មជ្ឈមណ្ឌលកូរ៉ែ សហ្វវែរ អេច អ ឌី

Korea Software HRD Center

ការស្វែងយល់ពី Class និង Object ក្នុង Java

ណែនាំដោយ : Dr. Kim Tae Kyung

Webcash

KOICA
Korea International Cooperation Agency

๑. Class and Class Structure

๒. Constructor

๓. This & this()

๔. Object Creation

១. Class and Class Structure

- អ្វីទៅជា Class ?
  - Class ជាប្លង់គំរូនៃ Object ដែលបង្ហាញអំពី behavior(method) and state (field) ដែល Object នៃ Class នោះមាន។
  - ឧទាហរណ៍: Class ម៉ូតូ ជាប្លង់ដីសាញ្ញានៃម៉ូតូ ដែលត្រូវមានលក្ខណៈ (Field) និង សកម្មភាព(Method) មួយកំប្លែរ សំរាប់យកទៅប្រើដើម្បីផលិតម៉ូតូជាក់ស្តែង(ជាObject) រាប់ 1000 គ្រឿង។

១. Class and Class Structure(ត)

- អ្វីទៅជា Object ?
  - Object នៅក្នុង real-world គឺជារបស់របរ វត្ថុអ្វីផ្សេងៗ ដែលមាន State និង Behavior ។
    - State: ជាលក្ខណៈសំគាល់ ឬពត៌មានផ្សេងៗ របស់ Object ។
    - Behavior: សកម្មភាពដែល Object អាចមាន។

ឧទាហរណ៍: Object ម៉ូតូ មាន -State: ពណ៌ក្រហម, Brand Honda, តំលៃ 1000 ដុល្លារ ។

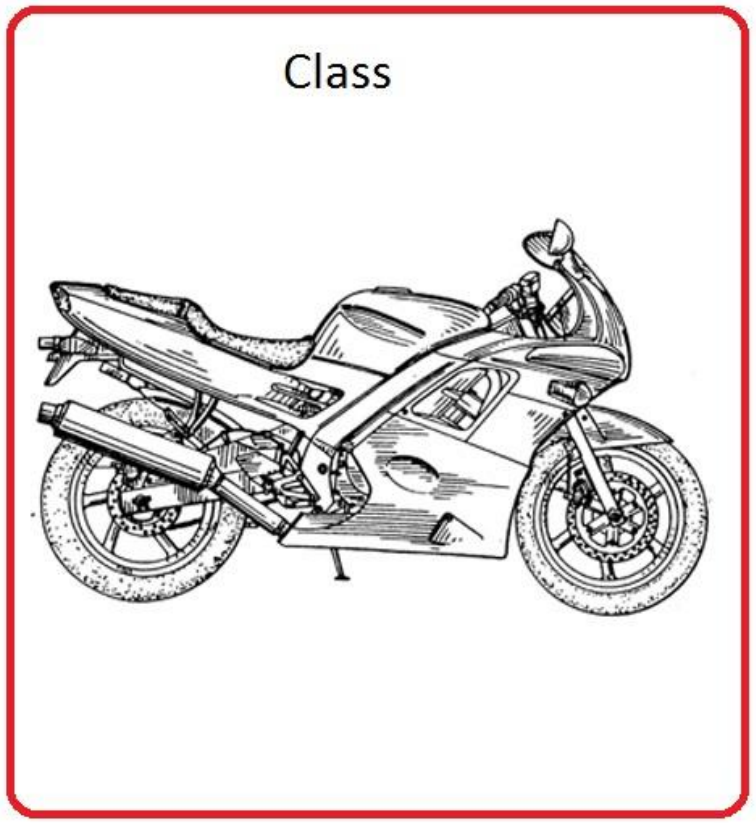
-Behavior: ជាន់ហ្វ្រាំង, មួលហ្គែរ, ចុចស៊ីប្លែរ, ។ល។

១. Class and Class Structure(ត)

- ឧទាហរណ៍៖ ខាងក្រោមនេះជារូបភាពបកស្រាយពីclass និង object

State:
Name
Wheels
Color
Speed
…..

Behavior:
Increase speed
Decrease speed
Brake
…..



១. Class and Class Structure(ត)

- Software-Object ស្រដៀងគ្នានឹង real-world Object
- គឺវាមាន **State** និង **Behavior** ដែរ :
  - Software Objects ផ្ទុក **State** ក្នុង **Fields** (ហៅថា **Variables** នៅក្នុងភាសា Programming ដទៃទៀត)។
  - Software Objects បង្ហាញសកម្មភាព **behavior** តាមរយៈ **Methods** (ហៅ **Functions** នៅក្នុងភាសា Programming ដទៃទៀត) ។

១. Class and Class Structure(ត)

- Syntax ក្នុងការបង្កើត class

```
[modifier] class Identifier
        {

        /..class members../


        }
```

១. Class and Class Structure(ត)

- Class Structure:
- ខាងក្រោមនេះ ជាការបង្កើត Class មួយ:

```
public class Dog{
        String breed;
        int age;
        String color;
        void barking(){
        }
        void hungry(){
        }
        void sleeping(){}
    }
}
```

Class head

Instance & Class variable / field

Method block

១. Class and Class Structure(ត)

- តើអ្វីជា class member?
- Class member គឺសំដៅទៅលើ variable & method ទាំងឡាយណាដែលនៅក្នុង class
- ឧទាហរណ៍៖

```
public class Car {
    int wheels, speed;// Variable
    public void increase_Speed(int speed) //Method
    {
        System.out.println("New speed is "+ speed);

    }


}
```

១. Class and Class Structure(ត)

- Class មានប្រភេទ Variable ដូចខាងក្រោម:

  1. Local variables: ជា Variables ដែលបង្កើតនៅក្នុង methods, constructors, ឬ blocks

     Variables នេះ នឹងត្រូវប្រកាស និងផ្តល់តំលៃនៅក្នុង method ហើយនឹង

     ត្រូវបញ្ចប់ នៅពេលដែល method បានធ្វើការចប់។

  2. Instance variables: ជា Variables ដែលបង្កើតនៅក្នុង Class តែនៅក្រៅ method។

     Variables នេះ ដំណើរការ Instantiate នៅពេលដែលគេ load class។

     Instance variable អាច access ពីក្នុង method, constructor, ឬ block

     ណាមួយរបស់ Class នោះបាន។

១. Class and Class Structure(ត)

3. **Class variables**: ជា Variables ដែលបង្កើតនៅក្នុង Class នៅក្រៅ method ជាមួយ Keyword Static ។

២. Constructor

- អ្វីទៅជា Constructor ?
  - Constructor ជាបណ្តុំនៃ statements ដែលប្រមូលផ្តុំគ្នា វាជាប្រភេទនៃ method ពិសេសមួយ ហើយវាដំណើរការឡើងនៅពេល មានការបង្កើត Object។
  - Constructor ប្រើដើម្បីបង្កើត Object និងផ្តល់តម្លៃដំបូងទៅឲ្យ Class Variable។
  - Constructor ត្រូវបានហៅដោយស្វ័យប្រវត្តិ នៅពេលគេប្រើ Keyword this() និង super
  - Class អាចមាន Constructor ច្រើនដែលត្រូវបានគេហៅថា Overload Constructor

២. Constructor(តរ)

Syntax:

```
ClassName ObjectName = new Constructor();

ClassName ObjectName = new Constructor(arg1, arg2,…);
```

ឧទាហរណ៍:

```
Student stu1 = new Student();

Student stu1 = new Student("001","Kakvey");

Student stu1 = new Student("001","Kakvey","Male","24");
```

២. Constructor (តទ)

➢ **ច្បាប់ក្នុងការប្រកាស Constructor** • ការប្រកាស constructor មានច្បាប់ ២ ៖ ▪ វាត្រូវតែមានឈ្មោះ ដូចទៅនឹង ឈ្មោះរបស់ Class ▪ វាមិនមាន Return Type ទេ	➢ **ប្រភេទនៃ constructors** • Constructor មាន ២ ប្រភេទ៖ ▪ Default constructor (no-arg constructor) ▪ Parameterized constructor

២. Constructor (តរ)

➢ **Java Default Constructor**

- Constructor ដែលគ្មាន parameter គឺត្រូវបានស្គាល់ជា default constructor

➢ **Syntax of default constructor:**

```
<modifier> <class_name>(){
    // body
}
```

២. Constructor (តរ)

ឧទាហរណ៍នៃ default constructor

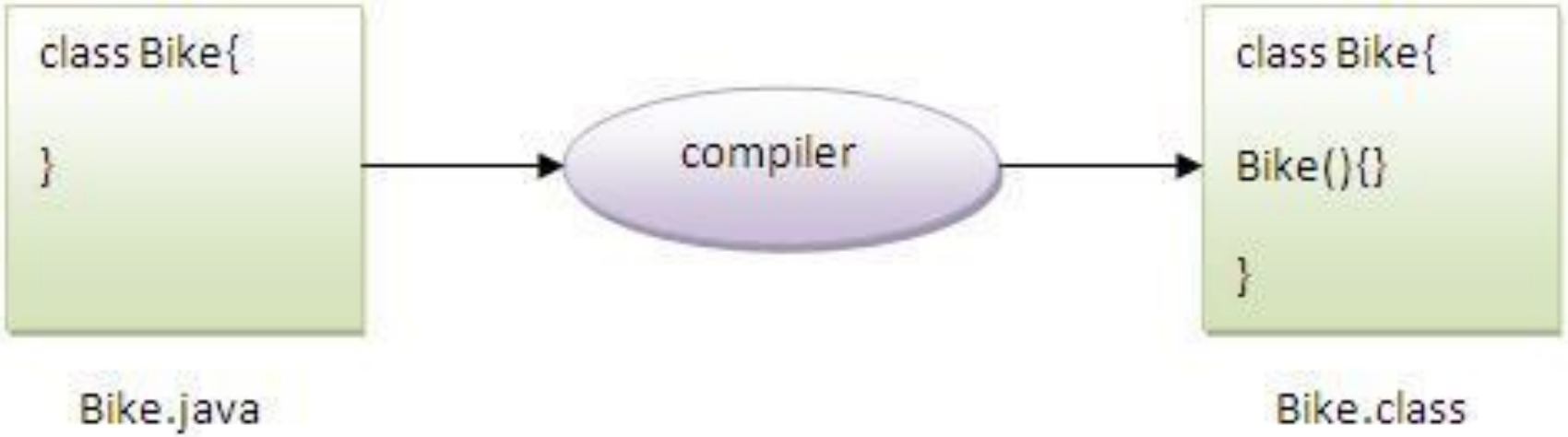
```
class Bike1 {
        public Bike1() {
        System.out.println("Bike is created");
        }

        public static void main(String[] args) {
        Bike1 b = new Bike1();
        }
}
```

Output: Bike is created

២. Constructor (តរ)

- ជាទូទៅ បើសិនជានៅក្នុង Class របស់យើងមិនមាន Constructor ទេ នោះ compiler នឹងបង្កើត default constructor មួយដោយស្វ័យប្រវត្តិ។



២. Constructor (ត)

តើគោលបំណងរបស់ការប្រើ default constructor ជាអ្វី?

- Default constructor គឺផ្តល់តម្លៃ default ទៅឲ្យ object ដូចជា 0, null etc និងផ្ទែកទៅតាមប្រភេទ។

ឧទាហរណ៍

```
class Student3{
    int id;
    String name;

void display(){
    System.out.println(id+" "+name);
    }
}
```

```
public static void main(String args[]){
    Student3 s1=new Student3();
    Student3 s2=new Student3();
    s1.display();
    s2.display();
}
```

Output: 0 null
0 null

២. Constructor (តស)

➢ Java parameterized constructor

- Constructor មួយដែលមាន parameters នៅក្នុង() គឺគេចាត់ទុកជា parameterized constructor។

- ហេតុអ្វីយើងប្រើ parameterized constructor?
  - Parameterized constructor គឺប្រើដើម្បីផ្តល់តម្លៃផ្សេងៗទៅកាន់ distinct objects។

២. Constructor (តៗ)

ឧទាហរណ៍

```
class Student4{
    int id;
    String name;

    Student4(int i,String n){
        id = i;
        name = n;
    }

    void display(){
         System.out.println(id+" "+name);
    }
}
```

```
public static void main(String args[]){
    Student4 s1 = new Student4(111,"Karan");
    Student4 s2 = new Student4(222,"Aryan");
    s1.display();
    s2.display();
}
```

Output: 111 Karan
222 Aryan

២. Constructor (ត)

➢ **Constructor Overloading in Java**

- Constructor overloading គឺជា technique មួយក្នុង Java ដែលអាចឲ្យ Class មួយអាចមាន constructors ច្រើន ហើយខុលគ្នាត្រង់ parameter lists។

๒. Constructor(តែ)

ឧទាហរណ៍

```
class Student5{
    int id;
    String name;
    int age;

    Student5(int i,String n){
          id = i;
          name = n;
    }
    Student5(int i,String n,int a){
          id = i;
          name = n;
          age=a;
    }
    void display(){System.out.println(id + " "
              + name +" " + age);}

    }
    public static void main(String args[]){
        Student5 s1 = new Student5(111,"Karan");
        Student5 s2 = new Student5(222,"Aryan",25);

        s1.display();
        s2.display();
       }
```

Output: 111 Karan 0
222 Aryan 25

២. Constructor (តរ)

ភាពខុសគ្នារវាង constructor និង method ក្នុង java

Java Constructor	Java Method
Constructor ត្រូវបានប្រើនៅពេលដែលចាប់ផ្តើមបង្កើត object	Method ត្រូវបានប្រើក្នុងការធ្វើសកម្មភាពរបស់ object.
Constructor មិនត្រូវមាន return type ទេ	Method អាចមាន return type
Constructor ត្រូវបានហៅជាលក្ខណៈ implicitly.	Method ត្រូវបានហៅជាលក្ខណៈ explicitly.
Java compiler ផ្តល់ default constructor មួយ បើយើងមិនមាន constructor ណាក្នុង class របស់យើង	Method មិនត្រូវបានផ្តល់ដោយ compiler ទេ
ឈ្មោះរបស់ Constructor ត្រូវតែដូចទៅនិងឈ្មោះ class	ឈ្មោះ Method អាចឬមិនអាចដូចទៅនិងឈ្មោះរបស់ class

៣. this & this( )

- អ្វីទៅជា this Keyword ?
  - This គឺជា Keyword មួយដែលត្រូវបានគេប្រើសម្រាប់នៅក្នុង Method ឬ Constructor នៃ Class។ This វាធ្វើការជា reference សម្រាប់ Current Object ដែល Method ឬ Constructor នៃ Object នោះត្រូវបានគេហៅមកប្រើប្រាស់។ ហើយវាអាចត្រូវបានគេប្រើប្រាស់ សម្រាប់ current object ផ្សេងៗទៀតដែលបិតនៅក្នុង Instance method ឬ constructor។
  - ក្នុងនោះវាចែកចេញជាពីរប្រភេទគឺ៖
    - ✓ This
    - ✓ This()

៣. this & this( )(ត)

- This គឺជា keyword ឬ reference variable ពិសេសមួយប្រើសម្រាប់សំដៅឲ្យ Current Object ឬ Instance Variable នៃ Class ផ្សេងៗទៀត។

- This() គឺជា Keyword មួយប្រើសម្រាប់ហៅ ឬ Access ទៅកាន់ Constructor មួយនៃ Class តែមួយហើយវាអាចត្រូវបានគេប្រើដើម្បីហៅ Overloaded constructor។

ចំណាំ៖ ហេតុផលសំខាន់ក្នុងការប្រើ This Keyword ពីព្រោះតែ Field មួយវាមានស្រមោល(ដូច) Method ឬ Constructor Parameter។

៣. this & this( )(ត)

➢ **តារាងប្រៀបធៀបរវាង This និងThis() Keyword**

THIS	THIS()
ត្រូវបានប្រើតែជាមួយ Object ឬ Instance variable នៃ Class តែប៉ុណ្ណោះ	ត្រូវបានប្រើតែជាមួយ Constructor ឬ Overloaded Constructor តែប៉ុណ្ណោះ
វាជា reference សម្រាប់ Current Object	ប្រើសម្រាប់ហៅ Constructor មួយពី Constructor ដទៃទៀតដែលឋិតនៅក្នុង Class តែមួយ
ដើម្បីញែកឲ្យដាច់រវាង Local and Instance variables នៅក្នុងការហៅ Method	

๓. this & this( )(๕)

ឧទាហរណ៍ ๑

```
public class Number {
    int num;

public void favorite(int num) {
    this.num = num;// observe this keyword here
}

public static void main(String[] args) {
    Number n1 = new Number();
    n1.favorite(8);
    System.out.println("Your favorite number is " + n1.num);
    }
}
```

៣. this & this( )(ត)

ឧទាហរណ៍ ២

```
public class Officer {

public static void main(String[] args) {
   Officer Re = new Officer(10000);
}

// default constructor 1
public Officer() {
    this("Group4"); // from 1 calling 2
}

// constructor 2 overloaded with string parameter
public Officer(String name) {
    System.out.println("Officer name is " + name);
}

// constructor 3 overloaded with int parameter
public Officer(int salary) {
    this();// from 3 calling 1
    System.out.println("Officer salary is $ " +
salary);
    }
}
```

៤. Object Creation

- យើងបង្កើត Object ចេញពី Class ។ Class មួយអាចបង្កើត Objects បានច្រើន ។ ក្នុងការបង្កើត objectមាន3ដំហានគឺ៖
  - **Declaration**: variable មួយប្រកាសជាមួយនឹងvariable name1ជាមួយobject type1។
  - **Instantiation**: ‘new’ keyword គឺប្រើប្រាស់សំរាប់បង្កើតobject។
  - **Initialization**: ‘new’ keyword គឺត្រូវបានធ្វើតាមការcallមួយទៅកាន់constructor.ហៅវាថាការ initializes new object។

๔. Object Creation(๓)

- ការបង្កើត Object មានដូចជា៖
  - ការ Declare Object

Syntax: <class name> <object name>;

Ex: Player ronaldo;

Class Name

Object name

៤. Object Creation(៥)

- ការបង្កើត Object ដោយប្រើ New keyword

```
<object name> = new <class name> ( [<parameter>] );
```

Ex: create ronaldo Object

```
ronaldo = new Player ();     // create objects

Player ronaldo = new Player();     // Declare & Create object
```

៤. Object Creation (ត)

- ការបង្កើត Object ហើយ initializes new object

```
ClassName ObjectName = new ClassName([Parameter]);
```

  - ClassName : គឺជាឈ្មោះរបស់ Class
  - ObjectName : គឺជាឈ្មោះរបស់ Object ដែលយើងបង្កើត
  - New : គឺជា Keyword នៃ Java Operator ប្រើសម្រាប់បង្កើត Object
  - ClassName([Parameter]) គឺជា Constructor នៃ Class ដែលមាន ឬ អត់ Parameter