

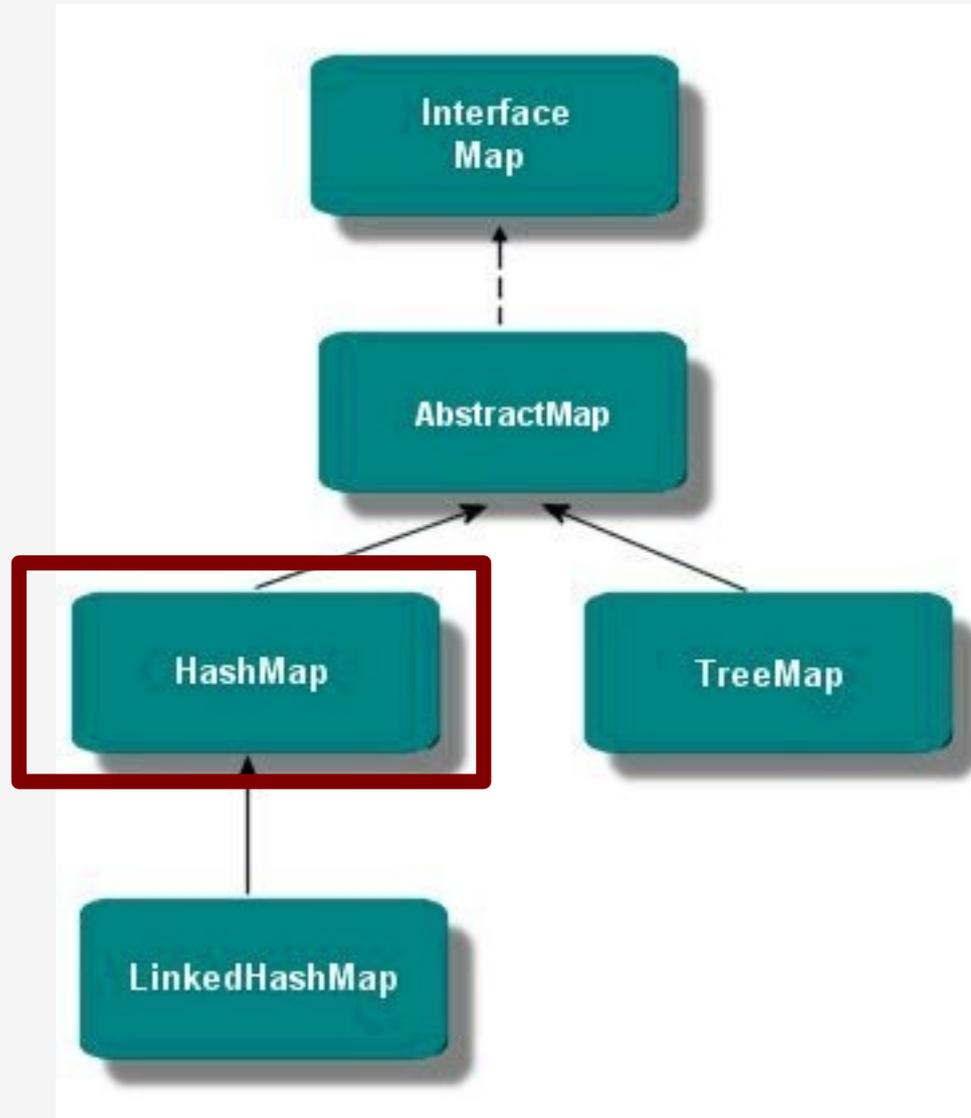


`java.util.HashMap`  
`java.util.Properties`

# java.util.HashMap<K,V>

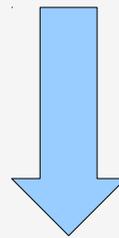
- Tableau associatif (implémenté en table de hachage)
- Permet d'associer des clés de type K à des valeurs de types V
- Implémente l'interface java.util.Map

# hiérarchie



# exemple

```
public static void main(String[] args) {  
  
    HashMap<String, String> contacts = new HashMap<String, String>();  
  
    contacts.put("nom", "06 0000 0000");  
    contacts.put("nom2", "09 0000 0000");  
    contacts.put("nom3", "01 0000 0000");  
  
    System.out.println(contacts);  
  
}
```

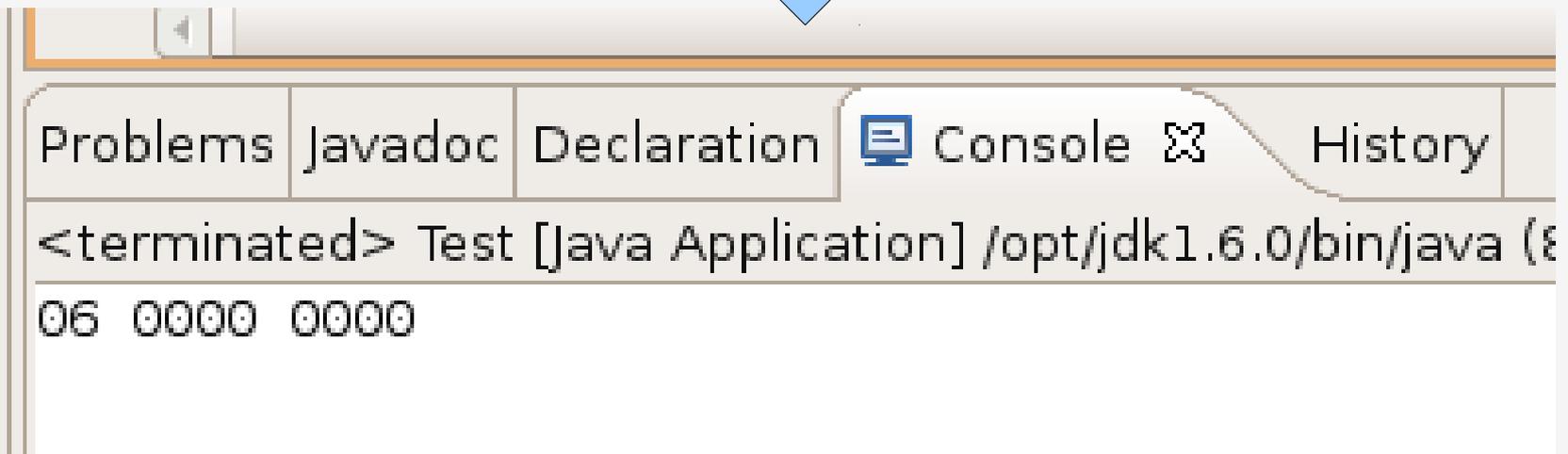
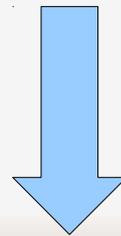


Problems javadoc Declaration  Console  History

```
<terminated> Test [Java Application] /opt/jdk1.6.0/bin/java (8 févr. 07 06:03:02)  
{nom3=01 0000 0000, nom2=09 0000 0000, nom=06 0000 0000}
```

# exemple

```
public static void main(String[] args) {  
  
    HashMap<String, String> contacts = new HashMap<String, String>();  
  
    contacts.put("nom", "06 0000 0000");  
    contacts.put("nom2", "09 0000 0000");  
    contacts.put("nom3", "01 0000 0000");  
  
    System.out.println(contacts.get("nom"));  
  
}
```



# Method Summary

|  |   |
|--|---|
| void   | <a href="#">clear()</a><br>Removes all mappings from this map.  |
| <a href="#">Object</a>                           | <a href="#">clone()</a><br>Returns a shallow copy of this <code>HashMap</code> instance: the keys and values themselves are not cloned.   |
| boolean  | <a href="#">containsKey(Object key)</a><br>Returns <code>true</code> if this map contains a mapping for the specified key.  |
| boolean  | <a href="#">containsValue(Object value)</a><br>Returns <code>true</code> if this map maps one or more keys to the specified value.  |
| <a href="#">Set&lt;Map.Entry&lt;K, V&gt;&gt;</a> | <a href="#">entrySet()</a><br>Returns a collection view of the mappings contained in this map.  |
| <a href="#">V</a>                                | <a href="#">get(Object key)</a><br>Returns the value to which the specified key is mapped in this identity hash map, or <code>null</code> if the map contains no mapping for this key.  |
| boolean  | <a href="#">isEmpty()</a><br>Returns <code>true</code> if this map contains no key-value mappings.  |
| <a href="#">Set&lt;K&gt;</a>                     | <a href="#">keySet()</a><br>Returns a set view of the keys contained in this map.   |
| <a href="#">V</a>                                | <a href="#">put(K key, V value)</a><br>Associates the specified value with the specified key in this map.   |
| void   | <a href="#">putAll(Map&lt;? extends K, ? extends V&gt; m)</a><br>Copies all of the mappings from the specified map to this map These mappings will replace any mappings that this map had for any of the keys currently in the specified map. |
| <a href="#">V</a>                                | <a href="#">remove(Object key)</a><br>Removes the mapping for this key from this map if present.  |
| int  | <a href="#">size()</a><br>Returns the number of key-value mappings in this map.   |
| <a href="#">Collection&lt;V&gt;</a>              | <a href="#">values()</a><br>Returns a collection view of the values contained in this map.  |

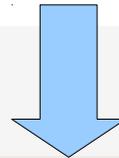
# Parcours d'une HashMap

```
HashMap<String, String> contacts = new HashMap<String, String>();

contacts.put("nom", "06 0000 0000");
contacts.put("nom2", "09 0000 0000");
contacts.put("nom3", "01 0000 0000");

for (Iterator<String> noms = contacts.keySet().iterator(); noms.hasNext();) {
    System.out.println(noms.next());
}

for (String number : contacts.values()) {
    System.out.println(number);
}
```



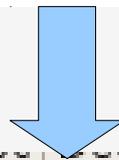
```
Problems | javadoc | Declaration | Console | History
<terminated> Test [Java Application] /opt/jdk1.6.0/bin/java (
nom3
nom2
nom
01 0000 0000
09 0000 0000
06 0000 0000
```

# Method Summary

|  |  |
|--|--|
| void   | <a href="#">clear()</a><br>Removes all mappings from this map.   |
| <a href="#">Object</a>                           | <a href="#">clone()</a><br>Returns a shallow copy of this <code>HashMap</code> instance: the keys and values themselves are not cloned.  |
| boolean  | <a href="#">containsKey(Object key)</a><br>Returns <code>true</code> if this map contains a mapping for the specified key.   |
| boolean  | <a href="#">containsValue(Object value)</a><br>Returns <code>true</code> if this map maps one or more keys to the specified value.   |
| <a href="#">Set&lt;Map.Entry&lt;K, V&gt;&gt;</a> | <a href="#">entrySet()</a><br>Returns a collection view of the mappings contained in this map.   |
| <a href="#">V</a>                                | <a href="#">get(Object key)</a><br>Returns the value to which the specified key is mapped in this identity hash map, or <code>null</code> if the map contains no mapping for this key.   |
| boolean  | <a href="#">isEmpty()</a><br>Returns <code>true</code> if this map contains no key-value mappings.   |
| <a href="#">Set&lt;K&gt;</a>                     | <a href="#">keySet()</a><br>Returns a set view of the keys contained in this map.  |
| <a href="#">V</a>                                | <a href="#">put(K key, V value)</a><br>Associates the specified value with the specified key in this map.  |
| void   | <a href="#">putAll(Map&lt;? extends K, ? extends V&gt; m)</a><br>Copies all of the mappings from the specified map to this map. These mappings will replace any mappings that this map had for any of the keys currently in the specified map. |
| <a href="#">V</a>                                | <a href="#">remove(Object key)</a><br>Removes the mapping for this key from this map if present.   |
| int  | <a href="#">size()</a><br>Returns the number of key-value mappings in this map.  |
| <a href="#">Collection&lt;V&gt;</a>              | <a href="#">values()</a><br>Returns a collection view of the values contained in this map.   |

# Parcours d'une HashMap

```
HashMap<String, String> contacts = new HashMap<String, String>();  
  
contacts.put("nom", "06 0000 0000");  
contacts.put("nom2", "09 0000 0000");  
contacts.put("nom3", "01 0000 0000");  
  
for (Map.Entry<String, String> entree : contacts.entrySet()) {  
    System.out.println(entree.getKey());  
    System.out.println(entree.getValue());  
    System.out.println();  
}
```



```
<terminated> Test [Java Application] /opt  
nom3  
01 0000 0000  
  
nom2  
09 0000 0000  
  
nom  
06 0000 0000
```

# java.util.Properties

- Sous classe de `java.util.Hashtable<Object, Object>`
- Permet de gérer des propriétés gérées sous forme clé -> valeur avec `String -> String`
- Possède des méthodes pour enregistrer les propriétés dans un fichier.
  - Permet par exemple d'appliquer une règle de programmation simple : « *ne pas coder en dur les paramètres* » de l'application.
- Certains paramètres de la JVM sont stockés dans un objet `Properties` : « `System.getProperties()` »
- Construction `Properties props = new Properties();`

# Method Summary

|                                 |   |
|---------------------------------|---|
| <a href="#">String</a>          | <a href="#">getProperty</a> ( <a href="#">String</a> key)<br>Searches for the property with the specified key in this property list.  |
| <a href="#">String</a>          | <a href="#">getProperty</a> ( <a href="#">String</a> key, <a href="#">String</a> defaultValue)<br>Searches for the property with the specified key in this property list.   |
| void                            | <a href="#">list</a> ( <a href="#">PrintStream</a> out)<br>Prints this property list out to the specified output stream.  |
| void                            | <a href="#">list</a> ( <a href="#">PrintWriter</a> out)<br>Prints this property list out to the specified output stream.  |
| void                            | <a href="#">load</a> ( <a href="#">InputStream</a> inStream)<br>Reads a property list (key and element pairs) from the input stream.  |
| void                            | <a href="#">loadFromXML</a> ( <a href="#">InputStream</a> in)<br>Loads all of the properties represented by the XML document on the specified input stream into this properties table.  |
| <a href="#">Enumeration</a> <?> | <a href="#">propertyNames</a> ()<br>Returns an enumeration of all the keys in this property list, including distinct keys in the default property list if a key of the same name has not already been found from the main properties list.  |
| void                            | <a href="#">save</a> ( <a href="#">OutputStream</a> out, <a href="#">String</a> comments)<br><b>Deprecated.</b> <i>This method does not throw an IOException if an I/O error occurs while saving the property list. The preferred way to save a properties list is via the <code>store(OutputStream out, String comments)</code> method or the <code>storeToXML(OutputStream os, String comment)</code> method.</i> |
| <a href="#">Object</a>          | <a href="#">setProperty</a> ( <a href="#">String</a> key, <a href="#">String</a> value)<br>Calls the Hashtable method put.  |
| void                            | <a href="#">store</a> ( <a href="#">OutputStream</a> out, <a href="#">String</a> comments)<br>Writes this property list (key and element pairs) in this Properties table to the output stream in a format suitable for loading into a Properties table using the <a href="#">load</a> method.   |
| void                            | <a href="#">storeToXML</a> ( <a href="#">OutputStream</a> os, <a href="#">String</a> comment)<br>Emits an XML document representing all of the properties contained in this table.  |
| void                            | <a href="#">storeToXML</a> ( <a href="#">OutputStream</a> os, <a href="#">String</a> comment, <a href="#">String</a> encoding)<br>Emits an XML document representing all of the properties contained in this table, using the specified encoding.   |

# Enregistrement : *store(OutputStream, String)*

```
public static void exempleDeSauvegarde() {  
    //Emplacement où sera stocké le fichier  
    String propertiesFileLocation = "myProperties.properties";  
  
    Properties myProps = new Properties();  
  
    myProps.setProperty("user.name", "hobby1");  
    myProps.setProperty("java.ide", "Eclipse");  
    myProps.setProperty("qqch", "valeur");  
  
    try (OutputStream out = new FileOutputStream(propertiesFileLocation)){  
        myProps.store(out, "un fichier properties de test");  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

# fichier résultat

 myProperties.properties 

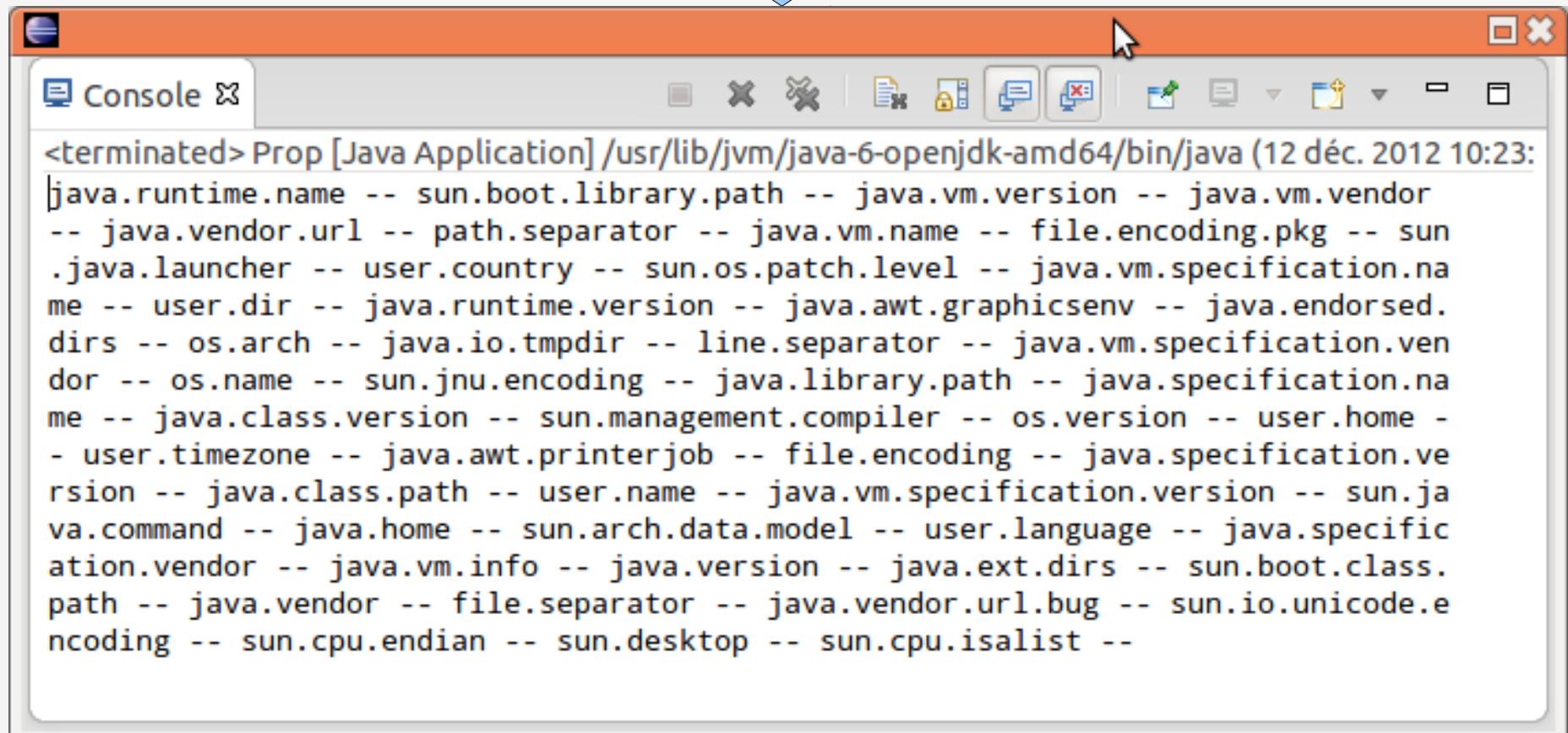
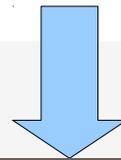
```
#un fichier properties de test  
#Mon Jan 18 16:43:14 CET 2016  
user.name=hobby1  
qqch=valeur  
java.ide=Eclipse
```

# lecture du fichier *load(InputStream)*

```
public static void exempleDeChargement() {  
    //Emplacement où sera stocké le fichier  
    String propertiesFileLocation = "myProperties.properties";  
  
    Properties myProps = new Properties();  
  
    try (InputStream in = new FileInputStream(propertiesFileLocation)){  
        myProps.load(in);  
    } catch (IOException e1) {  
        e1.printStackTrace();  
    }  
}
```

# Les propriétés de la JVM

```
myProps = System.getProperties();  
  
for (Object string : myProps.keySet()) {  
    System.out.print(string+" -- ");  
}
```



```
<terminated> Prop [Java Application] /usr/lib/jvm/java-6-openjdk-amd64/bin/java (12 déc. 2012 10:23:  
|java.runtime.name -- sun.boot.library.path -- java.vm.version -- java.vm.vendor  
-- java.vendor.url -- path.separator -- java.vm.name -- file.encoding.pkg -- sun  
.java.launcher -- user.country -- sun.os.patch.level -- java.vm.specification.na  
me -- user.dir -- java.runtime.version -- java.awt.graphicsenv -- java.endorsed.  
dirs -- os.arch -- java.io.tmpdir -- line.separator -- java.vm.specification.ven  
dor -- os.name -- sun.jnu.encoding -- java.library.path -- java.specification.na  
me -- java.class.version -- sun.management.compiler -- os.version -- user.home -  
- user.timezone -- java.awt.printerjob -- file.encoding -- java.specification.ve  
rsion -- java.class.path -- user.name -- java.vm.specification.version -- sun.ja  
va.command -- java.home -- sun.arch.data.model -- user.language -- java.specific  
ation.vendor -- java.vm.info -- java.version -- java.ext.dirs -- sun.boot.class.  
path -- java.vendor -- file.separator -- java.vendor.url.bug -- sun.io.unicode.e  
ncoding -- sun.cpu.endian -- sun.desktop -- sun.cpu.isalist --
```