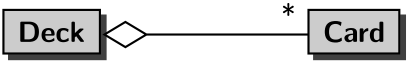
Collection



A Deck has multiple Cards.

List

Question

May a list contain duplicates?

List

Question

May a list contain duplicates?

Answer

Yes.

List

Question

May a list contain duplicates?

Answer

Yes.

Question

Are the elements of a list ordered?

List

Question

May a list contain duplicates?

Answer

Yes.

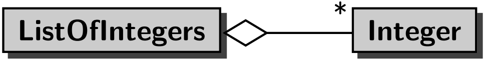
Question

Are the elements of a list ordered?

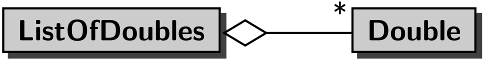
Answer

Yes.

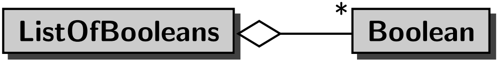
Number of millimeters of rain per day



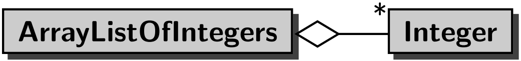
Score for each test



Whether there was any snow for each day

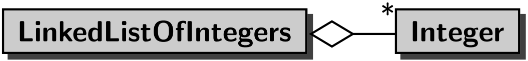


Number of millimeters of rain per day



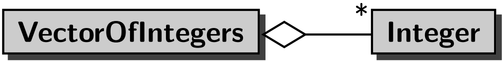
The list is implemented by means of an array.

Number of millimeters of rain per day



The list is implemented by means of a "links."

Number of millimeters of rain per day



The list is implemented by means of an array and multiple threads can manipulate the list at the same time.

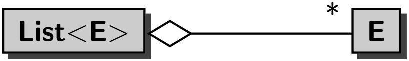
Lists

These different lists can be classified based on

- the type of the elements of the list (Integer, Double, Boolean, . . .) and
- the way the list is implemented (using an array, using “links,” . . .).

Lists

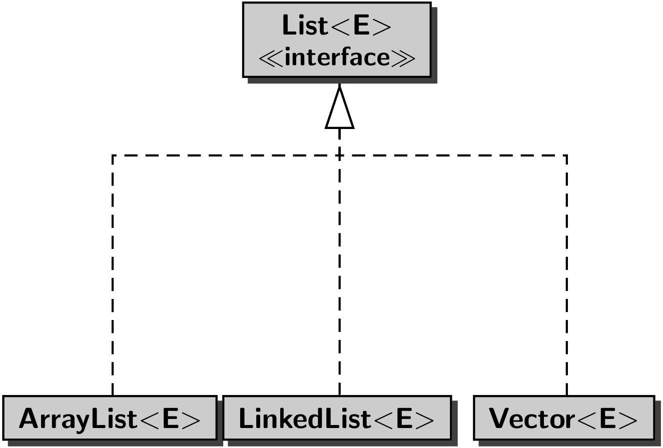
To abstract from the type of the elements of the list, we exploit generics.



E is a type parameter. The elements of the list are of type E.

Lists

To abstract from the way the list is implemented, we exploit interfaces.



Class versus Interface

interface	specification	what?
class	implementation	how?

Number of millimeters of rain per day

```
final int DAYS_PER_YEAR = 365;
List<Integer> rain =
   new ArrayList<Integer>(DAYS_PER_YEAR);
```

- The type of the elements is `Integer` and
- the list is implemented by means of an array.

Number of millimeters of rain per day

```
final int DAYS_PER_YEAR = 365;
List<Integer> rain =
   new ArrayList<Integer>(DAYS_PER_YEAR);
```

- The type of the elements is `Integer` and
- the list is implemented by means of an array.

Question

Why can we assign an object of type `ArrayList<Integer>` to a variable of type `List<Integer>`?

Number of millimeters of rain per day

```
final int DAYS_PER_YEAR = 365;
List<Integer> rain =
   new ArrayList<Integer>(DAYS_PER_YEAR);
```

- The type of the elements is `Integer` and
- the list is implemented by means of an array.

Question

Why can we assign an object of type `ArrayList<Integer>` to a variable of type `List<Integer>`?

Answer

Because the class `ArrayList<E>` implements the interface `List<E>`.

Score for each test

```
List<Double> tests = new LinkedList<Double>();
```

- The type of the elements is `Double` and
- the list is implemented by means of “links.”

ArrayList, LinkedList or Vector?

Depends on which operations on the list are performed.

Question

How many milliseconds does it take to add n elements to the end of a list?

ArrayList, LinkedList or Vector?

Depends on which operations on the list are performed.

Question

How many milliseconds does it take to add n elements to the end of a list?

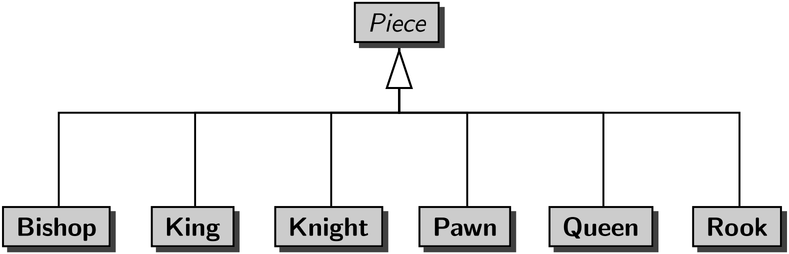
Answer

n	ArrayList	LinkedList	Vector
10^5	9	12	14
10^6	47	92	113
10^7	442	824	1041
2×10^7	913	1,650	2,076
3×10^7	1,350	143,616	3,230
4×10^7	2,527		4,103
5×10^7	2,689		6,119

ArrayList, LinkedList or Vector?

- Adding to or deleting from the beginning of a LinkedList is in general more efficient than adding to or deleting from the beginning of an ArrayList or Vector.
- Adding and deleting while traversing a LinkedList is in general more efficient than adding and deleting while traversing an ArrayList or Vector.
- In most other cases, ArrayList outperforms LinkedList and Vector.

Chess pieces



Row of a chess board

Question

How do you represent a row of a chess board?

Row of a chess board

Question

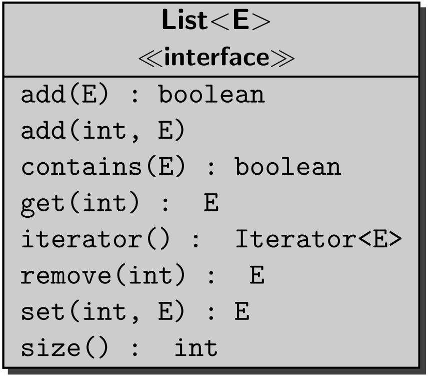
How do you represent a row of a chess board?

Answer

```
final int COLUMNS = 8;
List<Piece> row = new ArrayList<Piece>(COLUMNS);
```

- The type of the elements is `Piece` and
- the list is implemented by means of an array.

Methods of `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
```

Row of a Chess board

Question

Create an empty row of a chess board.

Row of a Chess board

Answer

```
final int COLUMNS = 8;
List<Piece> row = new ArrayList<Piece>(COLUMNS);
for (int c = 0; c < COLUMNS; c++)
{
   row.add(null);
}
```

Row of a Chess board

Question

Place a black rook on the first and the last square of the row.

Row of a Chess board

Answer

```
Rook rook = new Rook(Color.BLACK);
row.set(0, rook);
row.set(COLUMNS - 1, rook);
```

Row of a Chess board

Question

Place a white pawn on each square of the row.

Row of a Chess board

Answer

```
Pawn pawn = new Pawn(Color.WHITE);
for (int c = 0; c < COLUMNS; c++)
{
   row.set(c, pawn);
}
```

Row of a Chess board

Question

Print the row.

An empty square is represented by two spaces. A non-empty square is represented by the representation of the piece on that square. For example, a black king is represented by BK and a white queen is represented by WQ.

The squares are separated by a single space.

Row of a Chess board

Answer

```
StringBuffer representation = new StringBuffer();
for (Piece piece : row)
{
   if (piece == null) {
      representation.append("  ");
   }
   else
   {
      representation.append(piece.toString());
   }
   representation.append(" ");
}
output.println(representation.toString());
```

Row of a chess board

Question

How do you represent a chess board?

Row of a chess board

Question

How do you represent a chess board?

Answer

```
final int ROWS = 8;
List<List<Piece>> board = new ArrayList<List<Piece>>(ROWS);
```

- The type of the elements is `List<Piece>`, each representing a row of the board, and
- the list is implemented by means of an array.

Sets

Question

May a set contain duplicates?

Sets

Question

May a set contain duplicates?

Answer

No.

Sets

Question

May a set contain duplicates?

Answer

No.

Question

Are the elements of a set ordered?

Sets

Question

May a set contain duplicates?

Answer

No.

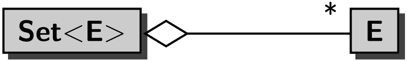
Question

Are the elements of a set ordered?

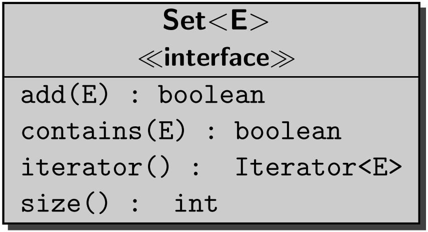
Answer

No.

Sets



Methods of Set

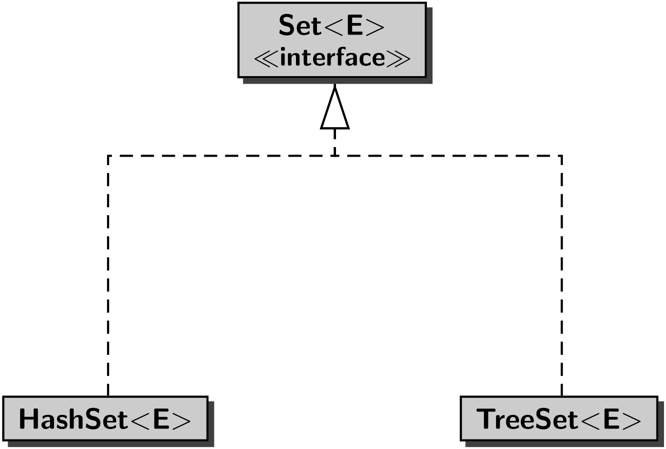


Set<E>

≪**interface**≫

```
add(E) : boolean
contains(E) : boolean
iterator() :  Iterator<E>
size() :  int
```

Sets



HashSet or TreeSet?

- Adding to or deleting from or searching in a HashSet is in general more efficient than adding to or deleting from or searching in a TreeSet.
- TreeSet keeps the elements sorted, but HashSet does not.

ITunes library

Problem

Given an iTunes library, determine whether each playlist of an iTunes library contains duplicates.