

Proudy v C++

Úvod

Konzolový vstup a výstup pomocí proudů (streamů)

- v jazyce C++ jsou definovány tři streamy (proudy) v knihovně `iostream`:
 - výstupní (na `stdout`) `cout`
 - vstupní (ze `stdin`) `cin`
 - chybový (na `stderr`) `cerr`
- k proudům se váží dva operátory `<<`, `>>`, jejichž význam je pro tento účel předefinován (tzv. **přetížení operátorů**)

- proudy `cout`, `cin`, `cerr` jsou objekty (tři proměnné objektového typu `ostream` a `istream`, již deklarované v knihovnách)
 - nejde tedy o příkazy
 - funkcionality je skryta v přetížení operátorů (tj. k přiřazení funkce k operátorům `<< a >>` vzhledem k `ostream` a `istream`)
- nic nebrání nadále používat funkce `printf`, `scanf`
- často se plete `<< a >>`

- příklad:

```
#include <iostream>
int main(void)
{
    int pl,pz;
    cout << "Zadejte pocet lidi a
zvirat: ";
    cin >> pl >> pz;
    cout << "Pocet lidi je " << pl <<
    ", pocet zvirat je " << pz << '.';
}
```

Mnemotechnická pomůcka

při výstupu data směřují
do streamu



```
cout << "Ahoj";
```

při vstupu data směřují **ze**
streamu do proměnné



```
cin >> pz;
```

Manipulátory

- slouží pro řízení vstupní a výstupní konverze
- jsou definovány v `iomanip`

```
// nový řádek
cout << "Ahoj" << endl;
// nastaví šestnáctkový výpis
cout << hex << x;
// vstup čísla v šestn. soustavě
cin >> hex >> a;
```

`dec`

`hex`

`oct`

`endl`

`setw(int n)`

`setfill(int c)`

`setprecision(int n)`

`showpos`

`noshowpos`

`boolalpha`

- **více v nápovědě**

dekadická konverze

šestnáctková konv.

osmičková konverze

konec řádku + “flush”

šířka položky *n* znaků

plnicí znak *c*

n desetinných míst

vypíše znaménko

zruší výpis znaménka

vypíše true/false

Jmenné prostory

(namespaces)

Prostory jmen

- při vkládání několika hlavičkových souborů může vzniknout kolize

zeleznice.h

```
const int x=10;  
typedef enum  
{  
    STOP, CONT  
} Stavy;
```

logika.h

```
const int x=5;  
typedef enum  
{  
    DELI, NEDELI  
} Stavy;
```

```
#include "zeleznice.h"
#include "logika.h"
void main(void)
{

    Stavy s1, s2;
    int a = x+3;
}
```



Které Stavy?

**ze zeleznice.h
nebo z logika.h**

- řešením je vnoření deklarace do prostoru jmen

zeleznice.h

```
namespace Zeleznice {  
    const int x=10;  
    typedef enum  
    {  
        STOP, CONT  
    } Stav; }  
}
```

logika.h

```
namespace Logika {  
    const int x=5;  
    typedef enum  
    {  
        DELI, NEDELI  
    } Stav; }  
}
```

- mimo deklarovaný prostor není jméno přístupné, musíme se odkázat na příslušný prostor jmen

Logika::Stav

```
#include "zeleznice.h"  
#include "logika.h"  
  
void main(void)  
{  
    Zeleznice::Stavy h1;  
    Logika::Stavy h2;  
    int a = Logika::x+3;  
}
```

- abychom nemuseli psát stále odkaz na prostor jmen, lze dosáhnout přímé viditelnosti pomocí direktivy **using**

```
#include "zeleznice.h"
#include "logika.h"

using namespace Zeleznice;
void main(void)
{
    Stavy h1;
    Logika::Stavy h2;
    using namespace Logika;
    int a = x+3;
}
```

- deklarace jmenných prostorů lze vnořovat

```
namespace Zeleznice {  
  
    const int x = 5;  
    namespace Vlak {  
        const int Delka_Vlaku = 500;  
    }  
  
    namespace ETCS {  
        const int ETCS_Level = 1;  
    }  
}
```

- do kódu zapíšeme

```
Zeleznice::x
```

```
Zeleznice ::Vlak::Delka_Vlaku
```

```
Zeleznice ::ETCS::ETCS_Level
```

- deklarace jsou otevřené, tj. deklarace lze přidávat

- v prvním souboru: **namespace** A { ... }

- ve druhém souboru: **namespace** A { ... }

- pro běžné objekty, např. `cout`, je definován jmenný prostor `std`

- nesmíme tedy zapomenout na počátku programu uvést

```
using namespace std;
```

- jména lze zkracovat pomocí aliasů:
 - **namespace** `Zel=Zeleznice;`

Vyzkoušejte...

```
#include <iostream>
#include <iomanip>
int main(int argc, char **argv)
{ int x;
  cout << "Ahoj, svete!" << endl;
  cout << hex << 65 << endl;
  cout << "Zadej cislo v osm. soustave: ";
  cin >> oct >> x;
  cout << dec << x << endl;
  cout << showpos << x << endl;
  cout << setw(5) << setfill('0') << x << endl;
  cout << boolalpha << true;
  return 0;
}
```

- tento kód nepůjde přeložit (překladač napíše, že nezná cout)

Vyzkoušejte...

```
#include <iostream>
#include <iomanip>
int main(int argc, char **argv)
{ int x;
  std::cout << "Ahoj, svete!" << std::endl;
  std::cout << std::hex << 65 << std::endl;
  ...
  return 0;
}
```

Vyzkoušejte...

```
#include <iostream>
#include <iomanip>

using namespace std;
int main(int argc, char **argv)
{ int x;
  cout << "Ahoj, svete!" << endl;
  cout << hex << 65 << endl;
  cout << "Zadej cislo v osm. soustave: ";
  cin >> oct >> x;
  cout << dec << x << endl;
  cout << showpos << x << endl;
  cout << setw(5) << setfill('0') << x << endl;
  cout << boolalpha << true;
  return 0;
}
```