

3. Essential Java Classes

5. Some Useful Classes

Class System (1 of 2)

- contains several useful class fields and methods
- it cannot be instantiated
- standard input, standard output, and error output streams
- gets and sets system properties

Class System (2 of 2)

- **exit(int status)** - terminates the currently running Java Virtual Machine
- **gc()** - runs the garbage collector
- **arraycopy(...)** - copies an array
- **console()** - Returns the unique Console object associated with the current JVM
- **nanoTime()** - Returns the current value of the running JVM high-resolution time

See

<http://docs.oracle.com/javase/7/docs/api/java/lang/System.html> for details

Class Runtime

- Allows to interface with the environment in which the application is running
- static **getRuntime()** - Returns the runtime object associated with the current Java application
- **freeMemory()** - returns the amount of free memory in the JVM
- **totalMemory()** - returns the total amount of memory in the JVM.

Example of String Command Execution

```
Runtime r = Runtime.getRuntime();  
try{  
    r.exec("C:\\Program Files\\Mozilla Firefox\\firefox.exe");  
}  
catch(Exception ex){  
    System.out.println(ex.getMessage());  
}
```

Class Properties

- represents a persistent set of properties as pairs of key-value
- details are here:

<http://docs.oracle.com/javase/tutorial/essential/environment/properties.html>

<http://docs.oracle.com/javase/7/docs/api/java/util/Properties.html>

Configuration File Example

```
dbpassword=pass&word  
database=localhost  
dbuser=vmo
```

Use Properties Example

```
Properties applicationProps = new Properties();  
in = new FileInputStream("appProperties");  
applicationProps.load(in);  
in.close();
```

```
String dbInfo = applicationProps. getProperty("database");
```


Class Object (1 of 2)

- The root of the class hierarchy
- Every class has Object as a superclass
- All objects, including arrays, implement the methods of this class
- See <http://docs.oracle.com/javase/7/docs/api/java/lang/Object.html> for details

Class Object (2 of 2)

- **equals(Object obj)** - indicates whether some other object is "equal to" this one
- **getClass()** - returns the runtime class of this Object
- **clone()** - creates and returns a copy of this object
- **toString()** - returns a string representation of the object

Class Random

- is used to generate a stream of pseudorandom numbers
- **Random(long seed)** - creates a new random number generator
- **nextInt()** - returns the next pseudorandom, uniformly distributed int value
- **nextInt(int n)** - returns a pseudorandom, uniformly distributed int value between 0 (inclusive) and the specified value (exclusive)

See

<http://docs.oracle.com/javase/7/docs/api/java/util/Random.html> for details

Class Math (1 of 2)

- Contains methods for performing basic numeric operations:
 - elementary exponential,
 - logarithm,
 - square root
 - trigonometric functions

See

<http://docs.oracle.com/javase/7/docs/api/java/lang/Math.html> for details

Class Math (2 of 2)

- `sqrt(double value)`
- `exp(double value)`
- `log(double value)`
- `power(double value, double p)`
- `sin(double value)`
- `sinh(double value)`
- `toRadians(double value)`