



មជ្ឈមណ្ឌលកូរ៉េ សហវ័ន អេច អ ឌី

Korea Software HRD Center

កម្មវិធីបង្រៀន Java Programming ជាភាសាខ្មែរ

Online Java Training Course

Advisor: Dr. Kim Tae Kyung



www.kshrd.com.kh

មេរៀនទី៤: **Abstraction**

៤.១. អ្វីជា Abstraction?

៤.២. Abstract Class

៤.៣. Abstract Method

១.១ . អ្វីជា Abstraction?

- **Abstraction:** ជាវិធីសាស្ត្រសំរាប់បង្កើត **Abstract Class** អោយមានលក្ខណៈ **OOP** ។
- យើងអាចបង្កើតនូវលក្ខណៈ **Abstraction** បាន តាមរយៈការបង្កើត **Abstract Class** ឬ **Interface** ។
- លក្ខណៈ **Abstraction** អាចអោយយើងកំនត់នូវដំណើរការ (**Method**) មួយ គត់ របស់ **Object** នៃ **Class** ដែលបានទទួលកេរ្តិ៍មរតក (**Inherit**) ។

១.២. Abstract Class

- **Abstract Class:** គឺជា **Class** មួយដែលត្រូវបានប្រកាសជា **Abstract** ដែលវាអាចមាននូវ **Abstract Methods** ឬ គ្មានបាន ។
- **Abstract Class:** មិនអាចបង្កើត **Object** បានទេ (**Instantiate**) តែ អាចតាងជា (**Reference**) **Subclass** បាន ។
- **Subclass:** អាចទទួលកេរ្តិ៍មរតកពី **Abstract Class** ដូច **Class** ធម្មតាដែរ ។

១.២. Abstract Class

```
//យើ ងមាន abstract class មួយ  
public abstract class Shape {  
    public abstract double area();  
    public abstract double circumference();  
}
```

```
//យើ ងមាន Class Rectangle ដែល extends ពី Class Shape  
public class Rectangle extends Shape {  
    private double width;  
    private double height;  
    @Override  
    public double area() {  
        return width * height;  
    }  
    @Override  
    public double circumference() {  
        return (width + height) * 2 ;  
    }  
}
```

១.៣. Abstract Method

- **Abstract Method:** គឺជា **Method** មួយដែលត្រូវបានប្រកាសជា **Abstract** ដោយគ្មានធ្វើការ **Implementation** (គ្មាននូវឃ្លាបបិទ និង ឃ្លាបបើក, និង បញ្ចប់ដោយ សញ្ញា ";") ។

☞ `Abstract void methodAbstract(double a, double b);`

១.៣. Abstract Method

- ប្រសិនបើ **Class** មួយមាននូវ **Abstract Methods Class** នោះត្រូវតែប្រកាសជា **Abstract Class** ។
- ប្រសិនបើ **Subclass** ជា **Class** ធម្មតា, ជាទូទៅត្រូវធ្វើការ **Implementation** សម្រាប់ **Abstract Methods** ទាំងអស់របស់ **Parent Class** ។
- ផ្ទុយទៅវិញ បើវាមិនធ្វើការ **Implementation Abstract Methods** ទាំងអស់នោះទេ ដូច្នោះយើងត្រូវប្រកាស **Subclass** នោះជា **Abstract Class** ផងដែរ ។

១.៣. Abstract Method

// Subclass ធម្មតាត្រូវធ្វើការ Implementation Abstract Method

```
public class TestAbstract extends Shape{
    @Override
    public double cirumference() {
        return 0;           // ធ្វើ ការ Implementation
    }
}
```

// ចំពោះ Subclass ជា Abstraction យើងអាចមិនធ្វើការ Implementation ក៏បាន

```
public class TestAbstract extends Shape{

    /*public double cirumference() {}*/
}
```


សមាជិក

ក្រុមអ្នកស្រាវជ្រាវ



ល. តាំង បូរិក

borith_tang@yahoo.com



ល. សៀង វិជ្ជាវិទ្ធ

rith.magnificent.9@gmail.com



ល. ចាន់ សិរិរតនៈ

chanseryratanak@yahoo.com



ល. ដារ៉ា ពោញចិត្ត

darapenhchet@gmail.com



ល. ចាន់ បូរ៉ា

borachann@gmail.com

ក្រុមផលិតវីដេអូ



ល. ឈុន បញ្ញាភាន់

chhunpanharath@gmail.com



ល. ព្រាប វិទ្ធី

Itpreap.vuthy@gmail.com



ល. ហង្ស បូរី

hongboreyrupp@gmail.com



ល. ហួ ឈុនឡេង

huochhunleng@yahoo.com

មេរៀនបន្ទាប់ នឹងធ្វើការបង្ហាញពី Encapsulation