.

Úplné if: **if** (*podmínka*) { *příkazy-1* } **else** { *příkazy-2* }

Je-li podmínka splněna, provedou se *příkazy-1*, jinak se provedou *příkazy-2*.

Příklad 1

- Napište znovu výpočet faktoriálu a ověřte, že uživatel zadá číslo n v rozmezí $0 \leq n \leq 12$
- Kontrolu shora provedeme, protože `int` je 4 bytový datový typ. Maximální jeho hodnota je 2 147 483 647 a $13! = 6\,227\,020\,800$
- Složitější podmínky pomocí operátorů konjunkce `&&` a disjunkce `||`

Podprogramy

- Podprogram
 - Část programu, kterou je možné volat (opakovaně) z různých míst kódu
 - Může mít parametry a vracet nějakou hodnotu
- Funkce
 - Podprogram, který vrací nějakou hodnotu
- Procedura
 - Podprogram, který nemá návratovou hodnotu
- Význam podprogramů
 - Zpřehlednění programu
 - Kód mohu používat na více místech (zásada neopakování se)

Funkce

- Definice
 - **Návratový_typ identifikátor(parametry) {příkazový blok}**
 - Návratový typ – int, bool, string, **void** (v případě procedury)
 - Identifikátor – název funkce
 - Funkce končí příkazem **return identifikátor;**
- Volání funkce
 - **Identifikátor(parametry);**

Příklad 2

Napište funkci **int faktorial(int n);**, kde **n** je parametr. Funkce bude vracet hodnotu **n!**.

Globální proměnná

- Proměnná deklarovaná mimo funkce
- Dostupná v celém souboru

Operátor podmíněného výrazu ?:

- Alternativa k jednoduchému if – else

```
if(podmínka) příkaz1;  
else příkaz2;
```

Lze napsat pomocí operátoru ?:

```
podmínka ? příkaz1 : příkaz2;
```


Ukázka vytvoření funkce **maximum**

- Vyzkoušíme různé parametry
- Různé návratové hodnoty
- Různé množství parametrů

Definice funkcí v samostatném souboru

- Přípona .h – obsahuje deklarace funkcí (nikoli definice)
- Přípona .cpp – obsahuje definice funkcí

Šablony – Jak se neopakovat

Úkoly

- Napište funkci **int nsd(int x, int y)**, která spočte a vrátí největšího společného dělitele dvou čísel.
- Napište funkci **int nsn(int x, int y)**, která spočte a vrátí nejmenší společný násobek čísel dvou čísel.