

Class: most important language feature that make OOP possible

is a programmer-defined type

↳ a value of a class type is an object/instance of the class

Java Programming: define classes → create objects



Computations
by object-to-object
communication
(method calls)

特点: modularity
Information-hiding
Code re-use

class definition:

members of
the object

①

class Name

②

instance variables
(states)

③

methods
(behaviors)

public

No modifier

info-hiding
abstraction
encapsulation

Java
Access
Modifiers

'+' public

same class

No modifier

same class/package

'-' private

same class/package/subclass

'#' protected

all

↑
can be in
dif package

parameters vs arguments:

Coding

`public void add (a, b) { } parameters`

`add (a, b); arguments`

Create an object from a particular class: `new`
`className objName = new className (param);`

"this" : when refers to the instance variables of this class
(must use if there's a same name in local variable & instance variable)

Local variable: variables inside a method,
only effective/valid inside that method



`Java does not have global variable`

Constructor: a method designed to create a new object and initialize the instance variables of the new object

```
public className (params) {  
    codes  
}
```

Note: Constructors must have same name of the class name

② no return value

③ can be overloaded regarding different given params

④ if a class do not have an explicit constructor, Java will define an default one

instance variable declaration:

<modifier> <type> <fieldName> = value ;

private	int	age	=	3 ;
public	double			
protected	String			
null	float			
	short			
	long			
	boolean			
	char			

→ Get a specific variable value from a class:
className.variableName

Method declaration:

heading → <modifier> <return val type> <methodName> (param) {
 [<statement>] ← body
}

→ Method calls:

ClassName.method(param);

Method Overloading {
ad-hoc polymorphism { same class
same name
different param list
can have different return type

{ Accessor (getter : getName();)
Mutator (setter : setName('Judy');)

Static Method: can be used without a calling object.

is a independent object of that class
is independent of objects of that class
→ static method can not access instance variables

<modifier> **static** <return val type> <methodName> (param) {
[<statement>]
}

Static variables: one copy of static variable per class.

→ Note: cannot refer to 'this'
as it's not part of an object

Package: a group of classes

→ import java.util.*;

→ create: group all classes together into a folder/directory

① add 'package packageName' to the top of each file

→ Naming: packageName shall be all lower case