

JAVA BEGINNER

COLLECTIONS FRAMEWORK



Abstract of New Technology

Website <http://www.antkh.com>

Tel 010 / 016 66 66 53

Prepared By Tum Sakal

Tel 087 36 31 30

Mail tumsakal.ts@gmail.com

Facebook [sak kal](#)

JAVA COLLECTIONS

- Java Collection គឺជា សំណុំនៃ interfaces និង classes សំរាប់ ជំនួយដល់ការគ្រប់គ្រង និងការរក្សាទុកទិន្នន័យក្នុងទំរង់នៃ data structure ណាមួយ។
- អត្ថប្រយោជន៍:
 - ជំនើការលឿនជាង Array
 - កាត់បន្ថយការសរសេរកូដ:

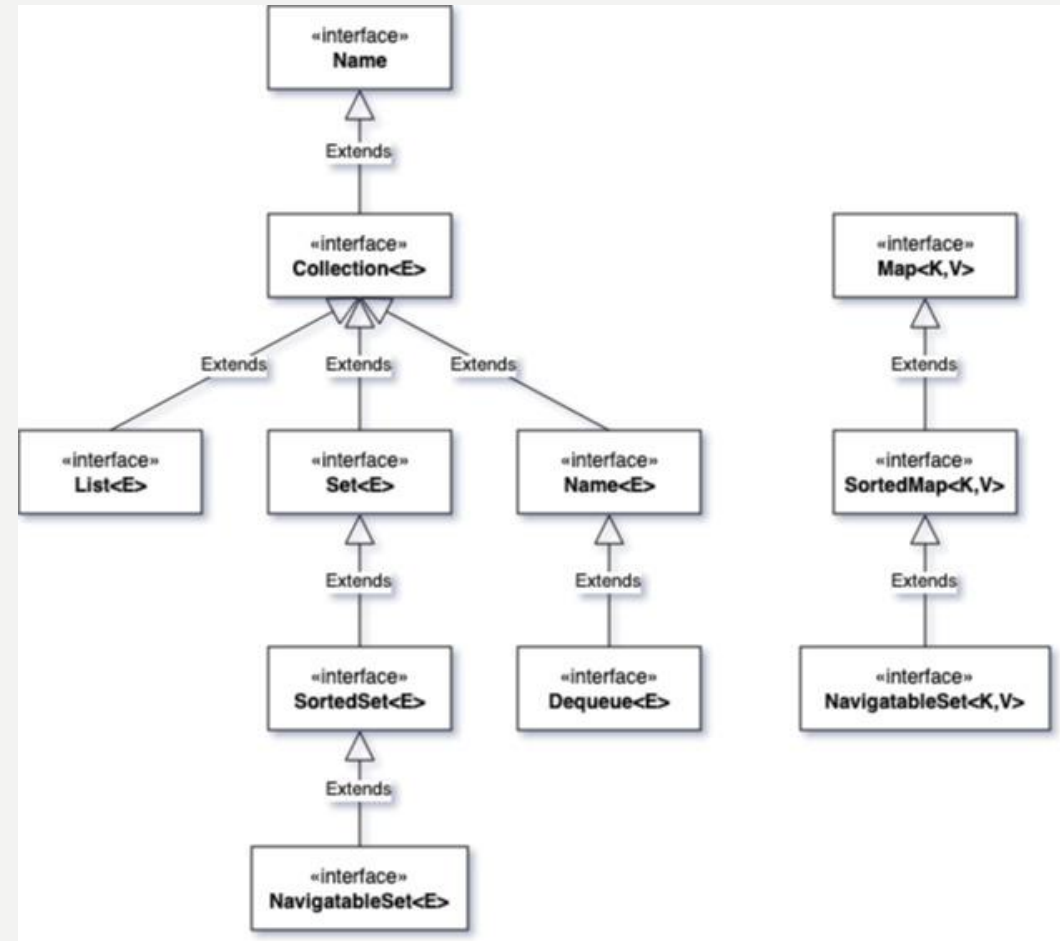
JAVA COLLECTION

- Java Collection បែងចែកជាបីផ្នែកគឺ
 - Interface
 - Implementation
 - និង Algorithm

CORE INTERFACE

- Java Collections មាន Interfaces សំខាន់ៗដូចជា

- Collection<E>
 - List<E>
 - Set<E>
 - Queue<E>
 - Dequeue<E>
- ផង Map<K, V>



COLLECTION<E>

- Collection<E> គឺជា Root Interface នៃ List, Set, Queue, Dequeue,...
- Collection<E> Interface រក្សាទុកទិន្នន័យក្នុងទំរង់ Sequence of elements

- Return methodName([parameter])
- boolean add(E element)
- boolean addAll(Collection<? extends E> coll)
- void clear()
- boolean contains(E element)
- boolean containsAll(Collection<? extends E> coll)
- boolean isEmpty()
- boolean remove(E element)
- boolean removeAll(Collection<?> coll)
- int size()
- Object[] toArray()
- T[] toArray(T[] array)

LIST<E>

- List<E> គឺជា Collection<E> Interface ដែល:
 - រក្សាទុកទិន្នន័យមានលំដាប់ (Insertion-Order)
 - ប្រើប្រាស់ Index: Zero-Based Index
 - Dynamic Size: អាចបន្ថែម រំលុបធាតុ
 - អាចផ្ទុកទិន្នន័យស្អាត និងទិន្នន័យទទេរ
- Class ដែល Implement List Interface មានដូចជា: `ArrayList<E>`, `LinkedList<E>`, `Stack`, `Vector`, ...។

data	data	data	data	data
0	1	2	3	4

- Return methodName([parameter])
- boolean add(int index, E element)
- boolean addAll(int index, Collection<? extends E> coll)
- E get(int index)
- int indexOf(Object obj)
- int lastIndexOf(Object obj)
- E remove(int index)
- E set(int index, E element)
- List<E> subList(int fromIndex, int toIndex)

ARRAYLIST

- ArrayList<E> គឺជា Class ដែល Implement “List<E>” Interface ក្នុងទំរង់នៃការរក្សាទុកទិន្នន័យបែប Array ដែលមានចំនួនធាតុមិនកំណត់ មានន័យថាអាចបន្ថយ រឺបន្ថែមបាន (Resizable-Array) ។

```
ArrayList<E> objectName = new ArrayList<>();
```

```
List<E> objectName = new ArrayList<>();
```

ARRAYLIST

```
List<Integer> list = new ArrayList<>();
```

```
list.add(10);
```

10

Size = 1

```
list.add(20);
```

10	20
----	----

Size = 2

```
list.add(30);
```

10	20	30
----	----	----

Size = 3

```
list.add(40);
```

10	20	30	40
----	----	----	----

Size = 4

```
list.add(50);
```

10	20	30	40	50
----	----	----	----	----

Size = 5

0 1 2 3 4

```
list.remove(2);
```

10	20	40	50
----	----	----	----

Size = 4

SET<E>

- Set<E> គឺជា Collection<E> Interface ដែល :
 - រក្សាទុកទិន្នន័យមិនមានលំដាប់ (No Insertion-Order)
 - រក្សាទុកទិន្នន័យមិនស្អាត
 - មិនប្រើប្រាស់ Index
- Class ដែល Implement Set Interface មានដូចជា: **HashSet<E>**, **LinkedHashSet<E>**, **TreeSet<E>**, ...។

data	data	data	data	data
------	------	------	------	------

HASHSET

- HashSet<E> គឺជា Class ដែល Implement “Set<E>” Interface

```
HashSet<E> objectName = new HashSet<>();
```

```
Set<E>          objectName = new HashSet<>();
```

HASHSET

```
Set<Integer> set = new HashSet<>();
```

```
set.add(10);
```

10

Size = 1

```
set.add(20);
```

20	10
----	----

Size = 2

```
set.add(30);
```

20	10	30
----	----	----

Size = 3

```
set.add(40);
```

20	40	10	30
----	----	----	----

Size = 4

```
set.add(50);
```

50	20	40	10	30
----	----	----	----	----

Size = 5

```
set.remove(20);
```

50	40	10	30
----	----	----	----

Size = 4

MAP<K,V>

- លក្ខណៈពិសេសរបស់ Map<K,V> Interface គឺ:

- រក្សាទុកទិន្នន័យមិនមានលំដាប់ (No Insertion-Order)
- រក្សាទុកទិន្នន័យមិនស្អាត
- ប្រើប្រាស់ Key ជំនួស Index
- រក្សាទុកទិន្នន័យក្នុងទំរង់គូរ Key និង Value

Key	Values
Key1	Value1 of Key1
Key2	Value of Key2
...	...
KeyN	Value of KeyN

- Class ដែល Implement Set Interface មានដូចជា: **HashMap**, **LinkedHashMap**, **TreeMap**, ...។

- Return methodName([parameter])
- void clear()
- boolean containsKey(Object key)
- boolean containsValue(Object value)
- V get(Object key)
- V getOrDefault(Object key, V defaultValue)
- boolean isEmpty()
- Set<V> keySet
- V put(K key, V value)
- V remove(Object key)
- int size()

HASHMAP

- `HashMap<K,V>` គឺជា Class ដែល Implement “`Map<K,V>`” Interface

`HashMap<K,V> objectName = new HashMap<>();`

`Map<K,V> objectName = new HashMap<>();`

HASHMAP

```
Map<Integer, String> numDictionary = new HashMap<>();
```

```
numDictionary.put("one", 1);
```

```
numDictionary.put("two", 2);
```

```
numDictionary.put("three", 3);
```

```
numDictionary.put("four", 4);
```

```
numDictionary.put("five", 5);
```

```
numDictionary.remove("two");
```

"one"	1
"two"	2
"three"	3
"four"	4
"five"	5