



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

ជំពូកទី ១០ Thread

មេរៀនទី១៖

Thread



មេរៀនទី១: Thread

១.១ អ្វីជា Thread ?

១.២ ហេតុអ្វីយើងត្រូវប្រើ Thread ?

១.៣ ការបង្កើត Thread

១.៤ Thread Synchronization

១.១ អ្វីជា Thread ?

- អ្នកប្រើប្រាស់កុំព្យូទ័រ អាចប្រើប្រាស់នូវកុំព្យូទ័រដែលអាចធ្វើការងារបានច្រើនក្នុងពេលតែមួយ ។ ជាទូទៅ ដូចជាអ្នកអាចធ្វើការបន្តនៅក្នុងកម្មវិធី **Microsoft Word**, ហើយយើងក៏អាចធ្វើការស្តាប់ចម្រៀង អាចធ្វើការផ្សេងៗជាច្រើន ទៀតផងដែរ ។ ជាងនេះផងដែរ កម្មវិធី **Microsoft Word** មួយក៏អាចធ្វើការងារច្រើននៅក្នុងពេលតែមួយ ។
- មុខងារដែលប្រតិបត្តិដោយឡែកខុសៗគ្នាទាំងអស់នេះហើយ គឺជា **Thread**។
- លក្ខណៈពិសេសមួយ ក្នុងចំណោមលក្ខណៈពិសេសទាំងអស់របស់ **Java** គឺវាអាចធ្វើការប្រើប្រាស់នូវ **Multi-Threading** គឺជាដំណើរការដែលធ្វើការ ប្រតិបត្តិច្រើននៅពេលជាមួយគ្នា។ នៅក្នុងមេរៀននេះ យើងនឹងបង្ហាញនូវ **Concept** នៃការប្រើប្រាស់ **Threads** និង របៀបបង្កើតនូវ **multi-Threading programs** នៅក្នុង **Java**.

១.២ ហេតុអ្វីយើងត្រូវប្រើ Thread ?

- ❑ **Multi-Threading** អាចធ្វើអោយកម្មវិធីរបស់យើងធ្វើសកម្មភាពកាន់តែច្រើនជាងមុន ព្រមទាំងមានល្បឿនលឿនផងដែរ។
- ❑ ឧទាហរណ៍ **Microsoft Office Word** វាអាចអោយយើងអាចធ្វើការរក្សាទុកនូវឯកសាររបស់យើងនៅពេលដែលយើងកំពុងតែព្យាយាមឯកសាររបស់យើង ។
- ❑ ភាគច្រើនគេប្រើ **Multi-Threading** នៅក្នុង **Games, Animation** ជាដើម។
- ❑ នៅក្នុងករណីនេះ **Multi-threaded Programs** ដំណើរការលឿនជាង **Single-threaded programs**។ ក្នុងនេះផងដែរ **Java** អាចផ្តល់អោយយើងធ្វើការបង្កើត និង ដំណើរការ **Thread** សម្រាប់ធ្វើការ **locking resources** ដើម្បីធ្វើការការពារនូវការប៉ះទង្គិចគ្នារវាង ទិន្នន័យ។

១.៣ ការបង្កើត Thread

- នៅក្នុងភាសា **Java** យើងអាចបង្កើត **Thread** បាន២របៀបគឺ
 - ☞ ដោយធ្វើការ **extends** ចេញពី **class Thread**
 - ☞ ដោយធ្វើការ **implement** ចេញពី **interface Runnable**
- **Thread class** មានផ្ទុកនូវ **Constructors** សម្រាប់ធ្វើការបង្កើតនូវ **Thread** សម្រាប់ធ្វើ ការងារ និង មាននូវ **methods** សម្រាប់ធ្វើការគ្រប់គ្រងទៅលើ **Threads**។
- **Thread class** មាននៅក្នុង **package java.lang**។

១.៣ ការបង្កើត Thread

- នេះគឺជា **methods** និងការប្រើប្រាស់របស់ Thread

Methods	ការប្រើប្រាស់
+Thread()	បង្កើតនូវ Thread ទទេមួយ
+Thread(task : Runnable)	បង្កើតនូវ Thread សម្រាប់ដោយផ្ដើមចេញពី Class ដែល Implements Runnable Interface
+start(): void	ចាប់ផ្ដើមដំណើរការនូវ Thread ដោយការហៅ method run
+isAlive(): Boolean	ពិនិត្យមើលថាតើ Thread នៅកំពុងដំណើរការឬក៏អត់
+setPriority(p: int): void	កំណត់នូវអាទិភាពរបស់ Thread (ពី ០ ទៅដល់ ១០)
+join(): void	រង់ចាំរហូតដល់ Thread បញ្ចប់
+sleep(millis: long): void	ដាក់អោយ Thread ដេកក្នុងរយៈពេលជានិច្ចជាក់លាក់ណាមួយ
+yield(): void	ផ្អាកនូវដំណើរការរបស់ Thread ណាមួយដោយអនុញ្ញាតអោយ Thread ដទៃទៀតដំណើរការ
+interrupt(): void	បញ្ឈប់នូវ Thread

១.៣ ការបង្កើត Thread

ឧទាហរណ៍

```
Java.Lang.Thread extends MyThread
// Custom thread class
public class MyThread extends Thread{
public MyThread(...){
}
// Override the run method in Runnable
public void run(){
// Tell system how to perform this
task
}
}
```

```
//main class
public class Main{
    public static void main(String
args[]){
    // Create a thread
    MyThread thread1 = new MyThread(...);
    // Start a thread
    thread1.start();
    // Create another thread
    MyThread thread2 = new MyThread(...);
    // Start a thread
    thread2.start();
    } }
}
```


១.៣ ការបង្កើត Thread

- **Runnable** គឺជា **interface** មួយ ។ យើងគ្រាន់តែធ្វើការ **implement** ដើម្បីបង្កើត Thread ជាការស្រេច។
- **Runnable** មាននៅក្នុង package `java.lang` ។

```
java.lang.Runnable implements TaskClass
// Custom task class
public class TaskClass implements Runnable{
public TaskClass(...){
}
// Implements the run method in Runnable
Public void run(){
// Tell system how to run custom thread
}
}
```

```
// Main class
public class Main{
public static void main(String args[]){
// Create an instance of TaskClass
TaskClass task = new TaskClass(...);
// Create a thread
Thread thread = new Thread(task);
// Start a thread
thread.start();}}
```

១.៣ Catching និង Handling Exception

- ឧទាហរណ៍ ៖ ប្រសិនបើយើងធ្វើការបង្កើត និង ដំណើរការនូវ ១០០ **Threads** ហើយ **Thread** នីមួយៗធ្វើការដាក់នូវ ចំនួនទឹកប្រាក់មួយចំនួនចូលទៅក្នុង **Account** មួយ។

```
public synchronized void deposit(double amount)
```

or

```
public void deposit(double amount){
```

```
    synchronized(this){        ...
```

```
    }
```

```
}
```

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 Files and I/O