

•	
•	(Concrete)
•	
•	
•	

가

가 ( )

가?

가?

가 가?

가

가 “ ”

가

가

2 가

: Vector , Stack, Hashtable, BitSet

Enumeration

2

가

C++ “ (STL, Standard Template Library)”  
(Generic Algorithms)” STL “

가

. 가

(queue)

-----  
:

LinkerList

(tail)

가

(head)

“

.(first in, first out)”

**2-1 :**

가

:

```
interface Queue
{
    void add(Object obj);
    Object Remove();
    int size();
}
```

가

2 가

“

”

.( 2-2 )

2-2:

Queue

class CircularArrayQueue **implements Queue**

```
{  
    CircularArrayQueue( int capacity) { ...}  
    public void add(Object obj){ ...}  
    public Object remove() { ..}  
    public int size() { ..}
```

```
    private Object[] elements;  
    private int head;  
    private int tail;
```

```
}
```

class LinkerListQueue **implements Queue**

```
{  
    LinkerListQueue () { ...}  
    public void add(Object obj){ ...}  
    public Object remove() { ..}  
    public int size() { ..}
```

```
    private Link head;  
    private Link tail;
```

```
}
```

가

(CircularArrayQueue )

**Queue** expressLane = new **CircularArrayQueue**(100);

ExpressLane.add( new Customer("Harry"));

LinkerListQueue 가

```
Queue expressLane = new LinkerListQueue ();
ExpressLana.add( new Customer("Harry"));
```

가?

가

add :

```
class CircularArrayQueue
{
    public void add(Object obj) throw CollectionFullExsception
    ...
}
```

CircularArrayQueue Queue

BoundedQueue Queue ? add 가

?

가

## Collection

. 가 :

Boolean add(Object obj)

Iterator iterator()

가 가 가 .

.

add

가 . add

가

true

flase

,

(set) 가

가

add

.

.

iterator

Iterator

. -

.

.

Iterator

3

가

.

Object next()

Boolean hasNext()

void remove()

next

.

next

NoSuchElementException

.

next

hasNext

가

.

가

true

.

hasnext 가 true

next

.

Iterator iter = c.iterator();

while(iter.hasNext())

{ Object obj = iter.next();

do something with obj

}

---

```

        :
Enumeration      nextElement      hasMoreElement      Iterator      next      hasNext      가
Enumeration
.
.
.

```

---

```

        , remove      next
remove      가      Iterator
.
.
.
remove      가 Iterator
가      .      가
.

```

```

. C++
.
I      a[i]
.      i++
.
.
(lookup)      next
.

```

```

next
.(      2-3)

```

**2-3:**

---

: Iterator.next    InputStream.read

. Read    next

remove    . Remove    next

가  
가 ,

```
Iterator it = c.iterator();  
it.next(); //  
it.remove(); //
```

next    remove    가    next  
remove    IllegalStateException

```
it.remove();  
it.remove(); // error
```

, next

```
it.remove();  
it.next();  
it.remove(); // OK
```

. 가 ,

print

```
Public static void print(Collection c)  
{    System.out.print("[");
```

```

        Iterator iter = c.iterator();
        While(iter.hasNext())
            System.out.println(iter.next() + " ");
        System.out.print("");
    }

```

---

```

        :
        System.out.println(c
        toString
        가

```

---

```

public static Boolean addAll(Collection to, Collection from)
{
    Iterator iter = from.iterator();
    boolean modified = false;
    while(iter.hasNext())
        if(to.add(iter.next()))
            modified = true;
    return modified;
}

```

```

        가
        add true
        Collection Iterator add next

```

```

        가 . addAll

```

```

Collection 가

```

```

가

```



## Collection

:

```
int size()
boolean isEmpty()
boolean contains(Object obj)
boolean containsAll(Collection c)
boolean equals(Object other)
boolean addAll(Collection from)
boolean remove(Object obj)
boolean removeAll(Collection c)
void clear()
boolean retainAll(Collection c)
Object[] toArray()
```

가 .

API

, Collection 가

AbstractCollection

( add iterator )

```
Public class AbstractCollection implements Collection
{
    ...
    public abstract Boolean add(Object obj);

    public Boolean addAll(Collection from)
    {
        Iterator iter = iterator();
        boolean modified = false;
        while(iter.hasNext())
            if(to.add(iter.next()))
                modified = true;
        return modified;
    }
    ...
}
```

AbstractCollection (extend)  
 add  
 AbstractCollection  
 addAll  
 가  
 가

## java.util.Collection

- Iterator iterator()  
 true
- int size()  
 true
- boolean isEmpty()  
 true
- boolean contains(Object obj)  
 obj  
 : obj  
 true
- boolean containsAll(Collection others)  
 others  
 : others  
 true
- boolean add(Object element)  
 element  
 : element  
 true
- boolean addAll(Collection other)  
 other  
 : other  
 true
- boolean remove(Object obj)  
 obj  
 : obj  
 true
- boolean removeAll(Collection other)

- true  
: other
- void clear()
- boolean retainAll(Collection other)

- true  
: other
- Object[] toArray()

## java.util.Iterator

- boolean hasNext()  
가 true
- Object next()
- NoSuchElementException
- Object remove()

(next )  
IllegalStateException

## (Concrete)

가

가

가

1

Vector

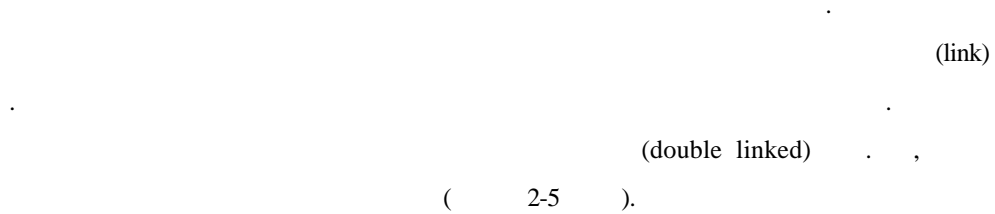
가

( 2-4 ).

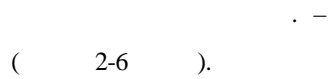
가

가

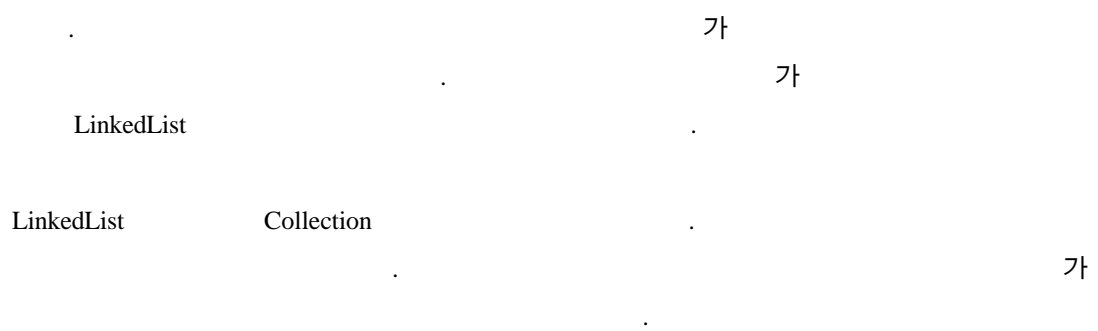
**2-4:**



**2-5 :**



**2-6:**



```
LinkedList staff = new LinkedList();
staff.add("Angela");
staff.add("Bob");
staff.add("Carl");
Iterator iter = staff.iterator();
for(int i=0; i<3; i++)
    System.out.println(iter.next());
```

```
iter.remove(); // remove last visited element
```

```

        가 (Ordered Collection) . LinkedList.add
        가 .
        가 . add
        가
        . , set
        . Iterator add
        . add
        .
        ListIterator

```

```
interface ListIterator extends Iterator
```

```
{
    void add(object);
    ...
}
```

```
Collection.add boolean . - add
        가 .
```

```

        가, ListIterator 가 —
Object previous()
Boolean hasPrevious()
-- . next previous

```

```
LinkedList listIterator ListIterator
```

```

        ListIterator iter = staff.listIterator();

```

```
add 가 . ,
        “Juliet” 가 ( 2-7 ).

```

```

        ListIterator iter = staff.listIterator();
        iter.next();
        iter.add(“Juliet”)

```

2-7:

가

add

가 .

가 .

ListIterator

add

. 가 가 . 가

( hasNext 가 flase ) 가

( tail) . 가 n 가 가

n+1 가 . N+1 가

. , 가 A,B,C

가 가 가 (| .).

|ABC

A|BC

AB|C

ABC|

-----  
: “ (Cursor)” . remove

. next remove

. previous .

remove .

add remove

.

-----  
set next previous

. 가 , .

ListIterator iter = list.listIterator();

Object oldValue = iter.next(); // return first element

가  
가  
가  
가

iter2.next 가 ConcurrentModificationException

```

    ConcurrentModificationException

```

=====

```

:
가,
    set
    set
Collection

```

---

```

LinkedList
    가,
        ListIterator
        Collection
        가
AbstractCollection
    [ A, B, C]
    contains
    , staff.contains("Harry")
    true
    가

```

---

```

:
    가
    가

```

```

LinkedList list = new LinkedList();
list.add(list); // add list itself
String contents = list.toString(); // dies with infinite recursion

```

---

```

n
n-1
가

```



LinkedList  
get  
:  
Object obj = list.get(n);

for( int i=0; i < list.size(); i++)  
Do something with list.get(i);

. LinkedList

-----  
: get : 가 size()/2  
-----

가

가 : nextIndex next  
. previousIndex previous  
, nextIndex -  
n 가  
list.listIterator(n) n  
, next list.get(n)

가 get() set()

가 ?

가

ArrayList

가 ArrayList

2-1

removeAll

|ACE |BDFG  
A|CE |BDFG  
AB|CE B|DFG  
....

System.out.println(a);  
a

## 2-1 : LinkedListTest.java

```
import java.util.*;  
  
public class LinkedListTest  
{ public static void main(String[] args)  
  { List a = new LinkedList();  
    a.add("Angela");  
    a.add("Carl");  
    a.add("Erica");  
  
    List b = new LinkedList();  
    b.add("Bob");  
    b.add("Doug");  
    b.add("Frances");  
    b.add("Gloria");  
  
    // merge the words from b into a  
  
    ListIterator aIter = a.listIterator();  
    Iterator bIter = b.iterator();  
  
    while (bIter.hasNext())  
    { if (aIter.hasNext()) aIter.next();
```

```

        aIter.add(bIter.next());
    }

    System.out.println(a);

    // remove every second word from b

    bIter = b.iterator();
    while (bIter.hasNext())
    { bIter.next(); // skip one element
      if (bIter.hasNext())
      { bIter.next(); // skip next element
        bIter.remove(); // remove that element
      }
    }

    System.out.println(b);

    // bulk operation: remove all words in b from a

    a.removeAll(b);

    System.out.println(a);
}
}

```

#### java.util.List

- ListIterator listIterator()
- ListIterator listIterator(int index)  
next  
: index
- void add(int i, Object element)  
가 .  
: index  
element 가
- void addAll(int i, Collection elements)  
가 .  
: index  
elements 가
- Object remove(int i)  
: index

- Object set(int i, Object element)

i : index  
element

- int indexOf(Object element)

-1 : element

- int lastIndexOf(Object element)

-1 : element

#### java.util.ListIterator

- void add(Object element)

가  
: element 가

- void set(Object element)

next previous  
가 next previous  
:  
: element

IllegalStateException

- boolean hasPrevious()

true

- Object previous()

NoSuchElementException

- int nextIndex()

next

- int previousIndex()

previous

```
java.util.LinkedList
```

- LinkedList()
- LinkedList(Collection elements)

가

: elements 가

- void addFirst(Object element)
- void addLast(Object element)

가

: element 가

- Object `getFirst()`
- Object `getLast()`

- Object removeFirst()
- Object removeLast()

List

## LinkedList

. List

(Ordered Collection)

2

$$\vdots$$

get      set

List

## ArrayList

- . ArrayList

Vector

•

Object[]

Vector

## ArrayList

?

가

. Vector

(Synchronized)

Vector

가

Vector

## ArrayList

## ArrayList

Vector

## ElementAt

setElementAt

get      set

가

가

가

가

가

( 2-8).

345 101 42  
.( 345 101 42 .)  
가  
(hash collesion)

가

가

2-8 :

가 가

150%  
 (prime number) 가  
 , 100 가 151  
 ,  
 (rehash) 가  
 (load factor) , 가  
 0.75 ( 0.75 가 .) 75%  
 2 0.75

가 set . add  
 가 가

HashSet  
 , HashSet 0.75 101

HashSet(int initialCapacity)  
 HashSet(int initialCapacity , float loadFactor)  
 add 가 . contains

( 2-2) System.in

가

( [www.gutenberg.net](http://www.gutenberg.net) )

“ ”

```
java SetTest < alice30.txt
```

가 .

.(  
.)

5,909

## 2-2: SetTest.java

```
import java.util.*;
import java.io.*;

public class SetTest
{ public static void main(String[] args)
  { Set words = new HashSet(59999);
    // set to HashSet or TreeSet
    long totalTime = 0;

    try
    { BufferedReader in = new
      BufferedReader(new InputStreamReader(System.in));
      String line;
      while ((line = in.readLine()) != null)
      { StringTokenizer tokenizer = new StringTokenizer(line);
        while (tokenizer.hasMoreTokens())
        { String word = tokenizer.nextToken();
          long callTime = System.currentTimeMillis();
          words.add(word);
          callTime = System.currentTimeMillis() - callTime;
          totalTime += callTime;
        }
      }
    }
    catch (IOException e)
    { System.out.println("Error " + e);
    }

    Iterator iter = words.iterator();
    while (iter.hasNext())
      System.out.println(iter.next());
    System.out.println(words.size()
      + " distinct words. " + totalTime + " milliseconds.");
  }
}
```



java.util.HashSet

- HashSet()

- HashSet(collection elements)

elements 가  
:

- HashSet(int initialCapacity)

initialCapacity

- HashSet(int initialCapacity , float loadFactor)

initialCapacity  
loadFactor

0.0

1.0

String

hashCode 가  
2-1 String hashCode

2-1 : hashCode

Hello	140207504
Harry	140013338
Hacker	884756206

hashcode Object

가

가

```
String s = "OK";
StringBuffer sb = new StringBuffer(s);
System.out.println(s.hashCode() + " " + sb.hashCode());
String t = "OK";
StringBuffer tb = new StringBuffer(t);
System.out.println(t.hashCode() + " " + tb.hashCode());
```

2-2

2-1 :

"OK" string	"OK" StringBuffer
3030	20526976
3030	20527144

s      t      .      sb      tb      .

StringBuffer      가      Object

hashCode

(      ).

가

,      Item      가      .

```
anItem = new Item("Toaster" , 49954);
```

가      .      ,

```
class Item
```

```
{      ....
```

```

public int hashCode()
{
    return 13*description.hashCode() + 17 * partNumber
}
...
private String description;
private int partNumber;
}

```

가 description  
가 .

, equals 가 . Object equals  
.  
equals

equals 가 .

```

class Item
{
    ....
    public boolean equals(Object other)
    {
        if(other != null && getClass() == other.getClass() )
        {
            Item otherItem= (Item) other;
            return description.equals(otherItem.description)
                && partNumber == otherItem.partNumber;
        }
        else
            return false;
    }
    ...
}

```

---

:	equals	hashCode	가	:	x.equals(y)가 true
x.hashCode()	y.hashCode()		.		

---

## java.lang.Object

- boolean equals(Object obj)  
가 . 가 true  
false .  
: obj  
(null false .)
- int hashCode()  
가 .

TreeSet 가 .  
(sorted collection) . 가  
(iterate) .  
, 가 가 .

```
TreeSort sorter = new TreeSet();  
sorter.add("Bob");  
sorter.add("Angela");  
sorter.add("Carl");  
Iterator iter = sorter.iterator();  
while(iter.hasNext()) System.out.println(iter.next());
```

: Angela Bob Carl.

.( - .) -

mit , ,

Introduction to Algorithms[MIT press, 1990] . 가

가 가 .  
가 가 n

$2^n$  , 1,000

가 10 .

TreeSet 가 HashSet 가 .  
TreeSet . - 2-3 .

2-3: 가

-----

28195	5909	5sec	7sec
466300	37545	75sec	98sec

-----

java.util.TreeSet

- TreeSet()
- TreeSet(Collection elements)

가 .

: elements 가

TreeSet ?

Comparable

가 . :

int compareTo(Object other)

a.compareTo(b) a b 가 0 , a 가 b , a 가 b  
.( >0 , 0, <0) .

Comparable

. String

. String

CompareTo

.

, Item .

```

class Item implements Comparable
{
    public int compareTo(Object other)
    {
        Item otherItem = (Item)other;
        return partNumber - otherItem.partNumber;
    }
    ...
}

```

compareTo 가 Comparable Object  
 가 compareTo ClassCastException  
 .( compareTo  
 .)  
 .  
 가 0,  
 .

---

: 가 . X 가  
 Y가 X - Y .

---

Comparable  
 가 .  
 ,  
 가 ? Comparable  
 가 ?

TreeSet Comparator  
 . Comparator  
 가 .

int compare(Object a, Object b)

compareTo compare a 가 b , 0 ,  
 .

(description)

Comparable

```
class ItemComparable implements Comparator
{
    public int compare(Object a, Object b)
    {
        Item itemA = (Item)a;
        Item itemB = (Item)b;
        String descrA = itemA.getDescription();
        String descrB = itemB.getDescription();
        return descrA.compareTo(descrB);
    }
}
```

```
ItemComparator comp = new ItemComparator();
TreeSet sortByDescription = new TreeSet(comp);
```

가

(function object)

:

```
SortedSet sortByDescription = new TreeSet(
    new Comparator()
    {
        public int compare(Object a, Object b)
        {
            Item itemA = (Item)a;
            Item itemB = (Item)b;
            String descrA = itemA.getDescription();
            String descrB = itemB.getDescription();
            return descrA.compareTo(descrB);
        }
    });
```





```

{ public Item(String aDescription, int aPartNumber)
  { description = aDescription;
    partNumber = aPartNumber;
  }

  public String getDescription()
  { return description;
  }

  public String toString()
  { return "[description=" + description
    + ", partNumber=" + partNumber + "]\n";
  }

  public boolean equals(Object other)
  { if (getClass() == other.getClass())
    { Item otherItem = (Item)other;
      return description.equals(otherItem.description)
        && partNumber == otherItem.partNumber;
    }
    else
      return false;
  }

  public int hashCode()
  { return 13 * description.hashCode() + 17 * partNumber;
  }

  public int compareTo(Object other)
  { Item otherItem = (Item)other;
    return partNumber - otherItem.partNumber;
  }

  private String description;
  private int partNumber;
}

```

#### java.lang.Comparable

- int compareTo(Object other)
 

this 가 other ,

0 this 가 other .

: other

#### java.util.Comparator

- int compareTo(Object a, Object b)
 

a 가 br , 0 a 가 b

.

: a,b

## java.util.SortedSet

- Comparator comparator()

Comparable

compareTo

null

.

- Object first()
- Object last()

가

.

## java.util.TreeSet

- TreeSet(Comparator c)

.

: c

- TreeSet(SortedSet elements)

가

.

.

: elements

가

.

(look up)

가

가

.

(look up)

—

가

. Map

.

/

가 ,

ID

Employee

: HashMap    TreeMap.

가?

가

```
HashMap staff = new HashMap();
Employee harry = new Employee("Harry Hacker");
staff.put("987-98-9996" , harry);
...
```

가  
Employee  
.( )

```
String s = "1411-16-2536";
e = (Employee)staff.get(s); // gets harry
```

가 get null .

put  
.(  
put null  
.)

remove() . size() .

.(  
(view) . Collection

가 : , ( ) /  
/ .

```
Set KeySet()
```

Collection values()

Set entrySet()

( Map.Entry .)

KeySet HashSet TreeSet . Set  
Set

. Set Collection .  
, Set 가  
. Collection  
.

, :

```
Set keys = map.keySet();
Iterator iter = keys.iterator();
while(iter.hasNext())
{
    Object key = iter.next();
    do something with key
}
```

-----  
:

. :

```
Set entries = staff.entrySet();
Iterator iter = entries.iterator();
while(iter.hasNext())
{
    Map.Entry entry = (Map.Entry)iter.next();
    Object key = entry.getKey();
    Object value = entry.getValue();
    Do something with key, value
}
```

-----  
remove

```

        .
        가
        .
        add
        가
        UnsupportedOperationException
        .
        /
        가
        .

```

```

-----
:      Hashtable      (      .)      (
      .)      가      .
.
.
-----

```

```

2-4      .      /      가      .      ,
      -
      .
      get
      .

```

## 2-4: MapTest.java

```

import java.util.*;

public class MapTest
{
    public static void main(String[] args)
    {
        Map staff = new HashMap();
        staff.put("144-25-5464", new Employee("Angela Hung"));
        staff.put("567-24-2546", new Employee("Harry Hacker"));
        staff.put("157-62-7935", new Employee("Gary Cooper"));
        staff.put("456-62-5527", new Employee("Francesca Cruz"));

        // print all entries

        System.out.println(staff);

        // remove an entry

        staff.remove("567-24-2546");

        // replace an entry

        staff.put("456-62-5527", new Employee("Francesca Miller"));

        // look up a value

        System.out.println(staff.get("157-62-7935"));

        // iterate through all entries
    }
}

```

```

        Set entries = staff.entrySet();
        Iterator iter = entries.iterator();
        while (iter.hasNext())
        {
            Map.Entry entry = (Map.Entry)iter.next();
            Object key = entry.getKey();
            Object value = entry.getValue();
            System.out.println("key=" + key + ", value=" + value);
        }
    }
}

class Employee
{
    public Employee(String n)
    {
        name = n;
        salary = 0;
    }

    public String toString()
    {
        return "[name=" + name + ", salary=" + salary + "]";
    }

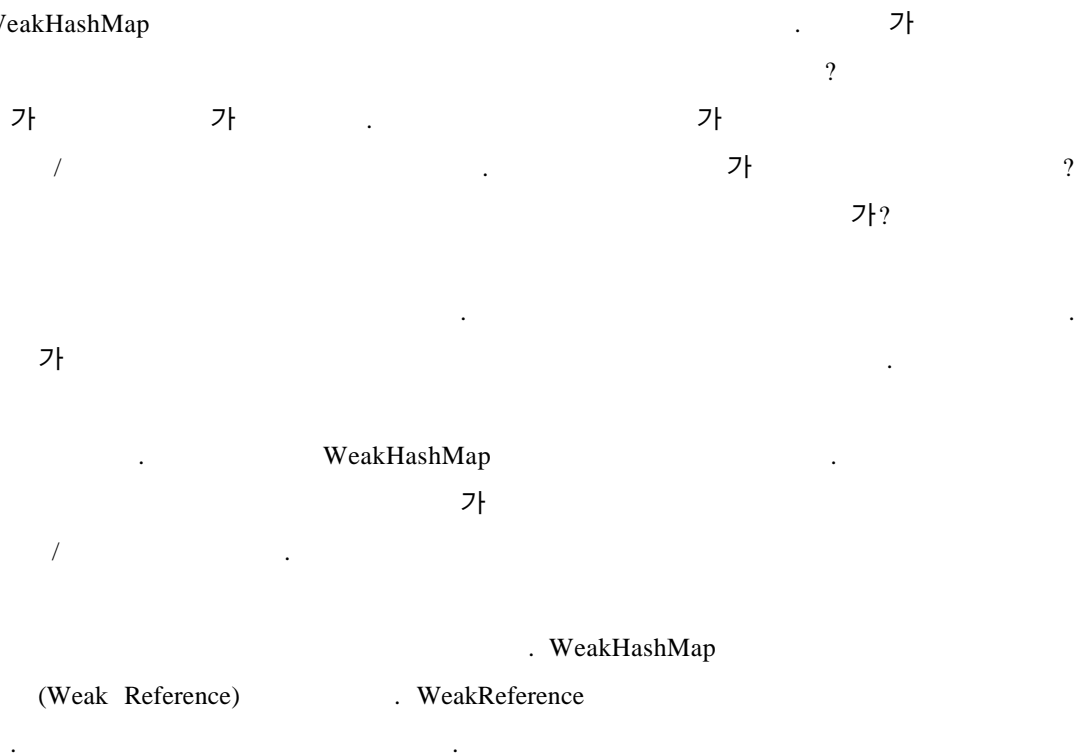
    public void setSalary(double s)
    {
        salary = s;
    }

    private String name;
    private double salary;
}

```

## WeakHashMap

WeakHashMap



WeakReference

WeakHashMap

가

WeakHashMap

가

가

WeakHashMap

java.util.map

- Object get(Object key)
 

가

가

null

:

key (null .)
- Object put(Object key , Object value)
 

가

가

가

key null (null .)

value (null .)
- void putAll(Map entries)
 

가

:

entries 가
- boolean containsKey(Object key)
 

가

true

key
- boolean ContainsValue(Object value)
 

true

value
- Set entrySet()
 

/ Map.Entry

가
- Set KeySet()
 

가

- Collection values()

가

#### java.util.Map.Entry

- Object getKey()
- Object getValue()
- Object setValue(Object value)

: value

#### java.util.HashMap

- HashMap()
- HashMap(Map entries)  
가  
: entries 가
- HashMap(int initialCapacity)
- HashMap(int initialCapacity , float loadFactor)

: initialCapacity  
loadFactor

1.0 0.75 0.0

#### java.util.WeakHashMap

- WeakHashMap()
- WeakHashMap(int initialCapacity)
- WeakHashMap(int initialCapacity , float loadFactor)

: initialCapacity  
loadFactor



1.0

null

가

2-9 :

가 :Collection Map.  
가 :

boolean add(Object element)

/ 가 put

boolean put(Object key , Object value)

get :

Object get(Object key)

List 가 . 가 .  
가 : List  
:

void add(int index , Object element)

Object get(int index)

void remove(int index)

ListIterator 가 :

void add(Object element)

가 Iterator next remove



:

AbstractCollection

AbstractList

AbstractSequentialList

AbstractSet

AbstractMap

6

:

LinkedList

ArrayList

HashSet

TreeSet

HashMap

TreeMap

2-10

**2-10**

Vector

Stack

Hashtable

Properties

.- 2-11 .

**2-11:**

```

        (    , Wrapper )
2-9          2-10          6          6
          5          .
.
Collection      Map
      KeySet
.
      가      .      KeySet      Set
.
.
.
.
.
      가      .      가
.
      . Vector
      가
      Vector      ArrayList
      가      . 가 ,
      가      가
.
.
.
      . 가 ,
Collection      synchronizedMap      Map
Map      :
.
HashMap hashMap = new HashMap();
Map = Collections.synchronizedMap(hashMap);
.
      ,      map      . Get      put
      -      가
.
.
      6      :
Collections.synchronizedCollection

```

Collections.synchronizedList  
Collections.synchronizedSet  
Collections.synchronizedSortedSet  
Collections.synchronizedMap  
Collections.synchronizedSortedMap

가

가

:

```
map = Collections.synchronizedMap(new HashMap())
```

가

가

가

```
Iterator iter = collection.iterator();  
while(iter.next() )  
    do something with iter.next();
```

next ConcurrentModificationException  
가 (lock)

```
synchronized(container)  
{  
    Iterator iter = collection.iterator();  
    while(iter.hasNext() )  
        do something with iter.next()  
}
```

가

가 , LinkedList List  
addFirst addLast 가



---

```

: API      , add      “ 가      ”
.      -      가
      가?
      가      .
      .
      .

```

---

```

Array      List      asList      가
.
.      ,
.
Card[] cardDeck      = new Card[52];
...
List cardList      = Arrays.asList(cardDeck);

ArrayList 가      get      set
.      (      add      remove
)      UnsupportedOperationException      .

(subrange)
. 가 ,      staff
가      10      19      가      .
subList      .

List group2 = staff.subList(10,20);

.- String
sunstring      .

. 가 ,      :

group2.clear(); // staff reduction

```



staff .

가

. SortedSet :

subSet(from,to)

headSet(to)

tailSet(from)

from to

. 가

subMap(from,to)

headMap (to)

tailMap (from)

.

### (LightWeight Collection Wrapper)

List 가

anObject n (illusion) .

Collection.nCopies(n,anObject);

- .

.

. 가 , “DEFAULT” 100

ArrayList .

ArrayList settings = new ArrayList(Collections.nCopies(100, “DEFAULTS”));

(List nCopies ) Set

가  
 . Collections.EMPTY\_LIST Collections.EMPTY\_SET  
 List Set singleton  
 :  
 ,

---

: JDK1.3 singletonList singletonMap EMPTY\_MAP 가 .

---

“ (optional)”  
 가  
 “ - -  
 ”  
 가  
 가 가 - 가  
 가 가 가 가

“ ” 가?  
 가  
 가

([www.objectspace.com](http://www.objectspace.com)) JGL

“ ”

java.util.Collections

- `static Collection synchronizedCollection(Collection c)`
- `static List synchronizedList(List c)`
- `static Set synchronizedSet(Set c)`
- `static SortedSet synchronizedSortedSet(SortedSet c)`
- `static Map synchronizedMap(Map c)`
- `static SortedMap synchronizedSortedMap(SortedMap c)`

$$\begin{array}{c} \vdots \\ \vdots \end{array} \quad \mathbf{C}$$

- `static Collection unmodifiableCollection(Collection c)`
- `static List unmodifiableList(List c)`
- `static Set unmodifiableSet(Set c)`
- `static SortedSet unmodifiableSortedSet(SortedSet c)`
- `static Map unmodifiableMap(Map c)`
- `static SortedMap unmodifiableSortedMap(SortedMap c)`

## UnsupportedOperationException

$$\begin{array}{c} \vdots \\ \vdots \end{array} \quad \mathbf{C}$$

- static List nCopies(int n, Object value)
- static Set singleton(Object value)

n

$$\vdots \quad n$$

value

- static final List EMPTY\_LIST
- static final Set EMPTY\_SET

```
java.util.Arrays
```

- static List asList(Object[] array)

가

가

: array

```
java.util.List
```

- List subList(int from , int to)

from  
to ( )

#### java.util.SortedSet

- SortedSet subSet(Object from , Object to)
- SortedSet headSet(Object to)
- SortedSet tailSet(Object from)

from  
to ( )

#### java.util.SortedMap

- SortedMap subMap(Object from , Object to)
- SortedMap headMap(Object to)
- SortedMap tailMap(Object from)

from  
to ( )

(intersection) 가

Set result = new HashSet(a);

가

RetainAll

result .retainAll(b);

b

. 가 ,  
ID 가  
가 .

```
Map staffMap = ...;  
Set terminatedIDs = ...;
```

ID .

```
StaffMap.keySet().removeAll(terminatedIDs);
```

.  
(sublist) (subset)  
. 가 , 10 가  
가 10 .

```
relocated.addAll(staff.subList(0,10));
```

(mutating operation) .

```
staff.subList(0,10).clear();
```

## API

API

, 가  
Vector :

```
Vector values = ...;
HashSet staff = new HashSet(values);
```

가

. 2 Vector List .

가 가 Arrays.asList  
:

```
String[] values = ...;
HashSet staff = new HashSet(Arrays.asList(values));
```

가

:

```
Vector values = new Vector(staff);
```

. toArray :

```
Object[] values = staff.toArray();
```

(Object) .  
:

```
String[] values = (String[])staff.toArray(); // Error!
```

toArray Object[] .  
. toArray 가 .  
가 0 .  
:

```
String[] value = (String[])staff.toArray(new String[0]);
```

-----  
: (String.class)Class toArray  
. API

“ (double doty)”

## java.util.Collection

- Object[] toArray(Object[] array)

가

가 null

array

가

가 null

가

: array

가

. 가 ,

가

```
if(a.length == 0) throw new NoSuchElementException();
```

```
Comparable largest = a[0];
```

```
for(int i=1; i < a.length; i++)
```

```
    if(largest.compareTo(a[i]) < 0) largest = a[i];
```

```
if(v.size() == 0) throw new NoSuchElementException();
```

```
Comparable largest = (Comparable)v.get(0);
```

```
for(int i=1; i < v.size() ; i++)
```

```
    if(largest.compareTo( (Comparable)v.get(i)) < 0) largest = v.get(i);
```

?

```

if(l.isEmpty())throw new NoSuchElementException();
Iterator iter = l.iterator();
Comparable largest = (Comparable)iter.next();
while(iter.hasNext() )
{
    Comparable next = (Comparable)iter.next();
    if(largest.compareTo(next) < 0) largest = next;
}

```

가 . 가 .  
가 가?  
?  
:

```

Object max(Comparable[] a)
Object max(Vector v)
Object max(LinkedList l)

```

가 가 .  
. get set  
.

Collection max

```

public static Object max(collection c)
{
    if(c.isEmpty()) throw new NoSuchElementException();
    Iterator iter = c.iterator();
    Comparable largest = (Comparable)iter.next();
    while(iter.hasNext())
    {
        Comparable next = (Comparable)iter.next();
        if(largest.compareTo(next) < 0) largest = next;
    }
    return largest;
}

```

,



```

LinkedList l;
Vector v;
Employee[] a;
....
largest = max(l);
largest = max(v);
largest = max(Arrays.asList(a));

```

C++

12

가

가

Collection

sort

List

```

List staff = new LinkedList();
// fill collection .:
Collections.sort(staff);

```

Comparable

가

Comparable

.(

.)

```

Collection.sort(staff,
    new Comparator()
    {
        public compare(Object a, Object b)

```

```

        {
            double salaryDifference = (Employee)a.getSalary()
                - (Employee)b.getSalary();
            if(salaryDifference < 0) return -1;
            if(salaryDifference > 0) return 1;
            return 0;
        }
    });

```

Collections.reverseOrder()  
 b.compareTo(a)가 ( Comparable  
 .)  
 가 ,

Collections.sort(staff, Collections.reverseOrder())

compareTo staff  
 sort 가

(merge sorting) ( 1998  
 , Robert Sedgwick Algorithms in C++, Parts1-4, P.366-369 )

(quick sort)  
 가.  
 가 가  
 ?

“ ” 가  
가 . 가 ,  
sort unmodifiableList .  
? 가 .

- set 가 (modifiable) .
- add remove 가 (resizable) .

Collection shuffle .  
가 ,

```
ArrayList cards = ...;  
Collections.shuffle(cards);
```

shuffle

2-5 1 49 49 Integer 6

## 2-5: ShuffleTest.java

```
import java.util.*;  
  
public class ShuffleTest  
{  
    public static void main(String[] args)  
    {  
        List numbers = new ArrayList(49);  
        for (int i = 1; i <= 49; i++)  
            numbers.add(new Integer(i));  
        Collections.shuffle(numbers);  
        List winningCombination = numbers.subList(0, 6);  
        Collections.sort(winningCombination);  
        System.out.println(winningCombination);  
    }  
}
```

java.util.Collections

- static void sort(List elements)
- static void sort(List elements, Comparator c)

$O(n \log n)$  . N  
: elements  
c

- static void shuffle(List elements)
- static void shuffle(List elements , Random r)

.  $O(n \cdot a(n))$   
n a(n)  
: elements  
r

- static Comparator reverseOrder()  
Comparable compareTo

가 .

. 가 , 1024 10  
(linear search) 512 .

Collections binarySearch

, (List )  
Comparable compareTo  
가 .

```
i = Collections.binarySearch(c , element);
i = Collections.binarySearch(c,element,comparator)
```

```

binarySearch(0, element);
    element;

insertionPoint = -i - 1;

-i, 0,

if (i < 0)
    c.add(-i - 1, element);

```

binarySearch 가

---

: , 가

가 binartSearch ,

가 AbstractSequentialList

AbstractSequentialList

binarySearch

가

---

#### java.util.Collections

- static int binarySearch(List elements, Object key)
  - static int binarySearch(List elements, Object key, Comparator c)
- elements 가 AbstractSequentialList

key

$O(a(n)\log n)$  n

a(n)

가 I .

-i-1 .

: elements  
key  
c

Collection 가 .

, , ?

가 가

Collections.max 가

#### java.util.Collections

- static Object min(Collection elements)
- static Object max(Collection elements)
- static Object min(Collection elements , Comparator c)
- static Object max(Collection elements , Comparator c)

가 .

: elements  
c

- static void copy(List to, List from)

가 .

: to  
from

- static void fill(List l, Object value)

: l  
value

- static void copy(List l)

O(n)

n

: 1

( )

가 . 가 ,

JcomboBox 가 .

:

```
void fillComboBox(JcomboBox comboBox, Vector choices)
{
    for(int i=0; i<choices.size(); i++)
        comboBox.addItem(choices.get(i) );
}
```

-

choices . choices 가

가 .

가 가

Collection

가 fillComboBox

```
void fillComboBox(JcomboBox comboBox, Collection choices)
{
    Iteration iter = choices.iteration();
    while(iter.hasNext())
        comboBox.addItem(iter.next() );
}
```

Arrays.asList

-----

:





```

        {
            return new
                AbstractList()
            {
                public Object get(int i)
                {
                    return comboBox.getItemAt(i);
                }
                public int size()
                {
                    return comboBox.getItemCount();
                }
            };
        }
    }

```

가 가

Hashtable Properties , Vector Stack  
BitSet .

## Hashtable

Hashtable HashMap 가  
. Vector Hashtable  
HashMap .

---

: t 가 Hashtable . Windows , HashTable  
가 . Windows

## (Enumeration)

Enumeration .

Enumeration hasMoreElements nextElement 가 .  
 Iterator hashNext next .

, Hashtable elements  
 :

```
Enumeration e = staff.elements();
while(e.hasMoreElements() )
{
    Employee e = (Employee) e.nextElement();
    ...
}
```

Collection.enumeration . ,

```
ArraySet stream = . . . ; // a sequence of input streams
SequenceInputStream in = new SequenceInputStream( Collections.enumeration(streams));
// the SequenceInputStream constructor expects an enumeration
```

-----  
 : C++ .  
 .  
 .  
 .  
 .  
 2  
 가 .  
 -----

java.util.Enumeration

- boolean hasMoreElements()  
 가 true .
- Object nextElement()  
 . hasMoreElements()가 false

## java.util.Hashtable

- Enumeration keys()
- Enumeration elements()

## java.util.Vector

- Enumeration elements()

가 가 .

- 
- 
- 

Properties .

. Unix DOS

가 . PC , AUTOEXEC.BAT 가 :

```
SET PROMPT=$P$G
```

```
SET TEMP=C:\Windows\Temp
```

```
SET CLASSPATH=c:\java\lib; .
```

```
Properties settings = new Properties();
settings.put("PROMPT", "$p$P");
settings.put("TEMP", "C:\\Windows\\Temp");
settings.put("CLASSPATH", "c:\\java\\lib;");
```

store

```
settings.store(System.out, "Environment settings");
```

```
#Environment settings
#Sun Jan 21 07:22:52 1996
CLASSPATH= c:\java\lib;.
TEMP=C:\Windows\Temp
PROMPT=$p$g
```

```
Properties :
System Properties
Properties /
```

```
import java.util.*;
```

```
public class System Info
```

```
{
    public static void main(String args[] )
    {
        Properties  systemProperties= System.getProperties();
        Enumeration enum = systemProperties.propertyNames();
        while(enum.hasMoreElements() )
        {
            String key = (String) enum.nextElement();
            System.out.println(key + "=" +
                               systemProperties.getProperty(key) );
        }
    }
}
```

}

. Properties

.(

.)

```
java.specification.namevalue is Java Platform API Specification
awt.toolkitvalue is sun.awt.windows.WToolkit
java.versionvalue is 1.2
java.awt.graphicsenvvalue is sun.awt.Win32GraphicsEnvironment
user.timezonevalue is Asia/Seoul
java.specification.versionvalue is 1.2
java.vm.vendorvalue is Sun Microsystems Inc.
user.homevalue is C:\WINDOWS
java.vm.specification.versionvalue is 1.0
os.archvalue is x86
java.awt.fontsvalue is
java.vendor.urlvalue is http://java.sun.com/
user.regionvalue is KR
file.encoding.pkgvalue is sun.io
java.homevalue is C:\JDK1.2\JRE
java.class.pathvalue is C:\MyDocuments\corejava;. ;C:\Program
Files\Exceed.95\hcljr.csv.zip;C:\Program Files\Exceed.95;C:\Program
Files\Kawa3.5\kawa\classes.zip;c:\jdk1.2\lib\tools.jar;c:\jdk1.2\jre\li
b\rt.jar
line.separatorvalue is
java.ext.dirsvalue is C:\JDK1.2\JRE\lib\ext
java.io.tmpdirvalue is C:\WINDOWS\TEMP\
os.namevalue is Windows 95
java.vendorvalue is Sun Microsystems Inc.
java.awt.printerjobvalue is sun.awt.windows.WPrinterJob
java.library.pathvalue is
C:\JDK1.2\BIN;. ;C:\WINDOWS\SYSTEM;C:\WINDOWS;c:\jdk1.2\bin;c:\jdk1.2\b
in;C:\WINDOWS;C:\WINDOWS\COMMAND;C:\JDK1.2\BIN
java.vm.specification.vendorvalue is Sun Microsystems Inc.
sun.io.unicode.encodingvalue is UnicodeLittle
file.encodingvalue is MS949
java.specification.vendorvalue is Sun Microsystems Inc.
user.languagevalue is ko
user.namevalue is jhpark
java.vendor.url.bugvalue is http://java.sun.com/cgi-bin/bugreport.cgi
java.vm.namevalue is Classic VM
java.class.versionvalue is 46.0
java.vm.specification.namevalue is Java Virtual Machine Specification
sun.boot.library.pathvalue is C:\JDK1.2\JRE\bin
os.versionvalue is 4.10
java.vm.versionvalue is 1.2
java.vm.infovalue is build JDK-1.2-V, native threads, symcjit
java.compilervalue is symcjit
path.separatorvalue is ;
```

```
file.separatorvalue is \
user.dirvalue is C:\MyDocuments\corejava
sun.boot.class.pathvalue is
C:\JDK1.2\JRE\lib\rt.jar;C:\JDK1.2\JRE\lib\i18n.jar;C:\JDK1.2\JRE\classes
```

```
-----
:
-----
```

```

                                . NotHelloWorld
                                CustomWorld.ini

```

- 
- 
- 
- 
- 

가 , .

Properties 가 가 , , 가

```
String font = settings.getProperty("FONT", "Courier");
```

```

                                "FONT" 가 , font
                                , font "Courier"

```

```

getProperty 가 가 ,
                                (lookup table)

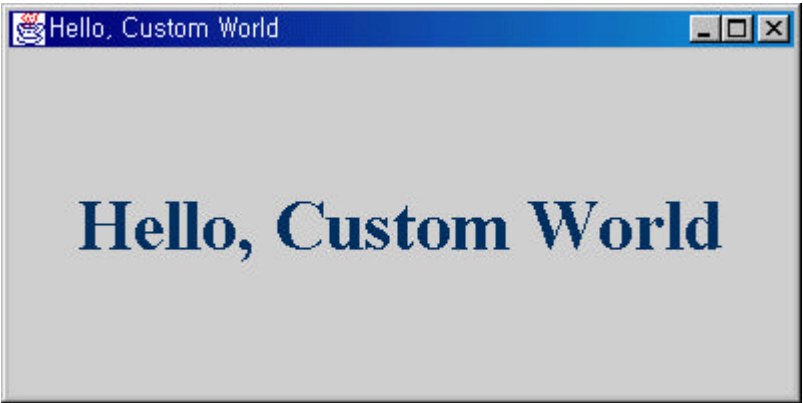
```

```
Properties defaultSettings = new Properties();
defaultSettings.put("FONT", "Courier");
defaultSettings.put("SIZE", "10");
```

```
defaultSettings.put("MESSAGE", "Hello, World");
...
Properties defaultSettings = new Properties(defaultSettings);
FileInputStream sf = new FileInputStream("CustomWorld.ini");
settings.load(sf);
...
```

```
defaultSettings
```

2-12 "Hello, World" .ini ( 2-12 ).



**2-12: Hello World**

```
#Environment settings
#Sun Jan 21 07:22:52 1996
FONT=Times New Roman
SIZE=400 200
MESSAGE=Hello, Custom World
COLOR=0 50 100
PTSIZE=36
```

---

:Properties Hashtable , Hashtable

Properties 가 , size 가  
 . (  
 .) 가 , keys  
 propertyName . put  
 .  
 “is-a” 가?  
 가?  
 가 , private  
 가 .

```
class Properties
{
    public String getProperty(String) { ... }
    public void put(String, String) { ... }
    ...
    private Hashtable nonDefaults;
    private Hashtable defaults;
}
```

Properties

“quick-and-dirty”

---

## 2-6 : CustomWorld.java

```
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import java.io.*;
import javax.swing.*;

public class CustomWorld
{
    public static void main(String[] args)
    {
        JFrame frame = new CustomWorldFrame();
        frame.show();
    }
}
```



```

}

class CustomWorldFrame extends JFrame
{
    public CustomWorldFrame()
    {
        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent e)
            {
                System.exit(0);
            }
        });

        Properties defaultSettings = new Properties();
        defaultSettings.put("FONT", "Monospaced");
        defaultSettings.put("SIZE", "300 200");
        defaultSettings.put("MESSAGE", "Hello, World");
        defaultSettings.put("COLOR", "0 50 50");
        defaultSettings.put("PTSIZE", "12");

        Properties settings = new Properties(defaultSettings);
        try
        {
            FileInputStream sf
                = new FileInputStream("CustomWorld.ini");
            settings.load(sf);
        }
        catch (FileNotFoundException e) {}
        catch (IOException e) {}

        StringTokenizer st = new StringTokenizer
            (settings.getProperty("COLOR"));
        int red = Integer.parseInt(st.nextToken());
        int green = Integer.parseInt(st.nextToken());
        int blue = Integer.parseInt(st.nextToken());

        Color foreground = new Color(red, green, blue);

        String name = settings.getProperty("FONT");
        int size = Integer.parseInt(settings.getProperty("PTSIZE"));
        Font f = new Font(name, Font.BOLD, size);

        st = new StringTokenizer(settings.getProperty("SIZE"));
        int hsize = Integer.parseInt(st.nextToken());
        int vsize = Integer.parseInt(st.nextToken());
        setSize(hsize, vsize);
        setTitle(settings.getProperty("MESSAGE"));

        getContentPane().add(new HelloWorldPanel(getTitle(),
            foreground, f), "Center");
    }
}

class HelloWorldPanel extends JPanel
{
    public HelloWorldPanel(String aMessage, Color aForeground,
        Font aFont)
    {
        message = aMessage;
        foreground = aForeground;
        font = aFont;
    }

    public void paintComponent(Graphics g)

```

```

{
    super.paintComponent(g);
    g.setColor(foreground);
    g.setFont(font);

    FontMetrics fm = g.getFontMetrics(font);
    int w = fm.stringWidth(message);

    Dimension d = getSize();
    int cx = (d.width - w) / 2;
    int cy = (d.height + fm.getHeight()) / 2 - fm.getDescent();

    g.drawString(message, cx, cy);
}

private Color foreground;
private Font font;
private String message;
}

```

#### java.util.Properties

- Properties()
  -
- Properties(Properties defaults)
  - 가
    - : defaults
- String getProperty(String key)
  - - : key
- String getProperty(String key, String defaultValue)
  - 가
    - ,
    - - : key
      - defaultValue 가
- void load(InputStream in) throws IOException
  - InputStream
    - : in

#### java.util.Stack

- void push(Object Item)

● Object pop() : item 가

● Object peek()

BitSet (set .-  
vector array 가 .)

Boolean ArrayList

BitSet , ,

, bucketOfBits BitSet 가

bucketOfBits.get(i);

i 가 (on) true false . 가 ,

bucketOfBits.set(i);

i (on) . ,

bucketOfBits.clear(i);

i (off) .

---

C++ : C++ vector<bool> BitSet 가 .

---

## java.util.BitSet

- `BitSet(int nbits)`

.  
: nbits

- `int length()`

“ ” : 가 + 1

- `boolean get(int bit)`

.  
: bit

- `void set(int bit)`

.  
: bit

- `void clear(int bit)`

.  
: bit

- `void and(BitSet set)`

AND.

: set

- `void or(BitSet set)`

OR.

: set

- `void xor(BitSet set)`

XOR.

: set

- `void andNot(BitSet set)`

.  
: set

**(sieve)**

, (prime number) “ ”  
. ( 2,3, 5 1 . )

가  
가 .(

.)

2 1,000,000 .( 78,498 가  
.)

, 100 가

가 2-7  
2-8 C++ .

---

: 가 가  
-166

96 가 , 98 .

C++ 5.4 : 3750  
JDK1.2.1 : 1640

가 C++ . C++  
bitset .  
가 bitset C++ 1090 .

, .

가 C++  
50% . 가 C++

---

## 2-7: Sieve.java

```
import java.util.*;

public class Sieve
{ public static final boolean PRINT = false;

  public static void main(String[] s)
  { int n = 1000000;
```

```

    long start = System.currentTimeMillis();
    BitSet b = new BitSet(n);
    int count = 0;
    int i;
    for (i = 2; i <= n; i++)
        b.set(i);
    i = 2;
    while (i * i <= n)
    {
        if (b.get(i))
        {
            if (PRINT) System.out.println(i);
            count++;
            int k = 2 * i;
            while (k <= n)
            {
                b.clear(k);
                k += i;
            }
        }
        i++;
    }
    while (i <= n)
    {
        if (b.get(i))
        {
            if (PRINT) System.out.println(i);
            count++;
        }
        i++;
    }
    long end = System.currentTimeMillis();
    System.out.println(count + " primes");
    System.out.println((end - start) + " milliseconds");
}
}

```

## 2-8 : Sieve.cpp

```

#ifndef AVOID_STANDARD_BITSET

#include <bitset>

#else

template<int N>
class bitset
{
public:
    bitset() : bits(new char[(N - 1) / 8 + 1]) {}

    bool test(int n)
    {
        return (bits[n >> 3] & (1 << (n & 7))) != 0;
    }

    void set(int n)
    {
        bits[n >> 3] |= 1 << (n & 7);
    }
}

#endif

```

```

    void reset(int n)
    { bits[n >> 3] &= ~(1 << (n & 7));
    }

private:
    char* bits;
};

#endif

#include <iostream>
#include <ctime>

using namespace std;

int main()
{
    const int N = 1000000;
    clock_t cstart = clock();

    bitset<N + 1> b;
    int count = 0;
    int i;
    for (i = 2; i <= N; i++)
        b.set(i);
    i = 2;
    while (i * i <= N)
    {
        if (b.test(i))
        {
            int k = 2 * i;
            while (k <= N)
            {
                b.reset(k);
                k += i;
            }
        }
        i++;
    }
    for (i = 2; i <= N; i++)
    {
        if (b.test(i))
        {
#ifdef PRINT
            cout << i << "\n";
#endif
            count++;
        }
    }

    clock_t cend = clock();
    double millis = 1000.0
        * (cend - cstart) / CLOCKS_PER_SEC;

    cout << count << " primes\n"
        << millis << " milliseconds\n";

    return 0;
}

```

