Модификаторы доступа в Java:

Как следует из названия, модификаторы доступа - это ключевые слова, которые блокируют использование переменной, класса, конструктора, метода или данных.

* Default
* Public
* Private
* Protected

Public access modifier:

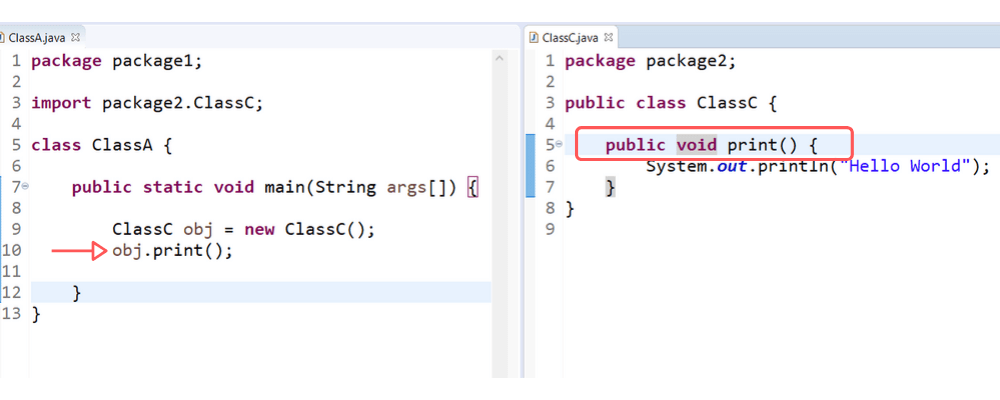
Мы можем сделать любую переменную, класс или данных общедоступными с помощью ключевого слова public.

Если вы делаете какой-либо элемент данных общедоступным, то к нему можно будет получить доступ из любой точки проекта.

(Public Access Modifier:

We can make any variable, class or data member public by using the public keyword.

If you’re making any data member public then it can be accessed from anywhere within a project.



Default access modifer:

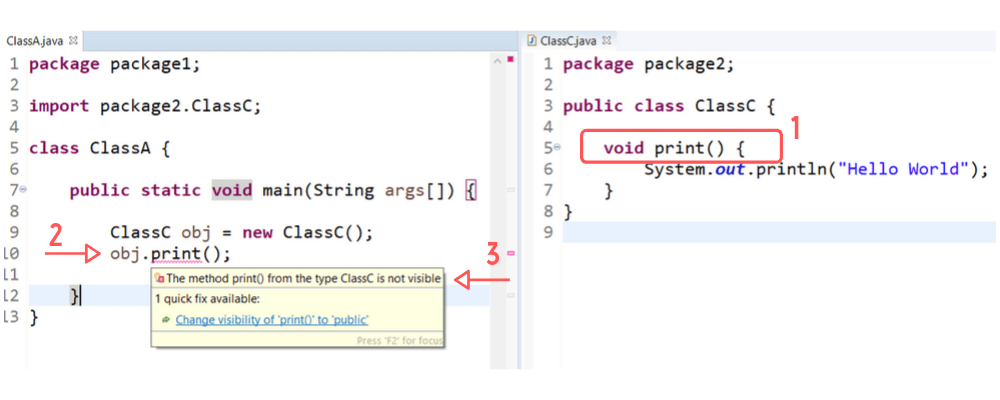
если мы не добавим модификаторы доступа к veribles, он будет Default по умолчанию

переменная по умолчанию, класс или член данных могут быть доступны только через один и тот же пакет

Default access modifier:

It will take default by default if we don’t declare access modifier to a variable

default variable, class or data member can only be accessed through the same package



**Protected access modifiers:**

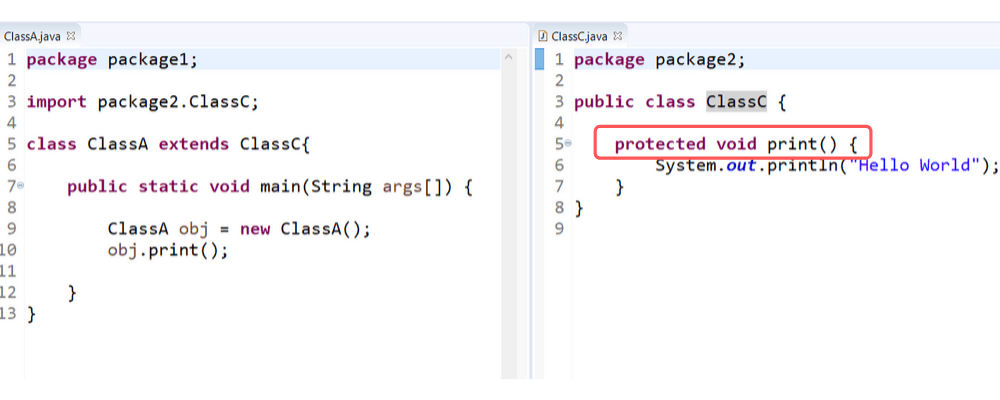
Мы можем сделать любую переменную, класс или член данных защищенными с помощью ключевого слова protected.

Если вы защищаете какой-либо элемент данных, то к нему можно будет получить доступ в тех же пакетах или подклассах.

Protected access modifier:

We can make any variable, class, or data member protected with the protected keyword.

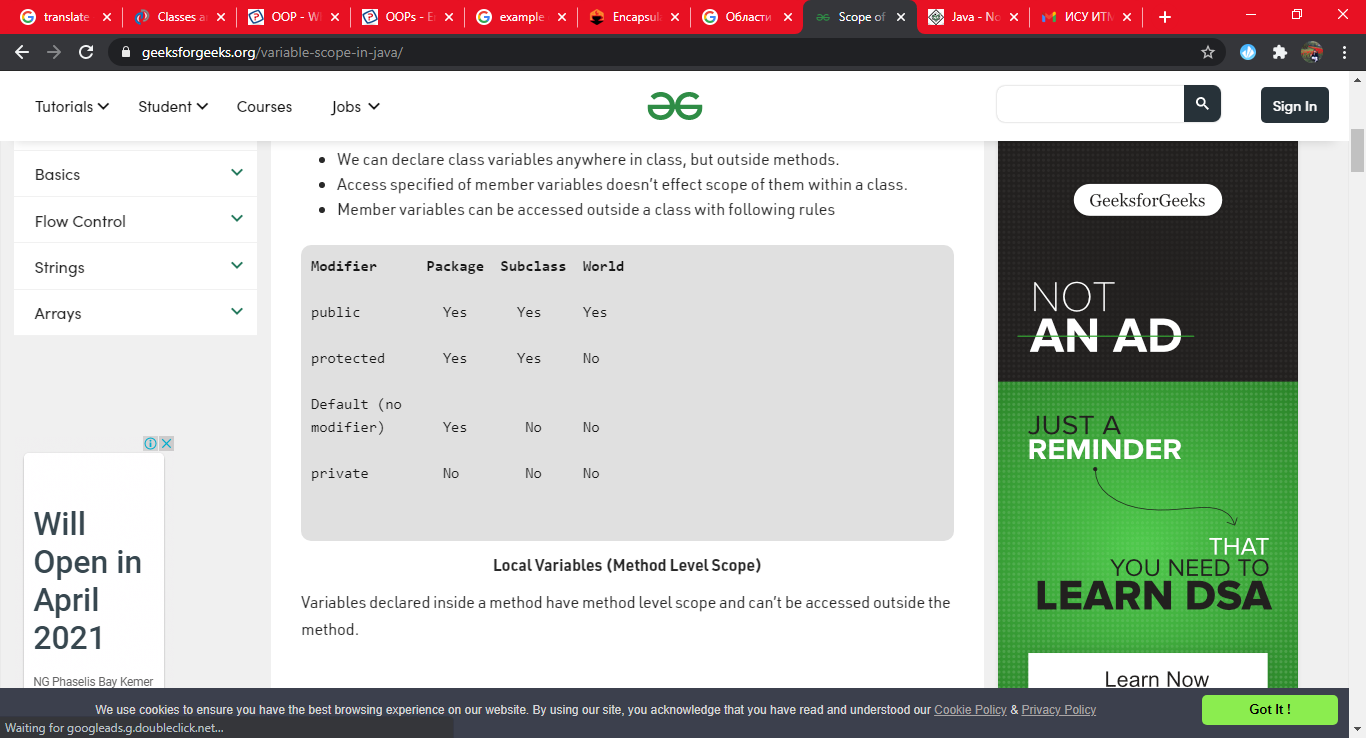
* If you are protecting any piece of data, then it can be accessed in the same packages or subclasses.



**Private**

Так как же скрыть данные? с помощью private access modifier. Если вы объявляете любого участника закрытым, то вы скрываете его от внешнего мира.

By using private access modifier we can save our data from showing to all



Scope of variables in java

Важные моменты об области видимости переменных в Java:

* Как правило, набор фигурных скобок {} определяет область действия.
* В Java мы обычно можем получить доступ к переменной, если она была определена в том же наборе скобок, что и код, который мы пишем, или в фигурных скобках внутри фигурных скобок, в которых была определена переменная.
* Любая переменная, определенная в классе вне любого метода, может использоваться всеми методами-членами.
* Чтобы переменная могла быть прочитана после завершения цикла, она должна быть объявлена ​​перед телом цикла.

**Important Points about Variable scope in Java:** 

* In general, a set of curly brackets { } defines a scope.
* In Java we can usually access a variable as long as it was defined within the same set of brackets as the code we are writing or within any curly brackets inside of the curly brackets where the variable was defined.
* Any variable defined in a class outside of any method can be used by all member methods.
* When a method has the same local variable as a member, this keyword can be used to reference the current class variable.
* For a variable to be read after the termination of a loop, It must be declared before the body of the loop.

Static переменные (static variable)

Ключевое слово static используется для создания переменных, которые будут существовать независимо от любых экземпляров, созданных для класса. Независимо от количества экземпляров класса существует только одна копия статической переменной.

Статические переменные также известны как переменные класса.

Static variables

The static keyword is used to create variables that will exist independently of any instances created for the class. Regardless of the number of instances of the class, there is only one copy of a static variable.

Static variables are also known as class variables..

Final переменные

Final переменная может быть явно инициализирована только один раз. Ссылочная переменная, объявленная как final, никогда не может быть переназначена для ссылки на другой объект.

Final variables

The last variable can only be explicitly initialized once. A reference variable declared final can never be reassigned to refer to another object.