

## Advanced Java Programming

Ohad Barzilay,  
Tel-Aviv University  
Spring '06

1

## Java Syntax & Structures

(full version in the course web site)

Identifiers, Keywords, Literals  
Separators and Operators

2

## Control Structures and statements

3

## Flow Control Statements

- if , if - else
- while , do - while
- for
- switch
- break , continue
- Labeled break , Labeled continue

Advanced Java Programming  
Ohad Barzilay

4

## Labeled continue and break

- In nested loops we might want to break/continue with respect to the outer loop
- As there is no goto in java – labels serves the missing feature

Advanced Java Programming  
Ohad Barzilay

5

## Labeled break

```
outer:
do {
    statement1;
    do {
        statement2;
        if ( condition ) {
            break outer;
        }
        statement3;
    } while ( test_expr );
    statement4;
} while ( test_expr );
```

Advanced Java Programming  
Ohad Barzilay

6

## Labeled continue

```
test:
do {
    statement1;
    do {
        statement2;
        if ( condition ) {
            continue test;
        }
        statement3;
    } while ( test_expr );

    statement4;
} while ( test_expr );
```

## Packages and Import statements

## Java API (Packages)

- Java comes with 3,000+ pre-designed components.
- The Java API is the library of classes supplied by Java.
- The classes in the Java API is separated into packages. Each package contains a set of classes that are related in some way.

## The Java API Packages

java.applet	java.net
java.awt	java.rmi
java.beans	java.security
java.io	java.sql
java.lang	java.text
java.math	java.util
...	

## Documentation:

<http://java.sun.com/j2se/1.5.0/docs/api/>

The screenshot shows the Java API documentation website. A callout box labeled "List of Packages" points to the "All Packages" section. Another callout box labeled "List of Classes" points to the "All Classes" section. A third callout box labeled "Details of Packages" points to the "Java 2 Platform Packages" section.

## java.lang

The screenshot shows the Java API documentation website for the java.lang package. A callout box labeled "Choose java.lang from list of Packages" points to the "Package java.lang" section. Another callout box labeled "List of Classes defined in Package" points to the "Classes" section. A third callout box labeled "Details of Classes" points to the "Class Summary" section.

# String Class

Overview Package **Class** Use Tree Deprecated Index Help Java™ 2 Platform  
Std. Ed. v1.4.0

DESCRIPTION: METHODS SUMMARY: MEMBER COMMENTS: SERIAL DATA: SERIAL VERSION: SERIAL USES:

java.lang  
**Class String**

java.lang.Object  
|-- java.lang.String

All Implemented Interfaces:  
CharSequence, Comparable, Serializable

public final class String  
extends Object  
implements CharSequence, Comparable, Serializable

The String class represents character strings. All string literals in Java programs, such as "abc", are implemented as instances of this class.

Strings are constant; that is, their values cannot be changed after they are created. String buffers support mutable strings. Because String objects are immutable they can be shared. For example:

Advanced Java Programming  
Olad Barzilay 13

# String Methods

Method Summary

char	<code>charAt(int index)</code>	Returns the character at the specified index.
boolean	<code>compareTo(Object o)</code>	Compares this String to another Object.
boolean	<code>compareTo(String anotherString)</code>	Compares two strings lexicographically.
boolean	<code>compareToIgnoreCase(String str)</code>	Compares two strings lexicographically, ignoring case considerations.
String	<code>concat(String str)</code>	Concatenates the specified string to the end of this string.
boolean	<code>contentEquals(StringBuffer sb)</code>	Returns true if and only if this String represents the same sequence of characters as the specified StringBuffer.
String	<code>copyValueOf(char[] data)</code>	Returns a String that represents the character sequence in the array specified.
String	<code>copyValueOf(char[] data, int offset, int count)</code>	Returns a String that represents the character sequence in the array specified.

Advanced Java Programming  
Olad Barzilay 14

# Importing Packages

- Using a class from the Java API can be accomplished by using its fully qualified name:  

```
java.util.Random random =  
    new java.util.Random();
```
- Or the class can be imported once with the import statement at the top of the file:  

```
import java.util.Random;  
... Random random = new Random();
```

# Importing Packages

- You can also import all the classes in a given package with a single import statement:  

```
import java.util.*;
```
- The `java.lang` package is automatically imported into every Java program.

# Arrays

# Arrays

- An array is an object that can be used to store a list of values.
- All array elements are of the same type (primitive or objects).
- Arrays have fixed sizes, set when the array is created.

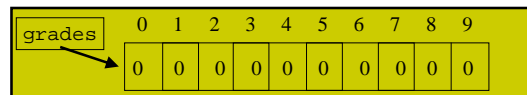
## Array Elements

- A particular value in an array is referenced using the array name followed by the index in brackets.
- As in C, a java Array of size n is indexed from 0 to n-1.
- The Java interpreter will throw an exception if an array index is out of bounds.

## Creating Arrays

- Array elements are initialized with there default values

```
int[] grades = new int[10];  
grades[3] = 70;  
grades[7] = 87;  
int i = 5;  
grades [i/2] = 39;
```



## Initialization list

- You can declare, construct, and initialize the array all in one statement:

```
int[] primes = {2,3,5,7,11,13,17,19};
```

- This declares an array of type `int`, constructs an array of 8 slots, and assigns the designated values into the array.

```
int[] [] values = {{1,2,5}, {4,3,2,1}, {11}};
```