

Java Collections – Some Example Programs [with System properties access]

Contains

```
import java.util.Hashtable;

public class Main {
    public static void main(String[] s) {
        Hashtable<String,String> table = new Hashtable<String,String>();
        table.put("key1", "value1");
        table.put("key2", "value2");
        table.put("key3", "value3");

        System.out.println(table.contains("value3"));
    }
}
```

containskey

```
import java.util.Hashtable;

public class Main {
    public static void main(String[] args) {
        Hashtable<String, String> ht = new Hashtable<String, String>();
        ht.put("1", "One");
        ht.put("2", "Two");
        ht.put("3", "Three");

        boolean blnExists = ht.containsKey("2");
        System.out.println("2 exists in Hashtable ? : " + blnExists);
    }
}
```

putAll

```
import java.util.Hashtable;

public class Main{
    public static void main(String[] s) {
        Hashtable<String,String> table = new Hashtable<String,String>();
        table.put("key1", "value1");
        table.put("key2", "value2");
        table.put("key3", "value3");
    }
}
```

```

        Hashtable<String,String> table2 = new Hashtable<String,String>();
        table2.put("key4", "value4");
        table2.put("key5", "value5");
        table2.put("key6", "value6");

        table2.putAll(table);

        System.out.println(table2);
    }
}

```

UsingKeySet

```

/**
 *Output:
 A: 4.34
 E: -9.08
 D: 9.22
 C: 8.0
 B: 3.22

 A's new balance: 1004.34
 */

import java.util.Hashtable;
import java.util.Iterator;
import java.util.Set;

public class MainClass {
    public static void main(String args[]) {
        Hashtable<String, Double> balance = new Hashtable<String, Double>();
;

        String str;
        double bal;

        balance.put("A", 4.34);
        balance.put("B", 3.22);
        balance.put("C", 8.00);
        balance.put("D", 9.22);
        balance.put("E", -9.08);

        Set<String> set = balance.keySet();

        Iterator<String> itr = set.iterator();
        while (itr.hasNext()) {
            str = itr.next();
            System.out.println(str + ": " + balance.get(str));
        }

        System.out.println();
    }
}

```

```

        bal = balance.get("A");
        balance.put("A", bal + 1000);
        System.out.println("A's new balance: " + balance.get("A"));
    }
}

import java.util.*;

public class TestMap {
    public static void main(String[] args) {
        // Create a HashMap
        HashMap hashMap = new HashMap();
        hashMap.put("Smith", new Loan(7, 15, 150000));
        hashMap.put("Anderson", new Loan(7.5, 30, 150000));
        hashMap.put("Lewis", new Loan(7.85, 30, 20000));
        hashMap.put("Cook", new Loan(7, 15, 100000));

        System.out.println("Display entries in HashMap");
        displayMapEntries(hashMap);

        // Create a TreeMap from the previous HashMap
        Map treeMap = new TreeMap(hashMap);
        System.out.println("\nDisplay entries in ascending order of key");
        displayMapEntries(treeMap);

        // Create a LinkedHashMap
        HashMap linkedHashMap = new LinkedHashMap(16, 0.75f, true);
        linkedHashMap.put("Smith", new Loan(7, 15, 150000));
        linkedHashMap.put("Anderson", new Loan(7.5, 30, 150000));
        linkedHashMap.put("Lewis", new Loan(7.85, 30, 20000));
        linkedHashMap.put("Cook", new Loan(7, 15, 100000));

        // Display the loan amount for Gerry K Lewis
        System.out.println("The loan amount for " + "Lewis is " +
            ((Loan)(linkedHashMap.get("Lewis"))).getLoanAmount());

        System.out.println("\nDisplay entries in LinkedHashMap");
        displayMapEntries(linkedHashMap);
    }

    public static void displayMapEntries(Map map) {
        // Get an entry set for the map
        Set entrySet = map.entrySet();

        // Get an iterator for the entry set
        Iterator iterator = entrySet.iterator();

        // Display mappings
        while (iterator.hasNext()) {
            System.out.println(iterator.next());
        }
    }
}

```

Output:

Display entries in HashMap

Cook=Loan@1ddeb3

Smith=Loan@a18aa2

Lewis=Loan@194ca6c

Anderson=Loan@17590db

Display entries in ascending order of key

Anderson=Loan@17590db

Cook=Loan@1ddeb3

Lewis=Loan@194ca6c

Smith=Loan@a18aa2

The loan amount for Lewis is 20000.0

Display entries in LinkedHashMap

Smith=Loan@1f12c4e

Anderson=Loan@93dee9

Cook=Loan@fabe9

Lewis=Loan@df6ccd

System Properties

Java maintains information about your system in the form of a Map of key-value pairs. You can get this Map as Properties instance and then use Map and TreeSet to display values in sorted order.

SysProps.java

```
import java.util.*;

public class SysProps {
    public static void main (String [] args) {

        Properties pr = System.getProperties();

        TreeSet propKeys = new TreeSet(pr.keySet()); // TreeSet sorts keys
        Object prop      = null;

        for (Iterator it = propKeys.iterator(); it.hasNext(); ) {
            prop = it.next(); // save value to use in lookup
            System.out.println(prop + "=" + pr.get(prop) );
        }
    }
}
```

Output:

```
awt.toolkit=sun.awt.windows.WToolkit
file.encoding=Cp1252
file.encoding.pkg=sun.io
file.separator=\
java.awt.graphicsenv=sun.awt.Win32GraphicsEnvironment
java.awt.printerjob=sun.awt.windows.WPrinterJob
java.class.path=.;c:\JMS\JMSeWin\lib\sound.jar;c:\JMS\JMSeWin\lib\jmf.jar;c:\JMS\JMSeWin\lib;. ;C:\j2sdk1.4.1_02\lib;C:\PROGRA~1\JMF21~1.1\lib\sound.jar;C:\junit3.8.1\junit.jar;C:\httpunit-1.5.4\lib\httpunit.jar;C:\httpunit-1.5.4\jars\Tidy.jar;C:\httpunit-1.5.4\jars\js.jar;C:\jakarta-tomcat-4.1.27\common\lib\servlet.jar;c:\fpc\smubrainstem;c:\jakarta-ant-1.5\lib\ant.jar;C:\PROGRA~1\JMF21~1.1\lib\jmf.jar;C:\PROGRA~1\JMF21~1.1\lib;C:\WINDOWS\java\classes;%JAXB_LIBS%\jaxb-api.jar;%JAXB_LIBS%\jaxb-ri.jar;%JAXB_LIBS%\jaxb-xjc.jar;%JAXB_LIBS%\jaxb-libs.jar;%JAXB_LIBS%\endorsed\xercesImpl.jar;%JUNITPERF_HOME%/lib/junitperf.jar;%JAXB_LIBS%\endorsed\xalan.jar;%JAXB_LIBS%\endorsed\sax.jar;%JAXB_LIBS%\endorsed\dom.jar;%JWSDP_LIBS%\jax-qname.jar;%JWSDP_LIBS%\namespace.jar;%JAXB_LIBS%\jaxp-api.jar;c:\WTK20\lib\midpapi.zip
java.class.version=48.0
java.endorsed.dirs=C:\j2sdk1.4.1_02\jre\lib\endorsed
java.ext.dirs=C:\j2sdk1.4.1_02\jre\lib\ext
java.home=C:\j2sdk1.4.1_02\jre
java.io.tmpdir=C:\DOCUME~1\FRANKC~1\LOCALS~1\Temp\
java.library.path=C:\j2sdk1.4.1_02\bin;. ;C:\WINDOWS\System32;C:\WINDOWS;C:\WINDOWS\system32;C:\WINDOWS\WBEM;C:\Perl\bin\;C
```

```
: \WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\Program
Files\Java\jdk1.5.0\bin;c:\j2sdskeel.3.1\bin;c:\jwsdp1.1\jakarta-ant-
1.5.1\bin;C:\PROGRA~1\COMMON~1\Odbc\FILEMA~1;C:\Program
Files\Real\Helix Producer
Basic;C:\NokiaJ2ME\Tools\Nokia_Developers_Suite_for_J2ME\bin;C:\Nokia\U
pdate_Manager\bin;C:\Program Files\CVSNT;C:\Program Files\Ulead
Systems\Ulead DVD MovieFactory 3.5 Suite Deluxe;C:\Program Files\Common
Files\Ulead Systems\MPEG;C:\Program Files\Common Files\Ulead
Systems\DVD;C:\jwsdp-1.1.PURE\jwsdp-shared\bin;C:\jwsdp-1.2\jwsdp-
shared\bin;C:\jwsdp1.1\bin;c:\jakarta-ant-
1.5\bin;C:\NokiaJ2ME\Tools\Nokia_Developers_Suite_for_J2ME\bin;C:\Nokia
\Update_Manager\bin;c:\jwsdp1.1\jwsdp-shared\bin;c:\jwsdp1.1\jaxrpc-
1.0.3\bin;C:\Program Files\TextPad 4
java.runtime.name=Java(TM) 2 Runtime Environment, Standard Edition
java.runtime.version=1.4.1_02-b06
java.specification.name=Java Platform API Specification
java.specification.vendor=Sun Microsystems Inc.
java.specification.version=1.4
java.util.prefs.PreferencesFactory=java.util.prefs.WindowsPreferencesFa
ctory
java.vendor=Sun Microsystems Inc.
java.vendor.url=http://java.sun.com/
java.vendor.url.bug=http://java.sun.com/cgi-bin/bugreport.cgi
java.version=1.4.1_02
java.vm.info=mixed mode
java.vm.name=Java HotSpot(TM) Client VM
java.vm.specification.name=Java Virtual Machine Specification
java.vm.specification.vendor=Sun Microsystems Inc.
java.vm.specification.version=1.0
java.vm.vendor=Sun Microsystems Inc.
java.vm.version=1.4.1_02-b06
line.separator=

os.arch=x86
os.name=Windows XP
os.version=5.1
path.separator=;
sun.arch.data.model=32
sun.boot.class.path=C:\j2sdk1.4.1_02\jre\lib\endorsed\dom.jar;C:\j2sdk1
.4.1_02\jre\lib\endorsed\sax.jar;C:\j2sdk1.4.1_02\jre\lib\endorsed\xala
n.jar;C:\j2sdk1.4.1_02\jre\lib\endorsed\xercesImpl.jar;C:\j2sdk1.4.1_02
\jre\lib\endorsed\xsltc.jar;C:\j2sdk1.4.1_02\jre\lib\rt.jar;C:\j2sdk1.4
.1_02\jre\lib\i18n.jar;C:\j2sdk1.4.1_02\jre\lib\sunrsasign.jar;C:\j2sdk
1.4.1_02\jre\lib\jsse.jar;C:\j2sdk1.4.1_02\jre\lib\jce.jar;C:\j2sdk1.4.
1_02\jre\lib\charsets.jar;C:\j2sdk1.4.1_02\jre\classes
sun.boot.library.path=C:\j2sdk1.4.1_02\jre\bin
sun.cpu.endian=little
sun.cpu.isalist=pentium i486 i386
sun.io.unicode.encoding=UnicodeLittle
sun.java2d.fontpath=
sun.os.patch.level=
user.country=US
user.dir=C:\Documents and Settings\frank coyle\My
Documents\7346.DistributedJava\Assignments.7346.Spr.04\A2.Timing Collec
tions
user.home=C:\Documents and Settings\frank coyle
user.language=en
```

```
user.name=frank coyle
user.timezone=
user.variant=
```

Tool completed successfully

Counting with HashMap

Counts words: StringTokenizer, HashMap

```
import java.util.*;

public class CountOccurrenceOfWords {
    public static void main(String[] args) {
        // Text in a string
        String text = "This year people will realize the more is not" +
            "necessarily better and will start to try to filter their" +
            "lives. RSS and other syndication formats continue their " +
            "march forward, but aggregators will become a way of life, " +
            "and better aggregators appear on the scene, providing " +
            "better control and help with information overload." ;

        // Create a hash map to hold words and key and count as value
        HashMap hashMap = new HashMap();

        StringTokenizer st = new StringTokenizer(text, " .!?");
        while (st.hasMoreTokens()) {
            String key = st.nextToken();

            if (hashMap.get(key) != null) {
                int value = ((Integer)hashMap.get(key)).intValue();
                value++;
                hashMap.put(key, new Integer(value));
            }
            else {
                hashMap.put(key, new Integer(1));
            }
        }

        // Create a tree map from the hash map
        TreeMap treeMap = new TreeMap(hashMap);

        // Get an entry set for the tree map
        Set entrySet = treeMap.entrySet();

        // Get an iterator for the entry set
        Iterator iterator = entrySet.iterator();

        // Display mappings
        System.out.println("Display words and their count in " +
```

```
        "ascending order of the words");

    // iterate over the collection
    while (iterator.hasNext()) {
        System.out.println(iterator.next());
    }
}
}
```

Output:

Display words and their count in ascending order of the words

```
Have=4
a=3
class=1
day=1
fun=1
good=3
visit=1
```