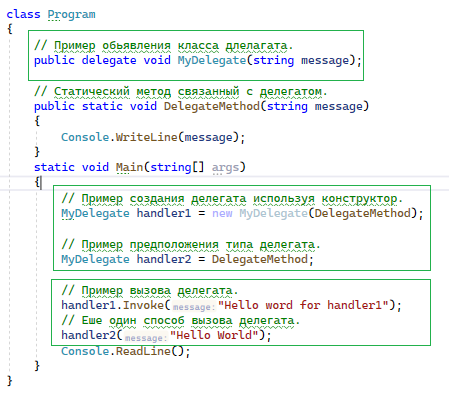
* 1. Делегаты(delegate), СОБЫТИЯ (event), анонимные методы

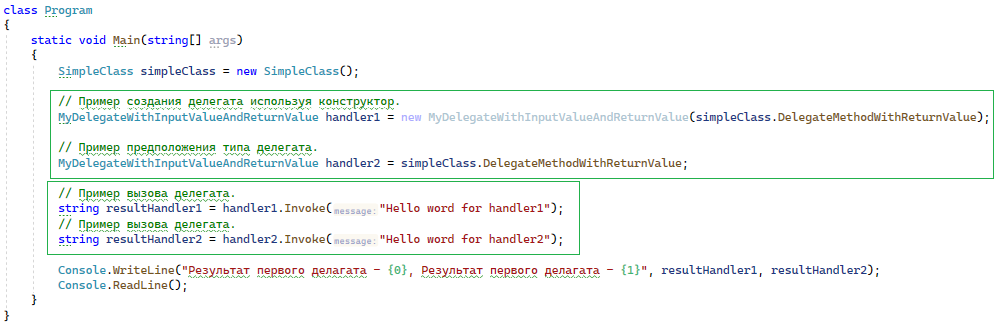
Делегат — это тип, который безопасно инкапсулирует метод, схожий с указателем функции в C и C++. В отличие от указателей функций в C делегаты объектно-ориентированы, типобезопасны и могут содержать ссылки на несколько методов. Создание делегата происходит про помощи ключевого слова delegate рис. 1.1. где задается сигнатура методов (тип и количество входных параметров, тип возвращаемого значения) которые могут быть назначены в этом делегате. На рис. 1.1 показан пример создания класса делегата MyDelegate. Методы который может быть назначен этому делегату должен принимать строковое значение и ничего не возвращать (void). Создать делегат можно к с помощью конструктора рис. 1.1 так и с помощью “предположения типа делегата”. Вызов методов на которые ссылается делегат осуществляется либо вызовом метода Invoke() рис. 1.1 (способ устарел) либо как вызов обычного метода. На рис 1.2 показан пример объявления класса делегата, возвращающего значение (string). На рис 1.3 показа пример работы с этим делегатом вызов метода, удовлетворяющего сигнатуре делегата рис 1.4 и получение значения.



*Рис. 1.1.* Пример работы с делегатом



*Рис. 1.2.* Сигнатура делегата, возвращающего значение



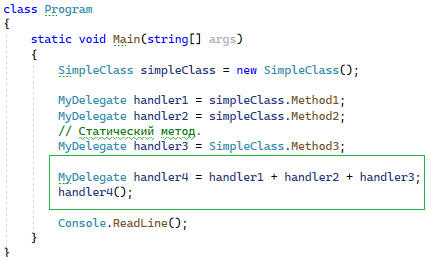
*Рис. 1.3.* Пример работы с делегатом, возвращающим значение

****

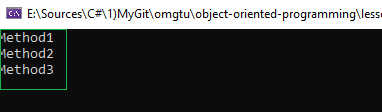
*Рис. 1.4.* Пример метода, удовлетворяющего сигнатуре делегата

* 1. Multicast delegate

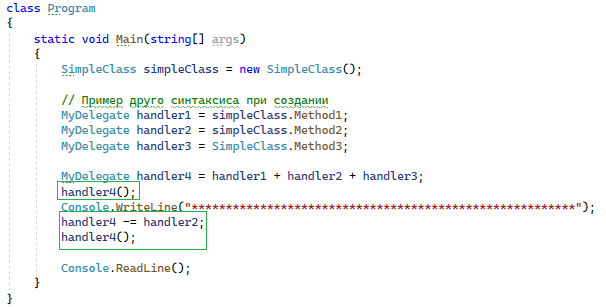
В отличие от указателей в C и С++ делегаты позволяют создавать так называемы Multicast делегаты или Групповые делегаты. Группой делегат содержит в себе силки на несколько методов удовлетворяющих сигнатуре делегата и момент вызова вызываются все эти методы по очереди. На рис 1.5 показано создание 3 делегатов handler1, handler2, handler3. Обратите внимание что делегат может быть связан как с обычным методом, так и со статическим. Далее мы создаем групповой делегат handler4 путем конкатенации 3 приводящих делегатов. Теперь при вызове handler4 будут вызываться все методы, связанные с этим делегатом рис 1.6. Отвязать один из обработчиков можно при помощи оператора -= что показано на рис 1.7. На рис 1.8 видно, как ведет себя групповой делегат до удаления handler2 и после. Если групповой делегат возвращает значение. То будет возвращено значение того метода, который был вызван последним рис 1.9, рис 1.10.



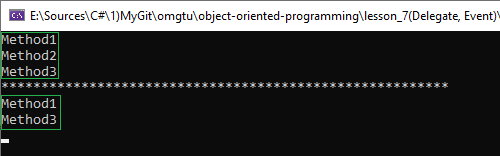
*Рис. 1.5.* Пример создание Multicast делегата



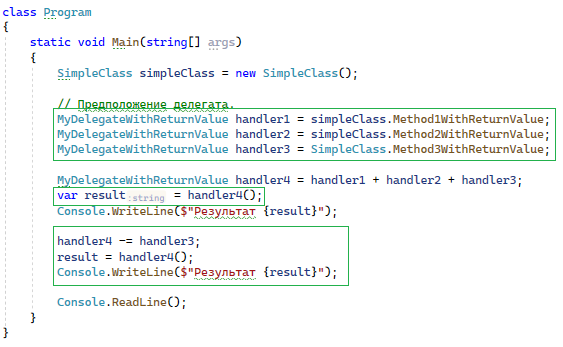
*Рис. 1.6.* Результат вызова Multicast делегата



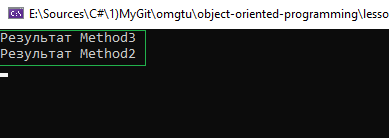
*Рис. 1.7.* Пример удаления метода из Multicast делегата



*Рис. 1.8.* Результат вызова Multicast делегата



*Рис. 1.9.* Пример Multicast делегата возвращающего значения



*Рис. 1.10.* Результат вызова Multicast делегата