Java Methods

#javatheeasyway

Learning java made easy.

/chauhansumitdev

Must Know

Java is a language containing predefined **keywords**.

Keywords are those words which are reserved by the java language. This means that these cannot be used as an **identifier** for any variable name, object name or class name.

Identifier is a unique name given to any entity of a program (names of variables, classes, objects).

examples:

class **UsingIdentifier**
{
int **uniqueName** = 20;
}

.....

UsingIdentifier = new UsingIdentifier();

[Know more about classes and objects in the previous post]

List of java keywords: ref: geekforgeeks.org

abstract
assert
boolean
break
byte
case
catch
char
class
continue
default
do
double
else
enum
extends
final
finally
float
for
null
if
implements
import
instanceof
int
interface
long
native
new
package
private
protected
public
return
short
static
strictfp
super
switch
synchronized
this
throw
Throws
transient
try
void
volatile
while

What Are Methods?

Methods adds behavior to the objects. It is a block of code that makes an object perform some function when called.

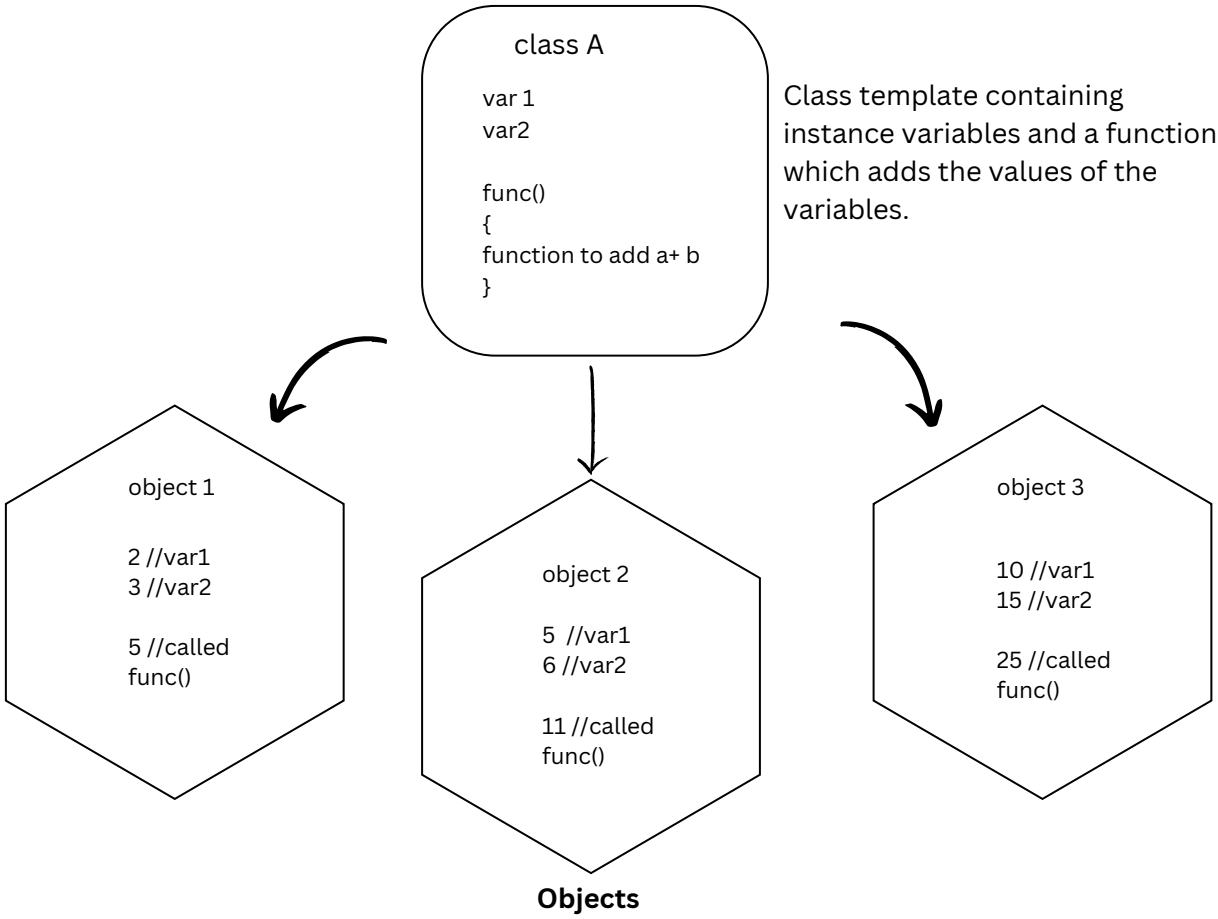
Consider the previous *Cylinder* class containing instance variables (radius and height), we have created many objects with different values for the instance variables. Now if we want to know the volume of each of the different *Cylinder* objects created we need to create a methods which outputs the volume of the object.

In other words methods are a block of code which gets executed when **called** thus adding functionality to the objects created.

Methods are also referred to as functions.

Calling a method is a process in which a specific method of the object is accessed using .(dot operator).

Example:



Categories Of Methods:

Predefined Methods

These methods are already defined in the java class.
Example: Java Math class contains a method sqrt(), which calculates the square root of the value.

Use Defined Methods

These are custom methods which are created for specific operations as required and are not already defined in the java library.
Example: There may be requirement of a function which calculates the final salary of an employ after all the deduction of taxes and addition of perks.

Syntax Of A Method

A function contains a group of keywords which are added as required, but the return type, name of the method along with a pair of parenthesis is compulsory. There are further two types of methods i.e. **Parameterized** and **Non-parameterized** methods.

Non-Parameterized Methods

These type of methods do not contain parameter(s) and does not accept arguments.

Parameterized Methods

These type of methods contain parameter(s) which accept **arguments** when called.

Arguments are the values passes along with method when it is being called.

access_specifiers define the visibility of the method to other parts of the program, classes and packages i.e. to which parts of the program the method is accessible. More on access specifiers is given below

Specifies whether the the method is **static** or **non static**. Static methods do not require their object of the class to which they belong to be created in order to access them.

return_type defines the type of value returned after the execution of the method

pair of **parenthesis** may or may not contain parameter(s) which accept argument(s) when called.

```
access_specifier | static/final | return_type | name_of_method()
{
    //statements to be exectued
}
```

statements are the lines of code which gets executed and performs a function.

name_of_method defines the unique name given to the method

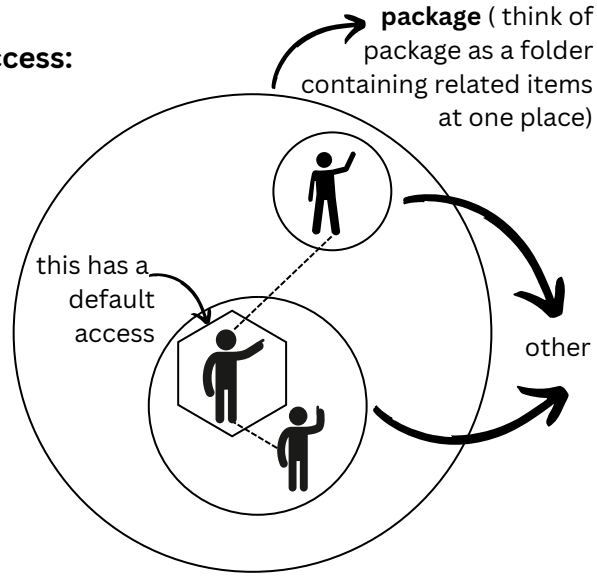
What are access specifiers?

Access specifiers are used to define the visibility of the method/variable to other parts of the same program or to other classes and packages. There are **4 types** of access specifiers in java.

default
public
private
protected

illustration:

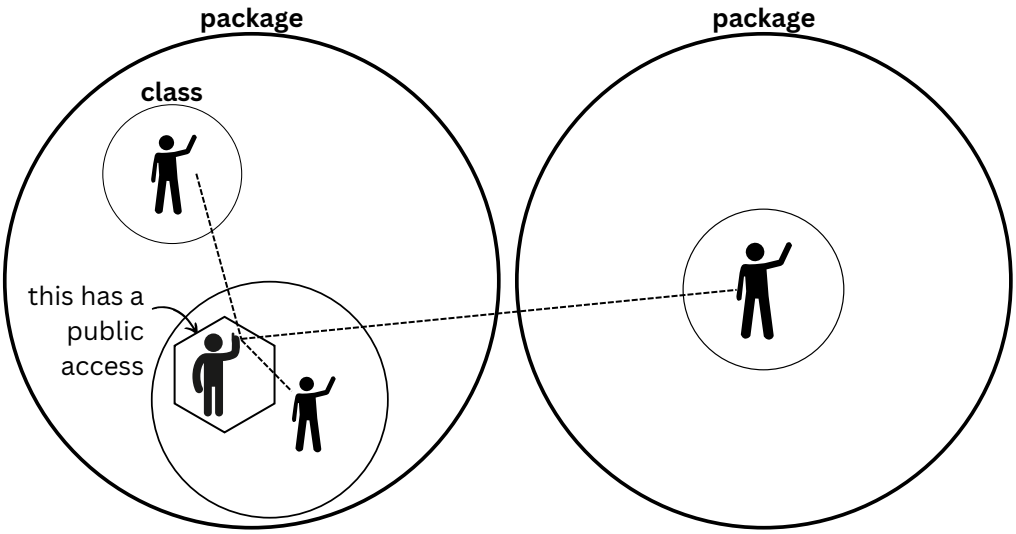
default access:



package (think of package as a folder containing related items at one place)

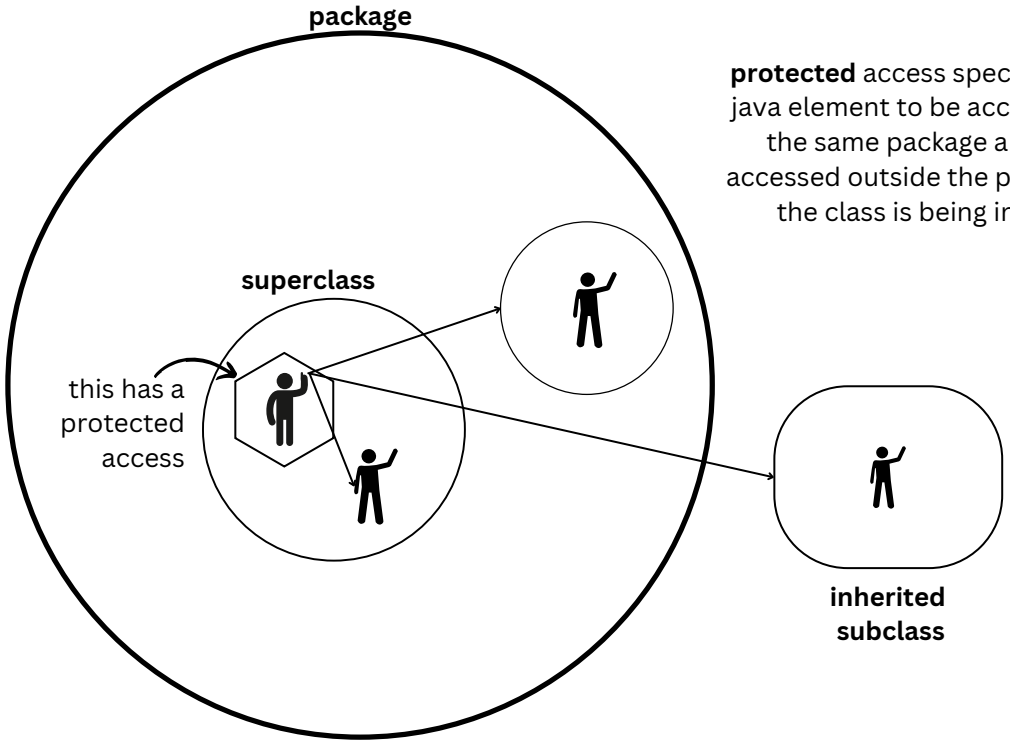
default access allows a java element to be accessed within the same class or within the same package. When no access specifier is specified, default access specifier is automatically assigned.

public access:



public access specifier allows a java element to be accessed from anywhere, either form the same program, class, package or outside classes and packages.

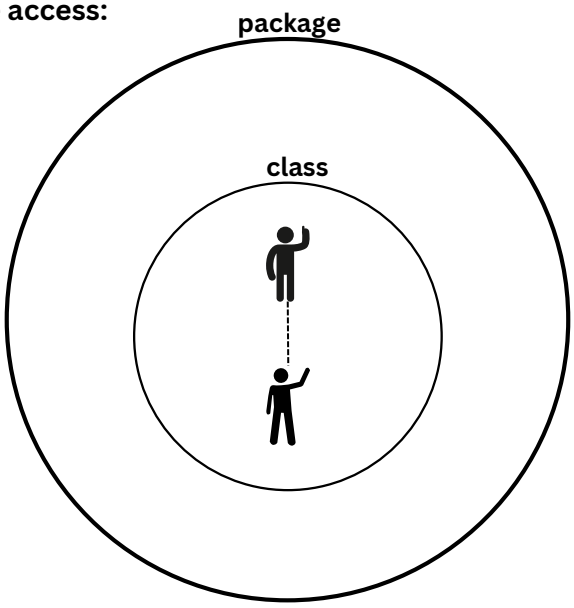
protected access:



protected access specifier allows a java element to be accessed within the same package and can be accessed outside the package when the class is being inherited

inheritance is the process by which a new class is created which inherits the features from its parent class called superclass.

private access:



private access specifier allows a java element to be accessed only within the same class and cannot be accessed by other classes and packages.

Practical Application Of Methods:

Let's consider our previous ongoing example of *Cylinder* class. Now we will be adding a method to calculate the volume of the cylinder created. We are also going to compile our knowledge and write a complete executable java program.

```
class Cylinder{

double radius;
double height;

void calcvolume()
{
double volume= 3.14*radius*radius*height;
System.out.println("The volume of cylinder is "+ volume);
}
}
```

```
class CylVolCalc
{
public static void main(String[] args)
{
Cylinder A=new Cylinder();
A.radius=10;
A.height=100;
A.calcvolume();
}
```

Process:
A very important point to note is that both the classes here are created in the same folder and class *CylVolCalc* is executed. All of the execution begins from the class containing the main method. As the main method gets executed an object of *Cylinder* class is created and values for radius and height are assigned using .(dot operator). A.calcvolume() calls the calcvolume() method created in object A, this in turn calculates the volume of the cylinder and stores in volume variable. The value stored is displayed on the screen using println() method.

Output:
The volume of cylinder is 31400.0

More objects with different outputs:

```
class CylVolCalc
{
public static void main(String[] args)
{
Cylinder B=new Cylinder();
B.radius=5;
B.height=50;
B.calcvolume();

Cylinder C=new Cylinder();
C.radius=3;
C.height=30;
C.calcvolume();

Cylinder D=new Cylinder();
D.radius=7;
D.height=70;
D.calcvolume();
}
```


Output:
The volume of cylinder is 3925.0
The volume of cylinder is 847.8
The volume of cylinder is 10770.2

/chauhansumitdev