



**មជ្ឈមណ្ឌលកូរ៉េ សហ្វ្វែរ អេច អ ឌី**

**Korea Software HRD Center**

**កម្មវិធីបង្រៀន Java Programming ជាកាសាខ្មែរ**

**Online Java Training Course**

**Advisor: Dr. Kim Tae Kyung**

---



[www.kshrd.com.kh](http://www.kshrd.com.kh)

ជំពូកទី ៧

Exception

មេរៀនទី១៖

Exception



# មេរៀនទី១៖ Exception

១.១ របៀបដើម

១.២ អ្វីទៅជា Exception?

១.៣ Catching និង Handling Exception

១.៤ Try with Resources

១.៥ របៀប throw Exception

១.៦ Exception Hierarchy

១.៧ បង្កើត Exception ផ្ទាល់ខ្លួន

# ១.១ សេចក្តីផ្តើម

ពេលកុំព្យូទ័រតំណើរការ



# ១.១ សេចក្តីផ្តើម

ពេល program តំណើរការ

ពេល program តំណើរការ

ដែល**មិនមាន**Exception

Handling



Exception

កើតឡើង



ពេល program តំណើរការ

ដែល**មាន**Exception

Handling



Exception

កើតឡើង



## ១.២ អ្វីទៅជា Exception?

- **Exception** គឺជា **event** រឺ **problem** ដែលកើតឡើងនៅពេល **program** កំពុងតំណើរការ។
- **Exception** អាចកើតមានឡើងនៅពេល:
  - ☞ អ្នកប្រើប្រាស់បញ្ជូនទិន្នន័យមិនត្រឹមត្រូវ
  - ☞ **File** ដែលត្រូវបើកប្រើរកមិនឃើញ
  - ☞ **Network connection** ត្រូវបានដាច់នៅពេលពាក់កណ្តាល **communications**
- ពេល **exception** កើតឡើង **program** ត្រូវ បានបញ្ចប់។

## ១.២ អ្វីទៅជា Exception?

- យើងអាចប្រើ **if clause** ដើម្បីការពារនៅ **error**

ឧទាហរណ៍:

```
int[] a={1,2,3};  
for(int i=0;i<5;i++){  
    if(i>=a.length)  
        break;  
    else  
        System.out.println(a[i]);  
}
```

- តន្ត្រីយើងអាចប្រើវិធីថ្មី: **exception handling** នៅពេលដែល **error** កើតឡើង នៅក្នុង **method** វាបង្កើតនូវ **object** មួយគឺ **Exception object**

## ១.៣ Catching និង Handling Exception

- **Try Block** : **Code** ដែលអាចកើតមាន **error** យើងត្រូវសរសេរវាក្នុង **Try Block**

ឧទាហរណ៍:

```
try{  
    //code  
}  
//catch and finally blocks
```

- នៅពេលដែល **error** កើតឡើងក្នុង **try block** វានឹង **throw exception object**  
**Exception object** ត្រូវបានចាប់ដោយ **exception handler**



## ១.៣ Catching និង Handling Exception

- **Catch Blocks:** catch block និមួយៗគឺជា exception handler ហើយ handler ក៏ជា exception ហើយវាទៅតាម ប្រភេទ(Type) របស់វា។

ឧទាហរណ៍:

```
try{  
    }catch(ExceptionType1 arg){}  
    catch(ExceptionType2 arg){}
```

- ចំណាំ:

- ☞ Catch block ផ្ទុកនៅ code ដែលវាគំណើរការនៅពេល exception កើតឡើង
- ☞ មិនមាន code នៅចន្លោះ end try block និង start catch block
- ☞ ExceptionType ត្រូវតែជាឈ្មោះរបស់ class ដែល interit ពី Throwable Class
- ☞ នៅក្នុង catch block មួយៗ អាច handle ប្រភេទ exception មួយ រឺ ច្រើន
- ☞ ex គឺជា parameter ដែលជា final យើងមិនអាចផ្តល់តម្លៃអោយវាបានទេ

## ១.៣ Catching និង Handling Exception

- **Finally Block :** **Finally Block** តែងតែ តំណើរការ នៅពេល **try block** តំណើរការ ។ យើងប្រើវាសម្រាប់ធ្វើការសម្អាត ។ យើងអាចដោះស្រាយវាដោយប្រើ **Finally Block** ។

```
BufferedReader bf = null;

try{

bf=new BufferedReader(new FileReader("C:\\\\FileName.txt"));

}chatch(FileNotFoundException ex){//code}catch(IOException ex){

//code}finally{if(bf!=null) bf.close();

//close he file if it is opened

}
```

## ១.៤ Try with Resources

- **JDK 7** បាននែណាំ **version** ថ្មីរបស់ **try statement** អោយស្គាល់ **try-with-resources statement**

Syntax:

```
try(resource-specification)
{
    //user the resource
}catch()
{...}
```

## ១.៤ Try with Resources

- **Try-with-resource statement** គឺជា **statement** ដែលប្រកាស មួយ រឺ ច្រើន **resources**។
- គ្រប់ **object** ដែល **implements** ពី `java.lang.AutoCloseable` or `java.io.Closeable` អាចបោះ parameter អោយ try statement។
- **Resource** គឺជា **object** ដែលប្រើក្នុង កម្មវិធី និង ត្រូវតែបិទនៅពេលកម្មវិធីត្រូវបានបញ្ចប់ ។
- **Try-with-resource statement** ធានាថា **resource** នីមួយៗត្រូវបានបិទនៅពេលកម្មវិធីត្រូវបានបញ្ចប់ យើងមិនចាំបាច់បិទ **resource** ទេ។

## ១.៥ របៀប throw Exception

- **throw keyword** ត្រូវបានប្រើដើម្បី **throw exception** ជាក់ណាមួយ
- **throw statement** នាមនាវាមាន argument តែមួយគត់គឺ: **throwable object**. មានតែ **object** របស់ **Throwable class** រឺក៏ **subclasses** របស់វា ប៉ុណ្ណោះត្រូវបានចោះចេញ
- **Program** ឈប់ដំណើរការ នៅពេលវាជួបជាមួយ **throw statement**, ហើយ **catch statement** ដែលនៅជាប់វាធ្វើការនែកមើលប្រភេទ **exception** ដែលត្រូវ គ្នានោះ។

## ១.៥ របៀប throw Exception

ទំងន់ទូទៅ:

```
throw someThrowableObject;
```

ឧទាហរណ៍:

```
throw new NullPointerException();
```

## ១.៥ របៀប throw Exception

**Throws** ត្រូវបានប្រើនៅពេលដែលយើងមិនចង់អោយ **method** មួយបង្កើត **exception** ឡើង.

ទំរង់ទូទៅ:

```
type method_name(parameter_list) throws exception_list{  
    //Defintion of method  
}
```

## ១.៥ របៀប throw Exception

### Throws keyword

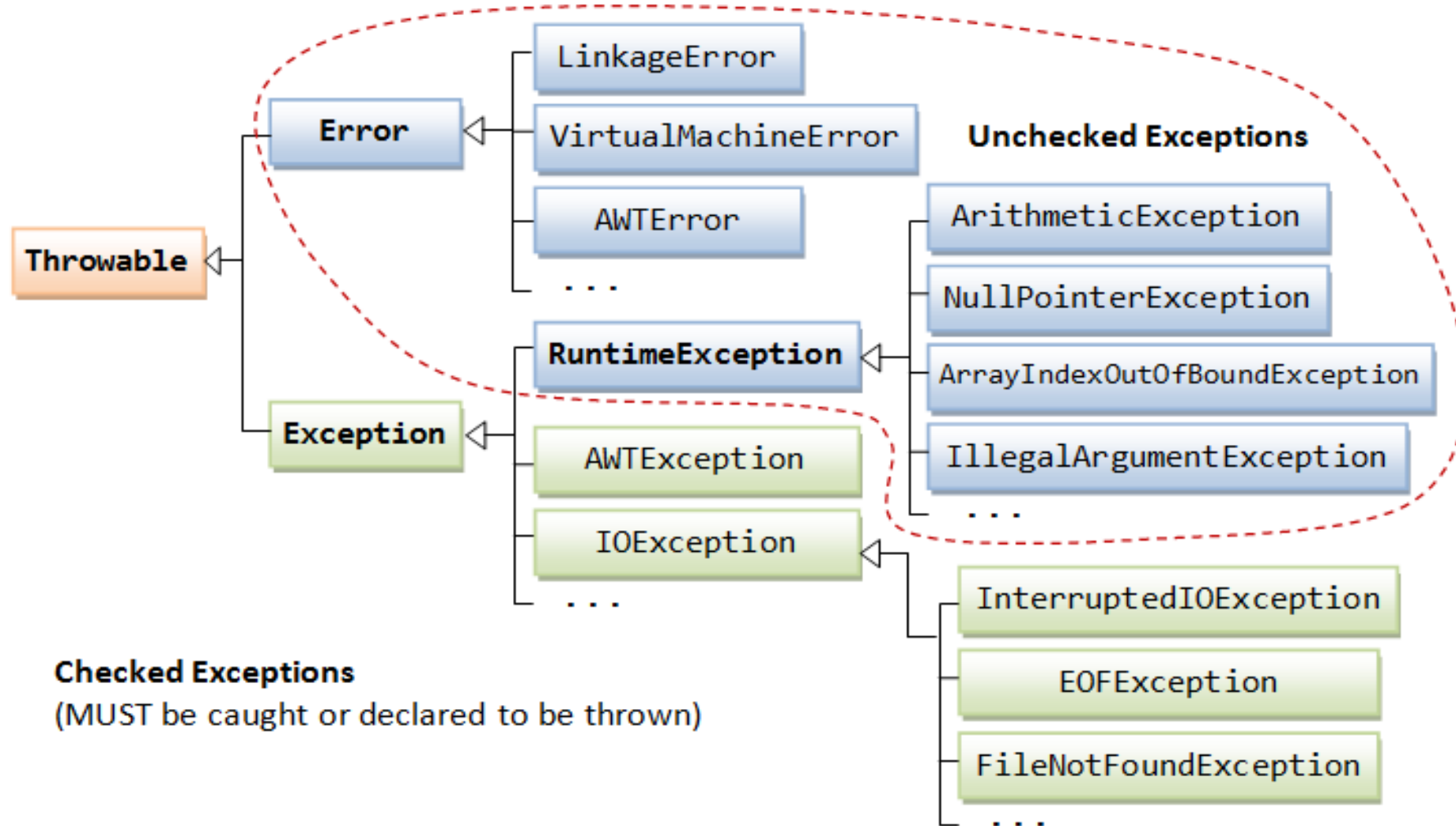
**Throws** ត្រូវបានប្រើនៅពេលដែលយើងមិនចង់អោយ **method** មួយ បង្កើត exception ឡើង.

ទំរង់ទូទៅ:

```
type method_name(parameter_list) throws exception_list{  
    //Defintion of method  
}
```



## ๑.๖ Exception Hierarchy



## ១.៦ Exception Hierarchy

- **Error Class:** គឺជា **error** ដែលកើតមានឡើងនៅដែលពេលមាន *dynamic Linking failure* ឬក៏ មាន hardware error ផ្សេងៗនៅក្នុង JVM(Java Virtual Machine).
- **Exception:** គឺបង្ហាញថាមានបញ្ហាបានកើតឡើង ប៉ុន្តែវាមិនធ្វើអោយមានបញ្ហាធ្ងន់ធ្ងរដល់ **system** នោះទេ.

## ១.៦ Exception Hierarchy

- **Exception** ត្រូវបានបែងចែកជាពីរគឺ **unchecked & checked**
- **Unchecked Exceptions** គឺមិនត្រូវការចាប់ ឬបញ្ជាក់ នៅពេល **program compile** ហើយវាជា **subclass** របស់ **RuntimeException**
- **Checked Exceptions** វាត្រូវតែចាប់ និងបញ្ជាក់ នៅពេល **compile time** ហើយវាជា **subclass** ផ្ទាល់របស់ **Exception**

## ១.៧ បង្កើត Exception ផ្ទាល់ខ្លួន

មានបីដំណាក់កាល (checked exception):

១. ត្រូវបង្កើត exception class: *Syntax*:

```
class UserDefinedEx extends Exception{  
    //statements  
}
```

២. Raise Error:*syntax*:

```
DataType methodName() throws UserDefinedEx{  
    //satements  
    throw UserefinedEx_Instance  
}
```

## ១.៧ បង្កើត Exception ផ្ទាល់ខ្លួន

### ៣. Catch Error:

```
try{  
    method that throws exception  
}catch(UserDefinedEx arg){  
    //statements  
}
```

# សមាជិក

## ក្រុមអ្នកស្រាវជ្រាវ



ល. គ្រី សុផាន់ណាត់  
krysophanatt@gmail.com



ល. ង៉ៅ ហ្គេចឡេង  
ngorgechleng@gmail.com



ល. ប្រាក់ ដារ  
prakda99@yahoo.com



ក. សុង ដារតនា  
songdarathana@gmail.com



ល. អេ កុសល  
longkosal7@gmail.com

## ក្រុមផលិតវីដេអូ



ល. ឈុន បញ្ហាភត្ត  
chhunpanharath@gmail.com



ល. ហង្ស បូរី  
houngboreyrupp@gmail.com



ល. ព្រាប វិទ្ធី  
Itpreap.vuthy@gmail.com



ល. ហួ ឈុនឡេង  
huochhunleng@yahoo.com

មេរៀនបន្ទាប់ និងធ្វើការបង្ហាញពី Thread