# Public

Мы имеем доступ к этому классу из других классов. Доступно всё и везде, это одна крайность. (Private – другая крайность).

# Protected

* Короче вообще бесполезная штука скорее всего.
* То же самое, что и default, но, когда мы наследуем данный класс (даже в другом пакете), данный класс становиться доступен.

# Private

* Не применим к обычному классу, но применим к вложенному. К такому вложенному классу я имею доступ только внутри основного класса.
* К такому методу/полю имеет доступ исключительно класс, в котором она объявлена. Ну вроде как да, объект созданный в мейне не имеет доступ к приватным полям класса, на образце которого был создан.
* Применим к полю, методу и конструктору.
* Если я делаю конструктор private (все конструкторы), то таким образом я запрещаю создавать экземпляры класса. Но! Я всё еще могу создать экземпляр внутри данного класса.

Можем сделать следующий финт ушами:

1. Создать в классе статичный и публичный метод, который возвращает экземпляр класса.
2. Сделать конструктор приватным

Теперь мы всё ещё можем получить экземпляр класса, но всё это дело находится под нашим контролем.

# Не указанный (Default)

Ставится сам по себе, если мы сами не прописали модификатор.

Даёт доступ ко всему внутри данного пакета.

Если я не могу поставить классу модификаторы private и protected, то какие вообще могу?

Чел заявляет, что если мы классу ставим модификатор (public, private, protected), то все поля, методы и конструкторы тоже по – умолчанию имеют такой модификатор, можно ли ему верить?