# **ФИТ НГУ, курс ОООП, осенний семестр**

# **Задание №0 (вводное). Раздельная компиляция и пространства имен (namespaces)**

### 

## ***Задача 0а***

* 1. Познакомьтесь с раздельной компиляцией и пространством имен в C++. В качестве примера используйте следующую программу:

**File *module1.h:***

#include <string>

namespace Module1

{

std::string getMyName();

}

**File *module1.cpp***

#include "module1.h"

namespace Module1

{

std::string getMyName()

{

std::string name = "John";

return name;

}

}

**File *module2.h:***

#include <string>

namespace Module2

{

std::string getMyName();

}

**File *module2.cpp***

#include "module2.h"

namespace Module2

{

std::string getMyName()

{

std::string name = "James";

return name;

}

}

**File *main.cpp:***

#include "module1.h"

#include "module2.h"

#include <iostream>

int main(int argc, char\*\* argv)

{

std::cout << "Hello world!" << "\n";

std::cout << Module1::getMyName() << "\n";

std::cout << Module2::getMyName() << "\n";

using namespace Module1;

std::cout << getMyName() << "\n"; // (A)

std::cout << Module2::getMyName() << "\n";

//using namespace Module2; // (B)

//std::cout << getMyName() << "\n"; // COMPILATION ERROR (C)

using Module2::getMyName;

std::cout << getMyName() << "\n"; // (D)

}

1. С тестовой программой нужно выполнить следующие действия:
2. Собрать программу и убедиться, что на каждый *\*.cpp* файл создается отдельный объектный файл с тем же именем (для Visual Studio, например, в папке Debug будут создаваться файлы с расширением *\*.obj*).
3. Убедиться, что при изменении одного *\*.cpp* файла и пересборке проекта обновляется только соответствующий ему объектный файл (дата изменения других объектных файлов останется прежней)
4. Объяснить, что выведется при выполнении строк с комментариями **(А)** и **(D)** в *main.cpp*
5. Убедиться, что раскомментирование строк **(B)** и **(C)** в *main.cpp* приводит к ошибке компиляции. Объяснить, почему эта ошибка происходит, и предложить пути её устранения.
6. Добавить в программу еще одну функцию *getMyName()*, возвращающую имя **Peter**. Обернуть её в еще одно пространство имён.
7. Объяснить, как можно избавиться от необходимости писать **std::cout** и вместо этого писать просто **cout**.

## ***Задача 0б***

Напишите программу, которая будет принимать в качестве аргумента имя текстового файла, и выводить CSV файл (<http://ru.wikipedia.org/wiki/CSV>) с колонками:

1. Слово.

2. Частота.

3. Частота (в %).

CSV файл должен быть упорядочен по убыванию частоты, то есть самые частые слова

должны идти в начале. Разделителями считать все символы кроме букв и цифр.

Аргументы командной строки программы:

**> word\_count.exe input.txt output.csv**

Требования к программе:

1. Для работы со строками используйте класс стандартной библиотеки *std::string* (см. <http://www.cplusplus.com/reference/string/string/>)
2. Работа с файлами должна осуществляться с помощью классов стандартной библиотеки из модуля *fstream* (<http://www.cplusplus.com/reference/fstream/>)
3. Строки из файла должны зачитываться с помощью метода *std::getline* (см. <http://www.cplusplus.com/reference/string/string/getline/>)
4. Сохраняйте зачитанные строки в контейнере стандартной библиотеки *std::list* (<http://www.cplusplus.com/reference/list/list/>), *std::map* (<http://www.cplusplus.com/reference/map/map/>)