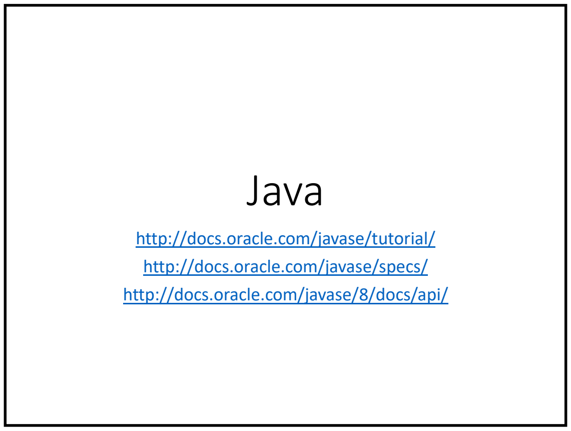
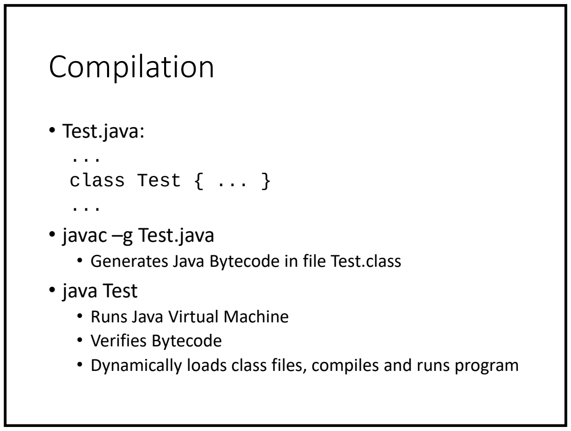


Java

http://docs.oracle.com/javase/tutorial/

http://docs.oracle.com/javase/specs/

http://docs.oracle.com/javase/8/docs/api/

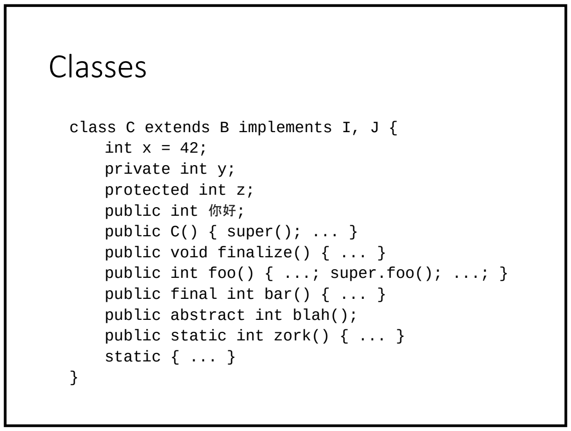


Compilation

- Test.java:

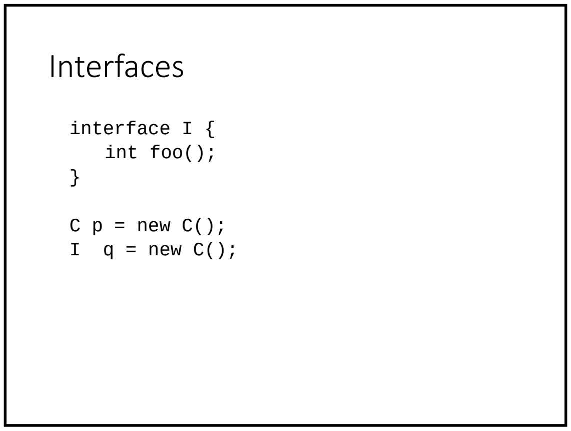
```
...
class Test { ... }
...
```

- javac –g Test.java
  - Generates Java Bytecode in file Test.class
- java Test
  - Runs Java Virtual Machine
  - Verifies Bytecode
  - Dynamically loads class files, compiles and runs program



Classes

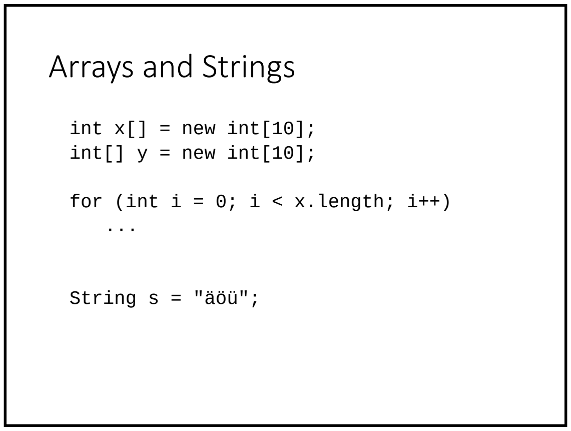
```
class C extends B implements I, J {
    int x = 42;
    private int y;
    protected int z;
    public int 你好;
    public C() { super(); ... }
    public void finalize() { ... }
    public int foo() { ...; super.foo(); ...; }
    public final int bar() { ... }
    public abstract int blah();
    public static int zork() { ... }
    static { ... }
}
```



Interfaces

```
interface I {
   int foo();
}

C p = new C();
I  q = new C();
```

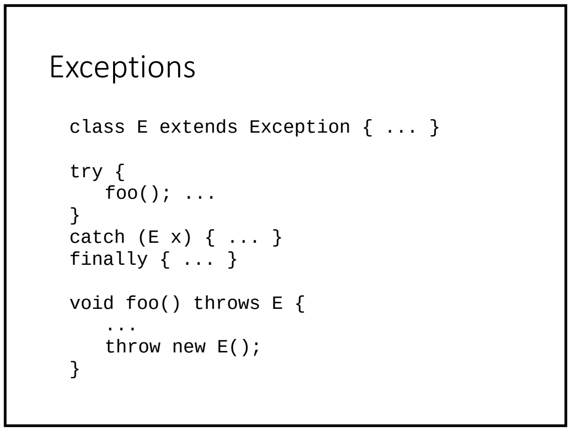


Arrays and Strings

```
int x[] = new int[10];
int[] y = new int[10];

for (int i = 0; i < x.length; i++)
   ...


String s = "äöü";
```

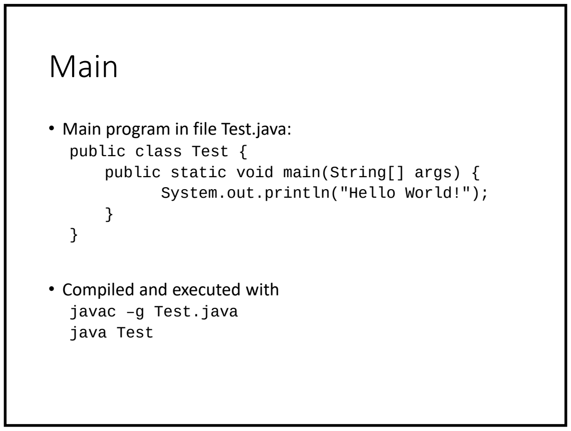


Exceptions

```
class E extends Exception { ... }

try {
   foo(); ...
}
catch (E x) { ... }
finally { ... }

void foo() throws E {
   ...
   throw new E();
}
```



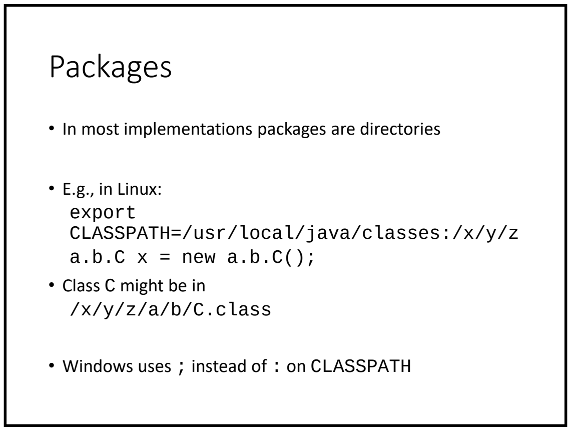
Main

- Main program in file Test.java:

```
public class Test {
    public static void main(String[] args) {
          System.out.println("Hello World!");
    }
}
```

- Compiled and executed with

```
javac –g Test.java
java Test
```



Packages

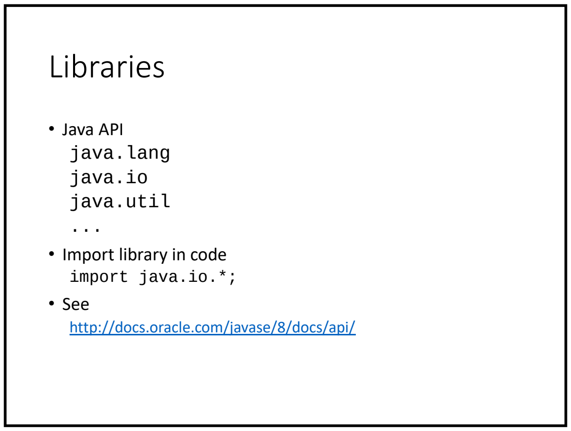
- In most implementations packages are directories
- E.g., in Linux:

```
export
CLASSPATH=/usr/local/java/classes:/x/y/z
a.b.C x = new a.b.C();
```

- Class C might be in

```
/x/y/z/a/b/C.class
```

- Windows uses `;` instead of `:` on CLASSPATH



Libraries

- Java API

```
java.lang
java.io
java.util
...
```

- Import library in code

```
import java.io.*;
```

- See
  http://docs.oracle.com/javase/8/docs/api/