Mitchell Ludwig

CPSC 501

Assignment 2

Tutorial 1

10015370

Version Control

Github

My git repository is hosted on github, where you can find the extensive logs.

Refactorings

hash f9a1c7: Added initial files

hash 0c6285: Applied Rename Method

hash 3ae6dc: Applied Extract Method

hash 167d84: Spiritually applied Extract Method, actually applied Extract Method. Completed project

Tests

hash dede5e: Publicized methods to make testing easier

hash e3964a: Added test for tabIn, tabOut, and forTheRecord

hash 0fcf33: Added test for getNiceName

hash 0c6285: Added more tests for isPrimitiveWrapper and methodParameters

hash 3ae6dc: Added test for printObject

hash 9c9bec: Added test for the actual inspect function

Unit Tests

```
import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertFalse;
import static org.junit.Assert.assertTrue;
import static org.junit.Assert.fail;

import org.junit.Test;

import asg2.Primes;

public class InspectorTest {

  @Test
  public void testTabs() {
        Inspector i = new Inspector();
        assertEquals(0, i._iTabLevel);
        i.tabIn();
        assertEquals(1, i._iTabLevel);
        System.out.println("Verify that there is 1 tab here");
        i.forTheRecordLn("Test");
        i.tabIn();
        assertEquals(2, i._iTabLevel);
        System.out.println("Verify that there are 2 tabs here");
        i.forTheRecordLn("Test for 2");
        i.tabIn();
        assertEquals(3, i._iTabLevel);
        i.tabIn();
        assertEquals(4, i._iTabLevel);
        i.tabOut();
        assertEquals(3, i._iTabLevel);
        i.tabOut();
        assertEquals(2, i._iTabLevel);
        i.tabOut();
        assertEquals(1, i._iTabLevel);
        i.tabOut();
        assertEquals(0, i._iTabLevel);
  }

  @Test
  public void testGetNiceName() {
        Inspector i = new Inspector();
        Primes p = new Primes(1000);
        Primes[] pa = new Primes[100];
        Primes[][] paa = new Primes[10][5];
        String sExpect = "asg2.Primes";
        assertEquals(sExpect, i.getNiceName(p.getClass()));
        assertEquals(sExpect + "[]", i.getNiceName(pa.getClass()));
		assertEquals(sExpect + "[][]", i.getNiceName(paa.getClass()));
		sExpect = "Interface java.util.Set";
		assertEquals(sExpect, i.getNiceName(java.util.Set.class));
		try {
			i.getNiceName(null);
			fail("This was supposed to die");
		} catch (Exception e) {
			// Intended behavior
		}
	}

	@Test
	public void testConstructorParameters() throws NoSuchMethodException,
SecurityException {
		Inspector i = new Inspector();
		Primes p = new Primes(1000);
		assertEquals("", i.constructorParameters(p.getClass().getConstructor(null)));
		assertEquals("int",
i.constructorParameters(p.getClass().getConstructor(int.class)));
	}

	@Test
	public void testMethodExceptions() throws NoSuchMethodException, SecurityException
{
		Inspector i = new Inspector();
		Primes p = new Primes(1000);
		assertEquals("throws java.lang.Exception",
i.methodExceptions(p.getClass().getMethod("areCoprime", int.class, int.class)));
		assertEquals("", i.methodExceptions(p.getClass().getMethod("isPrime",
int.class)));
	}

	@Test
	public void testMethodParameters() throws NoSuchMethodException,
SecurityException {
		Inspector i = new Inspector();
		Primes p = new Primes(1000);
		assertEquals("int, int",
i.methodParameters(p.getClass().getMethod("areCoprime", int.class, int.class)));
	}

	@Test
	public void testIsPrimitiveWrapper() {
		Inspector i = new Inspector();
		Primes p = new Primes(1000);
		Integer iWrapper = new Integer(5);
		assertTrue(i.isPrimitiveWrapper(iWrapper.getClass()));
		assertFalse(i.isPrimitiveWrapper(p.getClass()));
```

```
    }

    @Test
    public void testPrintObject() {
            Inspector i = new Inspector();
            Primes p = new Primes(1000);
            Integer iWrapper = new Integer(5);
            Integer[] iaWrapper = new Integer[] { 5, 4, 3, 2, 1 };
            Integer[][] iaaWrapper = new Integer[][] { { 5, 4, 3, 2, 1 }, { 5, 4, 3, 2, 1 } };
            System.out.println("Verify that it prints 5");
            i.printObjectOrArray(iWrapper);
            System.out.println();
            System.out.println("Verify that it prints [5, 4, 3, 2, 1]");
            i.printObjectOrArray(iaWrapper);
            System.out.println();
            System.out.println("Verify that it prints [[5, 4, 3, 2, 1], [5, 4, 3, 2, 1]]");
            i.printObjectOrArray(iaaWrapper);
            System.out.println();
    }

    @Test
    public void testAAInspection() {
            Inspector i = new Inspector();
            System.out.println("====== Verify that this looks OK =====");
            i.inspect(new Primes(100), true);

            System.out.println("======================================");
    }
}
```

Inspector

```
import java.lang.reflect.Array;

import java.lang.reflect.Constructor;

import java.lang.reflect.Field;

import java.lang.reflect.Method;

import java.lang.reflect.Modifier;

import java.util.Arrays;

import java.util.LinkedHashSet;


public class Inspector {

  public int                                   _iTabLevel       = 0;

  public Object                                _oInput;

  public Class<?>                              _cInput;

  public LinkedHashSet<Class<?>>       _clInheritanceHeirarchy;

  public boolean                               _bRecursive;


  public void inspect(Object obj, boolean recursive) {

          if (obj == null) {

                  System.out.println("Object is null");

                  return;

          }


          _bRecursive = recursive;

          _oInput = obj;

          _cInput = obj.getClass();

          _clInheritanceHeirarchy = getInheritanceHeirarchy();


          forTheRecordLn("Name: " + getNiceName(_cInput));

          tabIn();

          forTheRecordLn("Immediate Superclass: " + getNiceName(_cInput.getSuperclass()));

          printInterfaces();
		printConstructors();

		printMethods();

		printFields();

		tabOut();

	}


	public LinkedHashSet<Class<?>> getInheritanceHeirarchy() {

		LinkedHashSet<Class<?>> clCHL = new LinkedHashSet<Class<?>>();

		if (_cInput != null) {

			clCHL.addAll(superInterfaceHeirarchyList(_cInput));

			Class<?> cSuper = _cInput.getSuperclass();

			while (cSuper != null) {

				clCHL.add(cSuper);

				clCHL.addAll(superInterfaceHeirarchyList(cSuper));

				cSuper = cSuper.getSuperclass();

			}

		}

		return clCHL;

	}


	public LinkedHashSet<Class<?>> superInterfaceHeirarchyList(Class<?> c) {

		LinkedHashSet<Class<?>> clIHL = new LinkedHashSet<Class<?>>();

		for (Class<?> cInterface : c.getInterfaces()) {

			clIHL.add(cInterface);

			clIHL.addAll(superInterfaceHeirarchyList(cInterface));

		}

		return clIHL;

	}


	public void printInterfaces() {

		forTheRecordLn("Interfaces:");
```

```
			tabIn();
			for (Class<?> cInterface : _cInput.getInterfaces()) {
				forTheRecordLn(getNiceName(cInterface));
			}
			for (Class<?> cSuper : _clInheritanceHeirarchy) {
				for (Class<?> cInterface : cSuper.getInterfaces()) {
					forTheRecordLn("[From " + getNiceName(cSuper) + "] " +
getNiceName(cInterface));
				}
			}
			tabOut();
		}


		public void printConstructors() {
			forTheRecordLn("Constructors:");
			tabIn();
			for (Constructor<?> cConstr : _cInput.getConstructors()) {
				forTheRecordLn(Modifier.toString(cConstr.getModifiers()) + " " + getNiceName(_cInput)
+ "(" + constructorParameters(cConstr) + ")");
			}
			for (Class<?> c : _clInheritanceHeirarchy) {
				for (Constructor<?> cConstr : c.getConstructors()) {
					forTheRecordLn(Modifier.toString(cConstr.getModifiers()) + " " +
getNiceName(c) + "(" + constructorParameters(cConstr) + ")");
				}
			}
			tabOut();
		}


		public void printMethods() {
			forTheRecordLn("Methods:");
			tabIn();
		for (Method m : _cInput.getDeclaredMethods()) {

			forTheRecordLn(Modifier.toString(m.getModifiers()) + " " +
getNiceName(m.getReturnType()) + " " + m.getName() + "(" + methodParameters(m) + ") " +
methodExceptions(m));

		}

		for (Class<?> c : _clInheritanceHeirarchy) {

			for (Method m : c.getDeclaredMethods()) {

				forTheRecordLn("[From " + getNiceName(c) + "] " +
Modifier.toString(m.getModifiers()) + " " + getNiceName(m.getReturnType()) + " " + m.getName() + "(" +
methodParameters(m) + ") " + methodExceptions(m));

			}

		}

		tabOut();

	}


	public void printFields() {

		forTheRecordLn("Fields:");

		tabIn();

		printFields(_cInput, "");

		for (Class<?> c : _clInheritanceHeirarchy) {

			printFields(c, "[From " + getNiceName(c) + "] ");

		}

		tabOut();

	}


	public void printFields(Class<?> cObject, String sPrepend) {

		Object oFieldValue;

		for (Field f : cObject.getDeclaredFields()) {

			try {

				f.setAccessible(true);

				oFieldValue = f.get(_oInput);

				if (oFieldValue == null) {
```

```
                                        forTheRecordLn(sPrepend + Modifier.toString(f.getModifiers()) + " " 
+ getNiceName(f.getType()) + " " + f.getName() + " = null");

                                    } else {

                                        forTheRecord(sPrepend + Modifier.toString(f.getModifiers()) + " " + 
getNiceName(f.getType()) + " " + f.getName() + " = ");

                                        printObjectOrArray(oFieldValue);

                                        System.out.println();

                                    }

                            } catch (IllegalArgumentException e) {

                                    forTheRecordLn(sPrepend + Modifier.toString(f.getModifiers()) + " " + 
getNiceName(f.getType()) + " " + f.getName() + " = *IllegalArgument*");

                            } catch (IllegalAccessException e) {

                                    forTheRecordLn(sPrepend + Modifier.toString(f.getModifiers()) + " " + 
getNiceName(f.getType()) + " " + f.getName() + " = *IllegalAccess*");

                            }

                    }

            }



            public void printObjectOrArray(Object oInput) {

                    if (oInput == null) {

                            System.out.print("null");

                            return;

                    }

                    if (oInput.getClass().isArray()) {

                            printArray(oInput);

                    } else {

                            if (_bRecursive) {

                                    if (isPrimitiveWrapper(oInput.getClass())) {

                                            System.out.print(oInput);

                                            return;

                                    } else {

                                            Inspector i = new Inspector();

                                            System.out.println();
                    i._iTabLevel = this._iTabLevel + 1;
                    i.inspect(oInput, _bRecursive);
                    return;
                }
            } else {
                System.out.print(oInput);
                return;
            }
        }
    }

    public void printArray(Object oInput) {
        int iLength;
        Object[] oaInput;
        if (oInput instanceof Object[]) {
            System.out.print("[");
            oaInput = (Object[]) oInput;
            for (int i = 0; i < oaInput.length; i++) {
                printObjectOrArray(oaInput[i]);
                if (i != oaInput.length - 1) {
                    System.out.print(", ");
                }
            }
            System.out.print("]");
            return;
        } else {
            iLength = Array.getLength(oInput);
            Object[] objArr = new Object[iLength];
            for (int i = 0; i < iLength; i++) {
                objArr[i] = Array.get(oInput, i);
            }
```

```
                System.out.print(Arrays.deepToString(objArr));
                return;
            }
        }

        public boolean isPrimitiveWrapper(Class<?> c) {
            String sName = c.getName();

            switch (sName) {
                case "java.lang.Byte":
                case "java.lang.Short":
                case "java.lang.Integer":
                case "java.lang.Long":
                case "java.lang.Float":
                case "java.lang.Double":
                case "java.lang.Character":
                case "java.lang.Boolean":
                    return true;
                default:
            }
            return false;
        }

        public String getNiceName(Class<?> c) {
            if (c.isArray()) {
                return getNiceName(c.getComponentType()) + "[]";
            } else {
                if (c.isInterface()) {
                    return "Interface " + c.getName();
                } else {
                    return c.getName();
                }
        }
}

public String constructorParameters(Constructor<?> c) {
        StringBuffer sb = new StringBuffer();
        String sResult;
        for (Class<?> cParam : c.getParameterTypes()) {
                sb.append(getNiceName(cParam) + ", ");
        }
        sResult = sb.toString();
        if (sResult.isEmpty()) {
                return sResult;
        } else {
                return sResult.substring(0, sResult.length() - 2);
        }
}

public String methodExceptions(Method m) {
        StringBuffer sb = new StringBuffer();
        String sResult;
        for (Class<?> cException : m.getExceptionTypes()) {
                sb.append(getNiceName(cException) + ", ");
        }
        sResult = sb.toString();
        if (sResult.isEmpty()) {
                return sResult;
        } else {
                return "throws " + sResult.substring(0, sResult.length() - 2);
        }
}
```

```
public String methodParameters(Method m) {
		StringBuffer sb = new StringBuffer();
		String sResult;
		for (Class<?> cParam : m.getParameterTypes()) {
				sb.append(getNiceName(cParam) + ", ");
		}
		sResult = sb.toString();
		if (sResult.isEmpty()) {
				return sResult;
		} else {
				return sResult.substring(0, sResult.length() - 2);
		}
}

public void forTheRecordLn(String s) {
		for (int i = 0; i < _iTabLevel; i++) {
				System.out.print("\t");
		}
		System.out.println(s);
}

public void forTheRecord(String s) {
		for (int i = 0; i < _iTabLevel; i++) {
				System.out.print("\t");
		}
		System.out.print(s);
}

public void tabIn() {
		_iTabLevel++;
```

```
        }

        public void tabOut() {
                _iTabLevel--;
        }
    }
```

Output

```
Loading object inspector: Inspector
======================================================
Running Test: ClassA
Name: ClassA
        Immediate Superclass: java.lang.Object
        Interfaces:
                Interface java.io.Serializable
                Interface java.lang.Runnable
        Constructors:
                public ClassA()
                public ClassA(int)
                public java.lang.Object()
        Methods:
                public void run()
                public java.lang.String toString()
                public void setVal(int) throws java.lang.Exception
                public int getVal()
                private void printSomething()
                [From Interface java.lang.Runnable] public abstract void run()
                [From java.lang.Object] protected void finalize() throws java.lang.Throwable
                [From java.lang.Object] public final void wait(long, int) throws
java.lang.InterruptedException
                [From java.lang.Object] public final native void wait(long) throws
java.lang.InterruptedException
                [From java.lang.Object] public final void wait() throws java.lang.InterruptedException
                [From java.lang.Object] public boolean equals(java.lang.Object)
                [From java.lang.Object] public java.lang.String toString()
                [From java.lang.Object] public native int hashCode()
                [From java.lang.Object] public final native java.lang.Class getClass()
                [From java.lang.Object] protected native java.lang.Object clone() throws
java.lang.CloneNotSupportedException
                [From java.lang.Object] private static native void registerNatives()
                [From java.lang.Object] public final native void notify()
                [From java.lang.Object] public final native void notifyAll()
        Fields:
                private int val = 3
                private double val2 = 0.2
                private boolean val3 = true
======================================================
======================================================
Running Test: ClassA
Name: ClassA
        Immediate Superclass: java.lang.Object
        Interfaces:
                Interface java.io.Serializable
                Interface java.lang.Runnable
        Constructors:
                public ClassA()
                public ClassA(int)
                public java.lang.Object()
        Methods:
                public void run()
```

```
		public java.lang.String toString()
		public void setVal(int) throws java.lang.Exception
		public int getVal()
		private void printSomething()
		[From Interface java.lang.Runnable] public abstract void run()
		[From java.lang.Object] protected void finalize() throws java.lang.Throwable
		[From java.lang.Object] public final void wait(long, int) throws 
java.lang.InterruptedException
		[From java.lang.Object] public final native void wait(long) throws 
java.lang.InterruptedException
		[From java.lang.Object] public final void wait() throws java.lang.InterruptedException
		[From java.lang.Object] public boolean equals(java.lang.Object)
		[From java.lang.Object] public java.lang.String toString()
		[From java.lang.Object] public native int hashCode()
		[From java.lang.Object] public final native java.lang.Class getClass()
		[From java.lang.Object] protected native java.lang.Object clone() throws 
java.lang.CloneNotSupportedException
		[From java.lang.Object] private static native void registerNatives()
		[From java.lang.Object] public final native void notify()
		[From java.lang.Object] public final native void notifyAll()
	Fields:
		private int val = 12
		private double val2 = 0.2
		private boolean val3 = true
======================================================
======================================================
Running Test: ClassB
Name: ClassB
	Immediate Superclass: ClassC
	Interfaces:
		Interface java.lang.Runnable
		[From ClassC] Interface InterfaceA
		[From Interface InterfaceA] Interface InterfaceB
	Constructors:
		public ClassB()
		public ClassC()
		public ClassC(int, int)
		public ClassD()
		public ClassD(int)
		public java.lang.Object()
	Methods:
		public void run()
		public java.lang.String toString()
		public void func3(int)
		[From Interface java.lang.Runnable] public abstract void run()
		[From ClassC] public void run()
		[From ClassC] public java.lang.String toString()
		[From ClassC] public abstract void func3(int)
		[From ClassC] public void func0(int, boolean) throws java.lang.Exception
		[From ClassC] public void func1(int, double, boolean, java.lang.String) throws 
java.lang.Exception
		[From ClassC] public int func2(java.lang.String) throws java.lang.Exception, 
java.lang.ArithmeticException, java.lang.IllegalMonitorStateException
		[From Interface InterfaceA] public abstract void func1(int, double, boolean, 
java.lang.String) throws java.lang.Exception
		[From Interface InterfaceA] public abstract int func2(java.lang.String) throws 
java.lang.Exception, java.lang.ArithmeticException, java.lang.IllegalMonitorStateException
		[From Interface InterfaceB] public abstract void func0(int, boolean) throws 
java.lang.Exception
		[From ClassD] public java.lang.String toString()
		[From ClassD] public int getVal3()
		[From java.lang.Object] protected void finalize() throws java.lang.Throwable
		[From java.lang.Object] public final void wait(long, int) throws 
java.lang.InterruptedException
		[From java.lang.Object] public final native void wait(long) throws 
java.lang.InterruptedException
		[From java.lang.Object] public final void wait() throws java.lang.InterruptedException
		[From java.lang.Object] public boolean equals(java.lang.Object)
		[From java.lang.Object] public java.lang.String toString()
		[From java.lang.Object] public native int hashCode()
		[From java.lang.Object] public final native java.lang.Class getClass()
		[From java.lang.Object] protected native java.lang.Object clone() throws
java.lang.CloneNotSupportedException
		[From java.lang.Object] private static native void registerNatives()
		[From java.lang.Object] public final native void notify()
		[From java.lang.Object] public final native void notifyAll()
	Fields:
		private ClassA val =
			Name: ClassA
				Immediate Superclass: java.lang.Object
				Interfaces:
					Interface java.io.Serializable
					Interface java.lang.Runnable
				Constructors:
					public ClassA()
					public ClassA(int)
					public java.lang.Object()
				Methods:
					public void run()
					public java.lang.String toString()
					public void setVal(int) throws java.lang.Exception
					public int getVal()
					private void printSomething()
					[From Interface java.lang.Runnable] public abstract void run()
					[From java.lang.Object] protected void finalize() throws
java.lang.Throwable
					[From java.lang.Object] public final void wait(long, int) throws
java.lang.InterruptedException
					[From java.lang.Object] public final native void wait(long)
throws java.lang.InterruptedException
					[From java.lang.Object] public final void wait() throws
java.lang.InterruptedException
					[From java.lang.Object] public boolean equals(java.lang.Object)
					[From java.lang.Object] public java.lang.String toString()
					[From java.lang.Object] public native int hashCode()
					[From java.lang.Object] public final native java.lang.Class
getClass()
					[From java.lang.Object] protected native java.lang.Object clone()
throws java.lang.CloneNotSupportedException
					[From java.lang.Object] private static native void
registerNatives()
					[From java.lang.Object] public final native void notify()
					[From java.lang.Object] public final native void notifyAll()
				Fields:
					private int val = 3
					private double val2 = 0.2
					private boolean val3 = true

		private ClassA val2 =
			Name: ClassA
				Immediate Superclass: java.lang.Object
				Interfaces:
					Interface java.io.Serializable
					Interface java.lang.Runnable
				Constructors:
					public ClassA()
					public ClassA(int)
					public java.lang.Object()
				Methods:
					public void run()
					public java.lang.String toString()
					public void setVal(int) throws java.lang.Exception
					public int getVal()
					private void printSomething()
					[From Interface java.lang.Runnable] public abstract void run()
					[From java.lang.Object] protected void finalize() throws
java.lang.Throwable
```

```
                                        [From java.lang.Object] public final void wait(long, int) throws
java.lang.InterruptedException
                                        [From java.lang.Object] public final native void wait(long)
throws java.lang.InterruptedException
                                        [From java.lang.Object] public final void wait() throws
java.lang.InterruptedException
                                        [From java.lang.Object] public boolean equals(java.lang.Object)
                                        [From java.lang.Object] public java.lang.String toString()
                                        [From java.lang.Object] public native int hashCode()
                                        [From java.lang.Object] public final native java.lang.Class
getClass()
                                        [From java.lang.Object] protected native java.lang.Object clone()
throws java.lang.CloneNotSupportedException
                                        [From java.lang.Object] private static native void
registerNatives()
                                        [From java.lang.Object] public final native void notify()
                                        [From java.lang.Object] public final native void notifyAll()
                                Fields:
                                        private int val = 12
                                        private double val2 = 0.2
                                        private boolean val3 = true

                private ClassA val3 = null
                [From ClassC] private ClassA val2 =
                        Name: ClassA
                                Immediate Superclass: java.lang.Object
                                Interfaces:
                                        Interface java.io.Serializable
                                        Interface java.lang.Runnable
                                Constructors:
                                        public ClassA()
                                        public ClassA(int)
                                        public java.lang.Object()
                                Methods:
                                        public void run()
                                        public java.lang.String toString()
                                        public void setVal(int) throws java.lang.Exception
                                        public int getVal()
                                        private void printSomething()
                                        [From Interface java.lang.Runnable] public abstract void run()
                                        [From java.lang.Object] protected void finalize() throws
java.lang.Throwable
                                        [From java.lang.Object] public final void wait(long, int) throws
java.lang.InterruptedException
                                        [From java.lang.Object] public final native void wait(long)
throws java.lang.InterruptedException
                                        [From java.lang.Object] public final void wait() throws
java.lang.InterruptedException
                                        [From java.lang.Object] public boolean equals(java.lang.Object)
                                        [From java.lang.Object] public java.lang.String toString()
                                        [From java.lang.Object] public native int hashCode()
                                        [From java.lang.Object] public final native java.lang.Class
getClass()
                                        [From java.lang.Object] protected native java.lang.Object clone()
throws java.lang.CloneNotSupportedException
                                        [From java.lang.Object] private static native void
registerNatives()
                                        [From java.lang.Object] public final native void notify()
                                        [From java.lang.Object] public final native void notifyAll()
                                Fields:
                                        private int val = 100
                                        private double val2 = 0.2
                                        private boolean val3 = true

                [From ClassC] private ClassA val3 =
                        Name: ClassA
                                Immediate Superclass: java.lang.Object
                                Interfaces:
                                        Interface java.io.Serializable
                                        Interface java.lang.Runnable
                                Constructors:
                                        public ClassA()
                                        public ClassA(int)
                                        public java.lang.Object()
                                Methods:
                                        public void run()
                                        public java.lang.String toString()
                                        public void setVal(int) throws java.lang.Exception
                                        public int getVal()
                                        private void printSomething()
                                        [From Interface java.lang.Runnable] public abstract void run()
                                        [From java.lang.Object] protected void finalize() throws
java.lang.Throwable
                                        [From java.lang.Object] public final void wait(long, int) throws
java.lang.InterruptedException
                                        [From java.lang.Object] public final native void wait(long)
throws java.lang.InterruptedException
                                        [From java.lang.Object] public final void wait() throws
java.lang.InterruptedException
                                        [From java.lang.Object] public boolean equals(java.lang.Object)
                                        [From java.lang.Object] public java.lang.String toString()
                                        [From java.lang.Object] public native int hashCode()
                                        [From java.lang.Object] public final native java.lang.Class
getClass()
                                        [From java.lang.Object] protected native java.lang.Object clone()
throws java.lang.CloneNotSupportedException
                                        [From java.lang.Object] private static native void
registerNatives()
                                        [From java.lang.Object] public final native void notify()
                                        [From java.lang.Object] public final native void notifyAll()
                                Fields:
                                        private int val = 2
                                        private double val2 = 0.2
                                        private boolean val3 = true

                [From ClassC] private ClassA val4 =
                        Name: ClassA
                                Immediate Superclass: java.lang.Object
                                Interfaces:
                                        Interface java.io.Serializable
                                        Interface java.lang.Runnable
                                Constructors:
                                        public ClassA()
                                        public ClassA(int)
                                        public java.lang.Object()
                                Methods:
                                        public void run()
                                        public java.lang.String toString()
                                        public void setVal(int) throws java.lang.Exception
                                        public int getVal()
                                        private void printSomething()
                                        [From Interface java.lang.Runnable] public abstract void run()
                                        [From java.lang.Object] protected void finalize() throws
java.lang.Throwable
                                        [From java.lang.Object] public final void wait(long, int) throws
java.lang.InterruptedException
                                        [From java.lang.Object] public final native void wait(long)
throws java.lang.InterruptedException
                                        [From java.lang.Object] public final void wait() throws
java.lang.InterruptedException
                                        [From java.lang.Object] public boolean equals(java.lang.Object)
                                        [From java.lang.Object] public java.lang.String toString()
                                        [From java.lang.Object] public native int hashCode()
                                        [From java.lang.Object] public final native java.lang.Class
getClass()
                                        [From java.lang.Object] protected native java.lang.Object clone()
throws java.lang.CloneNotSupportedException
```

```
                                        [From java.lang.Object] private static native void
registerNatives()
                                        [From java.lang.Object] public final native void notify()
                                        [From java.lang.Object] public final native void notifyAll()
                                Fields:
                                        private int val = 3
                                        private double val2 = 0.2
                                        private boolean val3 = true

                [From ClassD] private ClassA val =
                        Name: ClassA
                                Immediate Superclass: java.lang.Object
                                Interfaces:
                                        Interface java.io.Serializable
                                        Interface java.lang.Runnable
                                Constructors:
                                        public ClassA()
                                        public ClassA(int)
                                        public java.lang.Object()
                                Methods:
                                        public void run()
                                        public java.lang.String toString()
                                        public void setVal(int) throws java.lang.Exception
                                        public int getVal()
                                        private void printSomething()
                                        [From Interface java.lang.Runnable] public abstract void run()
                                        [From java.lang.Object] protected void finalize() throws
java.lang.Throwable
                                        [From java.lang.Object] public final void wait(long, int) throws
java.lang.InterruptedException
                                        [From java.lang.Object] public final native void wait(long)
throws java.lang.InterruptedException
                                        [From java.lang.Object] public final void wait() throws
java.lang.InterruptedException
                                        [From java.lang.Object] public boolean equals(java.lang.Object)
                                        [From java.lang.Object] public java.lang.String toString()
                                        [From java.lang.Object] public native int hashCode()
                                        [From java.lang.Object] public final native java.lang.Class
getClass()
                                        [From java.lang.Object] protected native java.lang.Object clone()
throws java.lang.CloneNotSupportedException
                                        [From java.lang.Object] private static native void
registerNatives()
                                        [From java.lang.Object] public final native void notify()
                                        [From java.lang.Object] public final native void notifyAll()
                                Fields:
                                        private int val = 17
                                        private double val2 = 0.2
                                        private boolean val3 = true

                [From ClassD] private static ClassA val2 = null
                [From ClassD] private int val3 = 34
                [From ClassD] private ClassA[] vallarray = [null, null, null, null, null, null, null,
null, null, null]
======================================================
======================================================
Running Test: ClassD
Name: ClassD
        Immediate Superclass: java.lang.Object
        Interfaces:
        Constructors:
                public ClassD()
                public ClassD(int)
                public java.lang.Object()
        Methods:
                public java.lang.String toString()
                public int getVal3()
                [From java.lang.Object] protected void finalize() throws java.lang.Throwable
                [From java.lang.Object] public final void wait(long, int) throws
java.lang.InterruptedException
                [From java.lang.Object] public final native void wait(long) throws
java.lang.InterruptedException
                [From java.lang.Object] public final void wait() throws java.lang.InterruptedException
                [From java.lang.Object] public boolean equals(java.lang.Object)
                [From java.lang.Object] public java.lang.String toString()
                [From java.lang.Object] public native int hashCode()
                [From java.lang.Object] public final native java.lang.Class getClass()
                [From java.lang.Object] protected native java.lang.Object clone() throws
java.lang.CloneNotSupportedException
                [From java.lang.Object] private static native void registerNatives()
                [From java.lang.Object] public final native void notify()
                [From java.lang.Object] public final native void notifyAll()
        Fields:
                private ClassA val =
                        Name: ClassA
                                Immediate Superclass: java.lang.Object
                                Interfaces:
                                        Interface java.io.Serializable
                                        Interface java.lang.Runnable
                                Constructors:
                                        public ClassA()
                                        public ClassA(int)
                                        public java.lang.Object()
                                Methods:
                                        public void run()
                                        public java.lang.String toString()
                                        public void setVal(int) throws java.lang.Exception
                                        public int getVal()
                                        private void printSomething()
                                        [From Interface java.lang.Runnable] public abstract void run()
                                        [From java.lang.Object] protected void finalize() throws
java.lang.Throwable
                                        [From java.lang.Object] public final void wait(long, int) throws
java.lang.InterruptedException
                                        [From java.lang.Object] public final native void wait(long)
throws java.lang.InterruptedException
                                        [From java.lang.Object] public final void wait() throws
java.lang.InterruptedException
                                        [From java.lang.Object] public boolean equals(java.lang.Object)
                                        [From java.lang.Object] public java.lang.String toString()
                                        [From java.lang.Object] public native int hashCode()
                                        [From java.lang.Object] public final native java.lang.Class
getClass()
                                        [From java.lang.Object] protected native java.lang.Object clone()
throws java.lang.CloneNotSupportedException
                                        [From java.lang.Object] private static native void
registerNatives()
                                        [From java.lang.Object] public final native void notify()
                                        [From java.lang.Object] public final native void notifyAll()
                                Fields:
                                        private int val = 17
                                        private double val2 = 0.2
                                        private boolean val3 = true

                private static ClassA val2 = null
                private int val3 = 32
                private ClassA[] vallarray = [null, null, null, null, null, null, null, null, null, null]
======================================================
======================================================
Running Test: ClassD
Name: ClassD
        Immediate Superclass: java.lang.Object
        Interfaces:
        Constructors:
                public ClassD()
                public ClassD(int)
                public java.lang.Object()
        Methods:
                public java.lang.String toString()
                public int getVal3()
                [From java.lang.Object] protected void finalize() throws java.lang.Throwable
                [From java.lang.Object] public final void wait(long, int) throws
java.lang.InterruptedException
                [From java.lang.Object] public final native void wait(long) throws
java.lang.InterruptedException
                [From java.lang.Object] public final void wait() throws java.lang.InterruptedException
                [From java.lang.Object] public boolean equals(java.lang.Object)
                [From java.lang.Object] public java.lang.String toString()
                [From java.lang.Object] public native int hashCode()
                [From java.lang.Object] public final native java.lang.Class getClass()
                [From java.lang.Object] protected native java.lang.Object clone() throws
java.lang.CloneNotSupportedException
                [From java.lang.Object] private static native void registerNatives()
                [From java.lang.Object] public final native void notify()
                [From java.lang.Object] public final native void notifyAll()
        Fields:
                private ClassA val =
                        Name: ClassA
                                Immediate Superclass: java.lang.Object
                                Interfaces:
                                        Interface java.io.Serializable
                                        Interface java.lang.Runnable
                                Constructors:
                                        public ClassA()
                                        public ClassA(int)
                                        public java.lang.Object()
                                Methods:
                                        public void run()
                                        public java.lang.String toString()
                                        public void setVal(int) throws java.lang.Exception
                                        public int getVal()
                                        private void printSomething()
                                        [From Interface java.lang.Runnable] public abstract void run()
                                        [From java.lang.Object] protected void finalize() throws
java.lang.Throwable
                                        [From java.lang.Object] public final void wait(long, int) throws
java.lang.InterruptedException
                                        [From java.lang.Object] public final native void wait(long)
throws java.lang.InterruptedException
                                        [From java.lang.Object] public final void wait() throws
java.lang.InterruptedException
                                        [From java.lang.Object] public boolean equals(java.lang.Object)
                                        [From java.lang.Object] public java.lang.String toString()
                                        [From java.lang.Object] public native int hashCode()
                                        [From java.lang.Object] public final native java.lang.Class
getClass()
                                        [From java.lang.Object] protected native java.lang.Object clone()
throws java.lang.CloneNotSupportedException
                                        [From java.lang.Object] private static native void
registerNatives()
                                        [From java.lang.Object] public final native void notify()
                                        [From java.lang.Object] public final native void notifyAll()
                                Fields:
                                        private int val = 17
                                        private double val2 = 0.2
                                        private boolean val3 = true

                private static ClassA val2 = null
                private int val3 = 34
                private ClassA[] vallarray = [null, null, null, null, null, null, null, null, null, null]
======================================================
======================================================
Running Test: [LClassB;@7ef72e77
Name: ClassB[]
        Immediate Superclass: java.lang.Object
        Interfaces:
                Interface java.lang.Cloneable
                Interface java.io.Serializable
        Constructors:
                public java.lang.Object()
        Methods:
                [From java.lang.Object] protected void finalize() throws java.lang.Throwable
                [From java.lang.Object] public final void wait(long, int) throws
java.lang.InterruptedException
                [From java.lang.Object] public final native void wait(long) throws
java.lang.InterruptedException
                [From java.lang.Object] public final void wait() throws java.lang.InterruptedException
                [From java.lang.Object] public boolean equals(java.lang.Object)
                [From java.lang.Object] public java.lang.String toString()
                [From java.lang.Object] public native int hashCode()
                [From java.lang.Object] public final native java.lang.Class getClass()
                [From java.lang.Object] protected native java.lang.Object clone() throws
java.lang.CloneNotSupportedException
                [From java.lang.Object] private static native void registerNatives()
                [From java.lang.Object] public final native void notify()
                [From java.lang.Object] public final native void notifyAll()
        Fields:
======================================================
======================================================
Running Test: [[LClassB;@76d4d81
Name: ClassB[][]
        Immediate Superclass: java.lang.Object
        Interfaces:
                Interface java.lang.Cloneable
                Interface java.io.Serializable
        Constructors:
                public java.lang.Object()
        Methods:
                [From java.lang.Object] protected void finalize() throws java.lang.Throwable
                [From java.lang.Object] public final void wait(long, int) throws
java.lang.InterruptedException
                [From java.lang.Object] public final native void wait(long) throws
java.lang.InterruptedException
                [From java.lang.Object] public final void wait() throws java.lang.InterruptedException
                [From java.lang.Object] public boolean equals(java.lang.Object)
                [From java.lang.Object] public java.lang.String toString()
                [From java.lang.Object] public native int hashCode()
                [From java.lang.Object] public final native java.lang.Class getClass()
                [From java.lang.Object] protected native java.lang.Object clone() throws
java.lang.CloneNotSupportedException
                [From java.lang.Object] private static native void registerNatives()
                [From java.lang.Object] public final native void notify()
                [From java.lang.Object] public final native void notifyAll()
        Fields:
======================================================
======================================================
Running Test: Test String
Name: java.lang.String
        Immediate Superclass: java.lang.Object
        Interfaces:
                Interface java.io.Serializable
                Interface java.lang.Comparable
                Interface java.lang.CharSequence
        Constructors:
                public java.lang.String(byte[])
                public java.lang.String(byte[], int, int)
                public java.lang.String(byte[], java.nio.charset.Charset)
                public java.lang.String(byte[], java.lang.String)
                public java.lang.String(byte[], int, int, java.nio.charset.Charset)
                public java.lang.String(java.lang.StringBuilder)
                public java.lang.String(java.lang.StringBuffer)
                public java.lang.String(int[], int, int)
                public java.lang.String(char[], int, int)
                public java.lang.String(char[])
                public java.lang.String(java.lang.String)
                public java.lang.String()
                public java.lang.String(byte[], int, int, java.lang.String)
                public java.lang.String(byte[], int)
                public java.lang.String(byte[], int, int, int)
                public java.lang.Object()
        Methods:
                public boolean equals(java.lang.Object)
                public java.lang.String toString()
                public int hashCode()
                public int compareTo(java.lang.String)
                public volatile int compareTo(java.lang.Object)
                static int indexOf(char[], int, int, char[], int, int, int)
                public int indexOf(java.lang.String, int)
                public int indexOf(java.lang.String)
                public int indexOf(int, int)
                public int indexOf(int)
                public static java.lang.String valueOf(char[], int, int)
                public static java.lang.String valueOf(long)
                public static java.lang.String valueOf(float)
                public static java.lang.String valueOf(double)
                public static java.lang.String valueOf(java.lang.Object)
                public static java.lang.String valueOf(char[])
                public static java.lang.String valueOf(boolean)
                public static java.lang.String valueOf(int)
                public static java.lang.String valueOf(char)
                public int codePointBefore(int)
                public int codePointCount(int, int)
                public int compareToIgnoreCase(java.lang.String)
                public boolean contentEquals(Interface java.lang.CharSequence)
                public boolean contentEquals(java.lang.StringBuffer)
                public boolean equalsIgnoreCase(java.lang.String)
                private int indexOfSupplementary(int, int)
                public boolean matches(java.lang.String)
                public java.lang.String replace(Interface java.lang.CharSequence, Interface
java.lang.CharSequence)
                public java.lang.String replace(char, char)
                public java.lang.String replaceAll(java.lang.String, java.lang.String)
                public java.lang.String replaceFirst(java.lang.String, java.lang.String)
                public java.lang.String[] split(java.lang.String)
                public java.lang.String[] split(java.lang.String, int)
                public boolean startsWith(java.lang.String, int)
                public boolean startsWith(java.lang.String)
                public Interface java.lang.CharSequence subSequence(int, int)
                public java.lang.String substring(int, int)
                public java.lang.String substring(int)
                public char[] toCharArray()
                public java.lang.String toLowerCase(java.util.Locale)
                public java.lang.String toLowerCase()
                public java.lang.String toUpperCase(java.util.Locale)
                public java.lang.String toUpperCase()
                public java.lang.String trim()
                public char charAt(int)
                private static void checkBounds(byte[], int, int)
                public int codePointAt(int)
                public java.lang.String concat(java.lang.String)
                public boolean contains(Interface java.lang.CharSequence)
                public static java.lang.String copyValueOf(char[], int, int)
                public static java.lang.String copyValueOf(char[])
                public boolean endsWith(java.lang.String)
                public static transient java.lang.String format(java.lang.String, java.lang.Object[])
                public static transient java.lang.String format(java.util.Locale, java.lang.String,
java.lang.Object[])
                public byte[] getBytes()
                public void getBytes(int, int, byte[], int)
                public byte[] getBytes(java.lang.String) throws java.io.UnsupportedEncodingException
                public byte[] getBytes(java.nio.charset.Charset)
                public void getChars(int, int, char[], int)
                 void getChars(char[], int)
                 int hash32()
                public native java.lang.String intern()
                public boolean isEmpty()
                public int lastIndexOf(int)
                public int lastIndexOf(int, int)
                public int lastIndexOf(java.lang.String)
                static int lastIndexOf(char[], int, int, char[], int, int, int)
                public int lastIndexOf(java.lang.String, int)
                public int length()
                private int lastIndexOfSupplementary(int, int)
                public int offsetByCodePoints(int, int)
                public boolean regionMatches(int, java.lang.String, int, int)
                public boolean regionMatches(boolean, int, java.lang.String, int, int)
                [From Interface java.lang.Comparable] public abstract int compareTo(java.lang.Object)
                [From Interface java.lang.CharSequence] public abstract java.lang.String toString()
                [From Interface java.lang.CharSequence] public abstract Interface java.lang.CharSequence
subSequence(int, int)
                [From Interface java.lang.CharSequence] public abstract char charAt(int)
                [From Interface java.lang.CharSequence] public abstract int length()
                [From java.lang.Object] protected void finalize() throws java.lang.Throwable
                [From java.lang.Object] public final void wait(long, int) throws
java.lang.InterruptedException
                [From java.lang.Object] public final native void wait(long) throws
java.lang.InterruptedException
                [From java.lang.Object] public final void wait() throws java.lang.InterruptedException
                [From java.lang.Object] public boolean equals(java.lang.Object)
                [From java.lang.Object] public java.lang.String toString()
                [From java.lang.Object] public native int hashCode()
                [From java.lang.Object] public final native java.lang.Class getClass()
                [From java.lang.Object] protected native java.lang.Object clone() throws
java.lang.CloneNotSupportedException
                [From java.lang.Object] private static native void registerNatives()
                [From java.lang.Object] public final native void notify()
                [From java.lang.Object] public final native void notifyAll()
        Fields:
                private final char[] value = [T, e, s, t,  , S, t, r, i, n, g]
                private int hash = 0
                private static final long serialVersionUID = -6849794470754667710
                private static final java.io.ObjectStreamField[] serialPersistentFields = []
                public static final Interface java.util.Comparator CASE_INSENSITIVE_ORDER =
                        Name: java.lang.String$CaseInsensitiveComparator
                                Immediate Superclass: java.lang.Object
                                Interfaces:
                                        Interface java.util.Comparator
                                        Interface java.io.Serializable
                                Constructors:
                                        public java.lang.Object()
                                Methods:
                                        public int compare(java.lang.String, java.lang.String)
                                        public volatile int compare(java.lang.Object, java.lang.Object)
                                        [From Interface java.util.Comparator] public abstract boolean
equals(java.lang.Object)
                                        [From Interface java.util.Comparator] public abstract int
compare(java.lang.Object, java.lang.Object)
                                        [From java.lang.Object] protected void finalize() throws
java.lang.Throwable
                                        [From java.lang.Object] public final void wait(long, int) throws
java.lang.InterruptedException
                                        [From java.lang.Object] public final native void wait(long)
throws java.lang.InterruptedException
                                        [From java.lang.Object] public final void wait() throws
java.lang.InterruptedException
                                        [From java.lang.Object] public boolean equals(java.lang.Object)
                                        [From java.lang.Object] public java.lang.String toString()
                                        [From java.lang.Object] public native int hashCode()
                                        [From java.lang.Object] public final native java.lang.Class
getClass()
                                        [From java.lang.Object] protected native java.lang.Object clone()
throws java.lang.CloneNotSupportedException
```

```
                                        [From java.lang.Object] private static native void
registerNatives()
                                        [From java.lang.Object] public final native void notify()
                                        [From java.lang.Object] public final native void notifyAll()
                                Fields:
                                        private static final long serialVersionUID = 8575799808933029326

                private static final int HASHING_SEED = -5121256
                private transient int hash32 = 0
======================================================
```