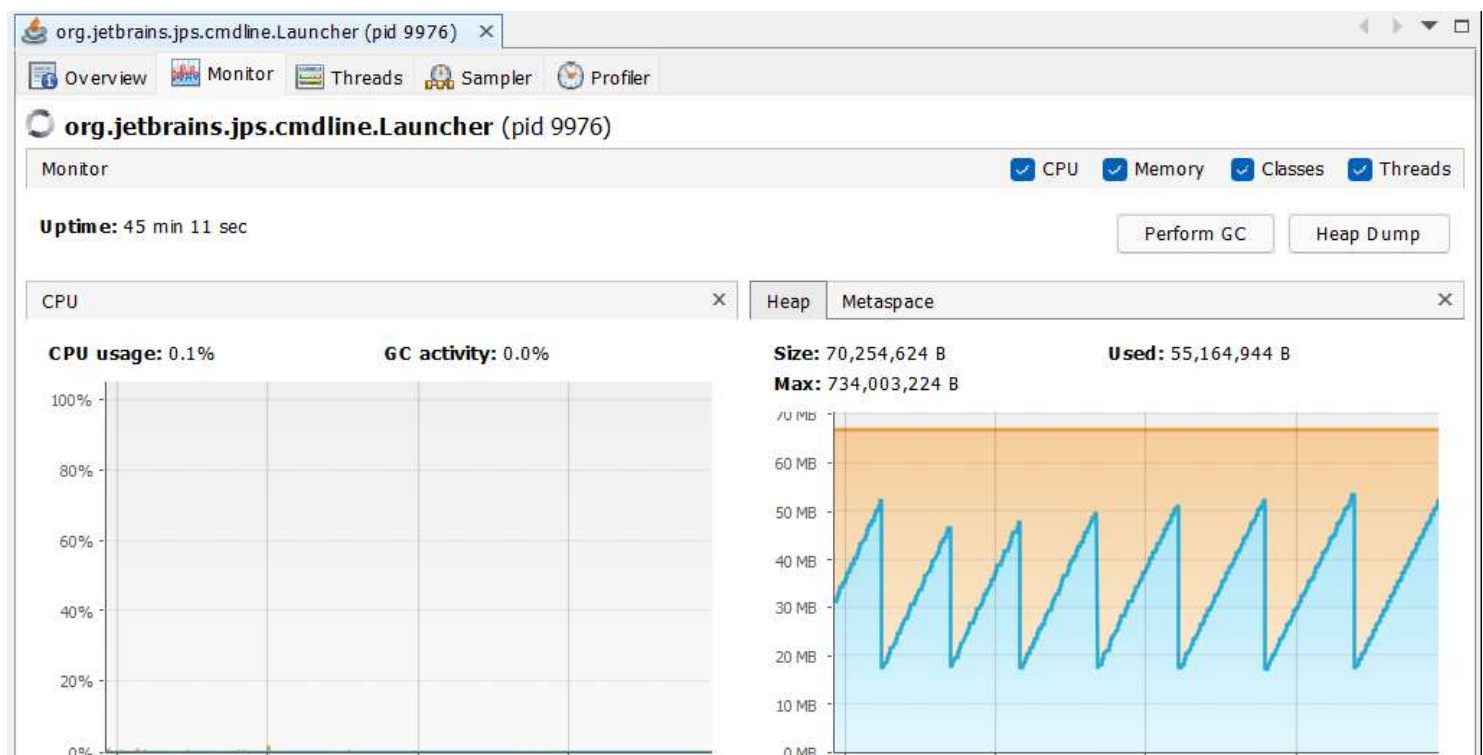
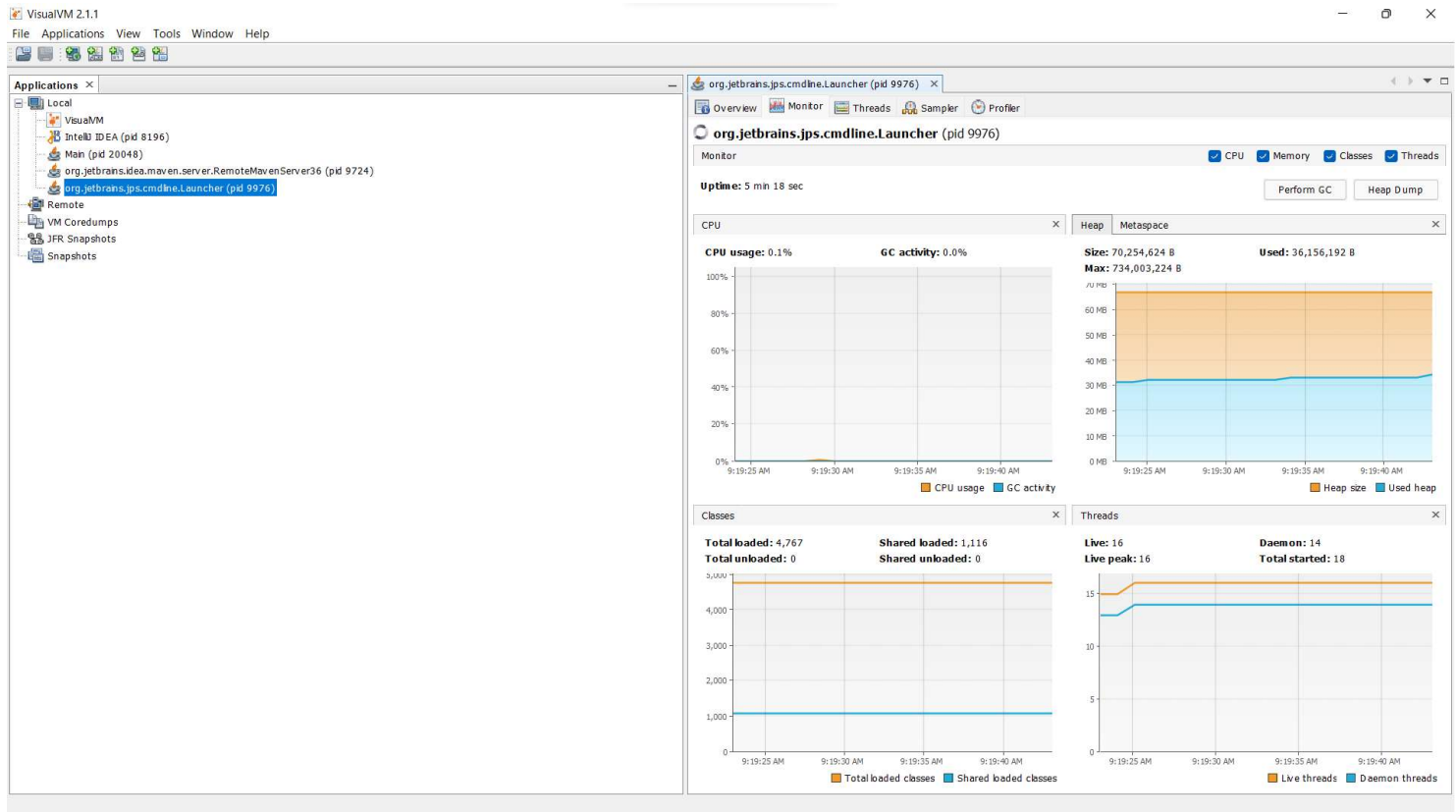
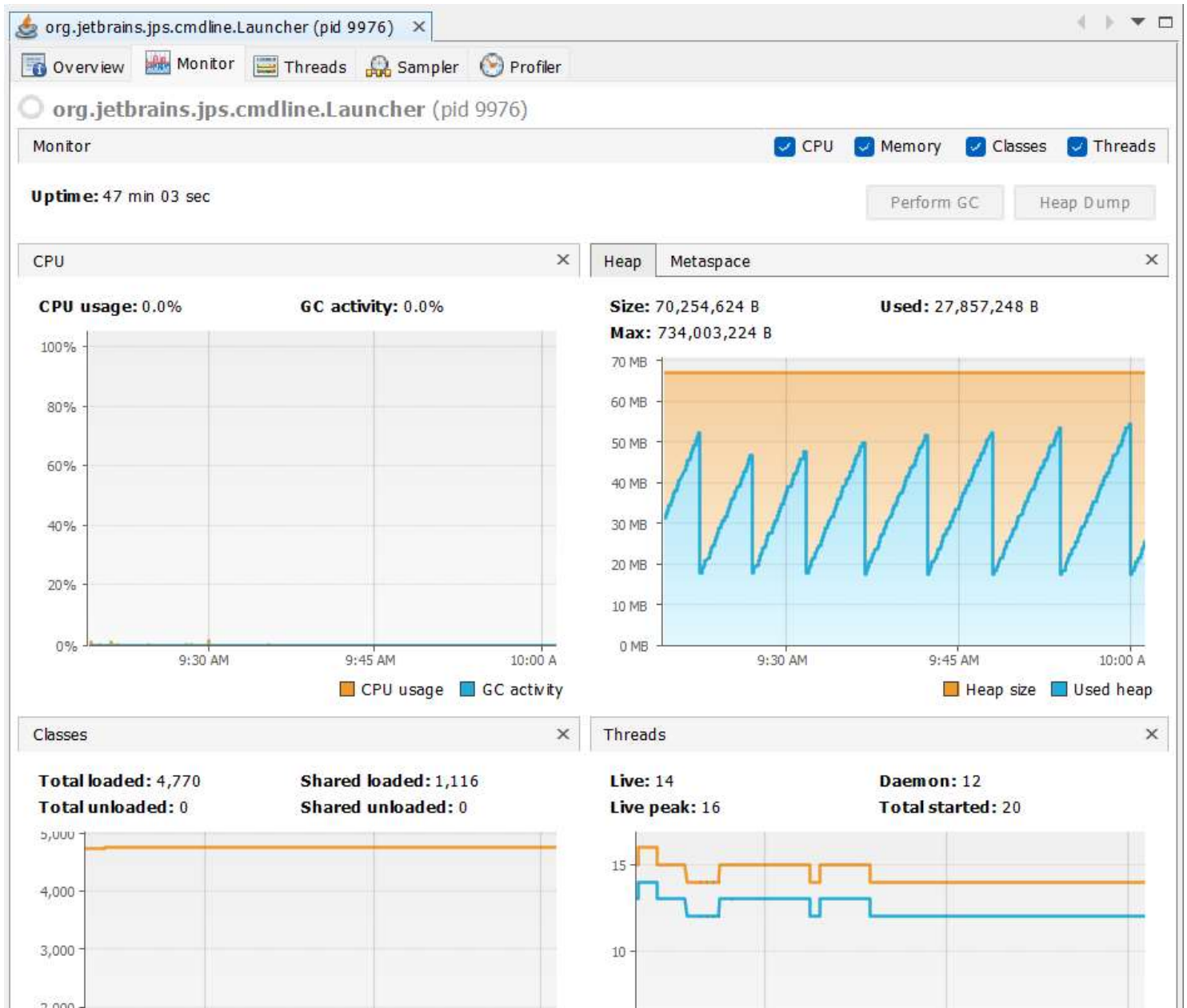
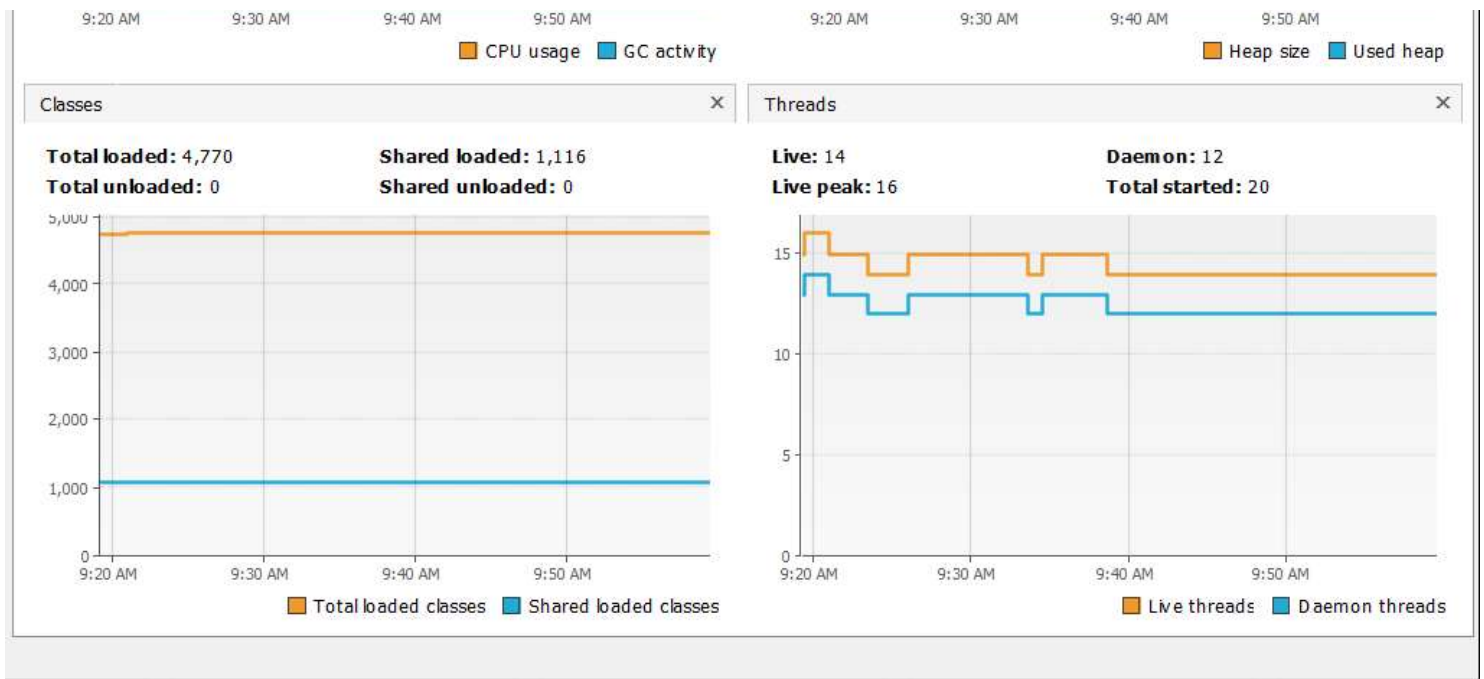


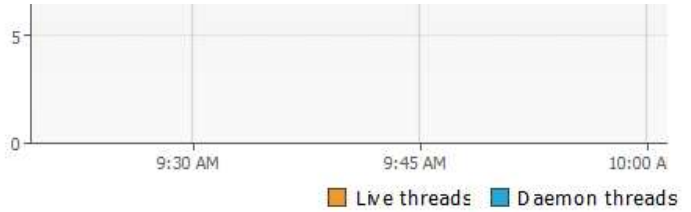
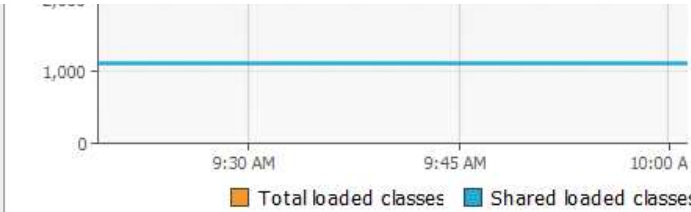
# Dynamic Analysis

It is noticed that from the monitor at the beginning of the program the program constantly has approximately 30mb used by memory. After running the tests, I ran a few buttons in the program, the memory used began to oscillate between 20-50mb.

The number of threads remains unchanged and the processor was used very little at first.







# SpotBugs Report

Produced using [SpotBugs](#)4.4.2.

Project: VVS-Project[untitled]

## Metrics

402 lines of code analyzed, in 8 classes, in 3 packages.

Metric	Total	Density*
High Priority Warnings	2	4.98
Medium Priority Warnings	7	17.41
Total Warnings	9	22.39

(\* Defects per Thousand lines of non-commenting source statements)

## Summary

Warning Type	Number
<a href="#">Bad practice Warnings</a>	2
<a href="#">Internationalization Warnings</a>	2
<a href="#">Malicious code vulnerability Warnings</a>	1
<a href="#">Performance Warnings</a>	1
<a href="#">Dodgy code Warnings</a>	3
Total	9

# Warnings

Click on each warning link to see a full description of the issue, and details of how to resolve it.

## Bad practice Warnings

Warning	Priority	Details
<a href="#">Method names should start with a lower case letter</a>	Medium	<p>The method name WebServerTest.PortNotOk() doesn't start with a lower case letter</p> <p>In file WebServerTest.java, lines 50 to 52 In class WebServerTest In method WebServerTest.PortNotOk() At WebServerTest.java:[lines 50-52]</p>
<a href="#">Method names should start with a lower case letter</a>	Medium	<p>The method name WebServerTest.PortOk() doesn't start with a lower case letter</p> <p>In file WebServerTest.java, lines 45 to 46 In class WebServerTest In method WebServerTest.PortOk() At WebServerTest.java:[lines 45-46]</p>

## Internationalization Warnings

Warning	Priority	Details
---------	----------	---------

[Reliance on default encoding](#)

High

Found reliance on default encoding in server.ServerController.clientHandle(Socket): new java.io.InputStreamReader(InputStream)

In file ServerController.java ServerController.java, line 117 117  
In class server.ServerController  
In method server.ServerController.clientHandle(Socket)  
Called method new java.io.InputStreamReader(InputStream)  
At ServerController.java:[line 117]  
At ServerController.java:[line 117]

[Reliance on default encoding](#)

High

Found reliance on default encoding in server.ServerController.respond(OutputStream, String, String, byte[]): String.getBytes()

In file ServerController.java ServerController.java, line 89 89  
In class server.ServerController  
In method server.ServerController.respond(OutputStream, String, String, byte[])  
Called method String.getBytes()  
At ServerController.java:[line 89]  
At ServerController.java:[line 89]

## Malicious code vulnerability Warnings

**Warning**

**Priority**

**Details**

[May expose internal representation by incorporating reference to mutable object](#)

Medium

new server.ServerController(WebServer) may expose internal representation by storing an externally mutable object into ServerController.webServer

In file ServerController.java ServerController.java, line 20 20  
In class server.ServerController  
In method new server.ServerController(WebServer)

Field server.ServerController.webServer  
Local variable named webServer  
At ServerController.java:[line 20]  
At ServerController.java:[line 20]

## Performance Warnings

Warning	Priority	Details
<a href="#">Method invokes inefficient new String() constructor</a>	Medium	WebServerTest.addPageOnLvllsOk() invokes inefficient new String() constructor  In file WebServerTest.java WebServerTest.java, line 120 120 In class WebServerTest In method WebServerTest.addPageOnLvllsOk() At WebServerTest.java:[line 120] At WebServerTest.java:[line 120]

## Dodgy code Warnings

Warning	Priority	Details
<a href="#">Dead store to local variable</a>	Medium	Dead store to serverSocket in ServerControllerTest.AcceptNotWorking()  In file ServerControllerTest.java ServerControllerTest.java, line 174 174 In class ServerControllerTest In method ServerControllerTest.AcceptNotWorking() Local variable named serverSocket

At ServerControllerTest.java:[line 174]  
At ServerControllerTest.java:[line 174]

[Dead store to local variable](#) Medium

Dead store to s2 in ServerControllerTest.TestPortNotAvailabe()

In file ServerControllerTest.java ServerControllerTest.java, line 55 55  
In class ServerControllerTest  
In method ServerControllerTest.TestPortNotAvailabe()  
Local variable named s2  
At ServerControllerTest.java:[line 55]  
At ServerControllerTest.java:[line 55]

[Dead store to local variable](#) Medium

Dead store to \$L2 in ServerControllerTest.TestServerSocketPortIsntOk()

In file ServerControllerTest.java ServerControllerTest.java, line 45 45  
In class ServerControllerTest  
In method ServerControllerTest.TestServerSocketPortIsntOk()  
Local variable stored in JVM register 2  
At ServerControllerTest.java:[line 45]  
At ServerControllerTest.java:[line 45]

## Warning Types

### Dead store to local variable

This instruction assigns a value to a local variable, but the value is not read or used in any subsequent instruction. Often, this indicates an error, because the value computed is never used.



Note that Sun's javac compiler often generates dead stores for final local variables. Because SpotBugs is a bytecode-based tool, there is no easy way to eliminate these false positives.

## **Method invokes inefficient new String() constructor**

Creating a new `java.lang.String` object using the no-argument constructor wastes memory because the object so created will be functionally indistinguishable from the empty string constant `""`. Java guarantees that identical string constants will be represented by the same `String` object. Therefore, you should just use the empty string constant directly.

## **Reliance on default encoding**

Found a call to a method which will perform a byte to `String` (or `String` to byte) conversion, and will assume that the default platform encoding is suitable. This will cause the application behaviour to vary between platforms. Use an alternative API and specify a charset name or `Charset` object explicitly.

## **May expose internal representation by incorporating reference to mutable object**

This code stores a reference to an externally mutable object into the internal representation of the object. If instances are accessed by untrusted code, and unchecked changes to the mutable object would compromise security or other important properties, you will need to do something different. Storing a copy of the object is better approach in many situations.

## **Method names should start with a lower case letter**

Methods should be verbs, in mixed case with the first letter lowercase, with the first letter of each internal word capitalized.

