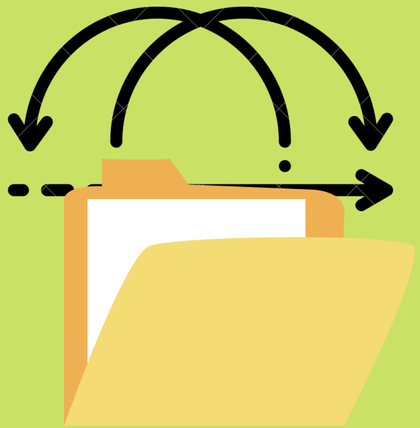
Class Modifiers

- Public
- Default
- Final
- Abstract
- Strict fp

Class Modifiers

public class

If a class declared as public then we can access that class from anywhere.
With in the package or outside the package.



within
same package

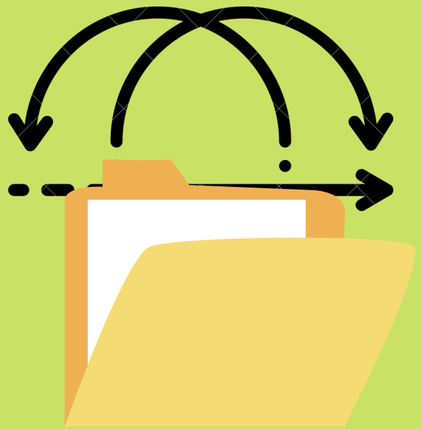
+

Class Modifiers

default class

If a class declared as the default then we can access that class only within the current package hence default access is also known as "package level access".

default class means giving no modifire to class



within
same package

Class Modifiers

final class

if we say that our class is final then we can not create its child class.

it will be final generation...no child !

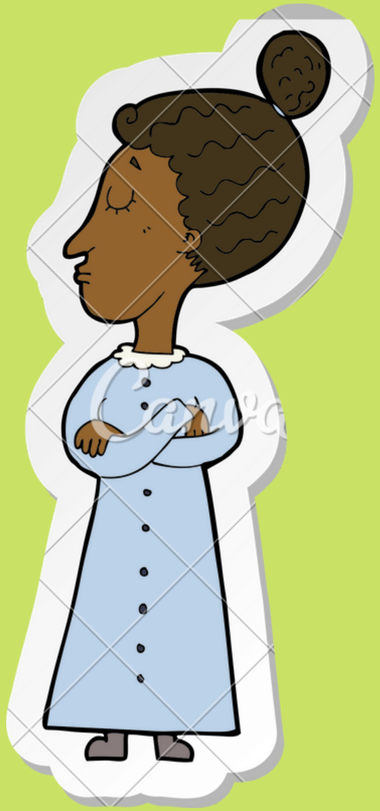
```
 2
 3 final public class Parent {
 4
 5     public static void property() {
 6         System.out.println("land, gold");
 7     }
 8
 9     public static void marry() {
10         System.out.println("mere dost ki beyti");
11     }
12 }
13
14 class Child extends Parent {
15
```

Remove 'final' modifier of 'Parent'

Class Modifiers

Abstract class

we can not create object of that class where class modifier is abstract



```
2
3  abstract public class Parent {
4
5      public static void property() {
6          System.out.println("land, gold");
7      }
8
9      public static void marry() {
10         System.out.println("mere dost ki beyti");
11     }
12 }
13
14 class Child {
15     Parent p = new Parent();
16         p.marry();
17 }
```

Cannot instantiate the type Parent

Class Modifiers

strictfp class

Usually the result of floating point of arithmetic is varing from platform to platform , to overcome this problem we should use strictfp modifier.

System.out.println(10/3);

p1	p2	IEEE754
3.333333333	3.333333	3.333

If a class declares as the Strictfp then every concrete method(which has body) of that class has to follow IEEE754 standard for floating point arithmetic, so we will get platform independent results.

Am i missing any class Modifier ?

ps- not talking about inner Class