# Úvod do programování

# **Lekce 1**

## *Základní pojmy*

**vytvoření spustitelného kódu**

- editor - psaní zdrojových souborů
- preprocesor - zpracování zdrojových souborů (vypuštění komentářů atd.)
- kompilátor (compiler) - překlad do objektového kódu s relativním adresováním
- linker - skládání objektových kódů do spustitelného souboru

**soubory**

- zdrojové (source) `*.cpp`
- hlavičkové (header) `*.h`

**ASCII kód**

- řídící znaky, např. znak konce řádku
- pomocné znaky, např. `+`
- číslice `0,1,2, ...`
- pomocné znaky, např. `@`
- velká písmena `A,B,C,...`
- pomocné znaky, např. `]`
- malá písmena `a,b,c,...`
- pomocné znaky, např. `~`

**identifikátory** - názvy proměnných, funkcí, atd.

- rozlišují se malá/velká písmena
- identifikátory můžou obsahovat podtržítko
- doporučují se dostatečně dlouhá popisná jména

**komentáře**

- blok `/* ... */`
- do konce řádku `// ...`

**základní typy proměnných**

- znakový `char`
- celočíselný `int,long int (long)`
- reálný `float,double,long double`

**konstanty**

- znakové `'a'`
- číselné `'2'`
- řetězcové `"abcd4"`
- speciální `'\n'`

**operátory**

- binární `+,-,*,/,%` (zbytek po dělení), `...`
- unární `++` (inkrementace), `--`, `...`
- přiřazovací `=,+=,-=,%=,=/,...`

- relační `<, >, <=, >=, ==` (rovnost), `!=`(nerovnost), `...`
- logické `&&` (logický součin), `||`(logický součet), `!` (negace), `...`

## *Vstup a výstup - standardní zařízení*

objekty zajišťující standardní vstup/výstup jsou definovány v hlavičkovém souboru `iostream` a jsou součástí standardní knihovny C++ a prostoru jmen (namespace) `std`

***formátovaný vstup a výstup***

- objekty `cin` a `cout` definují vstupní a výstupní proud (stream)
- přetížený (overloaded) operátor - volá různé funkce podle typu a počtu operandů
- pro formátovaný vstup/výstup jsou přetíženy operátory bitových posuvů `<< >>`

manipulátory - objekty modifikující proud

- `endl` - pošle znak konce řádku `'\n'` a zavolá `flush`
- `flush` - zapíše obsah výstupního bufferu

příklad: program přečte jeden znak za vstupu a opíše ho na výstup

```
#include<iostream>
using namespace std;
int main(void){          // kazdy program obsahuje funkci "main"
  char a;                // definice jedne promenne typu char
  cin >> a;              // precte znak a ulozi ho do promenne "a"
  cout << a << endl;     // opise znak z promenne "a" na vystup
}
```

poznámky:

- každý příkaz končí středníkem
- samotný středník je prázdný příkaz (nedělá nic)
- složené závorky vytvářejí z posloupnosti příkazů jeden složený příkaz
- funkce "main" je volána jako první, může ale volat další funkce

příklad: program přečte malé písmeno z klávesnice a konvertuje ho na velké písmeno

```
#include<iostream>
using namespace std;
int main(void){
  char a;
  int posun;
  cin >> a;
  posun = 'Z'-'z';  // vypocet "vzdalenosti" mezi velkym a malym
                    // pismenem v ASCII kodu
  a += posun;       // konverze na velke pismeno
  cout << a << endl;
}
```

poznámky:

- program využívá fakt, že rozdíl mezi ASCII kódem malého a velkého písmene je pro všechna písmena stejný
- aritmetické operace aplikované na znaky se provádějí s ASCII kódy (t.j. dochází k automatické konverzi mezi typy `char` a `int`.
- místo `a+=posun` lze také použít `a=a+posun`

příklad: program přečte délky odvěsen a vytiskne délku přepony na pět desetinných míst.

```
#include<iostream>
#include<cmath>
using namespace std;
int main(void){
  double a,b,c;
  cout << "Zadej delku strany a: ";     // opise retezec
  cin >> a;              // ulozi prectene cislo do promenne "a"
  cout << "Zadej delku strany b: ";
  cin >> b;
  c = sqrt(a*a+b*b);                    //vypocet delky prepony
  cout << "Delka prepony je " << c << " metru." << endl;
}
```

poznámka:

- matematické funkce, např. výše použitá druhá odmocnina `sqrt()`, jsou popsány v hlavičkovém souboru `cmath`

příklad: program přečte znak a vypíše odpovídající ASCII kód

```
#include<iostream>
using namespace std;
int main(void){
  char a;
  int d;
  cout << "Zadej znak: ";
  cin >> a;
  d = a;                 // konverze
  cout << "Znak '" << a << "' ma ASCII kod " << d << "." << endl;
}
```

stejný výsledek jinak:

```
#include<iostream>
using namespace std;
int main(void){
  char a;
  cout << "Zadej znak: ";
  cin >> a;
  cout << "ASCII kod: " << (int) a << endl;  // pretypovani
}
```

příklad konverze double na int: program vytiskne zvlášť celou a desetinnou část reálného čísla

```
#include<iostream>
using namespace std;
int main(void){
  double x,desetinna;
  int cela;
  cout << "Zadej cislo: ";
  cin >> x;
  cela = x;              // implicitni konverze
  desetinna = x - cela;
  cout << "Cela cast: " << cela << endl;
  cout << "Desetinna cast: " << desetinna << endl;
}
```

příklad: ověřit přesnost typů `float` a `double`, vypsat `1/3` s přesností na `20` desetinnych míst

```
#include<iostream>
#include<iomanip>
using namespace std;
int main(void){
  float x;
  double y,z;
  x = 1./3.;
  y = 1./3.;   // stejna konstanta ale jiny typ
  z = 1/3;     // pozor na celociselne deleni
  cout << fixed << setprecision(20);
  cout << "float:" << setw(30) << x << endl;
  cout << "double:" << setw(25) << y << endl;
  cout << "double:" << z << endl;
}
```

poznámky:

- číselné konstanty neceločíselných typů je nutno psát s desetinnou tečkou
- výraz `1/3` představuje celočíselné dělení, předá se celá část výsledku dělení t.j. nula
- zbytek po dělení lze získat operátorem `%`, např. `int zbytek; zbytek=7%4;` v proměnné `zbytek` bude `3` - často se používá pro ověření dělitelnosti apod.
- další užitečné manipulátory jsou definovány v `iomanip`, jako např. výše použité nastavení celkové šířky `setw()` a počtu desetinných míst `setprecision()`

***neformátovaný vstup a výstup***

`int cin.get()`

- přečte jeden znak ze standardního vstupu
- vrací `int` t.j. pracuje i s hodnotami mimo rozsah ASCII

`cout.put(int)`

- zapíše jeden znak na standardní výstup

příklad: opíše první tři zadané znaky

```
#include<iostream>
using namespace std;
int main(void){
  char a;
  cout << "Zadej 'a bc'" << endl;
  cin >> a; cout << a;
  cin >> a; cout << a;
  cin >> a; cout << a;
  cout << endl;
  cout << "Znovu zadej 'a bc'" << endl;
  cin.get();              // odstrani predchozi ENTER
  a = cin.get(); cout.put(a);
  a = cin.get(); cout.put(a);
  a = cin.get(); cout.put(a);
  cout.put('\n');
}
```

poznámky:

- operátor `>>` ignoruje bílé znaky, např. tabulátor, mezera
- tj. po zadání `'ab c'` je nejprve vypsáno `'abc'` pak `'a b'`, zkuste také tabulátor místo mezery
- `cin.get()` čte speciální znaky, včetně např. `'\n'`