មជ្ឈមណ្ឌលកូរ៉េ សហវែរ អេច អ ឌី

Korea Software HRD Center

កម្មវិធីបង្រៀន Java Programming ជាភាសាខ្មែរ

Online Java Training Course

Advisor: Dr. Kim Tae Kyung

Webcash

KOICA
Korea International Cooperation Agency

www.kshrd.com.kh

មេរៀនទី២៖ Regular Expressions

១.១ សេចក្តីផ្តើម

១.២ ហេតុអ្វីបានជាប្រើ Regex

១.៣ ប្រភេទរបស់ Regex

១.៤ ការប្រើប្រាស់ Backslashes

១.៥ របៀបប្រើប្រាស់ Regex

១.១ សេចក្តីផ្តើម

- **Regular Expression** គឺជាលំដាប់នៃតួអក្សរពិសេស (special sequence of characters)ដែលជួយអ្នកក្នុងការ៖
  - ☞ ស្វែងរកតួអក្សរ ឬសំណុំនៃតួអក្សរ ។
  - ☞ ផ្លាស់ប្តូរតួអក្សរដែលរកឃើញនោះទៅជាតួអក្សរមួយផ្សេងទៀតនៅក្នុងប្រយោគតែមួយ ។
- ពាក្យកាត់របស់ Regular Expression គឺជា regex ។
- Java បានផ្តល់នូវ package មួយគឺ java.util.regex សំរាប់ជាគំរូ (Pattern) ផ្គូរផ្គងជាមួយ Regex ។

១.១ សេចក្តីផ្តើម

- ក្នុង **java.util.regex package** មាន៣ Class សំខាន់ៗ៖

Pattern Class

Pattern object គឺជាតំណាងឲ្យតំនាក់កាល compiled នៃ Regular Expression . The Pattern class គឺមិនបានផ្តល់នូវ public constructors ទេ ។ ដើម្បីបង្កើតនូវ Pattern object បាន , ដំបូងអ្នកត្រូវធ្វើការហៅនូវ public static compile methods មួយដោយធ្វើការបោះ Regular Expression ជា Argument ឲ្យ។

Matcher Class

Matcher Object គឺជាម៉ាស៊ីនដែលធ្វើការបកប្រែគំរូ(Pattern) ដែល pattern object បានបង្កើតឡើង ជាមួយនឹងអក្សរ ឬលេខដែលបានបញ្ចូលជា Argument នៅក្នុង Matcher object ។ Matcher Class ដូចទៅនឹង Pattern class ដែរ គឺមិនមាន public constructors (មើលមេរៀន constructor)ទេ។

PatternSyntaxException

PatternSyntaxException object គឺប្រើសំរាប់ត្រួតពិនិត្យមើលថាតើ Pattern ដែលយើងសរសេរត្រូវលក្ខណៈដែរឬទេ ប្រសិនបើខុសវានឹងបង្ហាញ error ពេលយើងធ្វើការ compile ។

១.២ ហេតុអ្វីបានជាប្រើ Regex

- ស្ទើរតែគ្រប់ភាសា Programming ទាំងអស់ មានម៉ាស៊ីនដាច់ដោយឡែកមួយដែលសំរាប់ប្រើដើម្បីប្រតិបត្តិ (execute) Regex ។
- **Regex** ធ្វើប្រតិបត្តិ (execute) លឿនជាង index() Methods និង substring() Methods។
- ធ្វើឲ្យកូដមានរបៀបរៀបរយ កូដសរសេរតិច និងខ្លី ។

១.៣ ប្រភេទរបស់ Regex

- វិធីប្រើប្រាស់ Regex ត្រូវបានយកមកប្រើប្រាស់ស្ទើរតែគ្មានដែនកំណត់។ ភាសាមូលដ្ឋានសំរាប់ប្រើក្នុង Regex បានបែងចែកជា ៣ ថ្នាក់(Class)គឺ៖

Character Class

Quantifiers

Meta-characters

១.៣ ប្រភេទរបស់ Regex

- **Character Class:**
- ត្រូវបានប្រើសំរាប់កំនត់នូវអត្ថបទ(text) ថាជាលក្ខណៈអក្សរ ឬលេខ ឬមិនមែនអក្សរ មិនមែនលេខនៃគំរូ (pattern)

Regular Expression	និយមន័យ
.	សំរាប់លេខ, តួអក្សរ,និមិត្តសញ្ញាផ្សេងៗ ហើយមួយតួ
\d	សំរាប់គ្រប់លេខពី ០ ដល់ ៩
\D	សំរាប់មិនមែនជាលេខ
\s	សំរាប់អក្សរដែលភ្ជាប់ជាមួយការដកឃ្លា
\S	សំរាប់អក្សរដែលមិនភ្ជាប់ជាមួយការដកឃ្លា
\w	សំរាប់តួអក្សរពី a ដល់ z ពី A ដល់ Z និង ពី ០ ដល់ ៩
\W	មិនសំរាប់ពាក្យតួរអក្សរ : [^\w]

១.៣ ប្រភេទរបស់ Regex

- **Quantifiers:**
- Quantifiers អាចប្រើដើម្បីបញ្ជាក់ពីចំនួនឬប្រវែងដែលជាផ្នែកមួយនៃ Pattern គួរតែផ្គូផ្គងឬធ្វើឡើងវិញ។ Pattern និងចងទៅជាក្រុមកន្សោមខាងធ្វេងបន្ទាប់ពីខ្លួនរបស់វា។

ប្រភេទ	និយមន័យ
*	ស្វែងរកសូន្យដង ឬ ច្រើនដង
+	ស្វែងរក ១ដង ឬ ច្រើនដង
?	ស្វែងរក ១ដង ឬ ច្រើនដង
{n}	ស្វែងរកចំនួន n ដង
{n,}	ស្វែងរកយ៉ាងហោចណាស់ n ដង
{n,m}	ស្វែងរកយ៉ាងហោចណាស់ n ដង ប៉ុន្តែមិនច្រើនជាង m ដង

១.៣ ប្រភេទរបស់ Regex

- **Meta-characters:**
- Meta-character គឺត្រូវបានប្រើដើម្បីចងជាក្រុម ចែកជាក្រុម និងអនុវត្តប្រតិបត្តិការពិសេសក្នុងគំរូ(Pattern)។

Regular Expression	និយមន័យ
\	ចាកចេញទៅកាន់ meta-character បន្ទាប់
^	ស្វែងរកពីចំនុចចាប់ផ្តើមនៃបន្ទាត់
.	ស្វែងរកគ្រប់ប្រភេទតួអក្សរលើកលែងចូលទៅបន្ទាត់ថ្មី
$	ស្វែងរកចុងក្រោយនៃបន្ទាត់ (ឬមុននឹងចាប់ផ្តើមបន្ទាត់ថ្មី)
|	ជ្រើសរើសណាមួយក៏ត្រូវ (ស្មើរនឹង or)
()	សំរាប់ដាក់ជាក្រុម
[]	ផ្គូផ្គងគ្រប់តួអក្សរ ឬលេខ ឬសញ្ញាផ្សេងៗ ដែលមាននៅក្នុង “[]”

១.៤ ការប្រើប្រាស់ Backslashes

- នៅក្នុង Java ការប្រើ Regex ត្រូវដាក់ថែម "\" ចំពោះ Regex ណាដែលមាន "\" ហើយស្រាប់ ព្រោះនៅក្នុង Java មាន "\" ស្រាប់ ដូចច្នេះ Java អាចនឹងច្រឡំ នៅពេលប្រើប្រាស់ Regex

ឧទាហរណ៍៖

```
\w      ->      \\w
\s      ->      \\s
\d      ->      \\d
\       ->      \\\\
```

១.៥ របៀបប្រើប្រាស់ Regex

- ការប្រើប្រាស់ Regex មាន ២របៀប៖
  - ☞ របៀបទី១៖ ការប្រើប្រាស់ Methods ជាមួយនឹង String Class
  - ☞ របៀបទី២៖ ការប្រើប្រាស់ Class ក្នុង `java.util.regex package`

១.៥ របៀបប្រើប្រាស់ Regex

- **String** នៅក្នុង Java មាន Methods ស្រាប់ ដែលសំរាប់ប្រើប្រាស់ជាមួយនឹង Regex គឺ៖

Method	ណែនាំ
s.matches("regex")	បោះតំលៃជា True និង False នៅពេលដែលអក្សរដែលបានផ្គូផ្គងក្នុងអញ្ជាត s ទាំងមូលអាចផ្គូផ្គងត្រូវ
s.split("regex")	បង្កើត Array នៃអញ្ជាត s រួចកាត់ពាក្យ ឬ regex ដែលបោះឲ្យនោះចោល។ ពាក្យ ឬ regex ដែលកាត់នោះមិនត្រូវបានបង្ហាញនៅក្នុងលទ្ធផលទេ
s.replaceFirst("regex", "replacement")	ផ្លាស់ប្តូរនូវអក្សរដំបូងដែលបានរកឃើញមុនគេ
s.replaceAll("regex", "replacement")	ផ្លាស់ប្តូរនូវអក្សរទាំងអស់ដែលបានរកឃើញ

ចំណាំ:

`String.replace()` មិនស្គាល់ Regular Expression ឡើយ

១.៥ របៀបប្រើប្រាស់ Regex

- ក្រូវ ៖

១

- `Import java.util.regex.Pattern`
- `Import java.util.regex.Matcher`
- `Import java.util.regex.*`

២

- ការបង្កើត Pattern object ជាមួយនឹង Pattern Class។ Pattern object អនុញ្ញាតឲ្យបង្កើតនូវ Matcher object

៣

- ការបង្កើត Matcher object ជាមួយនឹង Matcher Class

៤

ក្នុង Matcher Class មាន២method សំរាប់ធ្វើការផ្គូផ្គង៖

- Matcher.matches() វាស្រដៀងទៅនឹង Method s.matches()(slice ខាងលើ) ដែលសំរាប់បញ្ជូលនូវអក្សរ ឬលេខ ឬសញ្ញាដែលចង់ផ្គូផ្គង
- Matcher.find() ស្វែងរកអក្សរ ឬលេខដែលមាននៅក្នុង Matcher.matches()

សមាជិក

ក្រុមអ្នកស្រាវជ្រាវ

ល. ក្រី សុផាន់ណាត់
krysophanatt@gmail.com

ល. ង៉ោ ហ្គេចឡេង
ngorgechleng@gmail.com

ក. សុង ដារតនា
songdarathana@gmail.com

ល. ប្រាក់ ដារ
prakda99@yahoo.com

ល. អេ កុសល
longkosal7@gmail.com

ក្រុមផលិតវីដេអូ

ល. ឈុន បញ្ញារតន៍
chhunpanharath@gmail.com

ល. ព្រាប វិទ្ធី
ltpreap.vuthy@gmail.com

ល. ហង្ស បូរី
houngboreyrupp@gmail.com

ល. ហួ ឈុនឡេង
huochhunleng@yahoo.com

មេរៀនបន្ទាប់ នឹងធ្វើការបង្ហាញពី Calender