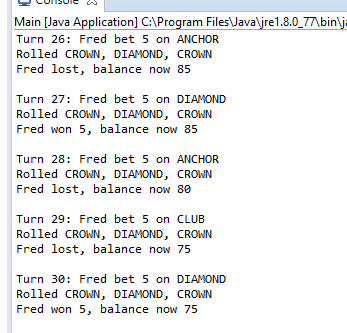
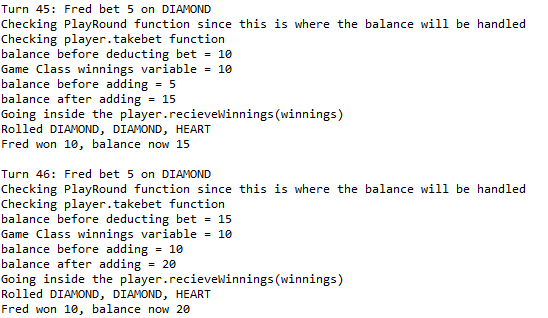
| **Test Name** | | check\_gamePlayBalance | | | |
| --- | --- | --- | --- | --- | --- |
| **Use Case Tested:** | | game\_balance | | | |
| **Test Description:** | | The test is conducted to identify if the game balance increases with the placed bet. | | | |
| **Pre-conditions** | | 1. Dice 1, 2 and 3. 2. Player = “Fred” 3. Balance > 0 && Balance < 200 4. New Game 5. totalWin = 0 6. totalLosses = 0 7. Limit == 0 || Limit > 0 | | | |
| **Post-conditions** | | Balance increases if player wins the turn. | | | |
| **Notes:** | | **The balance should increase if the player Fred wins a turn. Balance = balance + amountWon.** | | | |
| **Result (Pass/Fail/Warning/Incomplete)** | | **Fail** | | | |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | P | F |
|  | String name = "Fred";  **int** balance = 100;  **int** limit = 0;  player = **new** Player(name, balance);  player.setLimit(limit);  **int** bet = 5; | | Turn 17: Fred bet 5 on DIAMOND  Rolled ANCHOR, ANCHOR, HEART  Fred lost, balance now 35  Turn 18: Fred bet 5 on HEART  Rolled ANCHOR, ANCHOR, HEART  Fred won 5, balance now 40 |  | F |

Test Class 1 Output:

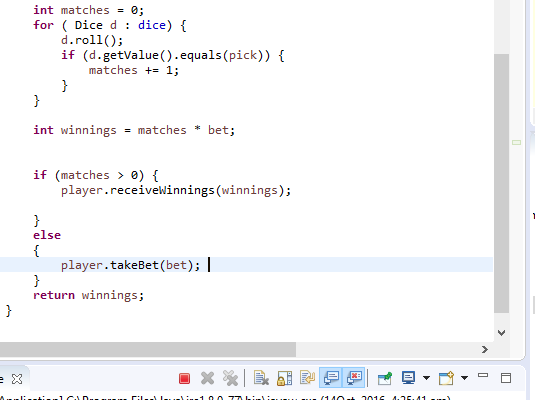


Debug Log for Test Case 1:

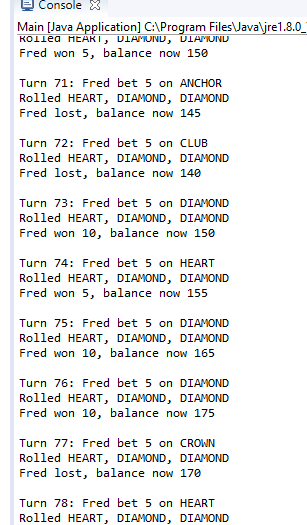


According to the Debug log takebet function inside the playRound method is deducting the balance by the bet before the recieveWinnings function thus the balance will always be deducted by bet after the winnings have been added thus to solve this takebet should be inside an if else statement.

Resolution for Test Case 1:

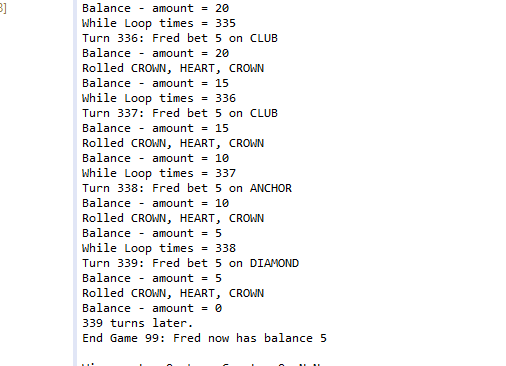


Resolution Output for Test Case 1:

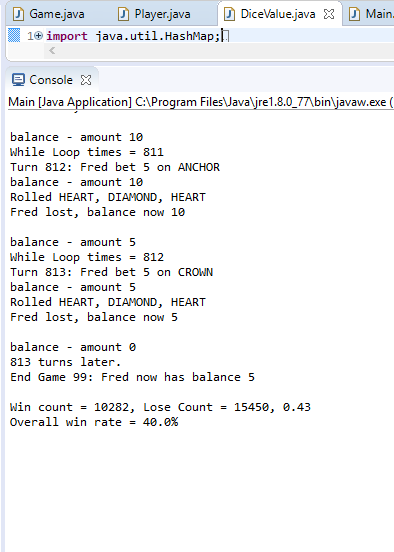


| **Test Name** | | checkGame\_endBalance | | | |
| --- | --- | --- | --- | --- | --- |
| **Use Case Tested:** | | game\_endbalance | | | |
| **Test Description:** | | This test is conducted in order to determine if the game end balance is equal to the limit. | | | |
| **Pre-conditions** | | String name = "Fred";  **int** balance = 100;  **int** limit = 0;  player = **new** Player(name, balance);  player.setLimit(limit); | | | |
| **Post-conditions** | | End Game 87: Fred now has balance 0 | | | |
| **Notes:** | | **The balance should equal to the limit stated** | | | |
| **Result (Pass/Fail/Warning/Incomplete)** | | **Fail** | | | |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | P | F |
|  | String name = "Jack";  **int** balance = 75;  **int** limit = 0;  player = **new** Player(name, balance);  player.setLimit(limit);  **int** bet = 5; | | End Game 32: Jack now has balance 0 |  | F |

Test Class 2 Output:

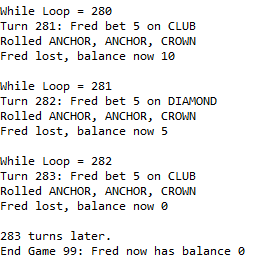


Debug Log for Test Case 2:

According to the Debug we can come to a resolution that

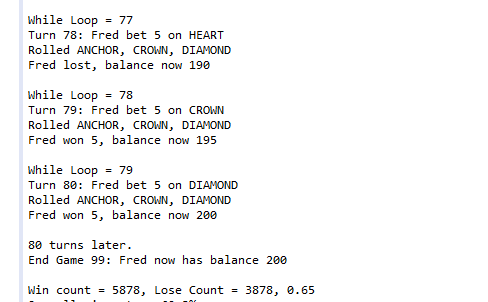
balance-amount >= limit since the value cannot be equal to limit until = is stated.

Resolution Output for Test Case 2:

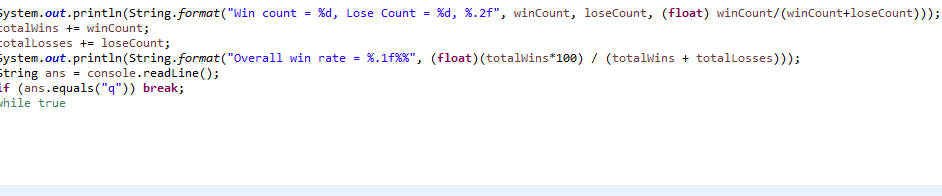


| **Test Name** | | Game win Ratio Test | | | |
| --- | --- | --- | --- | --- | --- |
| **Use Case Tested:** | | winRatio\_check | | | |
| **Test Description:** | | This test will determine if the game will provide the correct win ratio which should be approximate to 0.42. | | | |
| **Pre-conditions** | |  | | | |
| **Post-conditions** | | Win count = 1806, Lose Count = 2772, 0.41 | | | |
| **Notes:** | | **Overall Win Ratio should be approximate to 0.42** | | | |
| **Result (Pass/Fail/Warning/Incomplete)** | | **Fail** | | | |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | P | F |
|  |  | |  |  | F |

Test Class 3 Output:



Debug Log for Test Case 3:



The following formula needs to be adjusted to winCount \*108/(winCount+loseCount)/100 likewise no debug is necessary as the orgin of the problem can be sighted.

Resolution Test Case 3:

