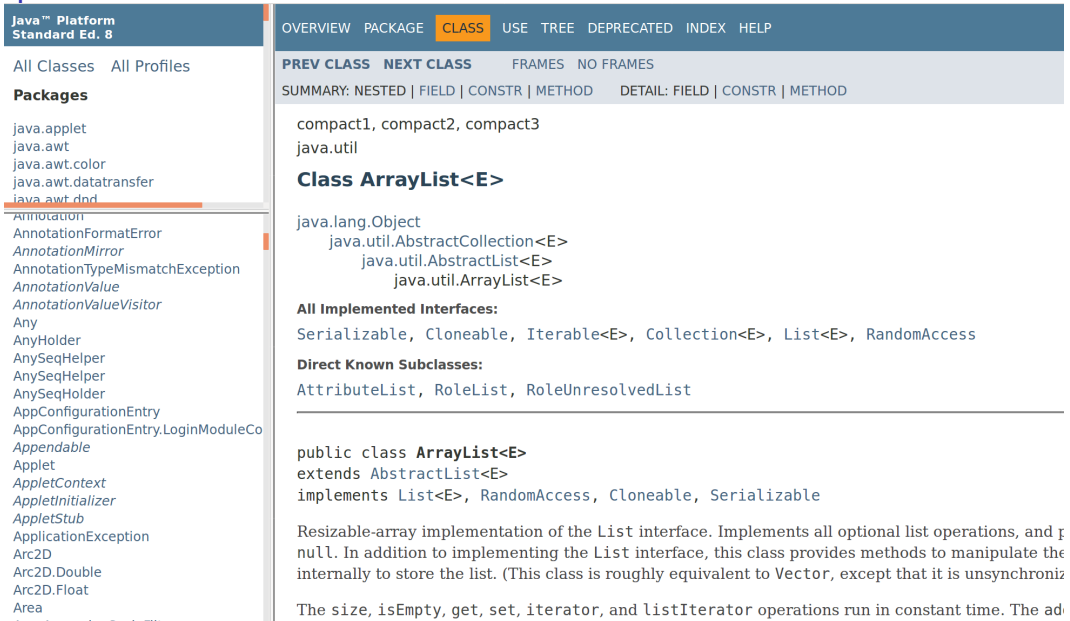
Javadoc

Kai Dührkop, Markus Fleischauer

Beispiel: Java API



Java™ Platform
Standard Ed. 8

All Classes All Profiles

Packages

java.applet
java.awt
java.awt.color
java.awt.datatransfer
java.awt.dnd

Annotation
AnnotationFormatError
AnnotationMirror
AnnotationTypeMismatchException
AnnotationValue
AnnotationValueVisitor
Any
AnyHolder
AnySeqHelper
AnySeqHelper
AnySeqHolder
AppConfigurationEntry
AppConfigurationEntry.LoginModuleCo
Appendable
Applet
AppletContext
AppletInitializer
AppletStub
ApplicationException
Arc2D
Arc2D.Double
Arc2D.Float
Area

OVERVIEW PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

compact1, compact2, compact3

java.util

Class ArrayList<E>

java.lang.Object
  java.util.AbstractCollection<E>
    java.util.AbstractList<E>
      java.util.ArrayList<E>

All Implemented Interfaces:

Serializable, Cloneable, Iterable<E>, Collection<E>, List<E>, RandomAccess

Direct Known Subclasses:

AttributeList, RoleList, RoleUnresolvedList

```
public class ArrayList<E>
extends AbstractList<E>
implements List<E>, RandomAccess, Cloneable, Serializable
```

Resizable-array implementation of the List interface. Implements all optional list operations, and p
null. In addition to implementing the List interface, this class provides methods to manipulate the
internally to store the list. (This class is roughly equivalent to Vector, except that it is unsynchroniz

The size, isEmpty, get, set, iterator, and listIterator operations run in constant time. The ad

```java
/**
  comments starting with **
  are parsed by javadoc
*/
class Example {

}
```

```
/**
* IDE may add leading * on each line.
* These are optional and have no special meaning.
*/
class Example {

}
```

```
/**
  Short documentation (one sentence).

  Long documentation. May contain multiple
  sentences and code examples.

  @author Kai Dührkop
*/
class Example {

}
```

```
class Sequence {
  /**
  * Position of the given character in the underlying alphabet.
  * Every sequence belongs to an alphabet, an ordered and indexed set of characters.
  * indexOf lookups the index of the given character in O(1) time.

    @param character the character which index should be looked up
    @return index of character in alphabet, or -1 if character is not contained
  */
  public int indexOf(char character) {

  }

}
```

```
class NucleotideSequence {
  /**
     {@inheritDoc}
  */
  public int indexOf(char character) {

  }

}
```

```
class NucleotideSequence {
  /**
  * Position of the given character in the DNA alphabet (A,C,G,T).
     @see Sequence#indexOf(char)
  */
  public int indexOf(char character) {

  }

}
```

```java
class Sequence {
  /**
   *  Position of the given character in the underlying alphabet.
   *  Every sequence belongs to an alphabet, an ordered and indexed set of characters.
   *  indexOf lookups the index of the given character in O(1) time.
   *  The alphabet of this sequence can be inspected
   *  by calling {@link Sequence#getAlphabetString()}.

     @param character the character which index should be looked up
     @return index of character in alphabet, or -1 if character is not contained
   */
  public int indexOf(char character) {

  }

}
```

```
class NucleotideSequence {
  /**
    ...

    @throws IllegalArgumentException if reading frame is not in {1, 2 or 3}.
  */
  public AminoAcidSequence translate(int readingFrame) {

  }

}
```

```
/**
 * One sentence description.
 * Multiple sentence description. Bla.

  <pre>{@code
  Example e = new Example();
  e.someExampleMethod();
  e.bla("This is how to use the Example class");
  }</pre>
*/
class Example {

}
```

```
class Example {

  /**
  * Some property description.
  */
  protected String someProperty;

  public String getSomeProperty() {return someProperty;}
  public void setSomeProperty(String property) {someProperty = property;}

}
```

- Getter/Setter müssen nicht extra dokumentiert werden
- Ansonsten: Alles was public ist muss dokumentiert werden. Möglichst auch alles was protected ist.
- Keine Trivial-Erklärungen (getChar gibt char zurück)

Integration in Gradle

- Javadoc Task ist Teil des Java plugins
  `apply plugin "java"`
- Javadoc Task legt die HTML Dokumentation an
  `gradle javadoc`
- Dokumentation zu Javadoc unter `https://docs.oracle.com/javase/1.5.0/docs/tooldocs/solaris/javadoc.html`

Classes

```
/**
 * Models a room.
 *
 * @author Thasso Griebel (thasso@minet.uni-jena.de)
 *
 */
public class Room {
}
```

Variables

```
/**
 * The Room number
 */
private int roomNumber;
```

Constructors

```
/**
 * Creates a new room of given size.
 * <p>
 * Size has to be &gt; 0
 * @param size
 */
public Room(int size){
       super();
       setSize(size);
       this.persons = new ArrayList<Person>();
}
```

Methods

```
/**
 * Adds a {@link Person} to this room. If the person is located
 * somewhere else, the Person is removed from its old location and
 * transfered to this room.
 *
 *
 * @param person the Person to add
 * @throws exception if the room is full
 * @return true if person successfully entered the room
 */
public synchronized boolean addPerson(Person person) throws Exception{
       if(person == null) return false;
       if(person.getLocation() != null ){
       }
       ......
       return persons.add(person);
}
```