**Software Testing**

**Project Report**

Session: Spring 2021

**Danish Hasan** MSCS20001

**Abu Bakar** MSCS-20013

**Musa Khan** MSCS-20065

**Awais** MSCS-20074

**Employee Time Reporting**

Department of Computer Science

**Information Technology University Lahore**

**Pakistan**

****

## Project Description

### Environment Setup

1. Download maven from here: <https://maven.apache.org/download.cgi>
2. Download and install the mysql workbench from here: <https://dev.mysql.com/downloads/installer/>
3. Download jdk1.8+
4. In the .\timesheet-master\build.bat, set the JAVA\_HOME to jdk path and similarly set MAVEN\_HOME to the maven path.
5. In the .\timesheet-master\run.bat, set the JAVA\_HOME and set CATALINA\_HOME to absolute path appended by ".\PaySystem\apache-tomcat-7.0.108-windows-x64\apache-tomcat-7.0.108".
6. Open Command prompt, navigate to project repository i.e .\Paysystem\timesheet-master\ and execute build.bat.
7. This will build the project.
8. Open mysql workbench and enter following two queries:
   1. drop database paysystem;
   2. create database paysystem;
9. When the database is created for first time, only execute the create query.
10. Execute run.bat.

### Description

The project is a lighter version of a pay system for managing the expenses of the employees.

* Adding the new employees in the database.
* Adding the time worked for a specific employee.
* Configuring the database settings.
* Managing the groups in the company.
* Generate the ADP reports of the employees.

### Application Running

After the local server is running, go to <http://localhost:8090/> or you can just go to the application <http://localhost:8090/PaySystem>



Enter the company name, and then click next.

Then you will be redirected to add information about the database. To avoid confusion, database username and database password are kept same.



You will be redirected to add username and password for the user purpose. These are also kept same.



You will be redirected to the login page.



After clicking login, Login using the username you set earlier.



After login you will be directed to the dashboard. Below is the full dashboard.



In the manage account section, you can add the wage.



In the manage employee section, you can add/delete the employees.



In the manage settings section, you can change the settings.



In the hour management section, you can add/delete/edit the hour types.



In the group management section, you can add/delete/edit the groups.



In the report section, you can generate the reports.



For the report generation, you can add the data for the employee.



After clicking the finalize data, a csv file is downloaded.

## White-Box Testing

**Function 1:**

Encodes a byte array into Base64 format.

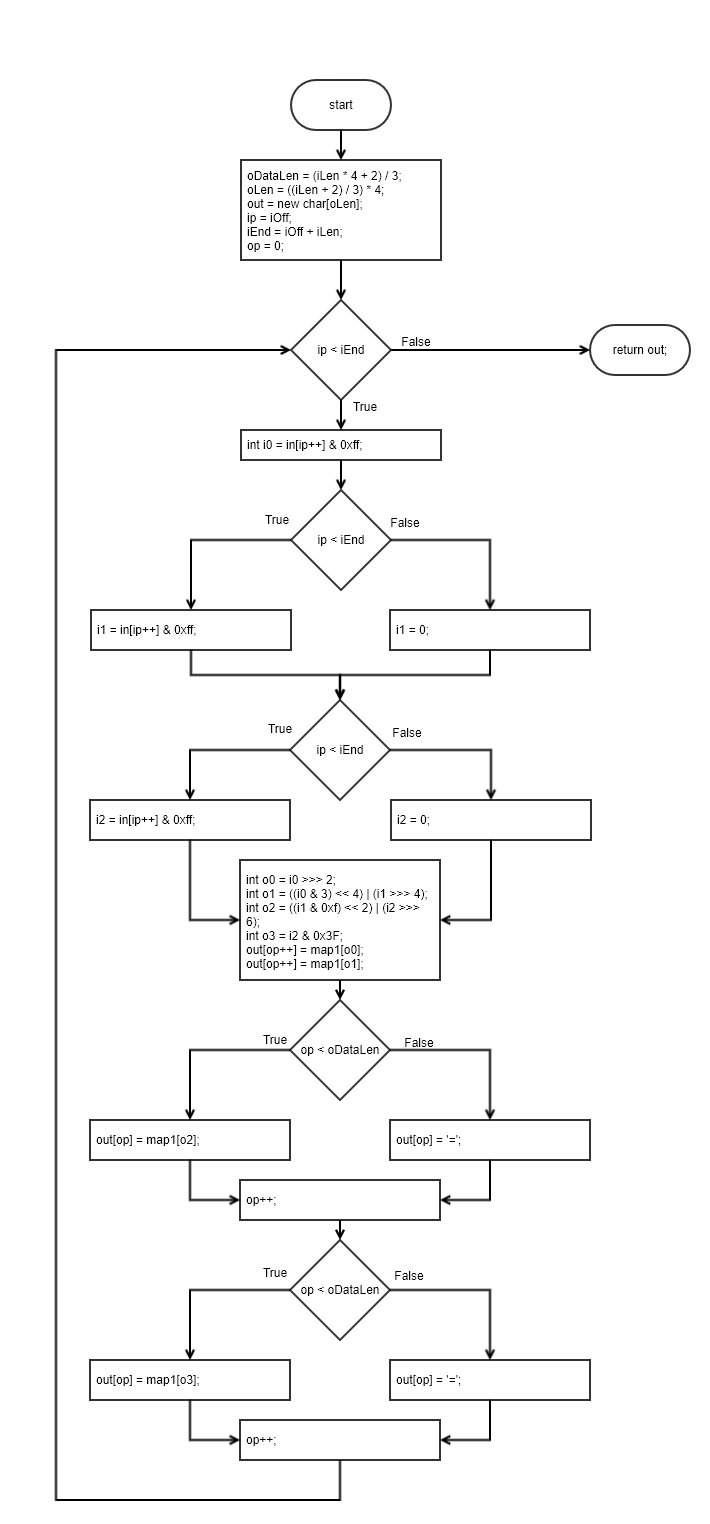
Note: map[] table is populated in another constructor function.

**Source Code:**

timesheet-master\src\main\java\timeSheet\util\properties\Base64Coder.java



**CFG:**

****

**Statement Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | In[] = {‘A’, ‘B’, ‘C’};  iOff = 0;  iLen = 3; | QUJD | QUJD | Pass | Covers all statements |

**Branch Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | In[] = {‘A’, ‘B’, ‘C’};  iOff = 0;  iLen = 3; | QUJD | QUJD | Pass | Covers 66TF, 68T, 69T, 76T, 78T |
| **2** | In[] = {‘A’, ‘B’, ‘C’};  iOff = 0;  iLen = 1; | QQ== | QQ== | Pass | Covers 66TF, 68F, 69F, 76F, 78F |

**Condition Coverage with Short Circuit Evaluation:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | In[] = {‘A’, ‘B’, ‘C’};  iOff = 0;  iLen = 3; | QUJD | QUJD | Pass | Covers 66TF, 68T, 69T, 76T, 78T |
| **2** | In[] = {‘A’, ‘B’, ‘C’};  iOff = 0;  iLen = 1; | QQ== | QQ== | Pass | Covers 66TF, 68F, 69F, 76F, 78F |

**Function 2:**

**Source Code:**

https://github.com/openjdk/jdk/tree/master/src/java.base/share/classes/java/time/Duration.java

**CFG:**

Diagram

Description automatically generated

**Statement Coverage:**

Line 414 exception case is not covered under sir’s guidance.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | text = “PT6H” | “6 hours” | “6 hours” | Pass | Covers statements from 391 to 395, 398 to 412 |
| **2** | text = “G3D” | “Exception” | “Exception” | Pass | Covers statement 419 |
| **3** | text = “-P2D” | “-2 days” | “-2 days” | Pass | Covers statement 396 |

**Branch Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | text = “PT6H” | “6 hours” | “6 hours” | Pass | Covers B393T, B395F, B404T |
| **2** | text = “G3D” | Exception | Exception | Pass | Covers B393F |
| **3** | text= “-PT6H3M” | “-6 Hours and -3 minutes” | “-6 Hours and -3 minutes” | Pass | Covers B393T, B395T |
| **4** | text= “PTDHM” | Exception | Exception | Pass | Covers B404F |

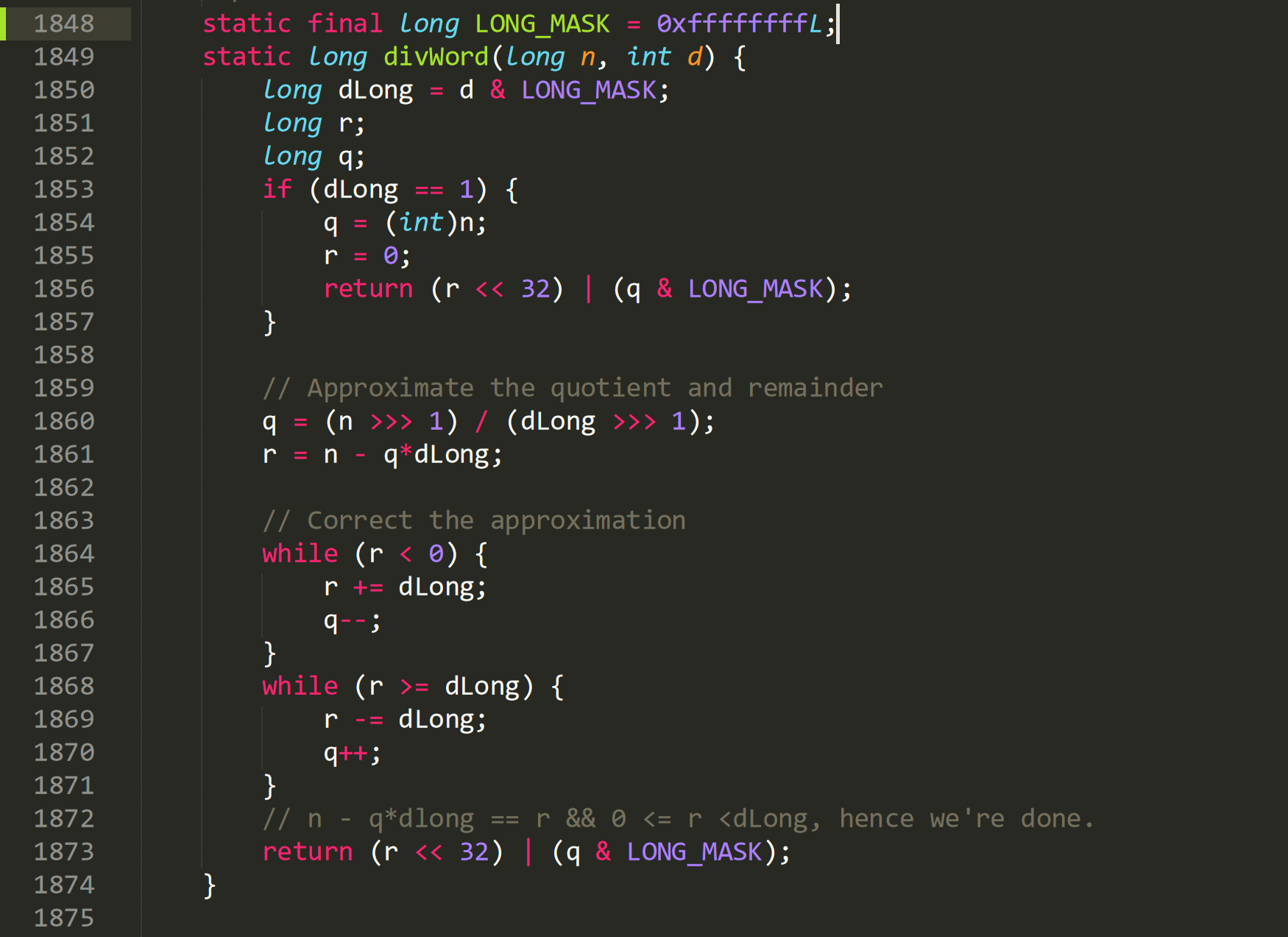
**Condition Coverage with Short Circuit Evaluation:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | text = “PT6H” | “6 hours” | “6 hours” | Pass | Covers C393T, C395F,  C404-1T |
| **2** | text = “G3D” | Exception | Exception | Pass | Covers C393F |
| **3** | text= “PT-6D6H” | “-6 Days and 6 Hours” | “-6 Days and 6 Hours” | Pass | Covers C393T, C395T, C404-1F, C404-2T |
| **4** | text= “PT-6D-6H6M” | “-6 Days and -6 Hours and 6 minutes” | “-6 Days and -6 Hours and 6 minutes” | Pass | Covers C393T, C395T, C404-1F, C404-2F, C404-3T |
| **5** | text= “PT-6D-6H-6M6S” | “-6 Days and -6 Hours and -6 minutes and 6 seconds” | “-6 Days and -6 Hours and -6 minutes and 6 seconds” | Pass | Covers C393T, C395T, C404-1F, C404-2F, C404-3F, C404-4T |
| **6** | text= “PT-6D-6H-6M-6S” | Exception | Exception | Pass | Covers C393T, C395T, C404-1F, C404-2F, C404-3F, C404-4F |

**Function 3:**

**Source Code:**

https://github.com/openjdk/jdk/tree/master/src/java.base/share/classes/java/math/ MutableBigInteger.java

****

**CFG:**

Diagram

Description automatically generated

**Statement Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | n = 16  d = 1 | 16 | 16 | Pass | Covers Statement 1850-1857 |
| **2** | n = 10  d = 3 | 4294967299 | 4294967299 | Pass | Covers Statement 1850,1851,1852, 1860-1868, 1873 |
| **3** | - | - | - | - | Statement 1869- 1870 I think this is a dead code, I could not find any such case in which the condition at 1868 becomes True |

**Branch Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | n = 16  d = 1 | 16 | 16 | Pass | Covers B1853T |
| **2** | n = 10  d = 3 | 4294967299 | 4294967299 | Pass | Covers B1853F , B1864TF, B1864F |
| **3** | - | - | - | - | Statement 1869- 1870 I think this is a dead code, I could not find any such case in which the condition at 1868 becomes True |

**Condition Coverage with Short Circuit Evaluation:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | n = 16  d = 1 | 16 | 16 | Pass | Covers C1853T |
| **2** | n = 10  d = 3 | 4294967299 | 4294967299 | Pass | Covers C1853F , C1864TF, C1864F |
| **3** | - | - | - | - | Statement 1869- 1870 I think this is a dead code, I could not find any such case in which the condition at 1868 becomes True |

**Function 4:**

Decodes a byte array from Base64 format.

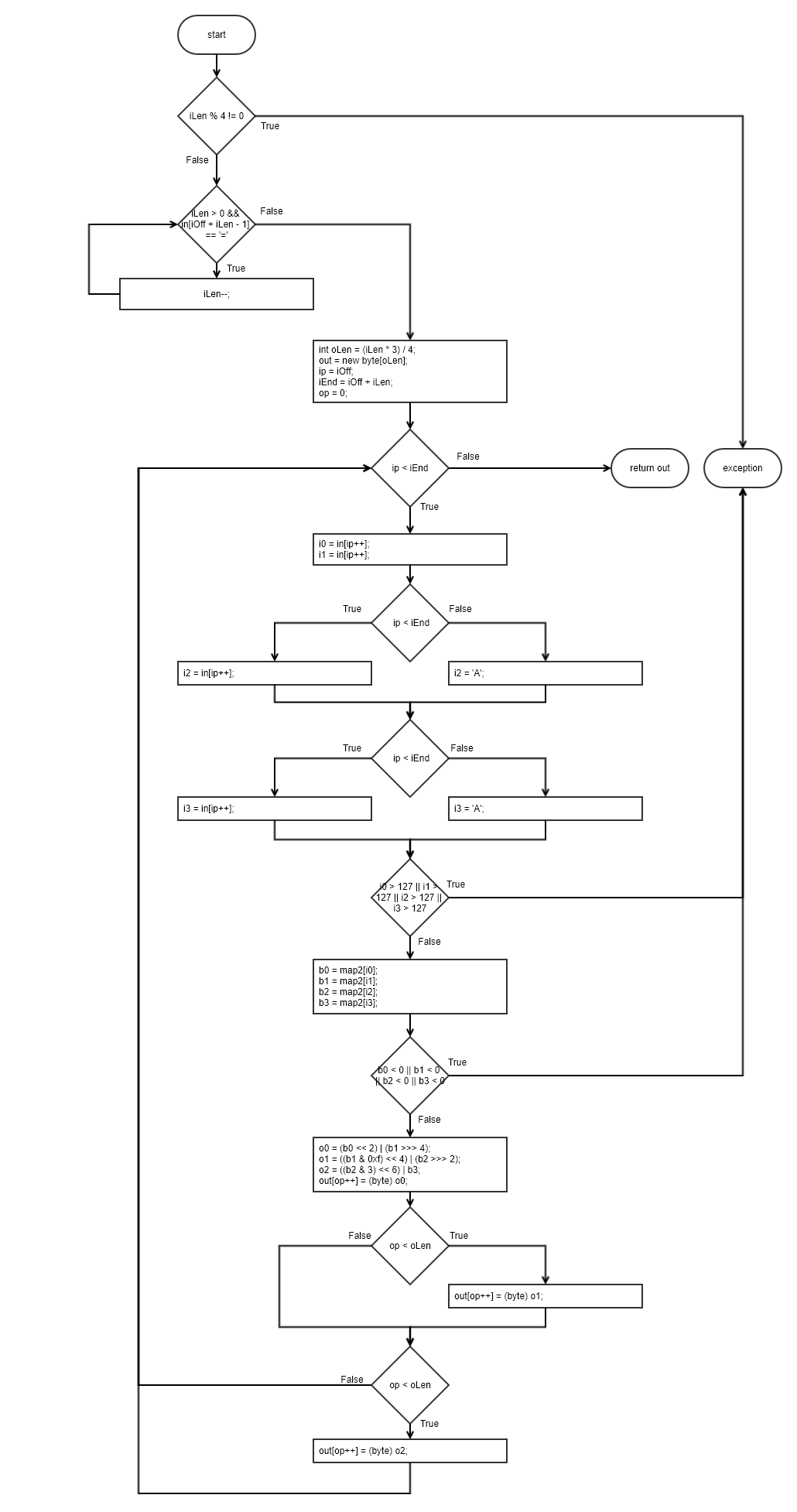
Note: map2[] table is populated in another constructor function.

**Source Code:**

timesheet-master\src\main\java\timeSheet\util\properties\Base64Coder.java



**CFG:**



**Statement Coverage:**

Exception cases are not covered under sir’s guidance.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | In[] = ‘QUJD’  iOff = 0  iLen = 4 | ‘ABC’ | ‘ABC’ | Pass | No padding |
| **2** | In[] = ‘QQ==’  iOff = 0  iLen = 4 | ‘A’ | ‘A’ | Pass | Padded with == |

**Branch Coverage:**

Exception cases are not covered under sir’s guidance.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | In[] = ‘QUJD’  iOff = 0  iLen = 4 | ‘ABC’ | ‘ABC’ | Pass | 109F, 115TF, 118T, 119T, 132T, 133T |
| **2** | In[] = ‘QQ==’  iOff = 0  iLen = 4 | ‘A’ | ‘A’ | Pass | 109TF, 115TF, 118F, 119F, 132F, 133F |

**Condition Coverage with Short Circuit Evaluation:**

Exception cases are not covered under sir’s guidance.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | In[] = ‘QUJD’  iOff = 0  iLen = 0 | Empty String | Empty String | Pass | 109aF, 115F |
| **2** | In[] = ‘QUJD’  iOff = 0  iLen = 4 | ‘ABC’ | ‘ABC’ | Pass | 109aT, 109bF, 115TF, 118T, 119T, 132T, 133T |
| **3** | In[] = ‘QQ==’  iOff = 0  iLen = 4 | ‘A’ | ‘A’ | Pass | 109aT, 109bTF, 115TF, 118F, 119F, 132F, 133F |

**Function 5:**

**Source Code:**

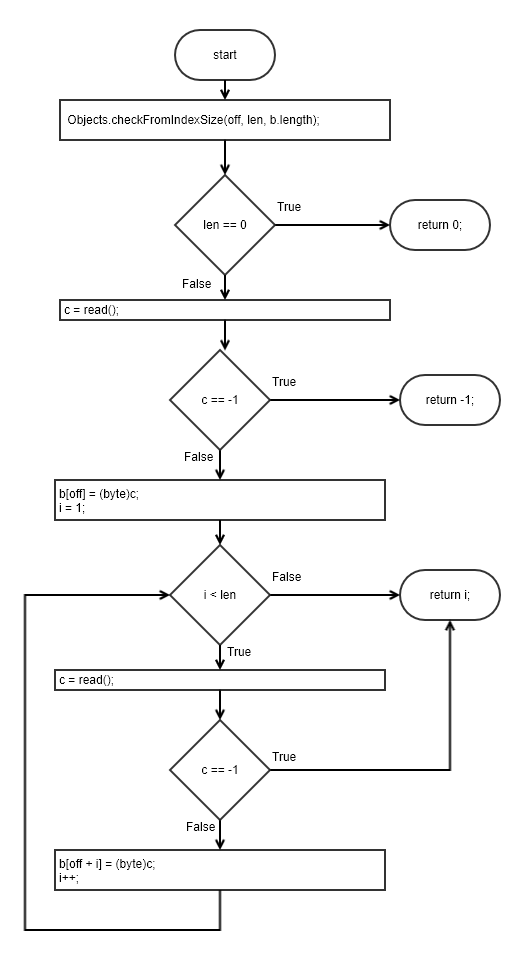
https://github.com/openjdk/jdk/blob/master/src/java.base/share/classes/java/io/InputStream.

Java

checkFromIndexSize and read are external APIs. checkFromIndexSize can be implemented as dummy stub while read is implemented as needed by each test case.



**CFG:**



**Statement Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | b[] = Empty Array  off = 0  len = 3 | 3,  b[] =‘ABC’ | 3,  b[] =‘ABC’ | Pass | External module API read() returns ‘A’, ‘B’, ‘C’ in consecutive calls. |
| **2** | b[] = Empty Array  off = 0  len = 0 | 0,  b[] = Empty Array | 0,  b[] = Empty Array | Pass | External module API read() is never called |
| **3** | b[] = Empty Array  off = 0  len = 3 | -1,  b[] = Empty Array | -1,  b[] = Empty Array | Pass | External module API read() returns -1 to notify an error at first call. |
| **4** | b[] = Empty Array  off = 0  len = 3 | 1,  b[] = ‘A’ | 1,  b[] = ‘A’ | Pass | External module API read() returns ‘A’, -1 in consecutive calls. |

**Branch Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | b[] = Empty Array  off = 0  len = 3 | 3,  b[] =‘ABC’ | 3,  b[] =‘ABC’ | Pass | External module API read() returns ‘A’, ‘B’, ‘C’ in consecutive calls.  280F, 285F, 292TF, 294F |
| **2** | b[] = Empty Array  off = 0  len = 0 | 0,  b[] = Empty Array | 0,  b[] = Empty Array | Pass | External module API read() is never called.  280T |
| **3** | b[] = Empty Array  off = 0  len = 3 | -1,  b[] = Empty Array | -1,  b[] = Empty Array | Pass | External module API read() returns -1 to notify an error at first call.  280F, 285T |
| **4** | b[] = Empty Array  off = 0  len = 3 | 1,  b[] = ‘A’ | 1,  b[] = ‘A’ | Pass | External module API read() returns ‘A’, -1 in consecutive calls.  280F, 285F, 292T, 294T |

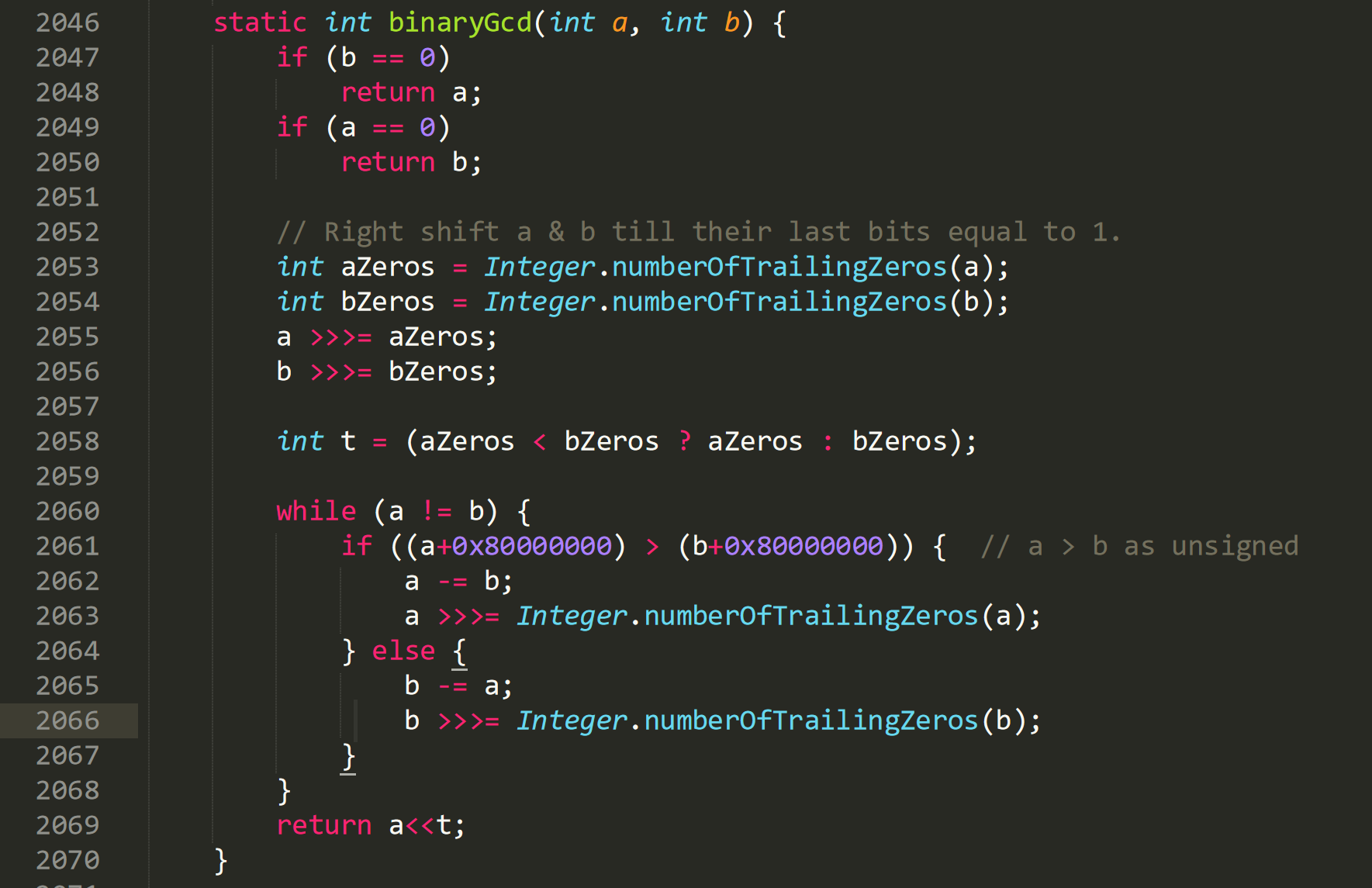
**Condition Coverage with Short Circuit Evaluation:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | b[] = Empty Array  off = 0  len = 3 | 3,  b[] =‘ABC’ | 3,  b[] =‘ABC’ | Pass | External module API read() returns ‘A’, ‘B’, ‘C’ in consecutive calls.  280F, 285F, 292TF, 294F |
| **2** | b[] = Empty Array  off = 0  len = 0 | 0,  b[] = Empty Array | 0,  b[] = Empty Array | Pass | External module API read() is never called.  280T |
| **3** | b[] = Empty Array  off = 0  len = 3 | -1,  b[] = Empty Array | -1,  b[] = Empty Array | Pass | External module API read() returns -1 to notify an error at first call.  280F, 285T |
| **4** | b[] = Empty Array  off = 0  len = 3 | 1,  b[] = ‘A’ | 1,  b[] = ‘A’ | Pass | External module API read() returns ‘A’, -1 in consecutive calls.  280F, 285F, 292T, 294T |

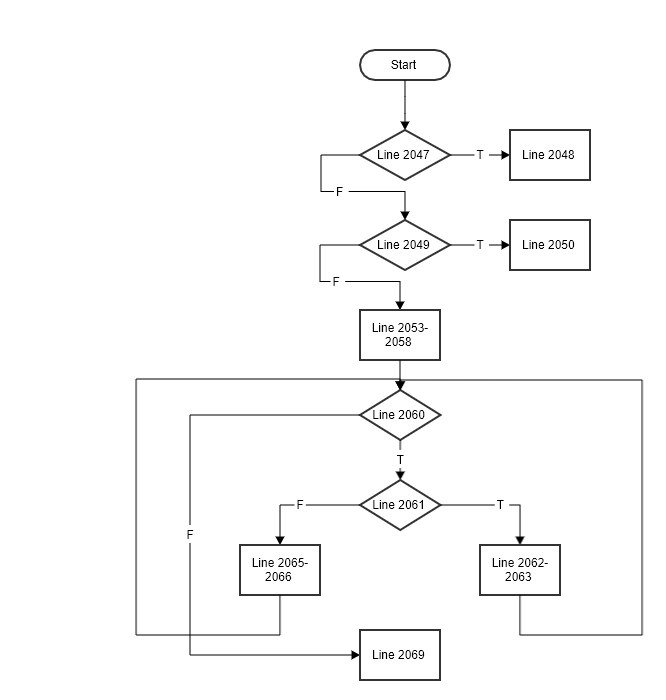
**Function 6:**

**Source Code:**

https://github.com/openjdk/jdk/tree/master/src/java.base/share/classes/java/math/ MutableBigInteger.java

****

**CFG:**

.

**Statement Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | a = 15 b = 0 | 15 | 15 | Pass | Covers statement 2047-2048 |
| **2** | a = 0  b =15 | 15 | 15 | Pass | Covers statement 2049-2050 |
| **3** | a = 98  b =56 | 14 | 14 | Pass | Covers statement 2047,2049, 2051-2069 |

**Branch Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | a = 15 b = 0 | 15 | 15 | Pass | Covers B2047T |
| **2** | a = 0  b =15 | 15 | 15 | Pass | Covers B2049T, B2047F |
| **3** | a = 98  b =56 | 14 | 14 | Pass | Covers B2047F, B2049F, B2060TF, B2061T |
| **4** | a = 56  b =98 | 14 | 14 | Pass | Covers B2047F, B2049F, B2060TF, B2061F |

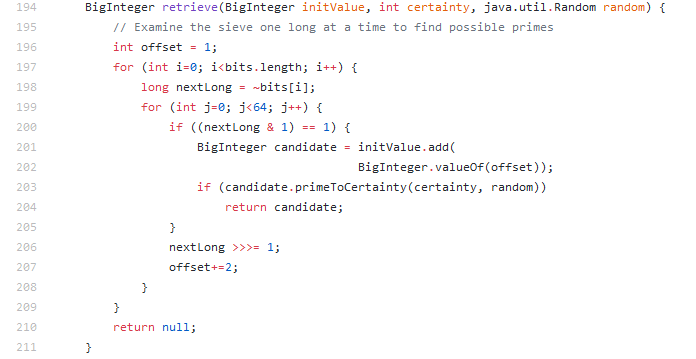
**Condition Coverage with Short Circuit Evaluation:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | a = 15 b = 0 | 15 | 15 | Pass | Covers C2047T |
| **2** | a = 0  b =15 | 15 | 15 | Pass | Covers C2049T, C2047F |
| **3** | a = 98  b =56 | 14 | 14 | Pass | Covers C2047F, C2049F, C2060TF, C2061T |
| **4** | a = 56  b =98 | 14 | 14 | Pass | Covers C2047F, C2049F, C2060TF, C2061F |

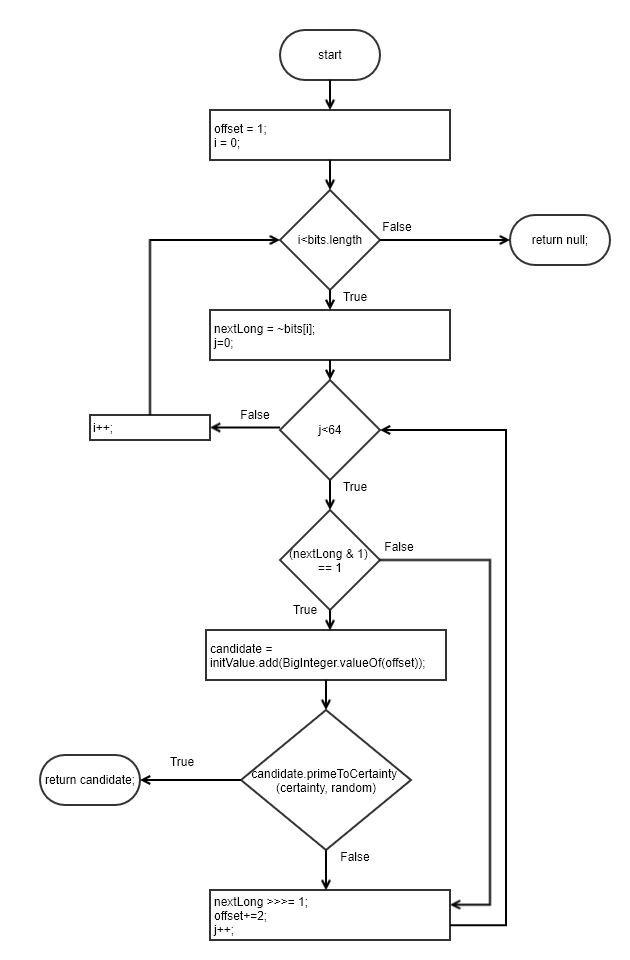
**Source Code:**

<https://github.com/openjdk/jdk/blob/master/src/java.base/share/classes/java/math/BitSieve.java>

bits are sieve bits where each bit represents a candidate odd integer. primeToCertainty is an external function which returns true if it is a prime with given probability.



**CFG:**



**Statement Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | initValue = 0;  certainity = 100;  random = 10  bits[] = b’11111010’ | 257 | 257 | Pass | Stub primeToCertainty shall return ‘False, True’ in consecutive calls. |
| **2** | initValue = 0;  certainity = 100;  random = 10  bits[] = b’11111111’ | null | null | Pass | Stub primeToCertainty shall never be called. |

**Branch Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | initValue = 0;  certainity = 100;  random = 10  bits[] = b’11111010’ | 257 | 257 | Pass | Stub primeToCertainty shall return ‘False, True’ in consecutive calls.  197T, 199TF, 200TF, 203TF |
| **2** | initValue = 0;  certainity = 100;  random = 10  bits[] = b’11111111’ | null | null | Pass | Stub primeToCertainty shall never be called.  197TF, 199TF, 200F |

**Condition Coverage with Short Circuit Evaluation:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **1** | initValue = 0;  certainity = 100;  random = 10  bits[] = b’11111010’ | 257 | 257 | Pass | Stub primeToCertainty shall return ‘False, True’ in consecutive calls.  197T, 199TF, 200TF, 203TF |
| **2** | initValue = 0;  certainity = 100;  random = 10  bits[] = b’11111111’ | null | null | Pass | Stub primeToCertainty shall never be called.  197TF, 199TF, 200F |

**Function 8:**

**Source Code:**

**CFG:**

Paste your CFG here.

**Statement Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **Cell 1** | Cell 2 | Cell 3 |  |  |  |
| **Cell 4** | Cell 5 | Cell 6 |  |  |  |

**Branch Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **Cell 1** | Cell 2 | Cell 3 |  |  |  |
| **Cell 4** | Cell 5 | Cell 6 |  |  |  |

**Condition Coverage with Short Circuit Evaluation:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **Cell 1** | Cell 2 | Cell 3 |  |  |  |
| **Cell 4** | Cell 5 | Cell 6 |  |  |  |

**Function 9:**

**Source Code:**

**CFG:**

Paste your CFG here.

**Statement Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **Cell 1** | Cell 2 | Cell 3 |  |  |  |
| **Cell 4** | Cell 5 | Cell 6 |  |  |  |

**Branch Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **Cell 1** | Cell 2 | Cell 3 |  |  |  |
| **Cell 4** | Cell 5 | Cell 6 |  |  |  |

**Condition Coverage with Short Circuit Evaluation:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **Cell 1** | Cell 2 | Cell 3 |  |  |  |
| **Cell 4** | Cell 5 | Cell 6 |  |  |  |

**Function 10:**

**Source Code:**

**CFG:**

Paste your CFG here.

**Statement Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **Cell 1** | Cell 2 | Cell 3 |  |  |  |
| **Cell 4** | Cell 5 | Cell 6 |  |  |  |

**Branch Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **Cell 1** | Cell 2 | Cell 3 |  |  |  |
| **Cell 4** | Cell 5 | Cell 6 |  |  |  |

**Condition Coverage with Short Circuit Evaluation:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **Cell 1** | Cell 2 | Cell 3 |  |  |  |
| **Cell 4** | Cell 5 | Cell 6 |  |  |  |

**Function 11:**

**Source Code:**

**CFG:**

Paste your CFG here.

**Statement Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **Cell 1** | Cell 2 | Cell 3 |  |  |  |
| **Cell 4** | Cell 5 | Cell 6 |  |  |  |

**Branch Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **Cell 1** | Cell 2 | Cell 3 |  |  |  |
| **Cell 4** | Cell 5 | Cell 6 |  |  |  |

**Condition Coverage with Short Circuit Evaluation:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **Cell 1** | Cell 2 | Cell 3 |  |  |  |
| **Cell 4** | Cell 5 | Cell 6 |  |  |  |

**Function 12:**

**Source Code:**

**CFG:**

Paste your CFG here.

**Statement Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **Cell 1** | Cell 2 | Cell 3 |  |  |  |
| **Cell 4** | Cell 5 | Cell 6 |  |  |  |

**Branch Coverage:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **Cell 1** | Cell 2 | Cell 3 |  |  |  |
| **Cell 4** | Cell 5 | Cell 6 |  |  |  |

**Condition Coverage with Short Circuit Evaluation:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Input** | **Output** | **Expected Output** | **Pass/Fail** | **Comments/Remarks** |
| **Cell 1** | Cell 2 | Cell 3 |  |  |  |
| **Cell 4** | Cell 5 | Cell 6 |  |  |  |