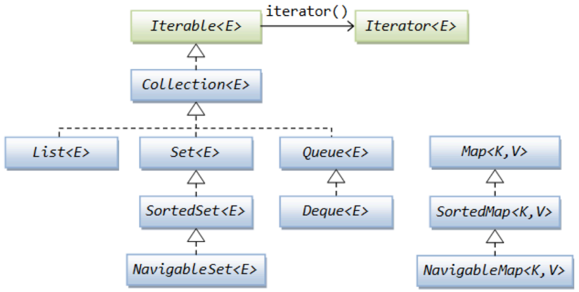
More Sophisticated Behaviour

Technical Support System V3.0

Produced by: Dr. Siobhán Drohan
Mr. Colm Dunphy
Mr. Diarmuid O’Connor
Dr. Frank Walsh

Java Collections Framework:



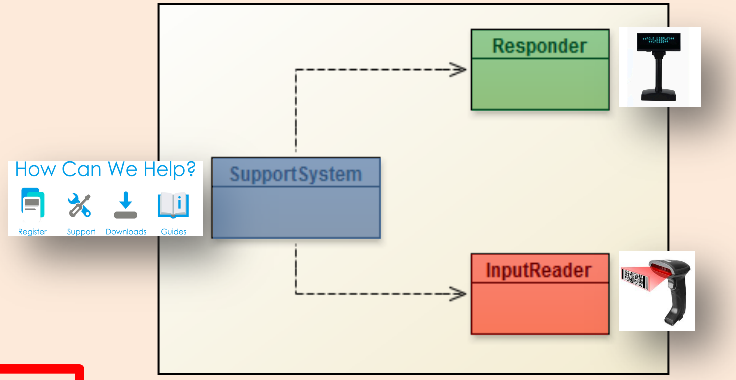
Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/

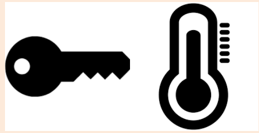
Topic List

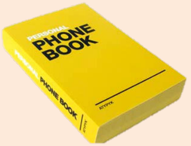
1. Recap: Technical Support System **V2**

2. Technical Support System **V3**
   - Overview
     - 3 classes:
       - **Responder**
       - **InputReader**
       - **SupportSystem**



3. Class Development
   - **Responder** class
     - Generating a related response
     - ArrayList
     - Map and **HashMap**





   - **InputReader** class
     - Tokenizing Strings
     - Set and **HashSet**

   - Responder class
     - Finishing the class

   - **SupportSystem** class
     - A small change.

Maps: (key=value) pairs

- Maps are collections
  - that contain pairs of values.

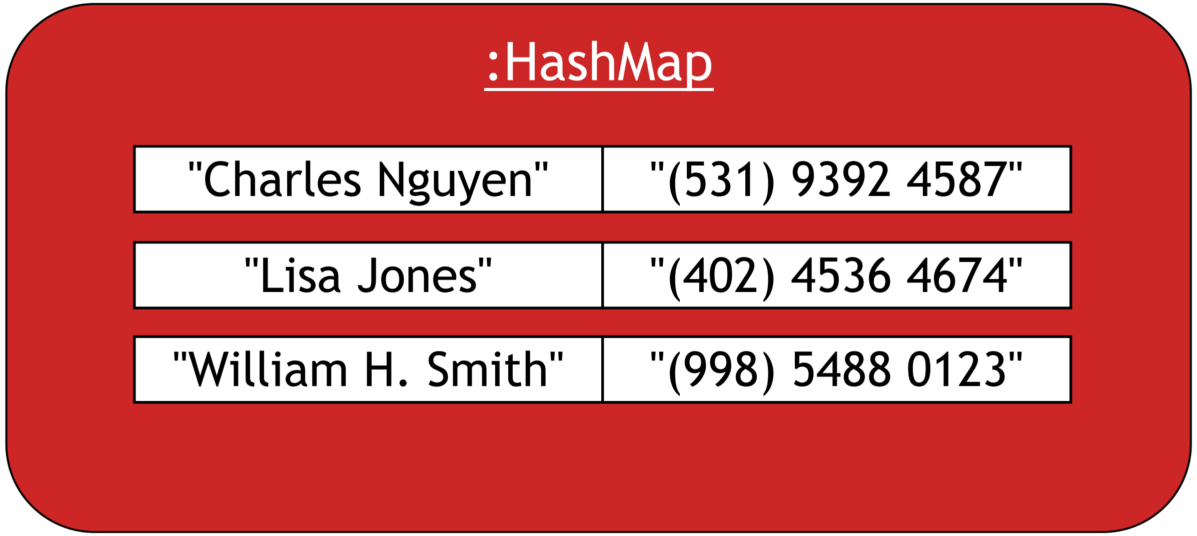
- **Pairs** consist of :
  - **key** 🔑
  - **value**. 🌡

Key	Value
Word	Response

- **Lookup** works by supplying a key, and retrieving a value.
  - E.g. telephone book
    - use the **name** to look up a **phone number.**

Using Maps

- A **MAP** with String keys & String values.



:HashMap

"Charles Nguyen"	"(531) 9392 4587"
"Lisa Jones"	"(402) 4536 4674"
"William H. Smith"	"(998) 5488 0123"

ArrayList **Vs** Map

ArrayList

1. each entry stores
   one object

2. you use an **integer index**
   to **lookup** the object

Map

1. each entry has a
   pair of objects (key=value).

2. you use the **key object**
   to **lookup** the value object

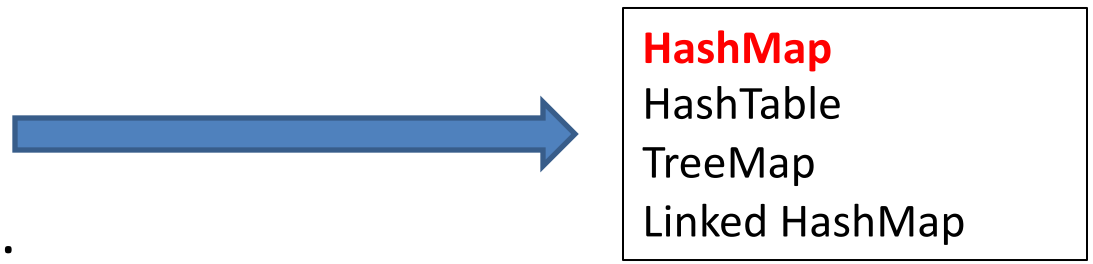
More on **Map**

- Maps are **ideal for one-way lookup using the key**.

- Using Maps to Look up a value associated with a key is easy!
  - However, **reverse lookup** is not so easy (finding a key for a value).
    - E.g. looking up a number in the phonebook, to find the persons name

- A map cannot contain duplicate keys;
  - **A key** can map to **at most one value**.

- Java provides 4 **Map** classes:
  - We will use the **HashMap** class.

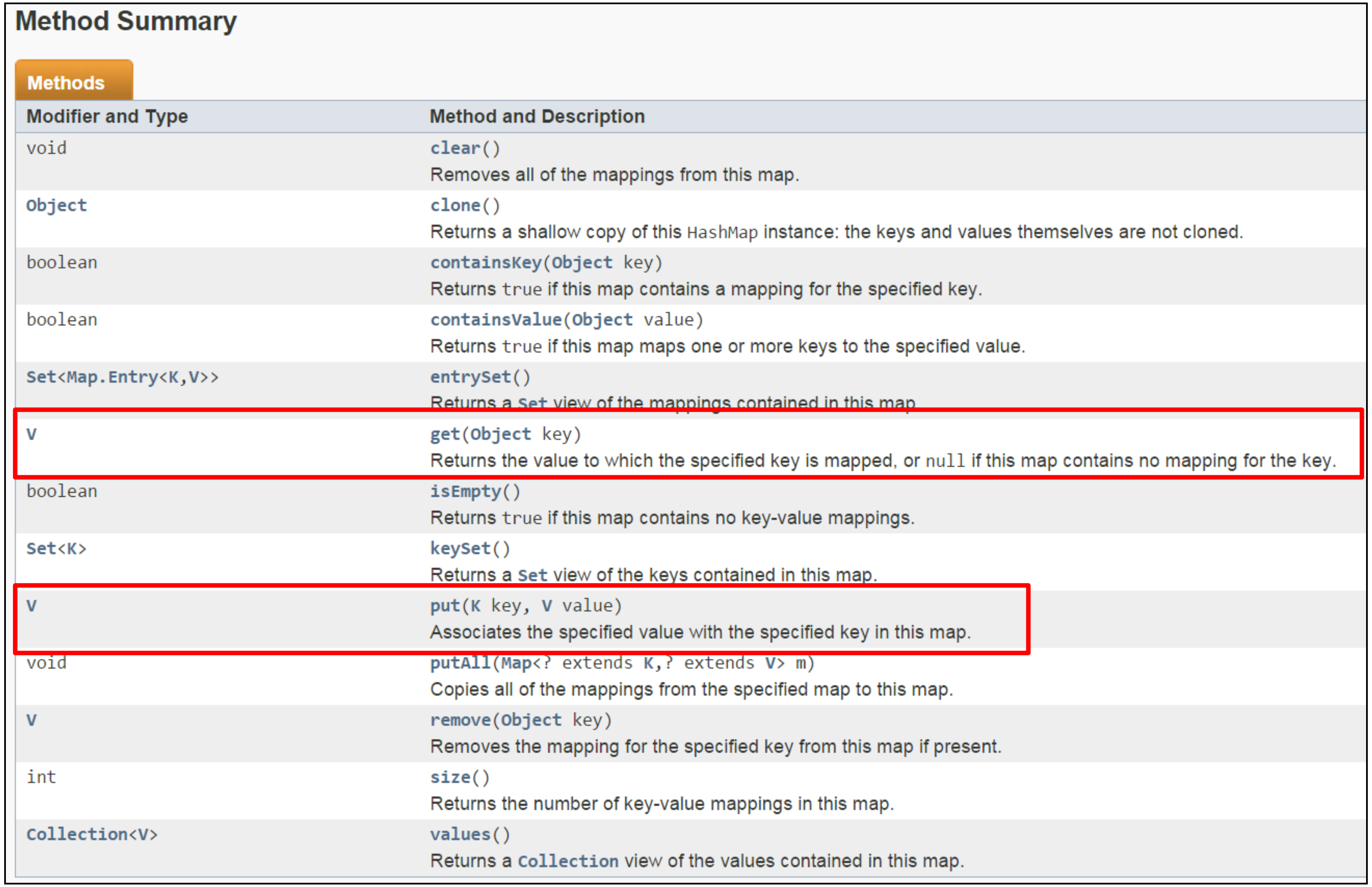


HashMap
HashTable
TreeMap
Linked HashMap

HashMap Methods

java.util

Class HashMap<K,V>



Method Summary

Methods

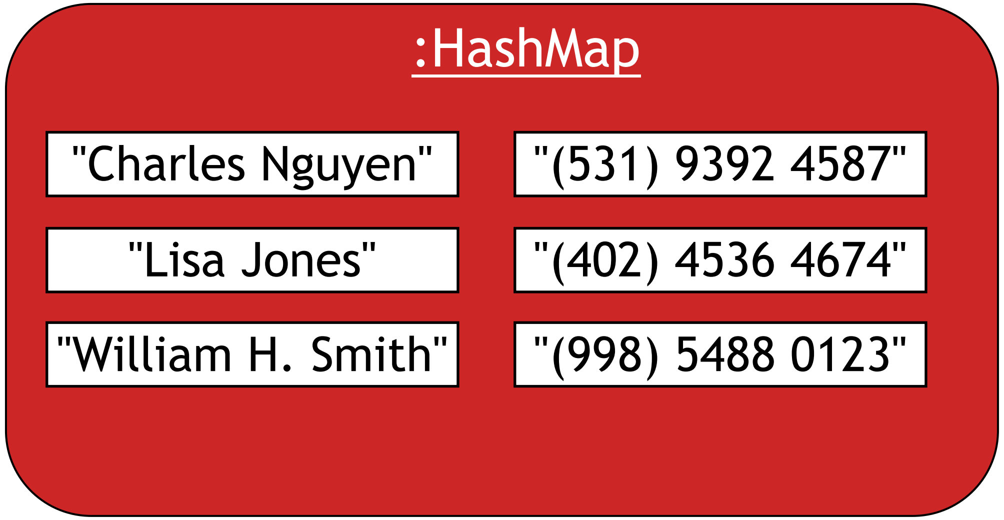
Modifier and Type	Method and Description
void	clear() Removes all of the mappings from this map.
Object	clone() Returns a shallow copy of this HashMap instance: the keys and values themselves are not cloned.
boolean	containsKey(Object key) Returns true if this map contains a mapping for the specified key.
boolean	containsValue(Object value) Returns true if this map maps one or more keys to the specified value.
Set<Map.Entry<K,V>>	entrySet() Returns a Set view of the mappings contained in this map.
V	get(Object key) Returns the value to which the specified key is mapped, or null if this map contains no mapping for the key.
boolean	isEmpty() Returns true if this map contains no key-value mappings.
Set<K>	keySet() Returns a Set view of the keys contained in this map.
V	put(K key, V value) Associates the specified value with the specified key in this map.
void	putAll(Map<? extends K,? extends V> m) Copies all of the mappings from the specified map to this map.
V	remove(Object key) Removes the mapping for the specified key from this map if present.
int	size() Returns the number of key-value mappings in this map.
Collection<V>	values() Returns a Collection view of the values contained in this map.

Using **HashMap**

```
HashMap <String, String> phoneBook = new HashMap<String, String>();
                                     // phoneBook is a hashmap of pairs of String objects.
phoneBook.put("Charles Nguyen", "(531) 9392 4587");
phoneBook.put("Lisa Jones", "(402) 4536 4674");
phoneBook.put("William H. Smith", "(998) 5488 0123");

String phoneNumber = phoneBook.get("Lisa Jones");
System.out.println(phoneNumber);
```

Lookup



:HashMap

"Charles Nguyen"	"(531) 9392 4587"
"Lisa Jones"	"(402) 4536 4674"
"William H. Smith"	"(998) 5488 0123"

```
Console Output:

(402) 4536 4674
```

HashMap in Tech Support System **V3**

In the **Responder** class,

we will now use **HashMap** to store **“Key-Value” pairs**

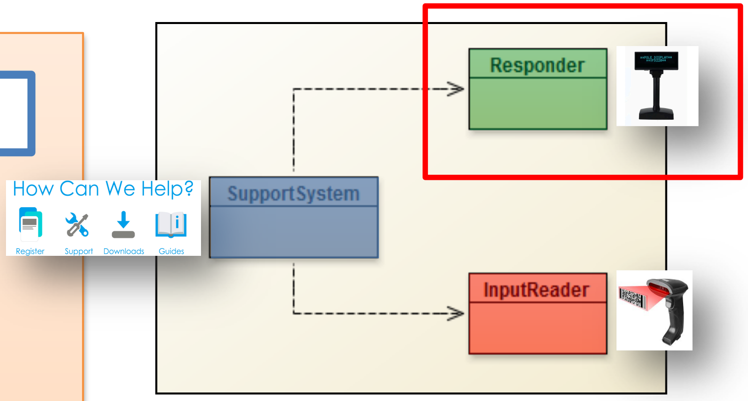
for context-sensitive responses e.g.

Key	Value
windows	This is a known bug to do with the Windows operating system. Please report it to Microsoft. There is nothing we can do about this.
slow	I think this has to do with your hardware. Upgrading your processor should solve all performance problems. Have you got a problem with our software?
bug	Well, you know, all software has some bugs. But our software engineers are working very hard to fix them. Can you describe the problem a bit further?
performance	Performance was quite adequate in all our tests. Are you running any other processes in the background?

```
private void fillResponseMap()
 {
    responseMap.put("crash",
                "Well, it never crashes on our system. It must have something\n" +
                "to do with your system. Tell me more about your configuration.");

    responseMap.put("crashes",
                "Well, it never crashes on our system. It must have something\n" +
                "to do with your system. Tell me more about your configuration.");
    responseMap.put("slow",
                "I think this has to do with your hardware. Upgrading your processor\n" +
                "should solve all performance problems. Have you got a problem with\n" +
                "our software?");
    responseMap.put("performance",
                "Performance was quite adequate in all our tests. Are you running\n" +
                "any other processes in the background?");
    responseMap.put("bug",
                "Well, you know, all software has some bugs. But our software engineers\n" +
                "are working very hard to fix them. Can you describe the problem a bit\n" +
                "further?");
    responseMap.put("buggy",
                "Well, you know, all software has some bugs. But our software engineers\n" +
                "are working very hard to fix them. Can you describe the problem a bit\n" +
                "further?");
    responseMap.put("windows",
                "This is a known bug to do with the Windows operating system. Please\n" +
                "report it to Microsoft. There is nothing we can do about this.");
    // and so on...
}
```

```
private HashMap<String, String> responseMap;
```



V3.0 Responder changes
(in red)

fillResponseMap()

```
responseMap.put (
          "crashes",

          "Well, it never crashes on our system. It must have something\n"
          + "to do with your system. Tell me more about your configuration.");
```

- Whenever someone enters the word “**crashes**”,
  - we can do a **lookup** and print the attached **response**.

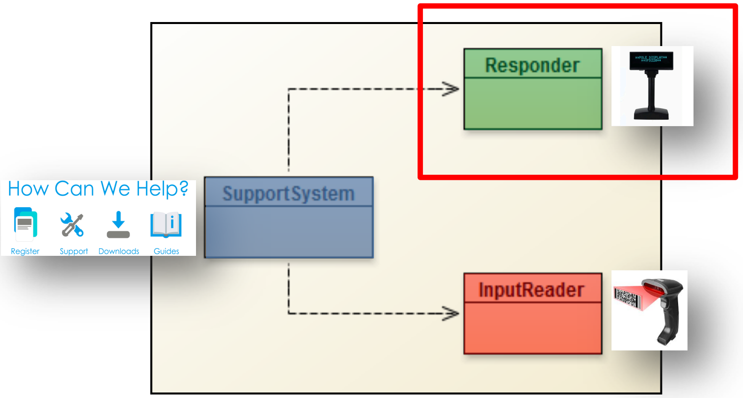
```
import java.util.HashMap;
import java.util.ArrayList;
import java.util.Random;

public class Responder
{
   // Used to map key words to responses.
   private HashMap<String, String> responseMap;

   // Default responses to use if we don't recognise a word.
   private ArrayList<String> defaultResponses;

   // For random responses
   private Random randomGenerator;

  public Responder()
   {
      responseMap = new HashMap<String, String>();
      fillResponseMap();
      defaultResponses = new ArrayList<String>();
      fillDefaultResponses();
      randomGenerator = new Random();
   }
```

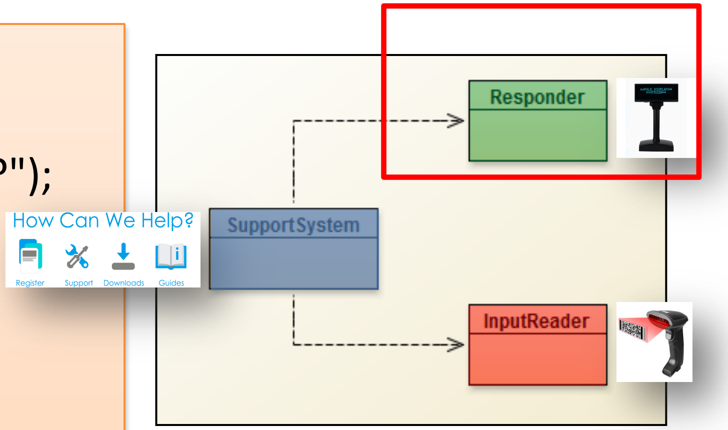


V3.0 Responder changes
(in red)

```
private void fillDefaultResponses() {

    defaultResponses.add("That sounds odd. Could you describe that problem in more detail?");
    defaultResponses.add("No other customer has ever complained about this before. \n" +
                         "What is your system configuration?");
    defaultResponses.add("That sounds interesting. Tell me more...");
    defaultResponses.add("I need a bit more information on that.");
    defaultResponses.add("Have you checked that you do not have a dll conflict?");
    defaultResponses.add("That is explained in the manual. Have you read the manual?");
    defaultResponses.add("Your description is a bit wishy-washy. Have you got an expert\n" +
                         "there with you who could describe this more precisely?");
    defaultResponses.add("That's not a bug, it's a feature!");
    defaultResponses.add("Could you elaborate on that?");
}

private String pickDefaultResponse()
{
    // Pick a random number for the index in the default response list.
    // The number will be between 0 (inclusive) and the size of the list (exclusive).
    int index = randomGenerator.nextInt( defaultResponses.size() );
    return defaultResponses.get(index);
}
```



V3.0 Responder changes **(in red)**

For what a default random response

Next we look at the context sensitive response

Any Questions?