

## Практическое задание №7 (перегрузка функций)

Изучить механизм перегрузки функций (см. материалы Лекции 6 и Лекции 7).

**Перегрузкой функций** (от англ. **overloading**) называют технику, обеспечивающую возможность ввести различную реализацию функций с одинаковыми именами, отличающихся списками параметров. Тип возвращаемого значения при разрешении перегрузки не учитывается.

1. Дана **статическая библиотека** (файл *Override.lib*), в которой реализованы различные варианты перегрузки функции `int F(T)`. В библиотеке содержится всего 21 вариант функции **F** для различных типов аргумента T. Каждая такая функция содержит единственную строку `return N`; (N, соответственно, имеет различные значения в функциях от 1 до 21).

Библиотека скомпилирована в конфигурации **Release** для **x86**-платформы.

2. Заголовочный файл (с прототипами перегруженных функций библиотеки) отсутствует.
3. Клиентская программа зависит от библиотеки *Override*, её код простейший, состоит из вызова функции **F** (например, с аргументом типа `bool`):

```
// Листинг 1: client.cpp
#include "F.h"

#include <iostream>

int main(int argc, char** argv)
{
    std::cout << F(true) << std::endl;
    return 0;
}
```

**Задача:** Восстановить отсутствующий заголовочный файл *F.h*, для чего необходимо найти все варианты (их 21) перегрузки функции **F**.

Для сдачи на проверку преподавателю должен быть представлен сформированный **заголовочный файл F.h**, в котором содержатся различные прототипы (варианты перегрузки) функции **F**, в комментарии к каждому прототипу указать, какое значение возвращает этот вариант перегрузки. Например:

```
// Листинг 2: Пример результирующего заголовочного файла F.h
#pragma once

int F(int);           // Возвращает 0
int F(другой тип);   // Возвращает 1
// и так далее
```

**Рекомендация.** При поиске очередного варианта перегрузки стоит закомментировать все ранее найденные прототипы во избежание неопределённости при компиляции.