

Богатов Р.Н.

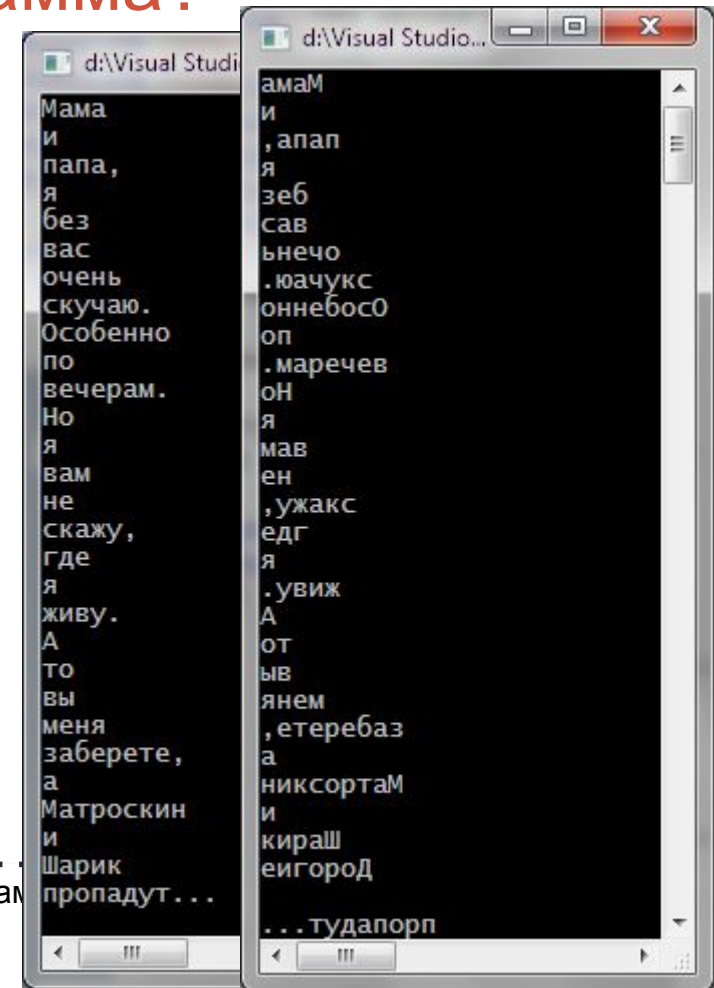
Программирование на языке высокого уровня

C++ ► Лекция 7 ► Модульное программирование

Кафедра АСОИУ ОмГТУ, 2012

Повтор. Что делает эта программа?

```
...
for (; !feof(f);)
{
    char c=' ';
    for (; !feof(f) && c==' ');
        c = fgetc(f);
    char s[500] = {c};
    int j=1;
    for (; !feof(f) && c!=' '; j++)
        s[j] = (c = fgetc(f));
    s[j-1]=0;
    printf( "%s\n", strrev(s) );
}
...
```



Мама и папа, я без вас очень скучаю. Особенно по вечерам. Но я вам не скажу, где я живу. А то вы меня заберете, а Матроскин и Шарик пропадут...

Дорогие папа и мама, вы меня теперь просто не узнаете. Хвост у меня крючком, уши торчком, нос холодный, и лохматость повысилась...

До свидания. Ваш сын - дядя Фёдор

файл myfile.txt

Разбивка кода (псевдомодульность)

```
#include <stdio.h>
#include <conio.h>
#include <locale.h>
```

```
void main()
{
```

```
    setlocale(LC_ALL, "Russian");
```

```
    printf( "(с) Вася Пупкин, каф. АСОИУ ОмГТУ, 2012\n\n" );
```

```
    #include "beginning.inl" // часто используемое начало
```

```
    printf("Это моя первая программа.\n" );
```

```
    for(int i=0; i<10; i++)
        printf("%d ", i);
```

```
    #include "ending.inl" // часто используемый конец
```

```
}
```

```
    printf("\n\nНажмите любую клавишу для выхода..." );
    getch();
```

файл beginning.inl

файл ending.inl



The screenshot shows a Windows command prompt window titled "d:\Visual Studio\Projects\C++\proga\Debug\proga.exe". The output of the program is displayed as follows:

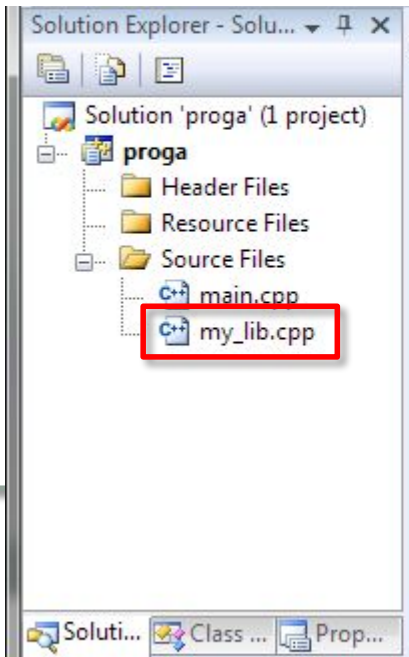
```
(с) Вася Пупкин, каф. АСОИУ ОмГТУ, 2012

Это моя первая программа.
0 1 2 3 4 5 6 7 8 9

Нажмите любую клавишу для выхода..._
```

Вынос функций в отдельный модуль

```
#include <stdio.h>
#include "my_lib.h"
```



```
printf("Это
for(int i=0
printf
my_ending
}
```

1. Компилятор видит в проекте два **cpp**-файла. Они не ссылаются друг на друга, поэтому компилируются отдельно в два **obj**-файла.

2. В **main.obj** компилятор оставляет «недоделанным» вызов двух неизвестных функций, про которые из **my_lib.h** компилятору известны только их декларации (входные и выходные параметры).

3. Редактор связей (ака «линковщик») получает задание собрать из двух **obj**-файлов программу (исполнимый модуль, **exe**-файл). Он находит в **main.obj** функцию **main**, которая должна быть главной, видит вызовы функций, которых в **main.obj** нет, находит их в **my_lib.obj**, подлинковывает (приклеивает) их и редактирует связи (настраивает адреса вызова этих функций). Программа готова.

Если есть **my_lib.obj** (был однажды скомпилирован), то его можно распространять вместе с **my_lib.h**, но без **my_lib.cpp** (например, чтобы скрыть исходный код, содержащий ноу-хау). В частности, эта программа будет благополучно компилироваться и линковаться, если удалить **my_lib.cpp**, но оставить **my_lib.obj**. Чтобы это сделать, нужно исключить из проекта ссылку на **my_lib.cpp** и добавить вместо него **my_lib.obj**.

Пример: простейшая база данных квитанций

...

```
void db_add(void)
```

...

```
void db_view(void)
```

...

```
void db_sum(void)
```

```
{
```

```
    printf( "\n\nПоиск по базе 'my_database.dat'\n\n" );
```

```
    int A = 0, B = 0;
```

```
    printf( "Введите диапазон номеров квитанций (через пробел): " );
```

```
    scanf( "%d%d", &A, &B );
```

```
    fflush(stdin);
```

```
    printf( "\n\tКвитанция: Сумма:\n" );
```

```
    fseek( f, 0, SEEK_SET );
```

```
    block t;
```

```
    double sum = 0;
```

```
    while( true ) {
```

```
        if( fread( &t, sizeof(block), 1, f ) !=1 ) break;
```

```
        if( t.number>=A && t.number<=B )
```

```
            printf( "\t%6d%10.2lf\n", t.number, t.sum ), sum += t.sum;
```

```
    }
```

```
    printf( "\nСумма по диапазону квитанций: %lg руб.\n\n", sum );
```

```
}
```

...

файл my_database.cpp

Домашнее задание

Написать модуль для работы с комплексными числами, в котором реализовать функции `cmplx_add`, `cmplx_sub`, `cmplx_mul` и `cmplx_div` (сложения, вычитания, умножения и деления), а также `cmplx_make` (формирования комплексного числа из вещественных действительной и мнимой частей), `cmplx_conj` (получения сопряжённого числа) и `cmplx_print` для вывода на экран.

```
// наброски кода
struct CMPLX
{
    double Re, Im;
};

CMPLX cmplx_make(double a, double b)
{
    CMPLX c = { a, b };
    return c;
}

CMPLX cmplx_add(CMPLX a, CMPLX b)
{
    return cmplx_make(a.Re + b.Re, a.Im + b.Im);
}

void main()
{
    CMPLX a = { -1, 3.4 };
    CMPLX b = { 2.5, -10 };
    cmplx_print( cmplx_add(a, b) );
}
```