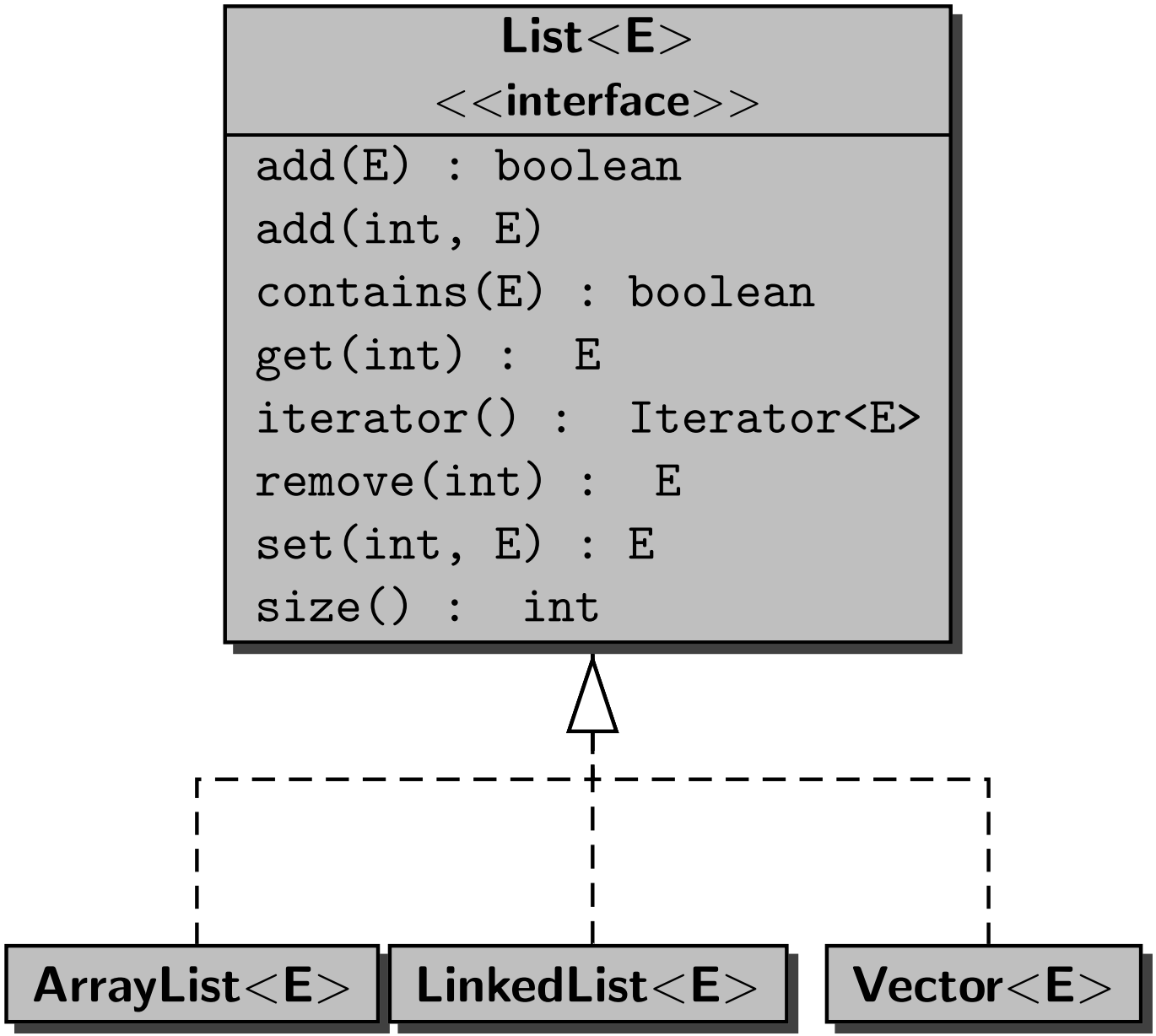
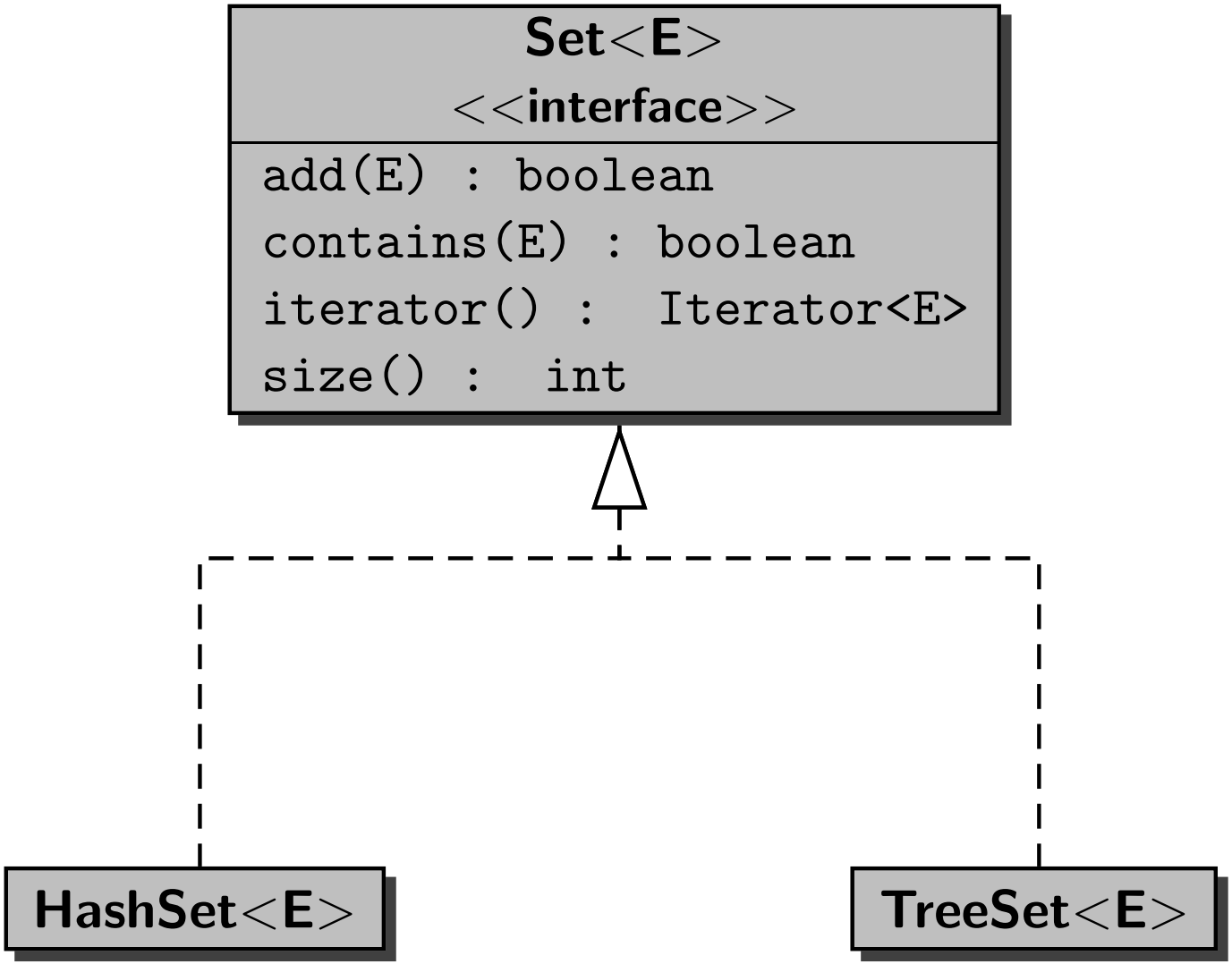
List


List<E>
<<interface>>
add(E) : boolean
add(int, E)
contains(E) : boolean
get(int) : E
iterator() : Iterator<E>
remove(int) : E
set(int, E) : E
size() : int
ArrayList<E>
LinkedList<E>
Vector<E>

Set


Set<E>
<<interface>>
add(E) : boolean
contains(E) : boolean
iterator() : Iterator<E>
size() : int
HashSet<E>
TreeSet<E>

Java Language Specification

Question

Prompt the user for a word using the prompt `Enter word:` and determine if that word occurs in the "Java Language Specification"?

Print the following: `The word "..." does not occur/occurs in the Java Language Specification.`

Remove the duplicates from the collection.

Answer

```
List<String> book = new ArrayList<String>();
...
Set<String> words = new TreeSet<String>(book);
boolean found = words.contains(search);
```

Answer

```
List<String> book = new ArrayList<String>();
...
Set<String> words = new TreeSet<String>(book);
boolean found = words.contains(search);
```

Question

Why can we use the constructor
`TreeSet(Collection<? extends E>)`
of the class `TreeSet<E>`?

Answer

```
List<String> book = new ArrayList<String>();
...
Set<String> words = new TreeSet<String>(book);
boolean found = words.contains(search);
```

Question

Why can we use the constructor
`TreeSet(Collection<? extends E>)`
of the class `TreeSet<E>`?

Answer

Because `Collection<E>` is a superinterface of `List<E>`.

Answer

```
Set<String> words = new TreeSet<String>(book);
boolean found = words.contains(search);
```

Java Language Specification

Answer

```
Set<String> words = new TreeSet<String>(book);
boolean found = words.contains(search);
```

Question

What is the time cost of `contains`?

Answer

```
Set<String> words = new TreeSet<String>(book);
boolean found = words.contains(search);
```

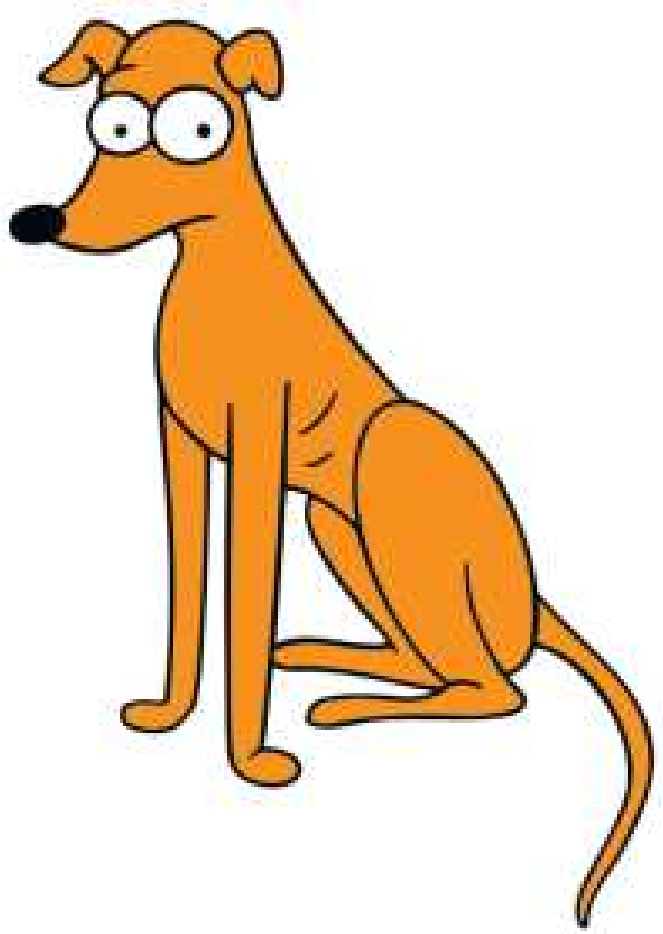
Question

What is the time cost of `contains`?

Answer

During early binding, the invocation `words.contains(search)` is bound to the method `contains(String)` of the interface `Set<String>`. During late binding, the invocation `words.contains(search)` is bound to the method `contains(String)` of the class `TreeSet<String>`. According to the API of the `TreeSet` class, the time cost of `contains` is $O(\log(n))$ where n is the size of the set.

Santa's Little Helper



Santa's Little Helper

Since the number of children in the world is increasing every year, Santa is in need of an app to maintain his list. Can you help him?

Numerous elves have already entered this list into a text file called list.txt. Santa needs you to develop an app that

- processes the text file,
- checks if a child is naughty or nice,
- adds a child to the list,
- changes a child's listing,
- lists all entries, and
- lists all nice children.

Map

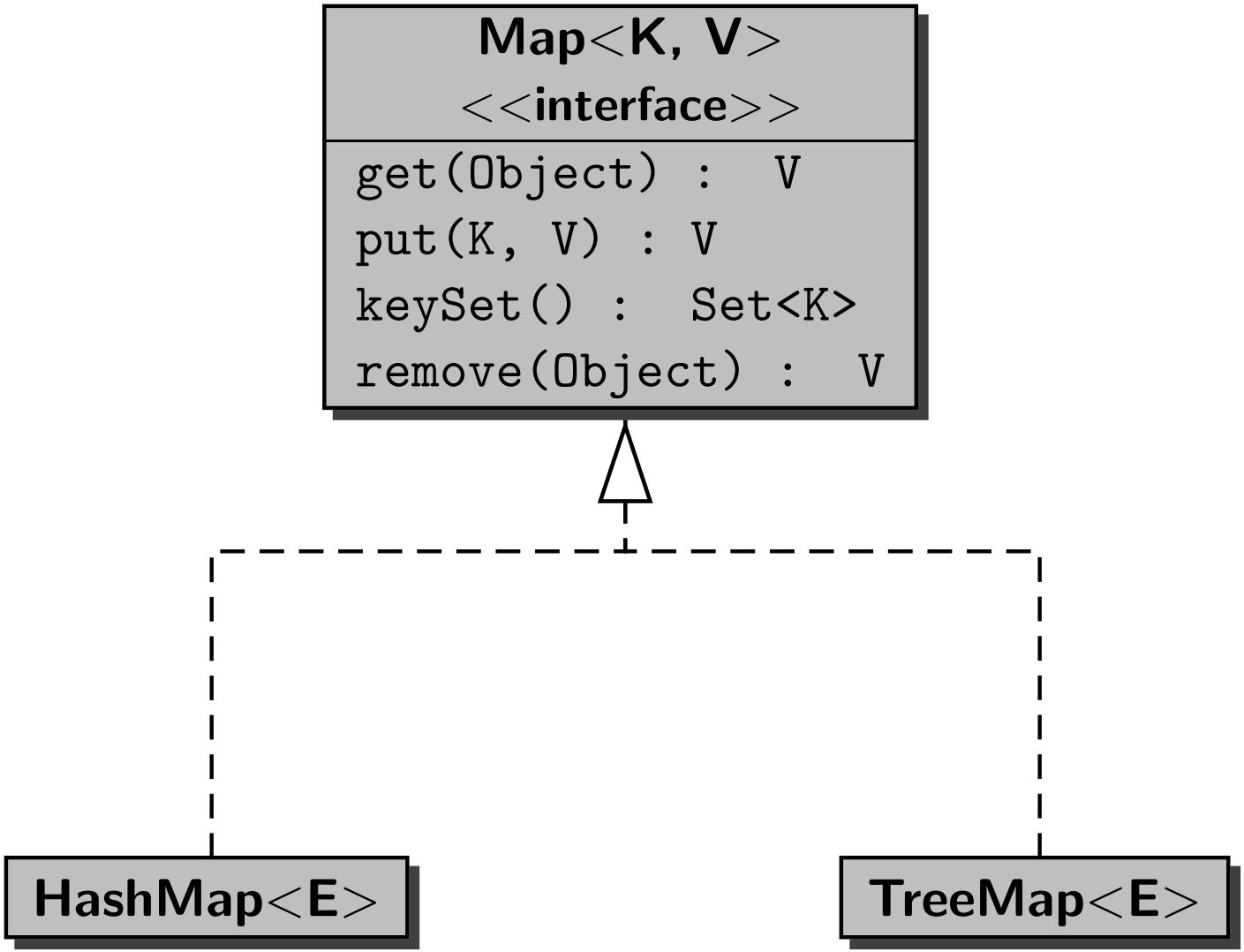
An object of type `Map<K, V>` is called a map.

The elements of type `K` are called keys and the elements of type `V` are called values.

An object of type `Map<K, V>` can be viewed as

- a function m from (the set of values of type) `K` to (the set of values of type) `V`, or
- a set M of pairs (k, v) where
  - k is a value of type `K`,
  - v is a value of type `V`,
  - if $(k, v_1) \in M$ and $(k, v_2) \in M$ then $v_1 = v_2$.

Map


Map<K, V>
<<interface>>
get(Object) : V
put(K, V) : V
keySet() : Set<K>
remove(Object) : V
HashMap<E>
TreeMap<E>

Empty Map

Question

Assume that the keys are integers and the values are strings. How do you create an empty map?

Empty Map

Question

Assume that the keys are integers and the values are strings. How do you create an empty map?

Answer

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

Empty Map

Question

Assume that the keys are integers and the values are strings. How do you create an empty map?

Answer

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

Question

Which function m does the object `map` represent?

Empty Map

Question

Assume that the keys are integers and the values are strings. How do you create an empty map?

Answer

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

Question

Which function m does the object `map` represent?

Answer

The function m is defined for each integer i by

$$m(i) = \texttt{null}.$$

Empty Map

Question

Assume that the keys are integers and the values are strings. How do you create an empty map?

Answer

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

Question

Which set M does the object `map` represent?

Empty Map

Question

Assume that the keys are integers and the values are strings. How do you create an empty map?

Answer

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

Question

Which set M does the object `map` represent?

Answer

The set M is defined by

$$M = \emptyset.$$

Get a Value

Question

How do you get the value to which integer `key` is mapped by `map`?

Get a Value

Question

How do you get the value to which integer `key` is mapped by `map`?

Answer

```
String value = map.get(key);
```

Get a Value

Question

How do you get the value to which integer `key` is mapped by `map`?

Answer

```
String value = map.get(key);
```

Question

Assume that the object `map` before the invocation of the method `get` is represented by the function m. Which function does the object `map` after the invocation represent?

Get a Value

Question

How do you get the value to which integer `key` is mapped by `map`?

Answer

```
String value = map.get(key);
```

Question

Assume that the object `map` before the invocation of the method `get` is represented by the function m. Which function does the object `map` after the invocation represent?

Answer

The same function, since the map does not change.

Put a Key and a Value

Question

How do you put the value `newValue` for the integer `key` into `map`?

Put a Key and a Value

Question

How do you put the value `newValue` for the integer `key` into `map`?

Answer

```
String oldValue = map.put(key, newValue);
```

Put a Key and a Value

Question

How do you put the value `newValue` for the integer `key` into `map`?

Answer

```
String oldValue = map.put(key, newValue);
```

Question

Assume that the object `map` before the invocation of the method `put` is represented by the function m. Which function does the object `map` after the invocation represent?

Put a Key and a Value

Question

How do you put the value `newValue` for the integer `key` into `map`?

Answer

```
String oldValue = map.put(key, newValue);
```

Question

Assume that the object `map` before the invocation of the method `put` is represented by the function m. Which function does the object `map` after the invocation represent?

Answer

The function m' is defined for each integer i by

$$m'(i) = \begin{cases} \texttt{newValue} & \text{if } i = \texttt{key} \\ m(i) & \text{otherwise} \end{cases}$$

Put a Key and a Value

Question

How do you put the value `newValue` for the integer `key` into `map`?

Answer

```
String oldValue = map.put(key, newValue);
```

Question

Assume that the object `map` before the invocation of the method `put` is represented by the set M. Which set does the object `map` after the invocation represent?

Put a Key and a Value

Question

How do you put the value `newValue` for the integer `key` into `map`?

Answer

```
String oldValue = map.put(key, newValue);
```

Question

Assume that the object `map` before the invocation of the method `put` is represented by the set M. Which set does the object `map` after the invocation represent?

Answer

The set M' is defined by

$$M' = (M \setminus \{(\texttt{key}, \texttt{oldValue})\}) \cup \{(\texttt{key}, \texttt{newValue})\}.$$

Remove a Key and its Value

Question

How do you remove an integer `key` and its value from `map`?

Remove a Key and its Value

Question

How do you remove an integer `key` and its value from `map`?

Answer

```
String value = map.remove(key);
```

Remove a Key and its Value

Question

How do you remove an integer `key` and its value from `map`?

Answer

```
String value = map.remove(key);
```

Question

Assume that the object `map` before the invocation of the method `remove` is represented by the function m. Which function does the object `map` after the invocation represent?

Remove a Key and its Value

Question

How do you remove an integer `key` and its value from `map`?

Answer

```
String value = map.remove(key);
```

Question

Assume that the object `map` before the invocation of the method `remove` is represented by the function m. Which function does the object `map` after the invocation represent?

Answer

The function m' is defined for each integer i by

$$m'(i) = \begin{cases} \texttt{null} & \text{if } i = \texttt{key} \\ m(i) & \text{otherwise} \end{cases}$$

Remove a Key and its Value

Question

How do you remove an integer `key` and its value from `map`?

Answer

```
String value = map.remove(key);
```

Question

Assume that the object `map` before the invocation of the method `remove` is represented by the set M. Which set does the object `map` after the invocation represent?

Remove a Key and its Value

Question

How do you remove an integer `key` and its value from `map`?

Answer

```
String value = map.remove(key);
```

Question

Assume that the object `map` before the invocation of the method `remove` is represented by the set M. Which set does the object `map` after the invocation represent?

Answer

The function M' is defined by

$$M' = M \setminus \{(\texttt{key}, \texttt{value})\}.$$

All Keys

Question

How do you get all keys of `map`?

All Keys

Question

How do you get all keys of `map`?

Answer

```
Set<Integer> keys = map.keySet();
```

All Keys

Question

How do you get all keys of `map`?

Answer

```
Set<Integer> keys = map.keySet();
```

Question

Assume that the object `map` before the invocation of the method `remove` is represented by the function m. Which set K does the object `keys` represent?

All Keys

Question

How do you get all keys of `map`?

Answer

```
Set<Integer> keys = map.keySet();
```

Question

Assume that the object `map` before the invocation of the method `remove` is represented by the function m. Which set K does the object `keys` represent?

Answer

The set K is defined by

$$K = \{i \mid m(i) \neq \texttt{null}\}.$$

All Keys

Question

How do you get all keys of `map`?

Answer

```
Set<Integer> keys = map.keySet();
```

Question

Assume that the object `map` before the invocation of the method `remove` is represented by the set M. Which set K does the object `keys` represent?

All Keys

Question

How do you get all keys of `map`?

Answer

```
Set<Integer> keys = map.keySet();
```

Question

Assume that the object `map` before the invocation of the method `remove` is represented by the set M. Which set K does the object `keys` represent?

Answer

The set K is defined by

$$K = \{i \mid \exists v : (i, v) \in M\}.$$

Santa's Little Helper

Since the number of children in the world is increasing every year, Santa is in need of an app to maintain his list. Can you help him?

Numerous elves have already entered this list into a text file called list.txt. Santa needs you to develop an app that

- processes the text file,
- checks if a child is naughty or nice,
- adds a child to the list,
- changes a child's listing,
- lists all entries, and
- lists all nice children.

Object Serialization

Rather than writing a string representation of an object to a file, we can also save the object to a file directly.

```
ObjectOutputStream objectOutput =
  new ObjectOutputStream(
     new FileOutputStream("list.dat"));
...
objectOutput.close();
```

Object Serialization

Rather than reading a string representation of an object from a file and creating the object, we can also read the object from a file directly.

```
ObjectInputStream objectInput =
   new ObjectInputStream(
      new FileInputStream("list.dat"));
Map<String, String> list =
   (Map) objectInput.readObject();
...
objectInput.close();
```

Exception Handling

Chapter 11

December 1, 2010

Sources of Crashes

- The user

```
Enter your choice: a
```

- The client

```
List<Integer> list = ...
for (int i = 0; i <= list.size(); i++)
{
   output.println(list.get(i));
}
```

- The implementer

```
import com.cheapbutquestionable.Integers;
...
int value = Integers.parseInt(input.nextInt());
```

- The runtime environment

```
List<String> list = ...
while (true) { list.add(new String("Hello")); }
```

API

Which exceptions a method may throw are specified in the API.

`E get(int index)`

Returns the element at the specified position in this list.

Parameters:

`index` – index of the element to return

Returns:

the element at the specified position in this list

Throws:

`IndexOutOfBoundsException` – if the index is out of range (`index < 0 || index >= size()`)

Preconditions versus Exceptions

Question

Why do we need exceptions? Can't we prevent crashes by introducing appropriate preconditions?

Preconditions versus Exceptions

Question

Why do we need exceptions? Can't we prevent crashes by introducing appropriate preconditions?

Answer

Introducing an appropriate precondition is not always practical and in some cases impossible.

Preconditions versus Exceptions

The method Double.valueOf(String) throws a NumberFormatException if the argument is not a parsable number.

If this exception were replaced with a precondition, the client would have to check that the argument is a parsable number. Although this can be done using a regular expression, as shown in the API of the method Double.valueOf(String), using exception handling is much easier.

Preconditions versus Exceptions

Each constructor throws an OutOfMemoryError when the Java Virtual Machine cannot allocate an object because it is out of memory, and no more memory could be made available by the garbage collector.

If this error were replaced with a precondition, the client would have to check if there would be sufficient memory before creating each object, which is obviously extremely tedious (if at all possible).

How to Handle Exceptions

Step 1: place a try block around the statement(s) that may throw the exception.

```
try
{
    ...
}
```

Step 2: place a catch block right after the try block.

```
catch (...Exception e)
{
    ...
}
```