



You:

```
/** * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * http://www.apache.org/licenses/LICENSE-2.0 * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. */ package org.apache.hadoop.util; import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.io.Text; import org.apache.hadoop.util.LineReader; import org.junit.Assert; import org.junit.Test; import java.io.ByteArrayInputStream; import java.io.IOException; public class ALineReaderTest { @Test public void testLineReaderBufferSizeConfig() throws IOException { // Set up Configuration conf = new Configuration(); // Test different buffer size values conf.set("io.file.buffer.size", "2048"); ByteArrayInputStream inputStream = new ByteArrayInputStream("Hello world!".getBytes()); // Exercise LineReader reader = new LineReader(inputStream, conf); // Verify Assert.assertEquals(2048, reader.getBufferSize()); // Repeat with different buffer sizes conf.set("io.file.buffer.size", "4096"); reader = new LineReader(inputStream, conf); Assert.assertEquals(4096, reader.getBufferSize()); conf.set("io.file.buffer.size", "8192"); reader = new LineReader(inputStream, conf); Assert.assertEquals(8192, reader.getBufferSize()); } @Test(expected = IllegalArgumentException.class) public void testInvalidBufferSize() throws IOException { Configuration conf = new Configuration(); // Invalid buffer size conf.set("io.file.buffer.size", "0"); ByteArrayInputStream inputStream = new ByteArrayInputStream("Hello world!".getBytes()); LineReader reader = new LineReader(inputStream, conf); } @Test public void testNegativeBufferSize() throws IOException { Configuration conf = new Configuration(); conf.set("io.file.buffer.size", "-1024"); ByteArrayInputStream inputStream = new ByteArrayInputStream("Hello world!".getBytes()); LineReader reader = new LineReader(inputStream, conf); // Should log a warning assertTrue(logsContainWarning()); } @Test public void testNonNumericBufferSize() throws IOException { Configuration conf = new Configuration(); conf.set("io.file.buffer.size", "text"); ByteArrayInputStream inputStream = new ByteArrayInputStream("Hello world!".getBytes()); LineReader reader = new LineReader(inputStream, conf); // Should log a warning assertTrue(logsContainWarning()); } } } [INFO] -----  
[ERROR] COMPILE ERROR : [INFO] ----- [ERROR]  
/home/nvadde2/hadoop/hadoop-common-project/hadoop-  
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[88,14] cannot find symbol symbol: method  
logsContainWarning() location: class org.apache.hadoop.util.ALineReaderTest [ERROR]  
/home/nvadde2/hadoop/hadoop-common-project/hadoop-  
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[103,14] cannot find symbol symbol: method  
logsContainWarning() location: class org.apache.hadoop.util.ALineReaderTest [INFO] 2 errors [INFO] -----  
----- [INFO] ----- [INFO] BUILD FAILURE  
[INFO] ----- [INFO] Total time: 27.243 s [INFO] Finished at: 2023-11-  
16T19:49:21-06:00 [INFO] ----- [ERROR] Failed to execute goal  
org.apache.maven.plugins:maven-compiler-plugin:3.10.1:testCompile (default-testCompile) on project hadoop-  
common: Compilation failure: Compilation failure: [ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-  
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[88,14] cannot find symbol [ERROR] symbol:  
method logsContainWarning() [ERROR] location: class org.apache.hadoop.util.ALineReaderTest [ERROR]  
/home/nvadde2/hadoop/hadoop-common-project/hadoop-  
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[103,14] cannot find symbol [ERROR] symbol:  
method logsContainWarning() [ERROR] location: class org.apache.hadoop.util.ALineReaderTest [ERROR] -> [Help 1]  
[ERROR] [ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch. [ERROR] Re-run Maven  
using the -X switch to enable full debug logging. [ERROR] [ERROR] For more information about the errors and  
possible solutions, please read the following articles: [ERROR] [Help 1]  
http://cwiki.apache.org/confluence/display/MAVEN/MojoFailureException
```



ChatGPT:

It looks like there are a couple of issues in your code and Maven build output.

1. The error `cannot find symbol` suggests that the method `logsContainWarning()` is not defined in the `ALineReaderTest` class. This method is referenced in the test methods `testNegativeBufferSize()` and

testNonNumericBufferSize(). You need to define this method in your test class.

Here's a simple example of what the method might look like:

```
java
private boolean logsContainWarning() {
    // Add your implementation to check if logs contain a warning
    // Return true if logs contain a warning, false otherwise
    return false;
}
```

2. In your test methods, there is a missing import for assertTrue. Make sure you import it from the appropriate testing framework. For example, if you are using JUnit, add the following import statement:

```
java
import static org.junit.Assert.assertTrue;
```

Make these corrections in your test class, and it should resolve the compilation errors.



You:

```
/** * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. */ package org.apache.hadoop.util; import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.io.Text; import org.apache.hadoop.util.LineReader; import org.junit.Assert; import org.junit.Test; import static org.junit.Assert.assertTrue; import java.io.ByteArrayInputStream; import java.io.IOException; public class ALineReaderTest { @Test public void testLineReaderBufferSizeConfig() throws IOException { // Set up Configuration conf = new Configuration(); // Test different buffer size values conf.set("io.file.buffer.size", "2048"); private boolean logsContainWarning() { // Add your implementation to check if logs contain a warning // Return true if logs contain a warning, false otherwise return false; } ByteArrayInputStream inputStream = new ByteArrayInputStream("Hello world!".getBytes()); // Exercise LineReader reader = new LineReader(inputStream, conf); // Verify Assert.assertEquals(2048, reader.getBufferSize()); // Repeat with different buffer sizes conf.set("io.file.buffer.size", "4096"); reader = new LineReader(inputStream, conf); Assert.assertEquals(4096, reader.getBufferSize()); conf.set("io.file.buffer.size", "8192"); reader = new LineReader(inputStream, conf); Assert.assertEquals(8192, reader.getBufferSize()); } @Test(expected = IllegalArgumentException.class) public void testInvalidBufferSize() throws IOException { Configuration conf = new Configuration(); // Invalid buffer size conf.set("io.file.buffer.size", "0"); ByteArrayInputStream inputStream = new ByteArrayInputStream("Hello world!".getBytes()); LineReader reader = new LineReader(inputStream, conf); } @Test public void testNegativeBufferSize() throws IOException { Configuration conf = new Configuration(); conf.set("io.file.buffer.size", "-1024"); ByteArrayInputStream inputStream = new ByteArrayInputStream("Hello world!".getBytes()); LineReader reader = new LineReader(inputStream, conf); // Should log a warning assertTrue(logsContainWarning()); } @Test public void testNonNumericBufferSize() throws IOException { Configuration conf = new Configuration(); conf.set("io.file.buffer.size", "text"); ByteArrayInputStream inputStream = new ByteArrayInputStream("Hello world!".getBytes()); LineReader reader = new LineReader(inputStream, conf); // Should log a warning assertTrue(logsContainWarning()); } } [INFO] -----
[ERROR] COMPILE ERROR : [INFO] ----- [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[42,5] illegal start of expression [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[54,24] <identifier> expected [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[54,25] illegal start of type [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
```

```
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[57,13] <identifier> expected [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[57,14] illegal start of type [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[59,11] <identifier> expected [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[61,24] <identifier> expected [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[61,25] illegal start of type [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[63,13] <identifier> expected [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[63,14] illegal start of type [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[65,11] <identifier> expected [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[67,24] <identifier> expected [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[67,25] illegal start of type [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[71,8] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[76,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[78,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[80,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[81,1] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[84,8] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[88,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[90,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[92,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[95,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[103,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[105,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[107,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[110,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[111,1] class, interface, or enum expected [INFO]
30 errors [INFO] ----- [INFO] -----
----- [INFO] BUILD FAILURE [INFO] ----- [INFO] Total time:
21.219 s [INFO] Finished at: 2023-11-16T19:59:55-06:00 [INFO] -----
- [ERROR] Failed to execute goal org.apache.maven.plugins:maven-compiler-plugin:3.10.1:testCompile (default-testCompile) on project hadoop-common: Compilation failure: Compilation failure: [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[42,5] illegal start of expression [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[54,24] <identifier> expected [ERROR]
```

```
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[54,25] illegal start of type [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[57,13] <identifier> expected [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[57,14] illegal start of type [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[59,11] <identifier> expected [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[61,24] <identifier> expected [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[61,25] illegal start of type [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[63,13] <identifier> expected [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[63,14] illegal start of type [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[65,11] <identifier> expected [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[67,24] <identifier> expected [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[67,25] illegal start of type [ERROR]
/home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[71,8] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[76,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[78,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[80,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[81,1] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[84,8] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[88,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[90,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[92,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[95,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[96,1] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[99,8] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[103,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[105,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[107,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[110,3] class, interface, or enum expected
[ERROR] /home/nvadde2/hadoop/hadoop-common-project/hadoop-
common/src/test/java/org/apache/hadoop/util/ALineReaderTest.java:[111,1] class, interface, or enum expected
[ERROR] -> [Help 1] [ERROR] [ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging. [ERROR] [ERROR] For more information
about the errors and possible solutions, please read the following articles: [ERROR] [Help 1]
http://cwiki.apache.org/confluence/display/MAVEN/MojoFailureException nvadde2@fa23-cs511-
009:~/hadoop/hadoop-common-project/hadoop-common$ give complete code to fix
```



It seems like there are several issues with your code. I'll provide a corrected version with comments to explain the changes.

```
java
package org.apache.hadoop.util;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.util.LineReader;
import org.junit.Assert;
import org.junit.Test;
import static org.junit.Assert.assertTrue;

import java.io.ByteArrayInputStream;
import java.io.IOException;

public class ALineReaderTest {

    // Define logsContainWarning method
    private boolean logsContainWarning() {
        // Add your implementation to check if logs contain a warning
        // Return true if logs contain a warning, false otherwise
        return false;
    }

    @Test
    public void testLineReaderBufferSizeConfig() throws IOException {
        // Set up
        Configuration conf = new Configuration();

        // Test different buffer size values
        conf.set("io.file.buffer.size", "2048");

        ByteArrayInputStream inputStream = new ByteArrayInputStream("Hello world!".getBytes());

        // Exercise
        LineReader reader = new LineReader(inputStream, conf);

        // Verify
        Assert.assertEquals(2048, reader.getBufferSize());

        // Repeat with different buffer sizes
        conf.set("io.file.buffer.size", "4096");

        reader = new LineReader(inputStream, conf);

        Assert.assertEquals(4096, reader.getBufferSize());

        conf.set("io.file.buffer.size", "8192");

        reader = new LineReader(inputStream, conf);

        Assert.assertEquals(8192, reader.getBufferSize());
    }
}
```

```
}

@Test(expected = IllegalArgumentException.class)
public void testInvalidBufferSize() throws IOException {
    Configuration conf = new Configuration();

    // Invalid buffer size
    conf.set("io.file.buffer.size", "0");

    ByteArrayInputStream inputStream = new ByteArrayInputStream("Hello world!".getBytes());

    // Exercise and Verify
    // This should throw an IllegalArgumentException
    new LineReader(inputStream, conf);
}

@Test
public void testNegativeBufferSize() throws IOException {
    Configuration conf = new Configuration();

    conf.set("io.file.buffer.size", "-1024");

    ByteArrayInputStream inputStream = new ByteArrayInputStream("Hello world!".getBytes());

    // Exercise
    LineReader reader = new LineReader(inputStream, conf);

    // Should log a warning
    assertTrue(logsContainWarning());
}

@Test
public void testNonNumericBufferSize() throws IOException {
    Configuration conf = new Configuration();

    conf.set("io.file.buffer.size", "text");

    ByteArrayInputStream inputStream = new ByteArrayInputStream("Hello world!".getBytes());

    // Exercise
    LineReader reader = new LineReader(inputStream, conf);

    // Should log a warning
    assertTrue(logsContainWarning());
}
```

Key changes:

1. Moved the `logsContainWarning()` method outside the `testLineReaderBufferSizeConfig()` method to make it accessible to other test methods.
2. Fixed the import for `assertTrue`.
3. Removed unnecessary imports.
4. Added comments to explain the changes.

Please note that the `logsContainWarning()` method is a placeholder, and you need to implement the actual logic to check if logs contain a warning.



You:

```
[INFO] ----- [INFO] T E S T S [INFO] -----
[INFO] Running org.apache.hadoop.util.ALineReaderTest [ERROR] Tests run: 4, Failures: 1, Errors: 2, Skipped: 0,
Time elapsed: 0.429 s <<< FAILURE! - in org.apache.hadoop.util.ALineReaderTest [ERROR]
testNonNumericBufferSize(org.apache.hadoop.util.ALineReaderTest) Time elapsed: 0.336 s <<< ERROR!
java.lang.NumberFormatException: For input string: "text" at
java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:65) at
java.base/java.lang.Integer.parseInt(Integer.java:652) at java.base/java.lang.Integer.parseInt(Integer.java:770) at
org.apache.hadoop.conf.Configuration.getInt(Configuration.java:1552) at org.apache.hadoop.util.LineReader.<init>
(LineReader.java:95) at
org.apache.hadoop.util.ALineReaderTest.testNonNumericBufferSize(ALineReaderTest.java:89) at
java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at
java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62) at
java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) at
java.base/java.lang.reflect.Method.invoke(Method.java:566) at
org.junit.runners.model.FrameworkMethod$1.runReflectiveCall(FrameworkMethod.java:59) at
org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:12) at
org.junit.runners.model.FrameworkMethod.invokeExplosively(FrameworkMethod.java:56) at
org.junit.internal.runners.statements.InvokeMethod.evaluate(InvokeMethod.java:17) at
org.junit.runners.ParentRunner$3.evaluate(ParentRunner.java:306) at
org.junit.runners.BlockJUnit4ClassRunner$1.evaluate(BlockJUnit4ClassRunner.java:100) at
org.junit.runners.ParentRunner.runLeaf(ParentRunner.java:366) at
org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:103) at
org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:63) at
org.junit.runners.ParentRunner$4.run(ParentRunner.java:331) at
org.junit.runners.ParentRunner$1.schedule(ParentRunner.java:79) at
org.junit.runners.ParentRunner.runChildren(ParentRunner.java:329) at
org.junit.runners.ParentRunner.access$100(ParentRunner.java:66) at
org.junit.runners.ParentRunner$2.evaluate(ParentRunner.java:293) at
org.junit.runners.ParentRunner$3.evaluate(ParentRunner.java:306) at
org.junit.runners.ParentRunner.run(ParentRunner.java:413) at
org.apache.maven.surefire.junit4.JUnit4Provider.execute(JUnit4Provider.java:365) at
org.apache.maven.surefire.junit4.JUnit4Provider.executeWithRerun(JUnit4Provider.java:273) at
org.apache.maven.surefire.junit4.JUnit4Provider.executeTestSet(JUnit4Provider.java:238) at
org.apache.maven.surefire.junit4.JUnit4Provider.invoke(JUnit4Provider.java:159) at
org.apache.maven.surefire.booter.ForkedBooter.invokeProviderInSameClassLoader(ForkedBooter.java:384) at
org.apache.maven.surefire.booter.ForkedBooter.runSuitesInProcess(ForkedBooter.java:345) at
org.apache.maven.surefire.booter.ForkedBooter.execute(ForkedBooter.java:126) at
org.apache.maven.surefire.booter.ForkedBooter.main(ForkedBooter.java:418) [ERROR]
testNegativeBufferSize(org.apache.hadoop.util.ALineReaderTest) Time elapsed: 0.011 s <<< ERROR!
java.lang.NegativeArraySizeException: -1024 at org.apache.hadoop.util.LineReader.<init>(LineReader.java:82) at
org.apache.hadoop.util.LineReader.<init>(LineReader.java:95) at
org.apache.hadoop.util.ALineReaderTest.testNegativeBufferSize(ALineReaderTest.java:74) at
java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at
java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62) at
java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) at
java.base/java.lang.reflect.Method.invoke(Method.java:566) at
org.junit.runners.model.FrameworkMethod$1.runReflectiveCall(FrameworkMethod.java:59) at
org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:12) at
org.junit.runners.model.FrameworkMethod.invokeExplosively(FrameworkMethod.java:56) at
org.junit.internal.runners.statements.InvokeMethod.evaluate(InvokeMethod.java:17) at
org.junit.runners.ParentRunner$3.evaluate(ParentRunner.java:306) at
org.junit.runners.BlockJUnit4ClassRunner$1.evaluate(BlockJUnit4ClassRunner.java:100) at
org.junit.runners.ParentRunner.runLeaf(ParentRunner.java:366) at
org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:103) at
org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:63) at
org.junit.runners.ParentRunner$4.run(ParentRunner.java:331) at
org.junit.runners.ParentRunner$1.schedule(ParentRunner.java:79) at
org.junit.runners.ParentRunner.runChildren(ParentRunner.java:329) at
```

```

org.junit.runners.ParentRunner.access$100(ParentRunner.java:66) at
org.junit.runners.ParentRunner$2.evaluate(ParentRunner.java:293) at
org.junit.runners.ParentRunner$3.evaluate(ParentRunner.java:306) at
org.junit.runners.ParentRunner.run(ParentRunner.java:413) at
org.apache.maven.surefire.junit4.JUnit4Provider.execute(JUnit4Provider.java:365) at
org.apache.maven.surefire.junit4.JUnit4Provider.executeWithRerun(JUnit4Provider.java:273) at
org.apache.maven.surefire.junit4.JUnit4Provider.executeTestSet(JUnit4Provider.java:238) at
org.apache.maven.surefire.junit4.JUnit4Provider.invoke(JUnit4Provider.java:159) at
org.apache.maven.surefire.booter.ForkedBooter.invokeProviderInSameClassLoader(ForkedBooter.java:384) at
org.apache.maven.surefire.booter.ForkedBooter.runSuitesInProcess(ForkedBooter.java:345) at
org.apache.maven.surefire.booter.ForkedBooter.execute(ForkedBooter.java:126) at
org.apache.maven.surefire.booter.ForkedBooter.main(ForkedBooter.java:418) [ERROR]
testInvalidBufferSize(org.apache.hadoop.util.ALineReaderTest) Time elapsed: 0.01 s <<< FAILURE!
java.lang.AssertionError: Expected exception: java.lang.IllegalArgumentException at
org.junit.internal.runners.statements.ExpectException.evaluate(ExpectException.java:34) at
org.junit.runners.ParentRunner$3.evaluate(ParentRunner.java:306) at
org.junit.runners.BlockJUnit4ClassRunner$1.evaluate(BlockJUnit4ClassRunner.java:100) at
org.junit.runners.ParentRunner.runLeaf(ParentRunner.java:366) at
org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:103) at
org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:63) at
org.junit.runners.ParentRunner$4.run(ParentRunner.java:331) at
org.junit.runners.ParentRunner$1.schedule(ParentRunner.java:79) at
org.junit.runners.ParentRunner.runChildren(ParentRunner.java:329) at
org.junit.runners.ParentRunner.access$100(ParentRunner.java:66) at
org.junit.runners.ParentRunner$2.evaluate(ParentRunner.java:293) at
org.junit.runners.ParentRunner$3.evaluate(ParentRunner.java:306) at
org.junit.runners.ParentRunner.run(ParentRunner.java:413) at
org.apache.maven.surefire.junit4.JUnit4Provider.execute(JUnit4Provider.java:365) at
org.apache.maven.surefire.junit4.JUnit4Provider.executeWithRerun(JUnit4Provider.java:273) at
org.apache.maven.surefire.junit4.JUnit4Provider.executeTestSet(JUnit4Provider.java:238) at
org.apache.maven.surefire.junit4.JUnit4Provider.invoke(JUnit4Provider.java:159) at
org.apache.maven.surefire.booter.ForkedBooter.invokeProviderInSameClassLoader(ForkedBooter.java:384) at
org.apache.maven.surefire.booter.ForkedBooter.runSuitesInProcess(ForkedBooter.java:345) at
org.apache.maven.surefire.booter.ForkedBooter.execute(ForkedBooter.java:126) at
org.apache.maven.surefire.booter.ForkedBooter.main(ForkedBooter.java:418) [INFO] [INFO] Results: [INFO] [ERROR]
Failures: [ERROR] ALineReaderTest.testInvalidBufferSize Expected exception: java.lang.IllegalArgumentException
[ERROR] Errors: [ERROR] ALineReaderTest.testNegativeBufferSize:74 » NegativeArraySize -1024 [ERROR]
ALineReaderTest.testNonNumericBufferSize:89 » NumberFormat For input string: "... [INFO] [ERROR] Tests run: 4,
Failures: 1, Errors: 2, Skipped: 0 [INFO] [ERROR] There are test failures. Please refer to
/home/nvadde2/hadoop/hadoop-common-project/hadoop-common/target/surefire-reports for the individual test results.
Please refer to dump files (if any exist) [date].dump, [date]-jvmRun[N].dump and [date].dumpstream. [INFO] [INFO] ---
maven-antrun-plugin:1.7:run (common-test-bats-driver) @ hadoop-common --- [INFO] Executing tasks main: [exec]
[exec] [exec] ERROR: bats not installed. Skipping bash tests. [exec] ERROR: Please install bats as soon as possible.
[exec] [exec] [INFO] Executed tasks [INFO] [INFO] --- jacoco-maven-plugin:0.8.6:report (report) @ hadoop-common ---
[INFO] Loading execution data file /home/nvadde2/hadoop/hadoop-common-project/hadoop-common/target/coverage-
reports/jacoco-ut.exec [INFO] Analyzed bundle 'Apache Hadoop Common' with 2160 classes [INFO] [INFO] --- jacoco-
maven-plugin:0.8.6:report (default-cli) @ hadoop-common --- [INFO] Skipping JaCoCo execution due to missing
execution data file. [INFO] ----- [INFO] BUILD SUCCESS [INFO] -----
----- [INFO] Total time: 34.185 s [INFO] Finished at: 2023-11-
16T20:02:11-06:00 [INFO] ----- SO i should implement the
logsContainWarning? if yes can you provide the same
-----
```



Certainly! You can implement the `logsContainWarning` method to check if logs contain a warning. Since you're working with test cases, you might want to use a logging framework like SLF4J. Below is an example implementation using SLF4J:

```
java
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class ALineReaderTest {

    // Logger instance
    private static final Logger logger = LoggerFactory.getLogger(ALineReaderTest.class);

    // ... (rest of your code)

    // Define logsContainWarning method
    private boolean logsContainWarning() {
        // Check if logs contain a warning
        // Replace "WARN" with the actual log level used for warnings in your application
        return logContainsLevel("WARN");
    }

    // Helper method to check if logs contain a specific log level
    private boolean logContainsLevel(String logLevel) {
        // Retrieve the logs and check if they contain the specified log level
        // This is a simple example, you might need to adapt it based on your logging framework
        String logs = captureLogs(); // Implement this method to capture logs

        return logs.contains(logLevel);
    }

    // Helper method to capture logs (replace with your logging framework)
    private String captureLogs() {
        // Use SLF4J's LoggerFactory to obtain the logger
        // This is just a placeholder, you should adapt it based on your logging framework
        // For SLF4J, you might need to configure an appender to capture logs during tests
        return "Example log with WARN level"; // Replace with your actual log capture logic
    }

    // ... (rest of your code)
}
```

In this example, `logsContainWarning` checks if the logs contain a warning level (you can customize the log level as needed). The `captureLogs` method is a placeholder – you should replace it with the actual logic to capture logs based on your logging framework. If you're using SLF4J, you might need to configure an appender to capture logs during tests.

Keep in mind that the specifics of capturing logs may depend on your logging framework and its configuration in your application.
