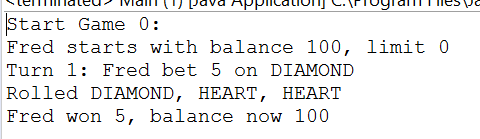
| **Test Name** | | | Test one match adds 1Xbet to balance. | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Use Case Tested:** | | | Play Game | | | |
| **Test Description:** | | | Tests that when the player makes one match, he receives 1X his bet added to his balance. | | | |
| **Pre-conditions** | | | Bug 4 must be resolved (producing new dice faces with each roll). There must be three turns where the player makes one match. | | | |
| **Post-conditions** | | | Player’s balance has increased by 1X his bet (for the turn identified). | | | |
| **Notes:** | |  | | | | |
| **Result (Pass/Fail/Warning/Incomplete)** | |  | | | | |
|  | **TEST STEP** | | | **EXPECTED TEST RESULTS** | P | F |
|  | Run Main.java | | | Console opens and results for 100 games are displayed in it. |  |  |
|  | Look at each turn and identify one where the player has made exactly one match. | | | There is such a turn. |  |  |
|  | Look at the previous balance from the end of the previous turn, the bet amount, and the balance at the end of the identified turn. | | | The balance for the end of the identified turn should be equal to the previous balance plus the bet amount. |  |  |
|  | Repeat steps 2-3 two more times to identify and examine different turns. | | | Same as steps 2-3. |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Data Table** | | | | | |
|  | **1** | **2** | **3** | **4** | **5** |
| N/A |  |  |  |  |  |

# Results

## Run 1



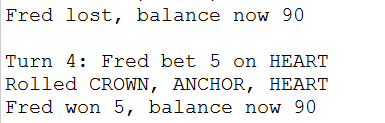
Balance before turn: 100

Balance after turn: 100

Expected: 105

Result: FAIL

## Run 2



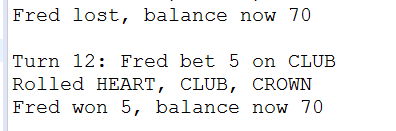
Balance before turn: 90

Balance after turn: 90

Expected: 95

Result: FAIL

## Run 3



Balance before turn: 70

Balance after turn: 70

Expected: 75

Result: FAIL