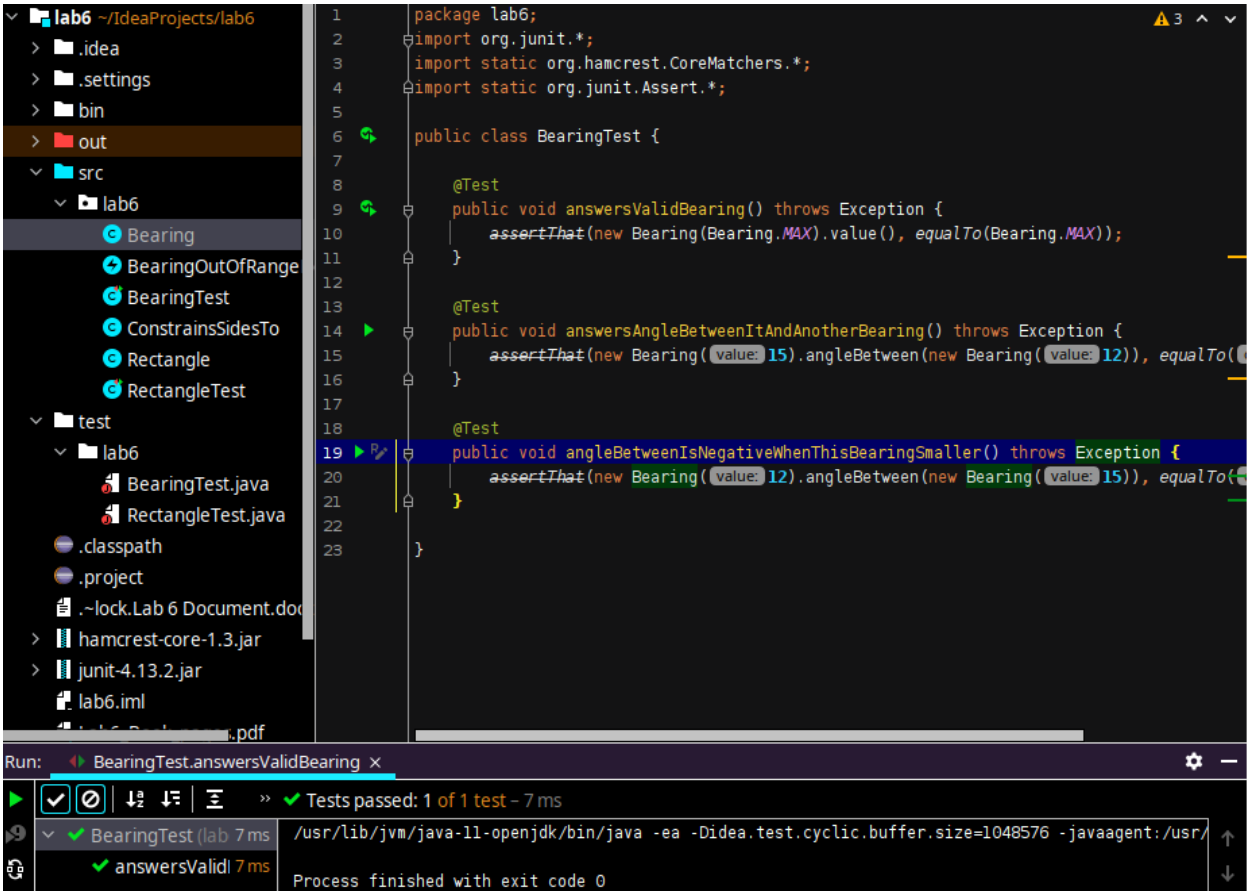
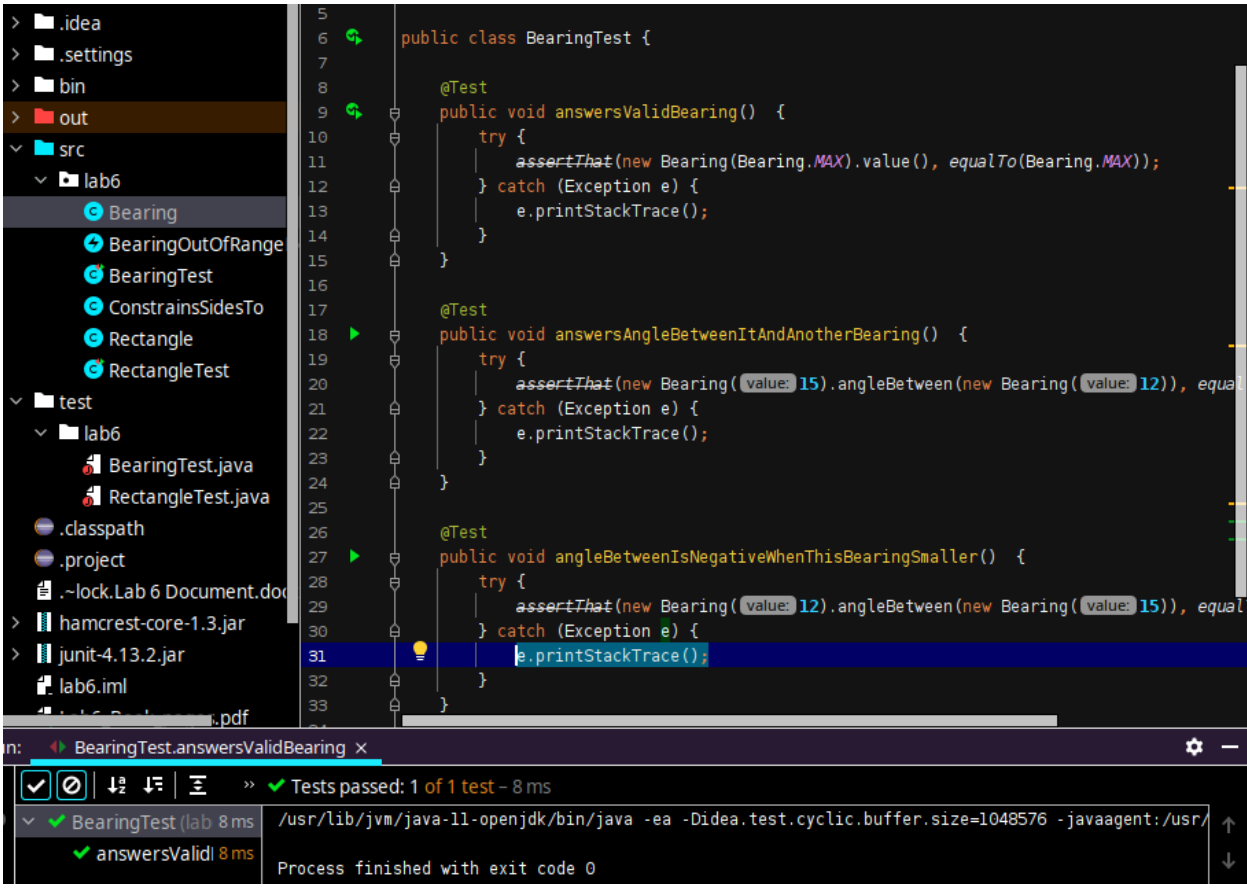
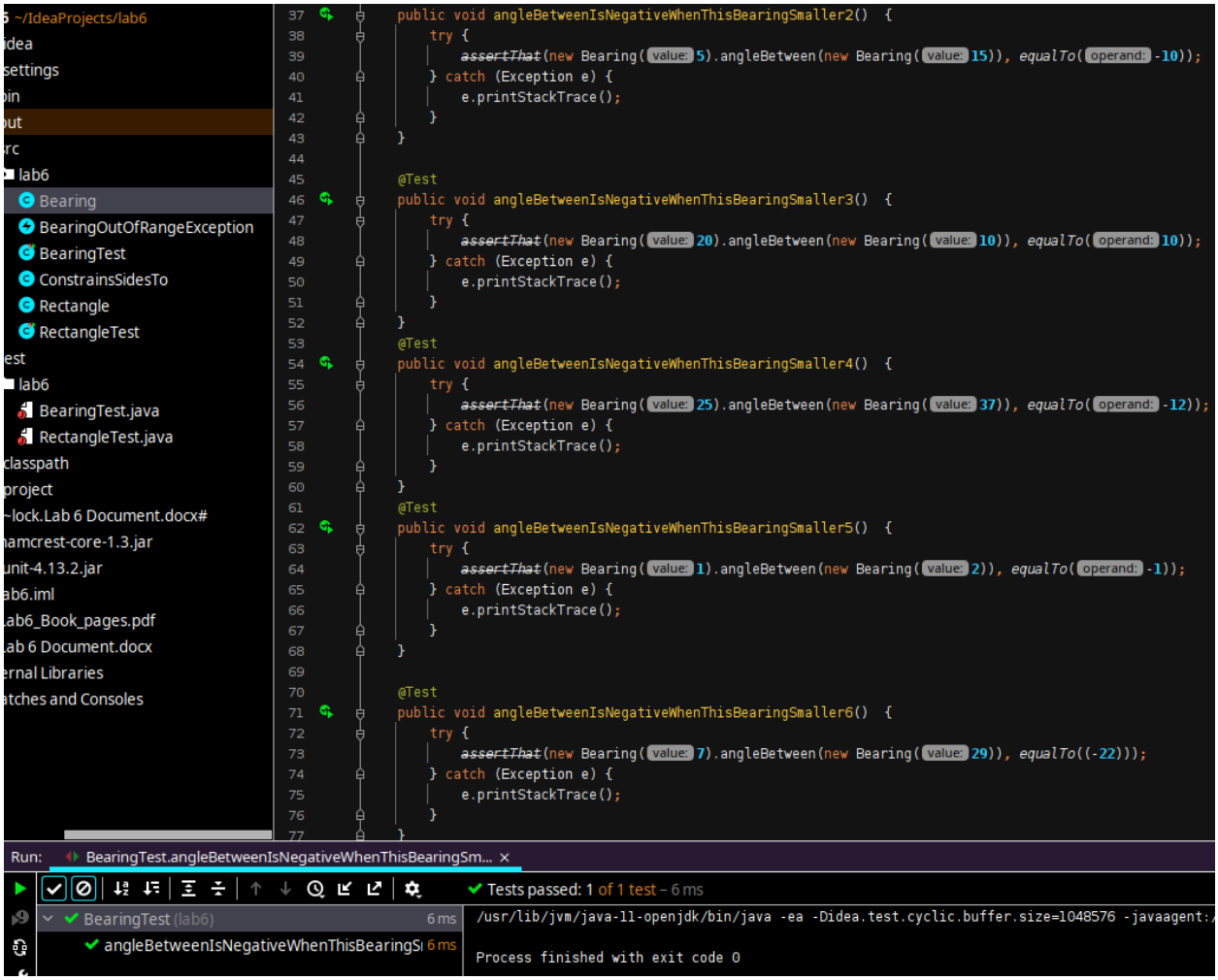
1. Part 1

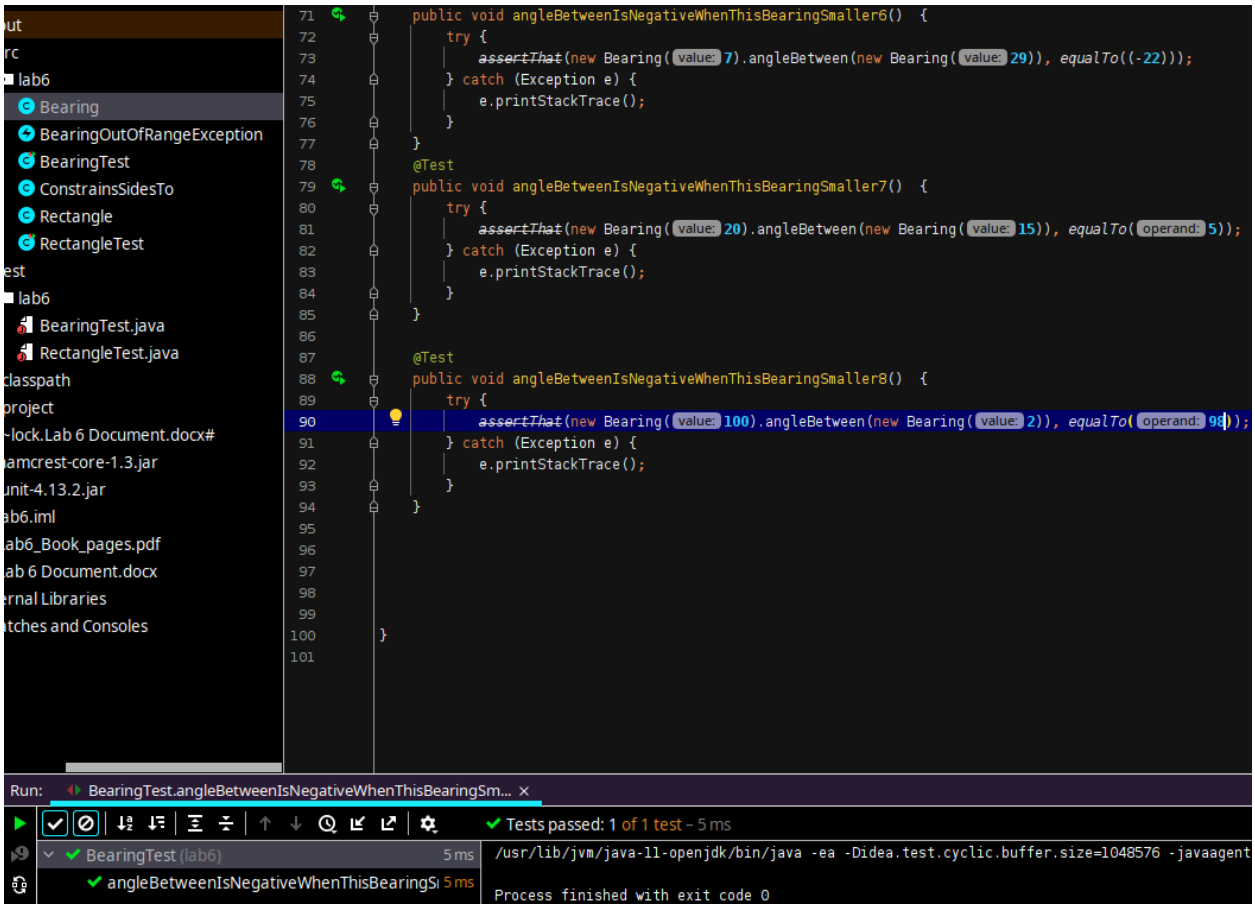
a. 
lab6 ~/IdeaProjects/lab6
.idea
.settings
bin
out
src
lab6
Bearing
BearingOutOfRange
BearingTest
ConstrainsSidesTo
Rectangle
RectangleTest
test
lab6
BearingTest.java
RectangleTest.java
.classpath
.project
.~lock.Lab 6 Document.doc
hamcrest-core-1.3.jar
junit-4.13.2.jar
lab6.iml
.pdf
package lab6;
import org.junit.*;
import static org.hamcrest.CoreMatchers.*;
import static org.junit.Assert.*;
public class BearingTest {
@Test
public void answersValidBearing() throws Exception {
assertThat(new Bearing(Bearing.MAX).value(), equalTo(Bearing.MAX));
}
@Test
public void answersAngleBetweenItAndAnotherBearing() throws Exception {
assertThat(new Bearing(value: 15).angleBetween(new Bearing(value: 12)), equalTo(
}
@Test
public void angleBetweenIsNegativeWhenThisBearingSmaller() throws Exception {
assertThat(new Bearing(value: 12).angleBetween(new Bearing(value: 15)), equalTo(
}
}
Run:
BearingTest.answersValidBearing
Tests passed: 1 of 1 test – 7 ms
BearingTest (lab 7 ms
answersValidi 7 ms
/usr/lib/jvm/java-11-openjdk/bin/java -ea -Didea.test.cyclic.buffer.size=1048576 -javaagent:/usr/
Process finished with exit code 0

b.


.idea
.settings
bin
out
src
lab6
Bearing
BearingOutOfRange
BearingTest
ConstrainsSidesTo
Rectangle
RectangleTest
test
lab6
BearingTest.java
RectangleTest.java
.classpath
.project
.~lock.Lab 6 Document.do
hamcrest-core-1.3.jar
junit-4.13.2.jar
lab6.iml
.pdf
public class BearingTest {
@Test
public void answersValidBearing() {
try {
assertThat(new Bearing(Bearing.MAX).value(), equalTo(Bearing.MAX));
} catch (Exception e) {
e.printStackTrace();
}
}
@Test
public void answersAngleBetweenItAndAnotherBearing() {
try {
assertThat(new Bearing(value: 15).angleBetween(new Bearing(value: 12)), equa
} catch (Exception e) {
e.printStackTrace();
}
}
@Test
public void angleBetweenIsNegativeWhenThisBearingSmaller() {
try {
assertThat(new Bearing(value: 12).angleBetween(new Bearing(value: 15)), equal
} catch (Exception e) {
e.printStackTrace();
}
}
BearingTest.answersValidBearing
Tests passed: 1 of 1 test – 8 ms
BearingTest (lab 8 ms
answersValidI 8 ms
/usr/lib/jvm/java-11-openjdk/bin/java -ea -Didea.test.cyclic.buffer.size=1048576 -javaagent:/usr/
Process finished with exit code 0

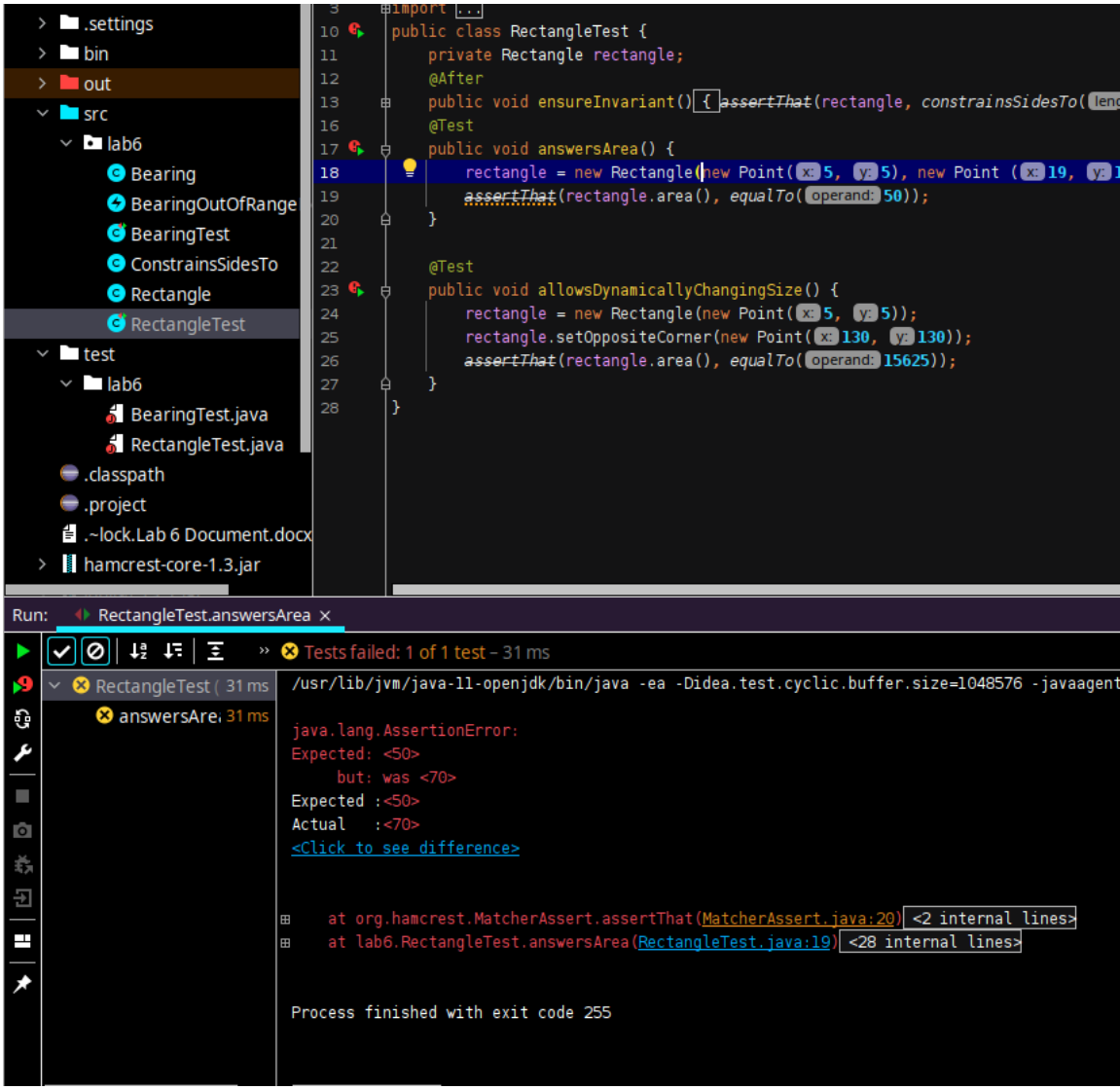
2.


~/IdeaProjects/lab6
idea
settings
bin
out
rc
lab6
Bearing
BearingOutOfRangeException
BearingTest
ConstrainsSidesTo
Rectangle
RectangleTest
est
lab6
BearingTest.java
RectangleTest.java
classpath
project
lock.Lab 6 Document.docx#
amcrest-core-1.3.jar
unit-4.13.2.jar
ab6.iml
ab6_Book_pages.pdf
ab 6 Document.docx
rnal Libraries
tches and Consoles
public void angleBetweenIsNegativeWhenThisBearingSmaller2() {
try {
assertThat(new Bearing(value: 5).angleBetween(new Bearing(value: 15)), equalTo(operand: -10));
} catch (Exception e) {
e.printStackTrace();
}
}
@Test
public void angleBetweenIsNegativeWhenThisBearingSmaller3() {
try {
assertThat(new Bearing(value: 20).angleBetween(new Bearing(value: 10)), equalTo(operand: 10));
} catch (Exception e) {
e.printStackTrace();
}
}
@Test
public void angleBetweenIsNegativeWhenThisBearingSmaller4() {
try {
assertThat(new Bearing(value: 25).angleBetween(new Bearing(value: 37)), equalTo(operand: -12));
} catch (Exception e) {
e.printStackTrace();
}
}
@Test
public void angleBetweenIsNegativeWhenThisBearingSmaller5() {
try {
assertThat(new Bearing(value: 1).angleBetween(new Bearing(value: 2)), equalTo(operand: -1));
} catch (Exception e) {
e.printStackTrace();
}
}
@Test
public void angleBetweenIsNegativeWhenThisBearingSmaller6() {
try {
assertThat(new Bearing(value: 7).angleBetween(new Bearing(value: 29)), equalTo((-22)));
} catch (Exception e) {
e.printStackTrace();
}
}
Run:
BearingTest.angleBetweenIsNegativeWhenThisBearingSm...
Tests passed: 1 of 1 test – 6 ms
BearingTest (lab6) 6 ms
angleBetweenIsNegativeWhenThisBearingS 6 ms
/usr/lib/jvm/java-11-openjdk/bin/java -ea -Didea.test.cyclic.buffer.size=1048576 -javaagent:
Process finished with exit code 0

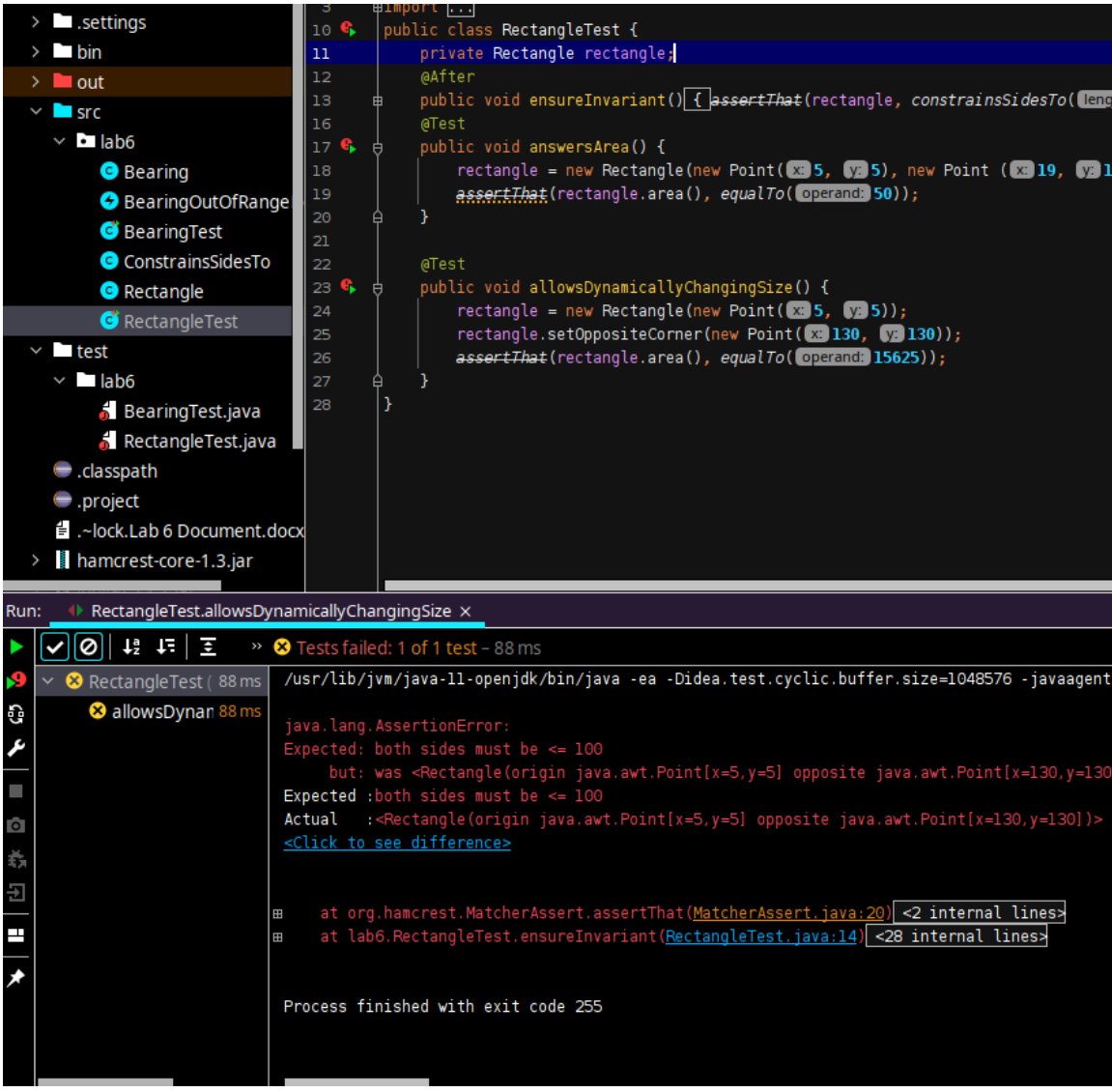


3. Part 3
    a. Tests

i.

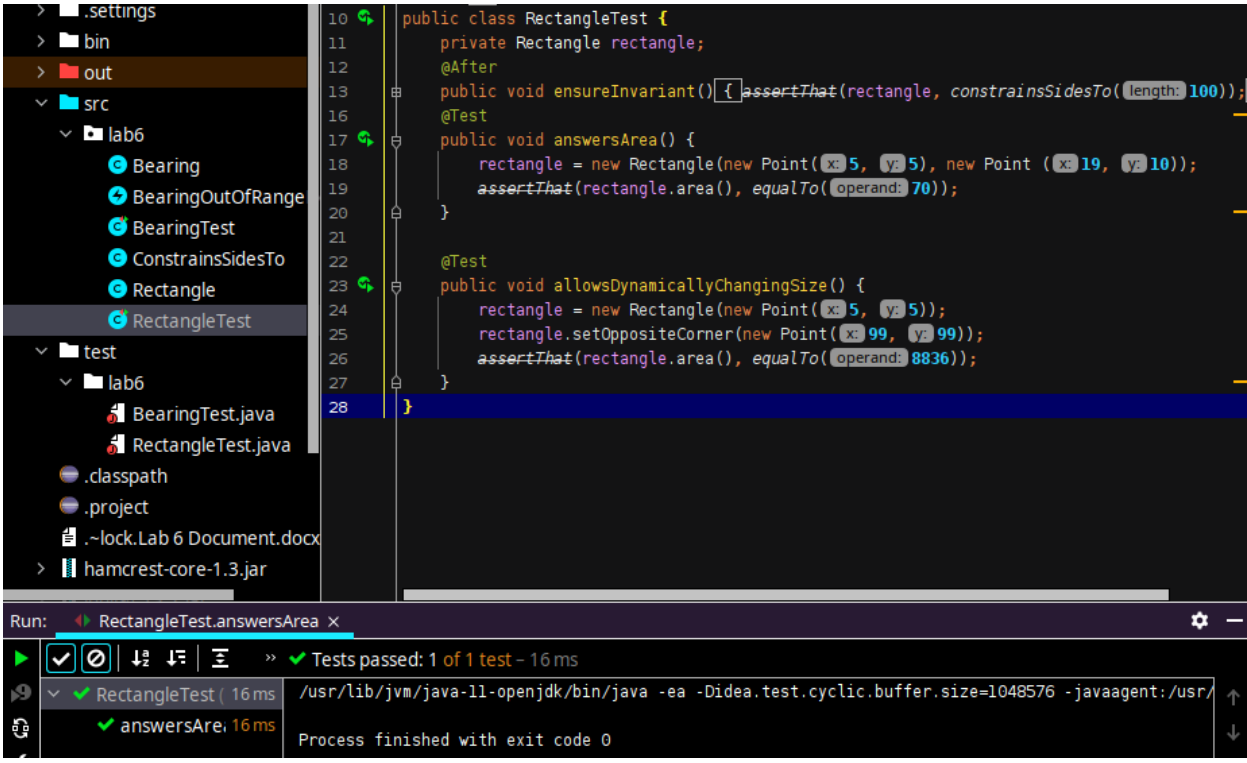

.settings
bin
out
src
lab6
Bearing
BearingOutOfRange
BearingTest
ConstrainsSidesTo
Rectangle
RectangleTest
test
lab6
BearingTest.java
RectangleTest.java
.classpath
.project
.~lock.Lab 6 Document.docx
hamcrest-core-1.3.jar
import ...
public class RectangleTest {
private Rectangle rectangle;
@After
public void ensureInvariant() { assertThat(rectangle, constrainsSidesTo(
@Test
public void answersArea() {
rectangle = new Rectangle(new Point(x: 5, y: 5), new Point (x: 19, y:
assertThat(rectangle.area(), equalTo(operand: 50));
}
@Test
public void allowsDynamicallyChangingSize() {
rectangle = new Rectangle(new Point(x: 5, y: 5));
rectangle.setOppositeCorner(new Point(x: 130, y: 130));
assertThat(rectangle.area(), equalTo(operand: 15625));
}
}
Run:
RectangleTest.answersArea
Tests failed: 1 of 1 test – 31 ms
RectangleTest (31 ms
answersAre, 31 ms
/usr/lib/jvm/java-11-openjdk/bin/java -ea -Didea.test.cyclic.buffer.size=1048576 -javaagent
java.lang.AssertionError:
Expected: <50>
but: was <70>
Expected :<50>
Actual :<70>
<Click to see difference>
at org.hamcrest.MatcherAssert.assertThat(MatcherAssert.java:20) <2 internal lines>
at lab6.RectangleTest.answersArea(RectangleTest.java:19) <28 internal lines>
Process finished with exit code 255

ii.

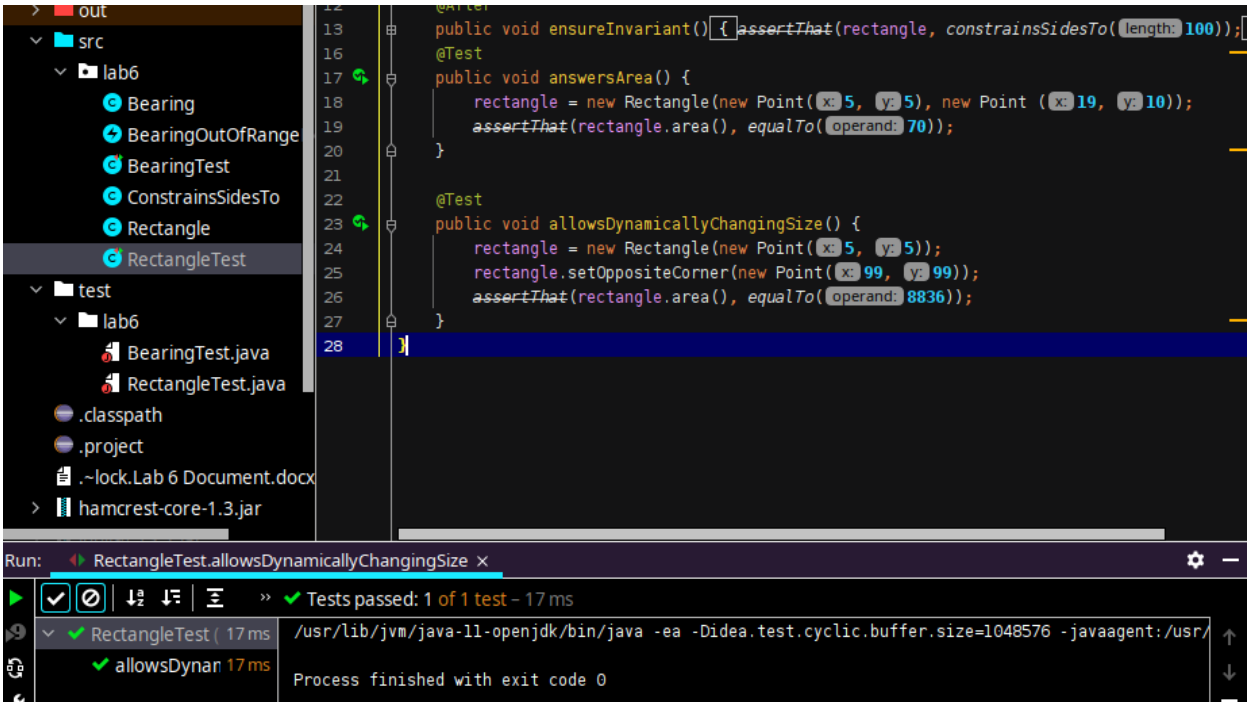


4. Fix Tests

a.



b.



5.

a. A throw exception throws a user-defined exception and allows a “fallback” when the code fails and proceeds to be an expected error when that exception occurs. This is why it fixes the code, because it allows the code to run even though there might be a potential problem, and allows it to be caught without causing any other problem related to the exception.

b. A try-catch method is code that also allows “fallback” but runs the code that is in the try method and if any exception is caught then it runs the code within the catch method. It fixes the code, by allowing you to run a separate part of the code when an expected error occurs. This allows the code to pass.
c. The try-catch will execute code that can throw exceptions, and use that exception to run additional code. Meanwhile, throwing an exception will allow you to throw an exception of a specific type but won’t allow you to run additional code when it's thrown.