Package utility

Interface List<E>

All Known Implementing Classes:

ArrayList, LinkedList

```
public interface List<E>
```

Method Summary

All Methods | **Instance Methods** | **Abstract Methods**

Modifier and Type	Method	Description
void	add(int index, E item)	
boolean	add(E item)	
void	clear()	
boolean	contains(E item)	
E	get(int index)	
int	indexOf(E item)	
boolean	isEmpty()	
Iterator<E>	iterator()	
E	remove(int index)	
boolean	remove(E item)	
E	set(int index, E item)	
int	size()	

Method Details

add

```
boolean add(E item)
```

add

```
void add(int index,
         E item)
```

clear

```
void clear()
```

contains

```
boolean contains(E item)
```

get

```
E get(int index)
```

indexOf

```
int indexOf(E item)
```

isEmpty

```
boolean isEmpty()
```

remove

```
E remove(int index)
```

remove

```
boolean remove(E item)
```

set

```
E set(int index,
      E item)
```

size

```
int size()
```

iterator

```
Iterator<E> iterator()
```

Package utility

Interface Iterator<E>

All Known Implementing Classes:

`LinkedList.LinkedIterator`

```
public interface Iterator<E>
```

Method Summary

All Methods | **Instance Methods** | **Abstract Methods**

Modifier and Type	Method	Description
`boolean`	`hasNext()`	
`E`	`next()`	
`void`	`remove()`	

Method Details

hasNext

```
boolean hasNext()
```

next

```
E next()
```

remove

```
void remove()
```

Package utility

Class ArrayList<E>

java.lang.Object
    utility.ArrayList<E>

All Implemented Interfaces:

List<E>

```
public class ArrayList<E>
extends java.lang.Object
implements List<E>
```

Field Summary

Fields

Modifier and Type	Field	Description
static int	DEFAULT_CAPACITY	

Constructor Summary

Constructors

Constructor	Description
ArrayList()	creates array list object
ArrayList(int capacity)	creates array list object for a specific capacity

Method Summary

All Methods **Instance Methods** **Concrete Methods**

Modifier and Type	Method	Description
void	add(int index, E item)	inserts the item at the given index in the list.
boolean	add(E item)	appends the item specified to the end of the list.
void	clear()	clears list of all elements, return size back to zero.
boolean	contains (E item)	searches for an item and returns true if in the array,

Modifier and Type	Method	Description
`void`	`ensureCapacity(int capacity)`	doubles the capacity of the underlying array, to ensure that it can hold the number of elements specified by the capacity requested.
`E`	`get(int index)`	returns the item at the specified position in the list.
`int`	`indexOf(E item)`	searches for an item and returns the first occurrence in the array, otherwise returns -1, if NOT found.
`boolean`	`isEmpty()`	returns true, if the list is empty,
`Iterator<E>`	`iterator()`	returns an object used to traverse the elements in list
`E`	`remove(int index)`	removes the item at the given index in the list.
`boolean`	`remove(E item)`	removes the first occurrence of the specified item from the list, if present.
`E`	`set(int index, E item)`	replaces the item at the specified position with the one passed.
`int`	`size()`	returns the number of the elements in the list.
`java.lang.String`	`toString()`	displays the contents of the list.

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Field Details

DEFAULT_CAPACITY

```
public static final int DEFAULT_CAPACITY
```

See Also:

Constant Field Values

Constructor Details

ArrayList

```
public ArrayList()
```

creates array list object

ArrayList

```
public ArrayList(int capacity)
```

creates array list object for a specific capacity

Parameters:

`capacity` - of the maximum list.

Method Details

size

```
public int size()
```

returns the number of the elements in the list.

Specified by:

`size` in interface `List<E>`

Returns:

size of the list.

add

```
public boolean add(E item)
```

appends the item specified to the end of the list.

Specified by:

`add` in interface `List<E>`

Parameters:

`item` - in the list

Returns:

boolean value if successful.

add

```
public void add(int index,
                E item)
```

inserts the item at the given index in the list.

Specified by:

`add` in interface `List<E>`

Parameters:

`index` - given in the list.

`item` - in the list.

clear

```
public void clear()
```

clears list of all elements, return size back to zero.

Specified by:

`clear` in interface `List<E>`

get

```
public E get(int index)
```

returns the item at the specified position in the list.

Specified by:

`get` in interface `List<E>`

Parameters:

`index` - of item in list.

Returns:

item at index.

remove

```
public E remove(int index)
```

removes the item at the given index in the list.

Specified by:

`remove` in interface `List<E>`

Parameters:

`index` - of item in list.

Returns:

old value in list.

remove

```
public boolean remove(E item)
```

removes the first occurrence of the specified item from the list, if present.

Specified by:

`remove` in interface `List<E>`

Parameters:

`item` - to remove from list.

Returns:

boolean value.

set

```
public E set(int index,
             E item)
```

replaces the item at the specified position with the one passed.

Specified by:

`set` in interface `List<E>`

Parameters:

`index` - to replace list item.

`item` - that replaces one in list.

Returns:

old item.

indexOf

```
public int indexOf(E item)
```

searches for an item and returns the first occurrence in the array, otherwise returns -1, if NOT found.

Specified by:

`indexOf` in interface `List<E>`

Parameters:

`item` - to search for in list.

Returns:

location of item, if found.

isEmpty

```
public boolean isEmpty()
```

returns true, if the list is empty,

Specified by:

`isEmpty` in interface `List<E>`

Returns:

boolean value

iterator

```
public Iterator<E> iterator()
```

returns an object used to traverse the elements in list

Specified by:

`iterator` in interface `List<E>`

Returns:

iterator for list

contains

```
public boolean contains(E item)
```

searches for an item and returns true if in the array,

Specified by:

`contains` in interface `List<E>`

Parameters:

`item` - to search for in list.

Returns:

boolean value.

ensureCapacity

```
public void ensureCapacity(int capacity)
```

doubles the capacity of the underlying array, to ensure that it can hold the number of elements specified by the capacity requested.

Parameters:

`capacity` -

toString

```
public java.lang.String toString()
```

displays the contents of the list.

Overrides:

`toString` in class `java.lang.Object`

Returns:

list

Package utility

Class LinkedList<E>

java.lang.Object
    utility.LinkedList<E>

All Implemented Interfaces:

`List<E>`

```
public class LinkedList<E>
extends java.lang.Object
implements List<E>
```

Nested Class Summary

Nested Classes

Modifier and Type	Class	Description
`class`	`LinkedList.LinkedIterator`	

Constructor Summary

Constructors

Constructor	Description
`LinkedList()`	creates linked list object

Method Summary

All Methods | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method	Description
`void`	`add(int index, E item)`	inserts the item at the given index in the list.
`boolean`	`add(E item)`	appends the item specified to the end of the list.
`void`	`clear()`	clears list of all elements, return size back to zero.
`boolean`	`contains(E item)`	searches for an item and returns true if in the array,

Modifier and Type	Method	Description

`E`	`get`
	`(int index)`
		returns the item at the specified position in the list.

`int`	`indexOf`
	`(E item)`
		searches for an item and returns the first occurrence in the array, otherwise returns -1, if NOT found.

`boolean`	`isEmpty()`	returns true, if the list is empty,
`Iterator<E>`	`iterator()`	returns an object used to traverse the elements in list

`E`	`remove`
	`(int index)`
		removes the item at the given index in the list.

`boolean`	`remove`
	`(E item)`
		removes the first occurrence of the specified item from the list, if present.

`E`	`set`
	`(int index,`
	`E item)`
		replaces the item at the specified position with the one passed.

`int`	`size()`	returns the number of the elements in the list.
`java.lang.String`	`toString()`	displays the contents of the list.

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructor Details

LinkedList

```
public LinkedList()
```

creates linked list object

Method Details

add

```
public boolean add(E item)
```

appends the item specified to the end of the list.

Specified by:

`add` in interface `List<E>`

Parameters:

`item` - in the list

Returns:

boolean value if successful.

add

```
public void add(int index,
                E item)
```

inserts the item at the given index in the list.

Specified by:

`add` in interface `List<E>`

Parameters:

`index` - given in the list.

`item` - in the list.

clear

```
public void clear()
```

clears list of all elements, return size back to zero.

Specified by:

`clear` in interface `List<E>`

contains

```
public boolean contains(E item)
```

searches for an item and returns true if in the array,

Specified by:

`contains` in interface `List<E>`

Parameters:

`item` - to search for in list.

Returns:

boolean value.

get

```
public E get(int index)
```

returns the item at the specified position in the list.

Specified by:

`get` in interface `List<E>`

Parameters:

`index` - of item in list.

Returns:

item at index.

indexOf

```
public int indexOf(E item)
```

searches for an item and returns the first occurrence in the array, otherwise returns -1, if NOT found.

Specified by:

`indexOf` in interface `List<E>`

Parameters:

`item` - to search for in list.

Returns:

location of item, if found.

isEmpty

```
public boolean isEmpty()
```

returns true, if the list is empty,

Specified by:

`isEmpty` in interface `List<E>`

Returns:

boolean value

iterator

```
public Iterator<E> iterator()
```

returns an object used to traverse the elements in list

Specified by:

`iterator` in interface `List<E>`

Returns:

iterator for list

remove

```
public E remove(int index)
```

removes the item at the given index in the list.

Specified by:

`remove` in interface `List<E>`

Parameters:

`index` - of item in list.

Returns:

old value in list.

remove

```
public boolean remove(E item)
```

removes the first occurrence of the specified item from the list, if present.

Specified by:

`remove` in interface `List<E>`

Parameters:

`item` - to remove from list.

Returns:

boolean value.

set

```
public E set(int index,
             E item)
```

replaces the item at the specified position with the one passed.

Specified by:

`set` in interface `List<E>`

Parameters:

`index` - to replace list item.

`item` - that replaces one in list.

Returns:

old item.

size

```
public int size()
```

returns the number of the elements in the list.

Specified by:

`size` in interface `List<E>`

Returns:

size of the list.

toString

```
public java.lang.String toString()
```

displays the contents of the list.

Overrides:

`toString` in class `java.lang.Object`

Returns:

string representation of list

Package utility

Class MyQueue<E>

java.lang.Object
    utility.MyQueue<E>

```
public class MyQueue<E>
extends java.lang.Object
```

Constructor Summary

Constructors

Constructor	Description
`MyQueue()`	

Method Summary

All Methods **Instance Methods** **Concrete Methods**

Modifier and Type	Method	Description
`boolean`	`add(E item)`	
`boolean`	`isEmpty()`	returns true, if the list is empty,
`E`	`peek()`	
`E`	`remove()`	
`int`	`size()`	returns the number of the elements in the list.
`java.lang.String`	`toString()`	displays the contents of the list.

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructor Details

MyQueue

```
public MyQueue()
```

Method Details

add

```
public boolean add(E item)
```

isEmpty

```
public boolean isEmpty()
```

returns true, if the list is empty,

Returns:

boolean value

peek

```
public E peek()
```

remove

```
public E remove()
```

size

```
public int size()
```

returns the number of the elements in the list.

Returns:

size of queue.

toString

```
public java.lang.String toString()
```

displays the contents of the list.

Overrides:

`toString` in class `java.lang.Object`

Returns:

queue

Package utility

Class MyStack<E>

java.lang.Object
    utility.MyStack<E>

```
public class MyStack<E>
extends java.lang.Object
```

Constructor Summary

Constructors

Constructor	Description
`MyStack()`	

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description
`boolean`	`isEmpty()`	returns true, if the list is empty,
`E`	`peek()`	
`E`	`pop()`	
`E`	`push(E item)`	
`int`	`size()`	returns the number of the elements in the list.
`java.lang.String`	`toString()`	displays the contents of the list.

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructor Details

MyStack

```
public MyStack()
```

Method Details

peek

```
public E peek()
```

pop

```
public E pop()
```

push

```
public E push(E item)
```

isEmpty

```
public boolean isEmpty()
```

returns true, if the list is empty,

Returns:

boolean value

size

```
public int size()
```

returns the number of the elements in the list.

Returns:

size of stack.

toString

```
public java.lang.String toString()
```

displays the contents of the list.

Overrides:

`toString` in class `java.lang.Object`

Returns:

stack

Package utility

Class Module

java.lang.Object
    utility.Module

```
public class Module
extends java.lang.Object
```

Constructor Summary

Constructors

Constructor	Description
`Module()`	

Method Summary

All Methods | **Static Methods** | **Concrete Methods**

Modifier and Type	Method	Description
`static <T extends java.lang.Comparable<T>> void`	`quickSort(List<T> list)`	
`static <T> void`	`swap(List<T> list, int a, int b)`	

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Details

Module

```
public Module()
```

Method Details

quickSort

```
public static <T extends java.lang.Comparable<T>>
void quickSort(List<T> list)
```

swap

```
public static <T> void swap(List<T> list,
                            int a,
                            int b)
```

Package activities

Class AlgoProgram

java.lang.Object
    activities.AlgoProgram

```
public class AlgoProgram
extends java.lang.Object
```

Constructor Summary

Constructors

Constructor	Description
`AlgoProgram()`	

Method Summary

All Methods | Static Methods | Concrete Methods

Modifier and Type	Method	Description
`static void`	`intro()`	
`static void`	`main(java.lang.String[] args)`	
`static void`	`testQuickSort()`	

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Details

AlgoProgram

```
public AlgoProgram()
```

Method Details

intro

```
public static void intro()
```

testQuickSort

```
public static void testQuickSort()
```

main

```
public static void main(java.lang.String[] args)
```

Package activities

Class Person

java.lang.Object
    activities.Person

All Implemented Interfaces:

`java.lang.Comparable<Person>`

```
public class Person
extends java.lang.Object
implements java.lang.Comparable<Person>
```

Constructor Summary

Constructors

Constructor	Description
`Person(java.lang.String name, int age)`	

Method Summary

All Methods | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method	Description
`int`	`compareTo(Person other)`	compares two people information
`java.lang.String`	`toString()`	represent the person's data
`int`	`yearsToRetirement()`	

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructor Details

Person

```
public Person(java.lang.String name,
              int age)
```

Method Details

yearsToRetirement

```
public int yearsToRetirement()
```

compareTo

```
public int compareTo(Person other)
```

compares two people information

Specified by:

`compareTo` in interface `java.lang.Comparable<Person>`

Parameters:

`other` - person

Returns:

the difference between two people

toString

```
public java.lang.String toString()
```

represent the person's data

Overrides:

`toString` in class `java.lang.Object`

Returns:

person's information

Package activities

Class ListPeopleProgram

java.lang.Object
    activities.ListPeopleProgram

```
public class ListPeopleProgram
extends java.lang.Object
```

Constructor Summary

Constructors

Constructor	Description
`ListPeopleProgram()`	

Method Summary

All Methods **Static Methods** **Concrete Methods**

Modifier and Type	Method	Description
`static void`	`main(java.lang.String[] args)`	
`static void`	`write(java.lang.String location, java.lang.StringBuilder builder)`	

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Details

ListPeopleProgram

```
public ListPeopleProgram()
```

Method Details

write

```
public static void write(java.lang.String location,
                         java.lang.StringBuilder builder)
                  throws java.io.IOException
```

Throws:

`java.io.IOException`

main

```
public static void main(java.lang.String[] args)
                 throws java.io.IOException
```

Throws:

`java.io.IOException`

Package activities

Class Patient

java.lang.Object
    activities.Patient

All Implemented Interfaces:

`java.lang.Comparable<Patient>`

```
public class Patient
extends java.lang.Object
implements java.lang.Comparable<Patient>
```

Constructor Summary

Constructors

Constructor	Description
`Patient(java.lang.String name, java.lang.String address, java.lang.String id, int priority)`	

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description
`int`	`compareTo(Patient other)`	compares two patient information
`java.lang.String`	`getInfo()`	represent the patient's information after medical examination
`java.lang.String`	`toString()`	represents the patient's information

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructor Details

Patient

```
public Patient(java.lang.String name,
               java.lang.String address,
               java.lang.String id,
               int priority)
```

Method Details

compareTo

```
public int compareTo(Patient other)
```

compares two patient information

Specified by:

`compareTo` in interface `java.lang.Comparable<Patient>`

Parameters:

`other` - patient

Returns:

the difference between two patient

getInfo

```
public java.lang.String getInfo()
```

represent the patient's information after medical examination

Returns:

the patient's data without priority number

toString

```
public java.lang.String toString()
```

represents the patient's information

Overrides:

`toString` in class `java.lang.Object`

Returns:

list of patient's data

Package activities

Class EmergencyRoomProgram

java.lang.Object
    activities.EmergencyRoomProgram

```
public class EmergencyRoomProgram
extends java.lang.Object
```

Constructor Summary

Constructors

Constructor	Description
`EmergencyRoomProgram()`	

Method Summary

All Methods | **Static Methods** | **Concrete Methods**

Modifier and Type	Method	Description
`static void`	`listEmergency()`	
`static void`	`main(java.lang.String[] args)`	

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Details

EmergencyRoomProgram

```
public EmergencyRoomProgram()
```

Method Details

listEmergency

```
public static void listEmergency()
                          throws java.io.IOException
```

Throws:

`java.io.IOException`

main

```
public static void main(java.lang.String[] args)
                 throws java.io.IOException
```

Throws:

`java.io.IOException`

Package activities

Class UndoRedoProgram

java.lang.Object
    activities.UndoRedoProgram

```
public class UndoRedoProgram
extends java.lang.Object
```

Constructor Summary

Constructors

Constructor	Description
`UndoRedoProgram()`	

Method Summary

All Methods | **Static Methods** | **Concrete Methods**

Modifier and Type	Method	Description
`static void`	`main(java.lang.String[] args)`	
`static void`	`UndoRedo()`	

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Details

UndoRedoProgram

```
public UndoRedoProgram()
```

Method Details

UndoRedo

```
public static void UndoRedo()
```

main

```
public static void main(java.lang.String[] args)
```