

Programování v C++

Namespace, Výjimky a IO 11

Pavel Fibich

pavel.fibich@prf.jcu.cz

8-12-2010

namespace

Namespace seskupuje třídy, funkce a objekty. Globální úroveň se tak může rozdělit do podčástí (sub-scopes).

```
#include <iostream>
using namespace std;
namespace first {
    int x = 5;
    int y = 10;
}
namespace second {
    double x = 3.1416;
    double y = 2.7183;
}
int main () {
    using namespace first;
    cout << x << endl; cout << y << endl;
    cout << second::x << endl;
    cout << second::y << endl;
    return 0;
}
```

výjimky

Výjimky umožňují reagovat na nečekané události (runtime chyby), např. `bad_alloc`, `bad_cast`, `ios_base::failure`, ..

```
#include <iostream>
using namespace std;

int main () {
    try {
        int* myarray= new int[1000000000000000];
    }
    catch (bad_alloc&) {
        cout << "Bad alloc exception!" << endl;
    }
    catch (...) {
        cout << "Obecná exception!" << endl;
    }
    return 0;
}
```

výjimky

Můžeme si udělat vlastní výjimku

```
#include <iostream>
#include <exception>
using namespace std;

class myexception: public exception {
    virtual const char* what() const throw() {
        return "My exception happened";
    }
} myex;

int main () {
    try {
        throw myex;
    }
    catch (exception& e) {
        cout << e.what() << endl;
    }
    return 0;
}
```

VV soubory

Třída `fstream`

- `ofstream` - jen zápis
 - `ifstream` - jen čtení
 - `fstream` - čtení i zápis
-

```
#include <iostream>
#include <fstream>
using namespace std;
```

```
int main () {
    ofstream myfile;
    myfile.open ("priklad.txt");
    myfile << "Text k zapisu do souboru.\n";
    myfile.close();
    return 0;
}
```

VV soubory - otevření

Módy pro otevření souboru `open (filename, mode);`

- `ios::binary` - binární
- `ios::ate` - pozice na konci souboru
- `ios::trunc` - starý soubor nahraď novým
- ...

```
myfile.open ("priklad.bin", ios::out | ios::app | ios::↵  
    binary);
```

```
if (myfile.is_open()) { ... // test jestli se povedlo ↵  
    otevreni
```

```
myfile.close(); // nakonci uzavrit soubor
```

VV soubory - čtení text. souboru

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

int main () {
    string line;
    ifstream myfile ("priklad.txt");
    if (myfile.is_open()) {
        while (! myfile.eof() ) {
            getline (myfile,line);
            cout << line << endl;
        }
        myfile.close();
    }
    else cout << "Soubor nejde otevrit";
    return 0;
}
```

VV soubory - flagy a pozice

Krom `eof()` můžeme testovat `fail()` (je `false` když se podařil zápis nebo čtení).

Příkazy pro pozicování `ifstream` končí **g** a pro `ofstream` končí **p**.

```
#include <iostream>
#include <fstream>
using namespace std;
```

```
int main () {
    long begin,end;
    ifstream myfile ("prikklad.txt");
    begin = myfile.tellg();
    myfile.seekg (0, ios::end);
    end = myfile.tellg();
    myfile.close();
    cout << "size is: " << (end-begin) << " bytes.\n";
    return 0;
}
```

proudy (streamy)

Přetěžování

```
ostream& operator<<(ostream& proud, komplex z)
{ return proud <<' ('<< z.real() << ',' << z.imag() <<')'; }
```

Vlastní manipulátor (např. místo endl)

```
ostream& oddel(ostream& proud)
{ return proud << "\n-*-*-*" << endl; }
```

Desetinná místa

```
double f =3.14159;
cout << setprecision (5) << f << endl;
```

Zarovnání

```
int n=-77;
cout.width(6);
cout << right << n << endl;
```

Ukol 5. alias projekt - podmínky

- odvedzení přes <http://moodle.prf.jcu.cz/course/view.php?id=16>
- 0-20 bodů
- na projekt máte 3 týdny
- odevzdáte zabalený (.zip nebo .tar.gz) soubor:
komentovaný zdrojový soubor `prijmeni_jmeno.cpp`, z něj
přeložený soubor (`prijmeni_jmeno.exe`) a `readme.txt`
- v `readme` bude vyplněna následující hlavička

AUTOR: (jmeno prijmeni, osobni cislo zacinajici B)

EMAIL:

SKUPINA: (kam chodite a jste zapsani na cviceni)

VERZE G++: (ziskate pomoci prikazu `g++ -v`)

PRIKAZ PREKLADU: (jaky prikazem byl program prelozen)

OS: (operacni system, napr. Win 7 64b)

POZNAMKY K IMPLEMENTACI:

Ukol 5. - zadání

Napište program na analýzu zdrojových kódů v C++. Program bude počítat speciální znaky, klíčová slova, počet komentářů a počet řádků v souboru zadaném jako parametr a výsledek uloží nakonec dalšího souboru na zvlášť řádek.

- Program bude akceptovat právě 2 argumenty (názvy souborů):
 - první bude název souboru kam program uloží výsledky, výsledky se budou ukládat na konec souboru
 - druhý bude název souboru k analýze
- Program otestuje existenci souborů a v případě neúspěchu skončí.
- Ve výstupním souboru bude v jednom řádku **název vstupního souboru, počet jeho řádků, počet komentářů** a počet výskytů následujících slov a znaků, počty budou oddělené mezerou a budou v uvedeném pořadí: `return`
`int string float char double if for while`
`void class cout = ; " . , ( ) < > [ ] { } -`
`+ %`

Ukol 5. - zadání

- Jednotlivý řádek ve výstupním souboru pak bude začínat např.

```
soubor1.txt 60 12 2 4 5 5 6 6 6 6 2 3
```

kde soubor1.txt ma 60 řádků, 12 komentářů, 2 výskyty return, 4 výskyty int, ..

- Počty slov a znaků nebudou zahrnovat výskyty v komentářích.
- Program bude mít alespoň 3 funkce.
- Komentář v komentáři se počítá za jeden komentář.
- To co program zapíše na konec výstupního souboru, vypíše ve stejném formátu i na standartní výstup (tj. pomocí `cout`).

Než začnete zvažte co použijete a podívejte se do referenční příručky. Ne všechno je třeba řešit hrubou silou. Řešte projekt po částech, pokud něco nezvládnete napište to do implementační poznámky.