**Name : sneh patel**

**Student ID : 11576000**

**Subject : ITC 205**

**Assignment : 4**

Table of Contents

[Replication 2](#_Toc496105381)

[Simplification 3](#_Toc496105382)

[1. Test to reproduces the bug 3](#_Toc496105383)

[2. Test to reproduces the bug 3](#_Toc496105384)

[3. Test to reproduces the bug 4](#_Toc496105385)

[Tracing 4](#_Toc496105386)

[1. Bug origin and hypotheses 4](#_Toc496105387)

[2. Bug origin and hypotheses 5](#_Toc496105388)

[3. Bug origin and hypotheses 6](#_Toc496105389)

[Resolution 6](#_Toc496105390)

[Correct the reported bug 1 6](#_Toc496105391)

[Correct the reported bug 2 7](#_Toc496105392)

[Correct the reported bug 3 8](#_Toc496105393)

[Test case 9](#_Toc496105394)

# Replication

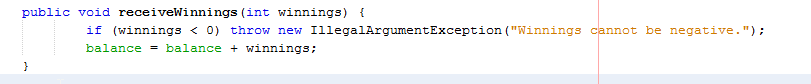
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case id** | **Unit to test** | **Assumptions** | **Test data** | **Steps to be executed** | **Expected result** | **Actual result** | **Comments** |
| 01 | If payer wins a match balance is not increase | This bug produces wrong calculation, whenever payer wins a match the win amount does not added to the current balance. | Turn 23: Fred bet 5 on CLUB  Rolled CLUB, ANCHOR, ANCHOR  Fred won 5, balance now 25 | Run the main.java file | balance now 30 | balance now 25 | In this programming there are some mathematical errors. |
| 02 | Player does not reach the last limit | This game will be stop when reached the balance limit 0, but in this game it will terminate the user when reached balance limit is 5. | Turn 50: Fred bet 5 on DIAMOND  Rolled ANCHOR, HEART, ANCHOR  Fred lost, balance now 5  50 Turns later.  End Game 99: Fred Now has balance 5 | Run the main.java file | Balance is 0 | Balance is 5 | There is must be one conditional statement, which need to be check the balance till 0. |
| 03 | The win and lose ratio should be approx. 0.42 but it does not appear. | The win and lose ration should calculate with appropriate value of 0.42 | Win count divided by total number of game played | Run the main.java file | Ratio should be 0.42 approx. | 40.2 | The 8% bias does not add to the calculation that may be the reason for wrong result. |

# Simplification

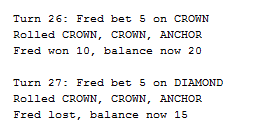
## 1. Test to reproduces the bug

If the user wins a match with balance 25, then the available balance does not update in the payer account.

Committed test code:



Output:



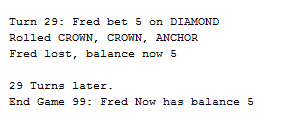
## 2. Test to reproduces the bug

If the payer wants to play the game till 0 balance than program automatically terminate whenever balance is 5.

Committed test code:



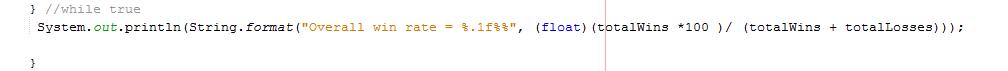
Output:



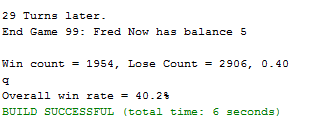
## 3. Test to reproduces the bug

The crown and anchor game have an approximate 8% bias, which is not calculate in the win ratio, therefore it produces wrong result.

Committed test code:



Output:



# Tracing

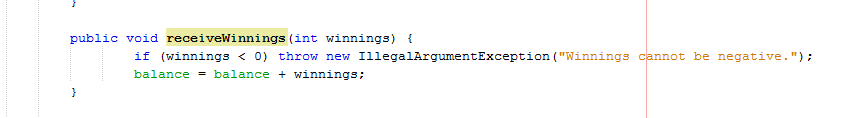
## 1. Bug origin and hypotheses

H1 (Alternative hypothesis): the method of adding the win balance is not being added to the current balance, is found.

H2 (Null hypothesis): the method of adding the win balance is not being added to the current balance, is not found.

In this tracing strategy, alternative hypothesis is successful. In this provided java programming the origin of this bug is found under player class.

Origin of the bug:



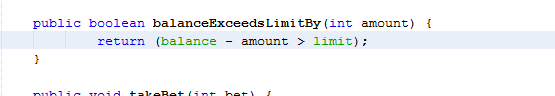
## 2. Bug origin and hypotheses

H1 (Alternative hypothesis): Limit set to 0, but game ends with player still with 5 (dollars) remaining, the bug is found in the provide source code.

H2 (Null hypothesis): Limit set to 0, but game ends with player still with 5 (dollars) remaining, the bug does not found in the provide source code.

The alternative hypothesis is achieved in this searching strategy, however this bug is found under the payer class.

Origin of the bug:



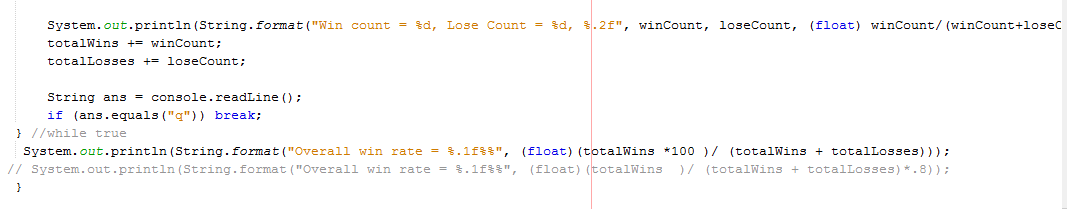
## 3. Bug origin and hypotheses

H1 (Alternative hypothesis): The win: (Win + Lose) ratio should approximately equal 0.42, but it’s not appropriately executed, this big is found.

H2 (Null hypothesis): The win: (Win + Lose) ratio should approximately equal 0.42, but it’s not appropriately executed, this big is not found.

This bug is found in main class in the given source code, therefore alternative hypothesis is true.

Origin of the bug:



# Resolution

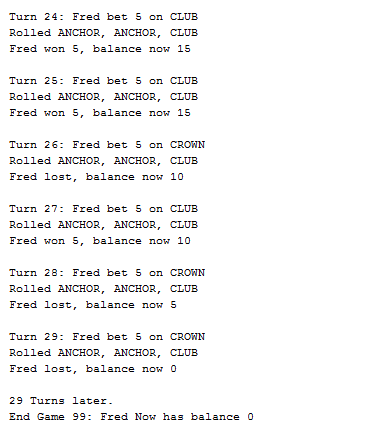
## Correct the reported bug 1

The following output of this program shows that the program is now eliminated this indented bug and produce correct result.



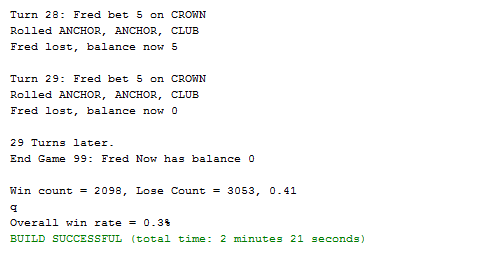
## Correct the reported bug 2

After eliminate this bug, a player can play the game till the 0 balance. However, it is shown in the following output.



## Correct the reported bug 3

After execution of entire game when a player reached the balance limit, system will generate a win ratio which should be approx. 0.42. However, in this section this result almost have a near value in comparison of previous value.



## Test case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case id** | **Test data** | **Expected result** | **Actual result** | **Output** |
| 01 | Turn 22: Fred bet 5 on ANCHOR  Rolled ANCHOR, ANCHOR, CLUB  Fred won 10, balance now 20 | balance 15 | balance now 20 |  |
| 02 | Turn 28: Fred bet 5 on CROWN  Rolled ANCHOR, ANCHOR, CLUB  Fred lost, balance now 5 | Balance is 0 | Balance is 0 |  |
| 03 | Win count divided by total number of game played | Win Ratio should be 0.42 approx. | 0.30 |  |

**Repository link** : https://github.com/sneh126846/ITC205\_project2.git