



**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)

## មេរៀនទី២៖ **Map Collection**

២.១ ស៊េចក្រឹត

២.២ Method ខ្លះៗនៃ Map Interface

២.៣ Map.Entry

២.៤ Hashtable

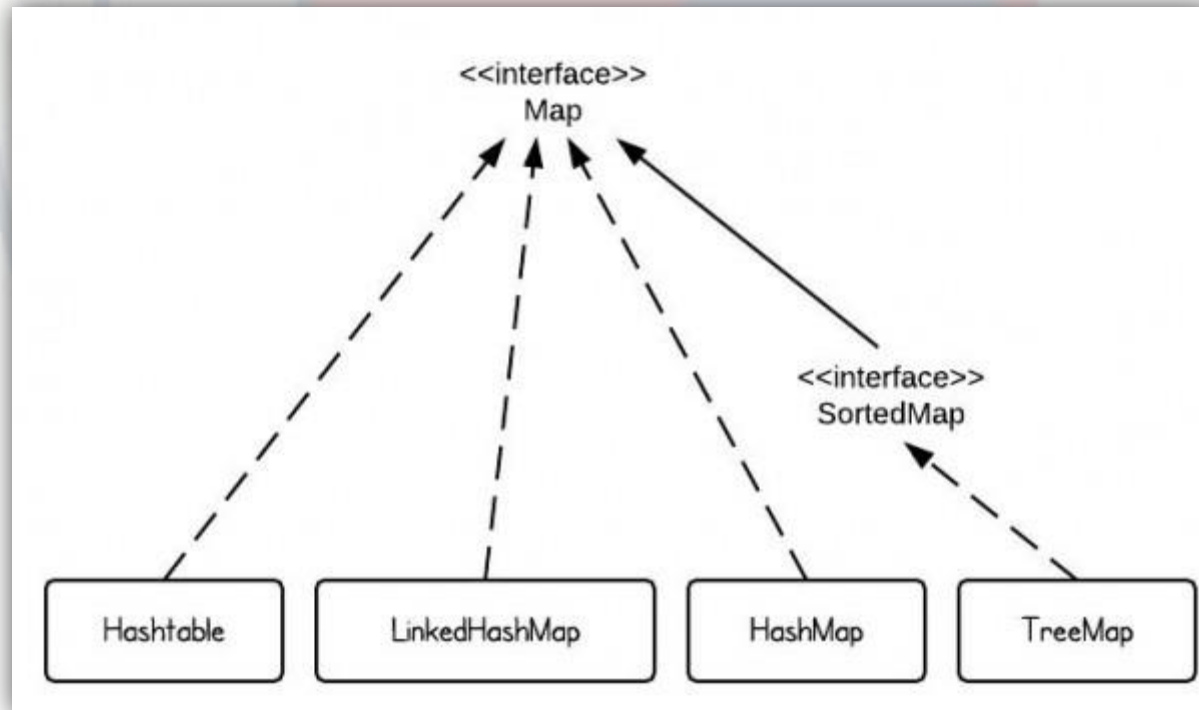
២.៥ LinkedHashMap

២.៦ HashMap

២.៧ TreeMap

## ២.១ សេចក្តីផ្តើម

- **Map** គឺជា **Object** ដែលភ្ជាប់ (**Maps**) **Keys** ទៅកាន់ **Values** របស់វា។
- **Map** គឺមិនអាចមាន **Key** ស្អន់គ្នានោះឡើយ ហើយរាល់ **Key** នីមួយៗ អាចភ្ជាប់ទៅកាន់ **Value** បានយ៉ាងច្រើនមួយ។



## ២.២ Method ខ្លះៗនៃ Map Interface

Method	Description
<code>void clear()</code>	លុបចោលនូវ ធាតុទាំងអស់ ដែលមាននៅក្នុង <b>Map</b> នោះ
<code>boolean containsKey(Object key)</code>	បោះតម្លៃ <b>true</b> ប្រសិនបើមាន <b>Key</b> នៅក្នុង <b>Map</b> នោះ
<code>boolean containsValue(Object value)</code>	បោះតម្លៃ <b>true</b> ប្រសិនបើមាន <b>Value</b> នៅក្នុង <b>Map</b> នោះ
<code>Set entrySet()</code>	បោះសំណុំនៃ <b>key</b> និង <b>value</b> ដែលមាននៅក្នុង <b>map</b> នោះ
<code>Object get(Object key)</code>	បោះ <b>value</b> នៃធាតុណាមួយដែលមាន <b>key</b> នោះ ហើយបោះតម្លៃ <b>null</b> ប្រសិនបើ <b>key</b> នោះមិនឃើញ
<code>Object put(Object key, Object value)</code>	បញ្ចូលធាតុមួយដែលមាន <b>key</b> និង <b>value</b> ហើយប្រសិនបើ <b>key</b> មានរួចហើយ វានឹងជំនួសធាតុដែលបានដាក់ជាថ្មីនោះ
<code>Object remove(Object key)</code>	លុប ធាតុជាក់លាក់ណាមួយ តាមរយៈ <b>key</b> ដែលផ្តល់ឱ្យ
<code>int size()</code>	បោះ ចំនួន ធាតុ ដែលមាននៅក្នុង <b>map</b>
<code>boolean isEmpty()</code>	បោះតម្លៃ <b>true</b> ប្រសិនបើ <b>map</b> នោះ ទទេ

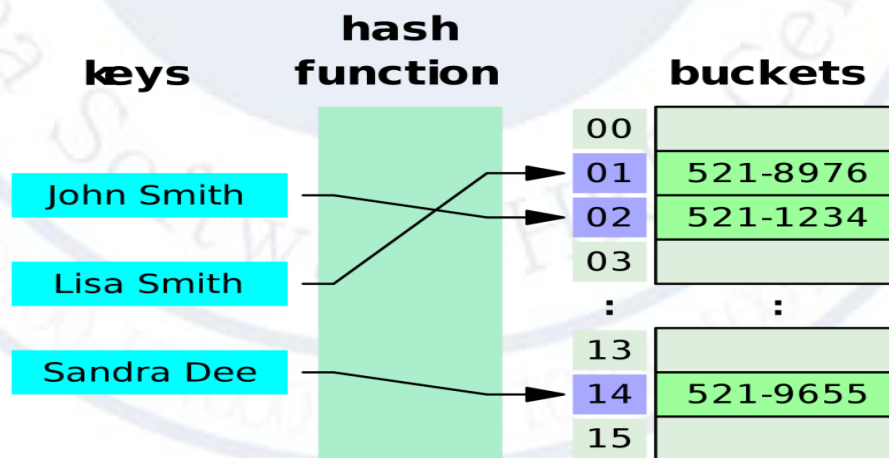
## ២.៣ Map.Entry

- **Entry** គឺជា **subinterface** របស់ **Map** ដូច្នោះហើយ យើងអាចដំណើរការវាបានដោយឈ្មោះ **Map.Entry** ដែលផ្តល់អោយ **methods** មួយចំនួន ដើម្បីទាញយក **key** និង **value**។

Method	Description
<b>getKey()</b>	ត្រូវបានប្រើដើម្បីទទួលបាន <b>Key</b> ដែលមាននៅក្នុង <b>Entry</b>
<b>getValue()</b>	ត្រូវបានប្រើដើម្បីទទួលបាន <b>Value</b> ដែលមាននៅក្នុង <b>Entry</b>

## ២.៤ Hashtable

- **Hashtable** ជាប្រភេទ **Collection** មួយដែលប្រើសម្រាប់រៀបចំ **key** ដែលបញ្ចូលទៅក្នុង **Map** ដោយមិនស្ថិតតាមលំដាប់លំដោយតាម **key** ដែលយើងបានបញ្ចូលឡើយ។ **Hashtable** អនុញ្ញាតឱ្យយើងអាចរក្សារទុកធាតុ និងទាញយកធាតុមកវិញបានតាមរយៈ **key** ដោយប្រើប្រាស់នូវ **function (algorithm)** ណាមួយរបស់ **map interface** ។ ម៉្យាងវិញទៀត **Bucket** វិញមានមុខងារសំរាប់ផ្គុំ **key and value** របស់ធាតុហើយ **key** ផ្សេងគ្នាអាចផ្គុំក្នុង **bucket** តែមួយបានផងដែរ ។ សូមពិនិត្យមើលរូបខាងក្រោម:



## ២.៤ Hashtable

Constructor Name	Description
<code>Hashtable ()</code>	<b>Constructor</b> ផ្តល់ឱ្យដំបូងគេ។
<code>Hashtable(int size)</code>	កំណត់ទំហំជាកលាក ។
<code>Hashtable(int size,float fillRatio)</code>	កំណត់ទំហំជាកលាកនិងមាន <b>fillRatio</b> សំរាប់កំណត់ <b>tradeoff</b> រវាងពេលវេលានិងទំហំហើយតំលៃរបស់វាចាប់ពី (0.0 to 1.0)។
<code>Hashtable(Map m):</code>	បង្កើត <b>Hashtable</b> ហើយចាប់ផ្តើមជាមួយធាតុ m។

## ២.៥ LinkedHashMap

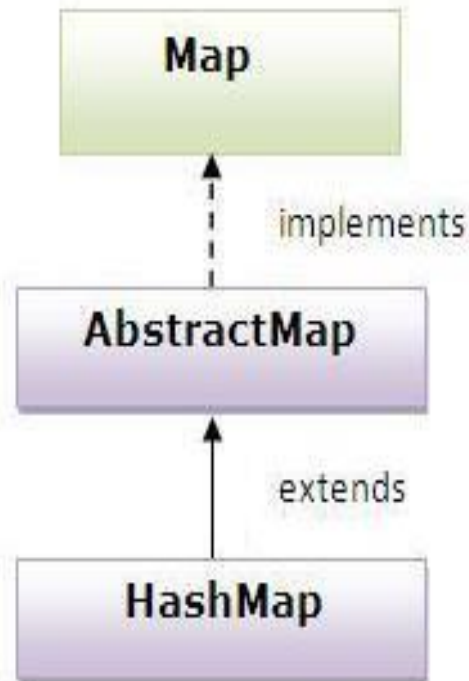
- **LinkedHashMap** ជា **Collection** មួយប្រភេទដែលប្រើសម្រាប់រៀបចំ **key** ដែលបញ្ចូលទៅក្នុង **Map** ឲ្យស្ថិតតាមលំដាប់លំដោយទៅតាម **key** ដែលយើងបានបញ្ចូល ។ ហើយនៅពេលអ្នកចង់យក **key/values** មកវិញក៏តាមលំដាប់លំដោយដែរ។

Constructor Name	Description
<code>LinkedHashMap( )</code>	ជាប្រភេទ <b>Default Constuctor</b> ។
<code>LinkedHashMap(Map m)</code>	ជា <b>Constructor</b> ដែលផ្ដើមជាមួយ ធាតុ <b>m</b> ។
<code>LinkedHashMap(int capacity)</code>	កំណត់ទំហំជាកំលាំង ។
<code>LinkedHashMap(int capacity, float fillRatio)</code>	កំណត់ទំហំជាកំលាំងនិងមាន <b>fillRatio</b> សំរាប់កំណត់ <b>tradeoff</b> រវាងពេលវេលានិងទំហំហើយតំលៃរបស់វាចាប់ពី <b>(0.0 to 1.0)</b> ។



## ២.៦ HashMap

- **HashMap implements** ពី **Map Interface** ហើយ **extends** ពី **AbstractMap class**។
- ការដំរៀបធាតុ ពុំបានរៀបតាមលំដាប់នោះទេ



## ២.៦ HashMap

- **HashMap class** មាន **constructor** ចំនួន៤ :
  - ☞ **HashMap()** : ជាប្រភេទ **default HashMap Constructor**
  - ☞ **HashMap(Map m)** : hashmap ប្រើប្រាស់ធាតុ m
  - ☞ **HashMap(int capacity)** : ជាប្រភេទ Constructor ដែលយើង Initialize តម្លៃឱ្យវា
  - ☞ **HashMap(int capacity, float fillRatio)** : ជាប្រភេទ Constructor ទី៤ ដែល អាចឱ្យយើង initialize តម្លៃ ទាំង Capacity និង Ratio ហើយវា Map ដោយប្រើ argument របស់វា ។

## ២.៦ HashMap

Name	Description
<code>void clear()</code>	លុបចោលនូវ <b>mappings</b> ទាំងអស់ពី <b>Map</b>
<code>Object clone()</code>	គឺជាការ <b>copy HashMap instance</b> ដែល <b>Key</b> និង <b>values</b> ក៏មិនបាន <b>Copy</b> នោះទេ
<code>boolean isEmpty()</code>	<b>Return</b> ពិត ប្រសិនបើ <b>Map contain</b> គ្មាន <b>Key-value</b> សំរាប់តភ្ជាប់
<code>Set keySet()</code>	<b>Return</b> សំណុំនៃ <b>Keys</b> ក្នុង <b>Map</b>
<code>Object remove(Object key)</code>	លុបចោល ការ តភ្ជាប់នៃ <b>Key</b>
<code>int size()</code>	<b>Return</b> ជាលេខនៃ <b>Key-value</b>
<code>Collection values()</code>	<b>Return</b> ជាតម្លៃ ជាប្រភេទ <b>Collection</b>

## ២.៧ TreeMap

- **TreeMap** ជាប្រភេទ **Collection** ដែល **values** របស់វា អាស្រ័យទៅលើ **key** ។  
ហើយវា **implements NavigableMap interface** ហើយ និង **extends** ពី **AbstractMap class** ។
- វាមិនអាចមាន **key** ដែលទទេបានឡើយ ប៉ុន្តែវាអាចមាន **values** ទទេជាច្រើន  
បានការដំរៀបធាតុគឺដូចទៅនឹង **HashMap** ដែរ ប៉ុន្តែវារៀបតាមលំដាប់កើន ពី  
តូចទៅធំ ។

# សមាជិក

## ក្រុមអ្នកស្រាវជ្រាវ



ល. ខេង ចាន់វិជ្ជា

vichea@rocketmail.com



ល. សែម ចិត្រា

sabaychitra84@gmail.com



ល. ឡៅ ស៊ុនឡេង

sunlenglao@gmail.com.com



ក. លីម សុខហេង

lim.sokheng1@gmail.



ល. យ៉ូ វណ្ណវិទូ

ravuthz@gmail.com

## ក្រុមផលិតវីដេអូ



ល. ឈុន បញ្ញាវត្ត

chhunpanharath@gmail.com



ល. ព្រាប វិទូ

Itpreap.vuthy@gmail.com



ល. ហង្ស បូរី

hounboreyrupp@gmail.com



ល. ហួ ឈុនឡេង

huochhunleng@yahoo.com

---

មេរៀនបន្ទាប់ និងធ្វើការបង្ហាញពី List Collection