



# បណ្ណបណ្ណាលក្នុង សម្បត្តិ អេច អ ឌី

## Korea Software HRD Center

**Generic**

ណែនាំដោយ : Dr. Kim Tae Kyung



<http://www.kshrd.com.kh>

# បាតកា

១. អ្វីទៅជា Generic ?

២. ប្រភេទនៃ Generic

៣. Generic Wildcard

៤. ប្រភេទនៃ Parameter របស់ Generic

# ១. អ្វីទៅជា Generic ?

- Generic គឺជា Feature មួយរបស់ដែលអាចអោយយើងប្រើនូវ Type-Safe Object
- គុណសម្បត្តិរបស់ Generic៖
  - Type-Safety : អាចអនុញ្ញាតអោយយើងផ្ទុកតែ object ដែលមាន DataType ដូចគ្នានៅក្នុង Collection
  - មិនត្រូវការ Type casting

// មុនពេលមាន Generic

```
List list = new ArrayList();  
list.add("hello");  
String s = (String) list.get(0); //typecasting
```

// ក្រោយពេលមាន Generic

```
List<String> list = new ArrayList<>();  
list.add("hello");  
String s = list.get(0);
```

# ១. អ្វីទៅជា Generic ? (ត)

- ពិនិត្យមើលនៅពេល Compile Time: ប្រសិនបើយើងដាក់បញ្ចូលនៅ DataType ដែលខុសពី DataType ដែលបានបញ្ជាក់ទៅ Generic វានឹងធ្វើអោយមាន Compile Time Error
- ប្រើប្រាស់ឡើងវិញនូវ Class ឬ Method ដែលជា Generic នៅក្នុងករណីដែលមាន DataType ផ្សេងគ្នា

## ២. ប្រភេទនៃ Generic

---

- ប្រភេទនៃ Generic មានដូចជា៖
  - Generic Class
  - Generic Method

## ២. ប្រភេទនៃ Generic (ត)

### ➤ Generic Class

```
class MyTeam<T>{  
    T obj;  
    void set(T obj){  
        this.obj=obj;  
    }  
    T get(){  
        return obj;  
    }  
}
```



## ២. ប្រភេទនៃ Generic (ត)

### ➤ Generic Method

```
public class Util {  
    public static <E> void printArray(E[] elements) {  
        for ( E element : elements){  
            System.out.println(element );  
        }  
        System.out.println();  
    }  
}
```

## ៣. Generic Wildcard

- Generic Wildcard អាចអោយយើងអាចកំណត់នៅ Generic Type មានលក្ខណៈជា Hierachy
- Generic Wildcard មាន ៣ប្រភេទ៖
  - `<?>` : Any Type
  - `<? extends DataType>` : អាចទទួលបាននូវ DataType ផ្ទាល់ ឬក៏ Child Class របស់វា
  - `<? super DataType>` : អាចទទួលបាននូវ DataType ផ្ទាល់ ឬក៏ Parent Class របស់វា



## ៣. Generic Wildcard (ត)

ឧទាហរណ៍៖

// Class

```
public class A { }
```

```
public class B extends A { }
```

```
public class C extends A { }
```

```
public void processElements(List<?> elements){  
    for(Object o : elements){  
        System.out.println(o);  
    }  
}
```

## M. Generic Wildcard (ñ)

**// Read**

```
public void processElements(List<? extends A> elements){  
    for(A a : elements){  
        System.out.println(a.getValue());  
    }  
}
```

**// Write**

```
public static void insertElements(List<? super A> list){  
    list.add(new A());  
    list.add(new B());  
    list.add(new C());  
}
```

## ៤. ប្រភេទនៃ Parameter របស់ Generic

---

- T - Type
- E - Element
- K - Key
- V - Value
- N - Number