# Классы памяти

Данная работа состоит из 2 частей (распределяемых по вариантам), которые должны быть выполнены в виде отдельных функций и модулей, и основной функции main, размещаемой в одном из модулей. Функция main должна демонстрировать работу заданных функций путём печати результатов их выполнения на экран. Режим демонстрации разрабатывается самостоятельно, основным условием является демонстрация всех ключевых моментов, указанных в задании.

*Таблица соответствия вариантов и заданий:*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вариант** | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* |
| **Задание 1** | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| **Задание 2** | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 |

*далее – циклически.*

Программа должна состоять из 2 или более программных модулей. В одном и них должна размещаться функция main(), а также функции из первого задания. Во втором модуле размещаются некоторые функции из задания 2.

Создание нескольких модулей подразумевает самостоятельное освоение работы с проектными файлами (добавление, удаление модулей).

## Задание 1.

Используя только статические и локальные переменные, создать модули, которые содержат:

Функция-счётчик. Возвращает целое значение, увеличивающееся на 1 при каждом вызове. Начальное значение равно 0.

## Задание 2.

Используя глобальные переменные, обеспечить передачу информации между функциями первого (main) и второго модулей. Второй модуль должен содержать 2 функции: add и sub, вычисляющих соответственно сумму и разность 2 входных данных: x и y. Точные определения функций (в зависимости от варианта) приведены ниже:

void Add(int x, int y);

int Sub();

***Указания:*** Входные данные и результат вычислений передаются различным образом в зависимости от варианта. Если возможна передача данных как обычных аргументов функции или возможно получить результат вычислений в виде возвращаемого функцией значения, нужно использовать этот метод. В противном случае, необходимо создать дополнительные глобальные переменные и пользоваться ими как временным хранилищем для передачи информации.

Например, в функцию int Func(int x, int y); данные можно передать в виде аргументов, а результат считать непосредственно как результат выполнения функции:

Result = Func(var1, var2);

Напротив, в случае функции void Func2(int x), мы можем передать в неё только один из аргументов, и не можем непосредственно считать результат (так как функция имеет тип void). В этом случае можно создать глобальные переменные, например Y и Res, и воспользоваться тем фактом, что, при правильном объявлении, область действия обеих этих переменных распространяется на функции обоих модулей, таким образом, в одной из функций мы можем записать значение в переменную, а в другой – считать.

Программа, демонстрирующая работу второго задания, должна обеспечивать ввод 2 значений с клавиатуры и выводить результат вычислений, полученный с помощью функций второго модуля.