**Билет 12**

Хорошо известное правило, называемое **законом Деметры**, гласит, что модуль не должен знать внутреннее устройство тех объектов, с которыми он работает. Обобщенно, закон Деметры является специальным случаем слабой связанности.

В более точной формулировке закон Деметры гласит, что метод f класса C должен ограничиваться вызовом методов следующих объектов:

* самого класса C;
* объекты, созданные f;
* объекты, переданные f в качестве аргумента;
* объекты, хранящиеся в переменной экземпляра C.

Метод не должен вызывать методы объектов, возвращаемых любыми из разрешенных функций. Другими словами, разговаривать можно с друзьями, но не с чужаками.

Говоря упрощённо, каждый программный модуль:

* должен обладать ограниченным знанием о других модулях: знать о модулях, которые имеют «непосредственное» отношение к этому модулю.
* должен взаимодействовать только с известными ему модулями «друзьями», не взаимодействовать с незнакомцами.
* обращаться только к непосредственным «друзьям».

Основной идеей является то, что объект должен иметь как можно меньше представления о структуре и свойствах чего угодно (включая собственные подкомпоненты).

Следующий код нарушает закон Деметры, потому что он вызывает функцию A() для возвращаемого значения C(), а затем вызывает B() для возвращаемого значения A(): C().A().B();

Еще **пример**:

Код A.B.Method() нарушает Закон Деметры, а код A.Method() является корректным.