import java.util.\*;

/\*\* This class implements some methods of the Java ArrayList class \*/

public class KWArrayList<E>

{

private static final int INITIAL\_CAPACITY = 10; // *The default initial capacity*

private E[] theData; // *array*

private int size; *// actual number of elements in the array*

private int capacity; *// array capacity*

*//default constructor*

@SupressWarnings("unchecked")

public KWArrayList ()

{

capacity = INITIAL\_CAPACITY;

theData = (E[]) new Object[capacity]; *// memory allocation*

size = 0; // *initially empty*

}

*// constructor with capacity*

@SupressWarnings("unchecked")

public KWArrayList (int cap)

{

capacity = cap;

theData = (E[]) new Object[capacity]; *// memory allocation*

size = 0; // *initially empty*

}

*// Return actual number of elements in the list*

public int size( )

{

return size;

}

*// Check whether list is empty or not*

public boolean isEmpty( )

{

return (size == 0);

}

// *Create a new array that is twice the capacity of the current array and*

// *then copy the contents of the new array*

private void reallocate ()

{

capacity \*= 2;

theData = Arrays.copyOf(theData, capacity);

}

*// Method to insert a new element at the end of the list*

public boolean add (E anEntry)

{

if (size >= capacity) *// Make sure there is space to add new item*

reallocate();

theData[size] = anEntry; *// insert item*

size++;

return true;

}

*// Method to insert a new element at the specified index in the list*

public void add (int index, E anEntry)

{

*// check validity of index*

if (index < 0 || index > size) {

throw new ArrayIndexOutOfBoundsException(index);

}

if (size >= capacity) *// Make sure there is space to add new item*

reallocate();

*// shift data in the array to make space for the new item*

for (int i = size; i > index; i--) {

theData[i] = theData[i-1];

}

theData[index] = anEntry; *// insert item*

size++;

}

*// Return the object at the specified index*

public E get (int index)

{

*// check validity of index*

if (index < 0 || index >= size) {

throw new ArrayIndexOutOfBoundsException(index);

}

return theData[index];

}

*// Replace the element at the specified index by the newValue and return*

*// the old value*

public E set (int index, E newValue)

{

if (index < 0 || index >= size) {

throw new ArrayIndexOutOfBoundsException(index);

}

E oldValue = theData[index];

theData[index] = newValue;

return oldValue;

}

*// Remove the element at the given index and return it*

public E remove (int index)

{

if (index < 0 || index >= size) {

throw new ArrayIndexOutOfBoundsException(index);

}

E returnValue = theData[index];

*// shift data in the array to delete element*

for (int i = index + 1; i < size; i++) {

theData[i-1] = theData[i];

}

size--;

return returnValue;

}

*// Method to search an object in the list. If found, return its location,*

*// else return -1. It is assumed that the class of obj implements equals*

*// method. The method finds the first occurrence of obj in the list.*

public int indexOf(E obj)

{

int index = -1;

for(int i = 0; i < size; i++)

if ( obj.equals(theData[i]) ){

index = i;

break;

}

return index;

}

*// Method to remove the first occurrence of the object from the list,*

*// if present and return true, else return false*

public boolean remove (E obj)

{

int index = indexOf(obj);

if (index != -1) {

remove(index); *// remove element at the index*

return true; *// obj found*

}

else

return false; *// obj not found*

}

*// Method to check whether an object is in the list. If found, return true,*

*// else return false. It is assumed that the class of obj implements equals*

*// method*

public boolean contains(E obj)

{

for(int i = 0; i < size; i++) {

if ( obj.equals(theData[i]) )

return true;

}

return false;

}

*// Method to delete all objects from the list and bring it to empty state*

public void clear( )

{

for(int i = 0; i < size; i++) {

theData[i] = null;

size = 0;

}

*// Return String representation of the list*

public String toString( )

{

String str = “”;

for(int i = 0; i < size; i++)

str = str + theData[i] + “\n”;

return str;

}

} //end class KWArrayList