# Exercise Labs 1: Maven

## 01.B - Maven – Hello World

App.java

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | package dblab.hello; | |  |  |  | | --- | |  | |  |  |  | | --- | | public class App { | |  |  |  | | --- | | public static void main( String[] args ) | |  |  |  | | --- | | { | |  |  |  | | --- | | System.out.println( "Hello World!" ); | |  |  |  | | --- | | if (args.length >0) | |  |  |  | | --- | | { | |  |  |  | | --- | | int i=0; | |  |  |  | | --- | | for (String argument : args) | |  |  |  | | --- | | { | |  |  |  | | --- | | System.out.format("argument [%d]: %s %n", i++, argument); | |  |  |  | | --- | | } | |  |  |  | | --- | | } | |  |  |  | | --- | | } | |  |   } |

pom.xml

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"> | |  |  |  | | --- | | <modelVersion>4.0.0</modelVersion> | |  |  |  | | --- | | <groupId>dblab</groupId> | |  |  |  | | --- | | <artifactId>hello</artifactId> | |  |  |  | | --- | | <version>0.0.1-SNAPSHOT</version> | |  |  |  | | --- | | <packaging>jar</packaging> | |  |  |  | | --- | |  | |  |  |  | | --- | | <name>hello</name> | |  |  |  | | --- | | <description>Exercise ISEP</description> | |  |  |  | | --- | |  | |  |  |  | | --- | | <url>http://maven.apache.org</url> | |  |  |  | | --- | |  | |  |  |  | | --- | | <properties> | |  |  |  | | --- | | <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding> | |  |  |  | | --- | | </properties> | |  |  |  | | --- | |  | |  |  |  | | --- | | <build> | |  |  |  | | --- | | <plugins> | |  |  |  | | --- | | <plugin> | |  |  |  | | --- | | <artifactId>maven-assembly-plugin</artifactId> | |  |  |  | | --- | | <version>2.3</version> | |  |  |  | | --- | | <configuration> | |  |  |  | | --- | | <descriptorRefs> | |  |  |  | | --- | | <descriptorRef>jar-with-dependencies</descriptorRef> | |  |  |  | | --- | | </descriptorRefs> | |  |  |  | | --- | | </configuration> | |  |  |  | | --- | | <executions> | |  |  |  | | --- | | <execution> | |  |  |  | | --- | | <id>make-assembly</id> <!-- this is used for inheritance merges --> | |  |  |  | | --- | | <phase>package</phase> <!-- bind to the packaging phase --> | |  |  |  | | --- | | <goals> | |  |  |  | | --- | | <goal>single</goal> | |  |  |  | | --- | | </goals> | |  |  |  | | --- | | </execution> | |  |  |  | | --- | | </executions> | |  |  |  | | --- | | </plugin> | |  |  |  | | --- | |  | |  |  |  | | --- | | <plugin> | |  |  |  | | --- | | <groupId>org.codehaus.mojo</groupId> | |  |  |  | | --- | | <artifactId>exec-maven-plugin</artifactId> | |  |  |  | | --- | | <version>1.2.1</version> | |  |  |  | | --- | | <executions> | |  |  |  | | --- | | <execution> | |  |  |  | | --- | | <goals> | |  |  |  | | --- | | <goal>java</goal> | |  |  |  | | --- | | </goals> | |  |  |  | | --- | | </execution> | |  |  |  | | --- | | </executions> | |  |  |  | | --- | | <configuration> | |  |  |  | | --- | | <mainClass>dblab.hello.App</mainClass> | |  |  |  | | --- | | <arguments> | |  |  |  | | --- | | <argument>foo</argument> | |  |  |  | | --- | | <argument>bar</argument> | |  |  |  | | --- | | </arguments> | |  |  |  | | --- | | </configuration> | |  |  |  | | --- | | </plugin> | |  |  |  | | --- | | </plugins> | |  |  |  | | --- | | </build> | |  |  |  | | --- | |  | |  |  |  | | --- | | <dependencies> | |  |  |  | | --- | | <dependency> | |  |  |  | | --- | | <groupId>junit</groupId> | |  |  |  | | --- | | <artifactId>junit</artifactId> | |  |  |  | | --- | | <version>3.8.1</version> | |  |  |  | | --- | | <scope>test</scope> | |  |  |  | | --- | | </dependency> | |  |  |  | | --- | |  | |  |  |  | | --- | | <!-- https://mvnrepository.com/artifact/log4j/log4j --> | |  |  |  | | --- | | <dependency> | |  |  |  | | --- | | <groupId>log4j</groupId> | |  |  |  | | --- | | <artifactId>log4j</artifactId> | |  |  |  | | --- | | <version>1.2.17</version> | |  |  |  | | --- | | </dependency> | |  |  |  | | --- | |  | |  |  |  | | --- | | </dependencies> | |  |  |  | | --- | |  | |  |   </project> |

AppTest.java

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | package dblab.hello; | |  |  |  | | --- | |  | |  |  |  | | --- | | import junit.framework.Test; | |  |  |  | | --- | | import junit.framework.TestCase; | |  |  |  | | --- | | import junit.framework.TestSuite; | |  |  |  | | --- | |  | |  |  |  | | --- | | /\*\* | |  |  |  | | --- | | \* Unit test for simple App. | |  |  |  | | --- | | \*/ | |  |  |  | | --- | | public class AppTest extends TestCase { | |  |  |  | | --- | |  | |  |  |  | | --- | | /\*\* | |  |  |  | | --- | | \* Create the test case | |  |  |  | | --- | | \* | |  |  |  | | --- | | \* @param testName name of the test case | |  |  |  | | --- | | \*/ | |  |  |  | | --- | | public AppTest( String testName ) | |  |  |  | | --- | | { | |  |  |  | | --- | | super( testName ); | |  |  |  | | --- | | } | |  |  |  | | --- | |  | |  |  |  | | --- | | /\*\* | |  |  |  | | --- | | \* @return the suite of tests being tested | |  |  |  | | --- | | \*/ | |  |  |  | | --- | | public static Test suite() | |  |  |  | | --- | | { | |  |  |  | | --- | | System.out.println("COUCOU"); | |  |  |  | | --- | | return new TestSuite( AppTest.class ); | |  |  |  | | --- | | } | |  |  |  | | --- | |  | |  |  |  | | --- | | /\*\* | |  |  |  | | --- | | \* Rigourous Test :-) | |  |  |  | | --- | | \*/ | |  |  |  | | --- | | public void testApp() | |  |  |  | | --- | | { | |  |  |  | | --- | | assertTrue( true ); | |  |  |  | | --- | | } | |  |  |  | | --- | |  | |  |  |  | | --- | |  | |  |   } |

## Exercise 01.C – Logging exercise 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | package ex01C; | |  |  |  | | --- | |  | |  |  |  | | --- | | import org.apache.log4j.\*; | |  |  |  | | --- | |  | |  |  |  | | --- | | public class Log4jBasics { | |  |  |  | | --- | | protected static Logger log = Logger.getLogger(Log4jBasics.class); | |  |  |  | | --- | |  | |  |  |  | | --- | | public static void main(String[] args) | |  |  |  | | --- | | { | |  |  |  | | --- | | log.debug("DEBUG: Cool !"); | |  |  |  | | --- | | log.info("INFO: Cool !"); | |  |  |  | | --- | | } | |  |   } |

|  |
| --- |
| Q1. In the lo4j.properties, change the level from INFO to DEBUG, what happens? When the level is INFO, only INFO related logs are displayed, as:  12:02:33,306 INFO Log4jBasics:11 - INFO: Cool !  If we change the level from INFO to DEBUG, the output is:  12:01:14,481 DEBUG Log4jBasics:10 - DEBUG: Cool !  12:01:14,482 INFO Log4jBasics:11 - INFO: Cool ! |

## Exercise 01.D – References

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | package ex01D; | |  |  |  | | --- | |  | |  |  |  | | --- | | public class C { | |  |  |  | | --- | | static void method1(int i, StringBuffer s) | |  |  |  | | --- | | { | |  |  |  | | --- | | i++; s.append("d"); | |  |  |  | | --- | | } | |  |  |  | | --- | |  | |  |  |  | | --- | | public static void main(String [] args) | |  |  |  | | --- | | { | |  |  |  | | --- | | int i = 0; | |  |  |  | | --- | | StringBuffer s = new StringBuffer("abc"); | |  |  |  | | --- | | method1(i, s); | |  |  |  | | --- | | System.out.println("i=" + i + ", s=" + s); | |  |  |  | | --- | | } | |  |   } |

## Exercise 01.E: Inheritance

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | package ex01E; | |  |  |  | | --- | |  | |  |  |  | | --- | | class A | |  |  |  | | --- | | { | |  |  |  | | --- | | int x; | |  |  |  | | --- | | void m() | |  |  |  | | --- | | { | |  |  |  | | --- | | System.out.println("Je suis dans la méthode m d'une instance de A"); | |  |  |  | | --- | | } | |  |  |  | | --- | | } | |  |  |  | | --- | |  | |  |  |  | | --- | | class B extends A | |  |  |  | | --- | | { | |  |  |  | | --- | | int x; | |  |  |  | | --- | | void m() { | |  |  |  | | --- | | System.out.println("Je suis dans la méthode m d'une instance de B"); | |  |  |  | | --- | | } | |  |  |  | | --- | | } | |  |  |  | | --- | |  | |  |  |  | | --- | | class C extends B | |  |  |  | | --- | | { | |  |  |  | | --- | | int x, a; | |  |  |  | | --- | | void m() { | |  |  |  | | --- | | System.out.println("Je suis dans la méthode m d'une instance de C"); | |  |  |  | | --- | | } | |  |  |  | | --- | |  | |  |  |  | | --- | | void test() { | |  |  |  | | --- | | a = super.x; | |  |  |  | | --- | | // a = super.super.x; // super.super is not possible, as it breaks encapsulation. You shouldn't be able to bypass the parent class's behavior. | |  |  |  | | --- | | a = ((B)this).x; | |  |  |  | | --- | | a = ((A)this).x; | |  |  |  | | --- | | super.m(); | |  |  |  | | --- | | // super.super.m(); | |  |  |  | | --- | | ((B)this).m(); // (1) | |  |  |  | | --- | | } | |  |  |  | | --- | | } | |  |  |  | | --- | |  | |  |  |  | | --- | | public class InheritanceExample { | |  |  |  | | --- | | public static void main(String [] args) { | |  |  |  | | --- | | C c = new C(); | |  |  |  | | --- | | c.test(); | |  |  |  | | --- | | } | |  |   } |

|  |
| --- |
| Q1. Which m() method is called in (1)? m() method in class C  Q2. Is it the expected result?  Yes  Q3. Which OOP implementation type fits to that result?  Virtual function overloading |

## Exercise 01.F: toString() overloading

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | package ex01F; | |  |  |  | | --- | |  | |  |  |  | | --- | | import org.apache.log4j.Logger; | |  |  |  | | --- | |  | |  |  |  | | --- | | public class Circle | |  |  |  | | --- | | { | |  |  |  | | --- | | protected static Logger log = Logger.getLogger(Circle.class); | |  |  |  | | --- | |  | |  |  |  | | --- | | int x; | |  |  |  | | --- | | int y; | |  |  |  | | --- | | int radius; | |  |  |  | | --- | |  | |  |  |  | | --- | | public Circle(int x, int y, int radius) | |  |  |  | | --- | | { | |  |  |  | | --- | | this.x = x; | |  |  |  | | --- | | this.y = y; | |  |  |  | | --- | | this.radius = radius; | |  |  |  | | --- | | } | |  |  |  | | --- | |  | |  |  |  | | --- | | public String toString() | |  |  |  | | --- | | { | |  |  |  | | --- | | return( | |  |  |  | | --- | | String.format("Circle with center (%d,%d) and radius %d (Perimter is %,.2f)", | |  |  |  | | --- | | this.x, | |  |  |  | | --- | | this.y, | |  |  |  | | --- | | this.radius, | |  |  |  | | --- | | (2 \* java.lang.Math.PI \* this.radius) | |  |  |  | | --- | | ) | |  |  |  | | --- | | ); | |  |  |  | | --- | | } | |  |   } |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | package ex01F; | |  |  |  | | --- | |  | |  |  |  | | --- | | public class ToStringOverloading { | |  |  |  | | --- | | public static void main(String[] args) | |  |  |  | | --- | | { | |  |  |  | | --- | | Circle c1 = new Circle(0,0, 5); | |  |  |  | | --- | | Circle c2 = new Circle(0,0, 3); | |  |  |  | | --- | |  | |  |  |  | | --- | | System.out.println("C1 => " + c1); | |  |  |  | | --- | | System.out.println("C2 => " + c2); | |  |  |  | | --- | | } | |  |   } |

|  |
| --- |
| Q1. What shows up? Why is the method toString() called without being named? java.lang.Object is a super class of any class by default which results the string representation of an object. In our case, the Circle class overrides the toString() implementation of java.lang.Object class.  Q2. If you rename the method toString(), what happens?  It calls the default toString() method from java.lang.Object |

# Labs 2: Java Standard API

## java.util

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | package ex021; | |  |  |  | | --- | |  | |  |  |  | | --- | | import java.io.File; | |  |  |  | | --- | | import java.util.Enumeration; | |  |  |  | | --- | | import java.util.zip.ZipEntry; | |  |  |  | | --- | | import java.util.zip.ZipFile; | |  |  |  | | --- | | import java.util.Date; | |  |  |  | | --- | | import java.io.FileNotFoundException; | |  |  |  | | --- | | import java.io.IOException; | |  |  |  | | --- | |  | |  |  |  | | --- | | public class JavaUtilZipExample { | |  |  |  | | --- | | private static ZipFile zipFile = null; | |  |  |  | | --- | |  | |  |  |  | | --- | | public static void main(String [] args) throws IOException { | |  |  |  | | --- | | try { | |  |  |  | | --- | | ClassLoader classLoader = JavaUtilZipExample.class.getClassLoader(); | |  |  |  | | --- | | File file = new File(classLoader.getResource("SampleZipFile.zip").getFile()); | |  |  |  | | --- | |  | |  |  |  | | --- | | // ZipFile has problems parsing file path with spaces (%20) | |  |  |  | | --- | | zipFile = new ZipFile(file.getAbsolutePath().replace("%20", " ")); | |  |  |  | | --- | |  | |  |  |  | | --- | | Enumeration<? extends ZipEntry> entries = zipFile .entries(); | |  |  |  | | --- | |  | |  |  |  | | --- | | while(entries.hasMoreElements()){ | |  |  |  | | --- | | ZipEntry entry = entries.nextElement(); | |  |  |  | | --- | | if(entry.isDirectory()){ | |  |  |  | | --- | | System.out.println("Dir Name : " + entry.getName()); | |  |  |  | | --- | | } else { | |  |  |  | | --- | | Date date = new Date(entry.getTime()); | |  |  |  | | --- | |  | |  |  |  | | --- | | System.out.println("File Name : " + entry.getName() + | |  |  |  | | --- | | "\t| Date : " + date.getDate()+ | |  |  |  | | --- | | "\t| Hour : " + date.getHours() + | |  |  |  | | --- | | "\t| Size : " + entry.getSize() + | |  |  |  | | --- | | "\t| Packed Size : " + entry.getCompressedSize()); | |  |  |  | | --- | | } | |  |  |  | | --- | | } | |  |  |  | | --- | | } catch(FileNotFoundException ex) { | |  |  |  | | --- | | System.out.println("File not found : " + ex.getMessage()); | |  |  |  | | --- | | } catch(IOException ex) { | |  |  |  | | --- | | System.out.println("Exception : " + ex.getMessage()); | |  |  |  | | --- | | } finally { | |  |  |  | | --- | | if (zipFile != null) zipFile.close(); | |  |  |  | | --- | | } | |  |  |  | | --- | | } | |  |   } |

|  |
| --- |
| Q1. Why are these compression classes included in the Java language? Provides classes for reading and writing the standard ZIP formats. It can compress as well as decompress the ZIP formats. |

## File copy

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | package ex022; | |  |  |  | | --- | |  | |  |  |  | | --- | | import java.io.BufferedInputStream; | |  |  |  | | --- | | import java.io.BufferedOutputStream; | |  |  |  | | --- | | import java.io.File; | |  |  |  | | --- | | import java.io.FileInputStream; | |  |  |  | | --- | | import java.io.FileNotFoundException; | |  |  |  | | --- | | import java.io.FileOutputStream; | |  |  |  | | --- | | import java.io.IOException; | |  |  |  | | --- | |  | |  |  |  | | --- | | class FileCopier { | |  |  |  | | --- | |  | |  |  |  | | --- | | private String \_source; | |  |  |  | | --- | | private String \_destination; | |  |  |  | | --- | | private BufferedInputStream \_bin = null; | |  |  |  | | --- | | private BufferedOutputStream \_bout = null; | |  |  |  | | --- | |  | |  |  |  | | --- | | FileCopier(String sourceFilePath, String destinationFilePath) { | |  |  |  | | --- | | \_source = sourceFilePath; | |  |  |  | | --- | | \_destination = destinationFilePath; | |  |  |  | | --- | | } | |  |  |  | | --- | |  | |  |  |  | | --- | | boolean startCopy() throws IOException { | |  |  |  | | --- | | try { | |  |  |  | | --- | | \_bin = new BufferedInputStream(new FileInputStream(\_source)); | |  |  |  | | --- | | \_bout = new BufferedOutputStream(new FileOutputStream(\_destination)); | |  |  |  | | --- | |  | |  |  |  | | --- | | byte[] buffer = new byte[32 \* 1024]; | |  |  |  | | --- | | int numBytes; | |  |  |  | | --- | | while ((numBytes = \_bin.read(buffer))!= -1) | |  |  |  | | --- | | \_bout.write(buffer, 0, numBytes); | |  |  |  | | --- | |  | |  |  |  | | --- | | System.out.println(\_source + " is successfully copied to "+ \_destination); | |  |  |  | | --- | | return true; | |  |  |  | | --- | | } catch(FileNotFoundException ex) { | |  |  |  | | --- | | System.out.println("File not found : " + ex.getMessage()); | |  |  |  | | --- | | return false; | |  |  |  | | --- | | } catch(IOException ex) { | |  |  |  | | --- | | System.out.println("Exception : " + ex.getMessage()); | |  |  |  | | --- | | return false; | |  |  |  | | --- | | } finally { | |  |  |  | | --- | | if (\_bin != null) \_bin.close(); | |  |  |  | | --- | | if (\_bout != null) \_bout.close(); | |  |  |  | | --- | | } | |  |  |  | | --- | | } | |  |  |  | | --- | |  | |  |  |  | | --- | | } | |  |  |  | | --- | |  | |  |  |  | | --- | | public class FileCopyExample { | |  |  |  | | --- | |  | |  |  |  | | --- | | public static void main(String[] args) throws IOException { | |  |  |  | | --- | |  | |  |  |  | | --- | | // get file from resources | |  |  |  | | --- | | ClassLoader classLoader = FileCopyExample.class.getClassLoader(); | |  |  |  | | --- | | File sourceFile = new File(classLoader.getResource("SampleFileCopy.txt").getFile()); | |  |  |  | | --- | |  | |  |  |  | | --- | | // get the desktop path | |  |  |  | | --- | | String desktopPath = System.getProperty("user.home") + "\\Desktop"; | |  |  |  | | --- | |  | |  |  |  | | --- | | FileCopier fc = new FileCopier(sourceFile.getAbsolutePath().replace("%20", " "), desktopPath + "\\" + "OutputFile.txt"); | |  |  |  | | --- | | fc.startCopy(); | |  |  |  | | --- | |  | |  |  |  | | --- | | } | |  |  |  | | --- | |  | |  |   } |

|  |
| --- |
| Q1. Why is it interesting to use buffered streams? The BufferedStreams class provides buffering to your streams. Buffering can speed up IO quite a bit. Rather than read/write one byte at a time from the network or disk, the BufferedStream reads/writes a larger block at a time into an internal buffer. When you read a byte from the BufferedInputStream you are therefore reading it from its internal buffer. When the buffer is fully read, the BufferedInputStream reads another larger block of data into the buffer. This is typically much faster than reading a single byte at a time from an InputStream, especially for disk access and larger data amounts. |

## Word counter

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | |  | | package ex023; | | |  | |  |  | | --- | |  | |  |  |  | | --- | | import java.io.FileReader; | |  |  |  | | --- | | import java.io.IOException; | |  |  |  | | --- | | import java.io.BufferedReader; | |  |  |  | | --- | | import java.io.File; | |  |  |  | | --- | | import java.io.FileNotFoundException; | |  |  |  | | --- | | import java.io.FileWriter; | |  |  |  | | --- | | import java.util.StringTokenizer; | |  |  |  | | --- | |  | |  |  |  | | --- | | class WordCounter { | |  |  |  | | --- | |  | |  |  |  | | --- | | private String \_source; | |  |  |  | | --- | | private String \_log; | |  |  |  | | --- | | private FileWriter \_fw = null; | |  |  |  | | --- | | private BufferedReader \_br = null; | |  |  |  | | --- | |  | |  |  |  | | --- | | WordCounter(String sourceFilePath, String logFilePath) { | |  |  |  | | --- | | \_source = sourceFilePath; | |  |  |  | | --- | | \_log = logFilePath; | |  |  |  | | --- | | } | |  |  |  | | --- | |  | |  |  |  | | --- | | boolean start() throws IOException { | |  |  |  | | --- | | try { | |  |  |  | | --- | | \_br = new BufferedReader(new FileReader(\_source)); | |  |  |  | | --- | | String line = "", str = ""; | |  |  |  | | --- | | int words = 0; | |  |  |  | | --- | | while ((line = \_br.readLine()) != null) { | |  |  |  | | --- | | str += line + " "; | |  |  |  | | --- | | } | |  |  |  | | --- | |  | |  |  |  | | --- | | StringTokenizer st = new StringTokenizer(str); | |  |  |  | | --- | | while (st.hasMoreTokens()) { | |  |  |  | | --- | | st.nextToken(); | |  |  |  | | --- | | words++; | |  |  |  | | --- | | } | |  |  |  | | --- | |  | |  |  |  | | --- | | \_fw = new FileWriter(\_log); | |  |  |  | | --- | | \_fw.write(\_source + " has " + words + " words"); | |  |  |  | | --- | | System.out.println("Log file created at : " + \_log); | |  |  |  | | --- | | return true; | |  |  |  | | --- | | } catch(FileNotFoundException ex) { | |  |  |  | | --- | | System.out.println("File not found : " + ex.getMessage()); | |  |  |  | | --- | | return false; | |  |  |  | | --- | | } catch(IOException ex) { | |  |  |  | | --- | | System.out.println("Exception : " + ex.getMessage()); | |  |  |  | | --- | | return false; | |  |  |  | | --- | | } finally { | |  |  |  | | --- | | if (\_br != null) \_br.close(); | |  |  |  | | --- | | if (\_fw != null) \_fw.close(); | |  |  |  | | --- | | } | |  |  |  | | --- | | } | |  |  |  | | --- | | } | |  |  |  | | --- | |  | |  |  |  | | --- | | public class WordCounterExample { | |  |  |  | | --- | |  | |  |  |  | | --- | | public static void main(String[] args) throws IOException { | |  |  |  | | --- | |  | |  |  |  | | --- | | // get file from resources | |  |  |  | | --- | | ClassLoader classLoader = WordCounterExample.class.getClassLoader(); | |  |  |  | | --- | | File sourceFile = new File(classLoader.getResource("iso\_8859-1.txt").getFile()); | |  |  |  | | --- | |  | |  |  |  | | --- | | // get the desktop path | |  |  |  | | --- | | String desktopPath = System.getProperty("user.home") + "\\Desktop"; | |  |  |  | | --- | |  | |  |  |  | | --- | | WordCounter wc = new WordCounter(sourceFile.getAbsolutePath().replace("%20", " "), desktopPath + "\\" + "WordCount.log"); | |  |  |  | | --- | | wc.start(); | |  |  |  | | --- | |  | |  |  |  | | --- | | } | |  |  |  | | --- | |  | |  |   } |

|  |  |
| --- | --- |
| |  | | --- | |  |  Q1. What is the difference between this stream Reader/Writer et InputStream/OutputStream? OutputStream classes writes to the target byte by byte where as Writer classes writes to the target character by character Writers provide more flexibility in that they can write characters and even strings while taking a special encoding into account. Q2. Which text network protocols do you know?Q3. How can the objects discovered here be relevant in the analysis of such protocols? Whenever you want to handle the binary data then you need to use the InputStream/OutputStream classes (generally, everything that contains Stream in its name).  That's why there's a FileInputStream and a FileOutputStream: those read from and write to files and they handle binary data.  Whenever you want to handle text data, then you need to use the Reader/Writer classes. |