**Professional Programming Practice**

**ITC 515– Assignment 4**

**Roshan Basnet - 11620446**

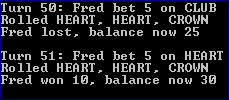
**BUG 1 – Incorrect balance increase on winning**

**Description:** The balance does not increase correctly on winnings.

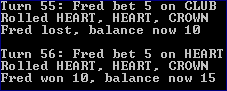
1. **Replication.**

Some of the screenshots are listed below which best describing the bug. According to them, the balance does not increase when the player wins a game.

In Turn 50, player lost and the balance is 25. In turn 51 he wins 10, the balance should be 25 + 10 = 35 but it is showing only 30.

****

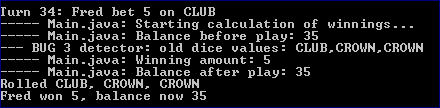
Or from output no 2. In turn 15, after winning of 10, the balance is only 15 instead of being 20.

****

**2. Simplification.**

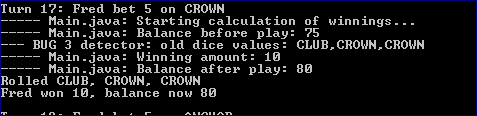
The bug can be detected by the Using unit testing method. In the Main.java class, printing out the balance before and after the game is played. In case winning is greater than 0 and the balance after game played less than the old balance plus the winning, the bug occur inside the playRound() method:

And here is the result when we run the program:



It can be witnessed that at the turn 34 balance before play is 35, Fred wins 5 but the balance after play still remains 35.

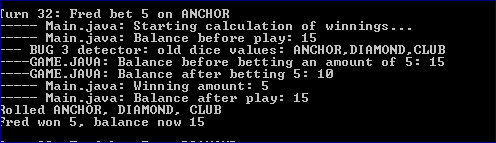
In case player wins 10 the balance rises only by 5:



Therefore, we can conclude that the bug occurs in the playRound function.

In the playRound method, we also add some debugging messages to print out the winnings:

And run the game again:



So we can see that the bet is deducted from the player’s balance after the player.takeBet(bet) is called. And then the winnings are added to the balance exclude the bet (which should be returned to the balance). This stage, bugging point has been identified.

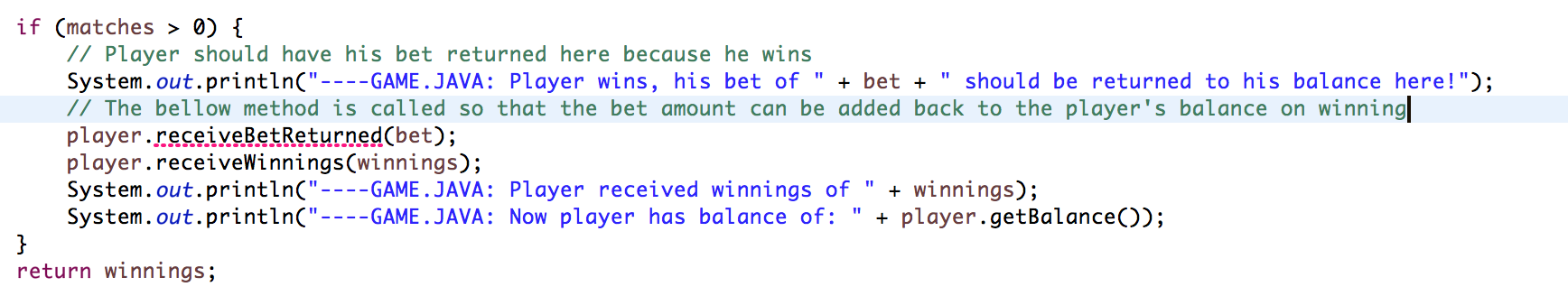
**3. Tracing.**

Following all debugging messages, we can clearly see the point of bug. It occurs only when player wins a game, in other words, in case matches variable is greater than 0. At there, player should be returned his bet together with his winnings:

Obviously, the variable balance is affected. At this point, the balance should be added more an amount of bet.

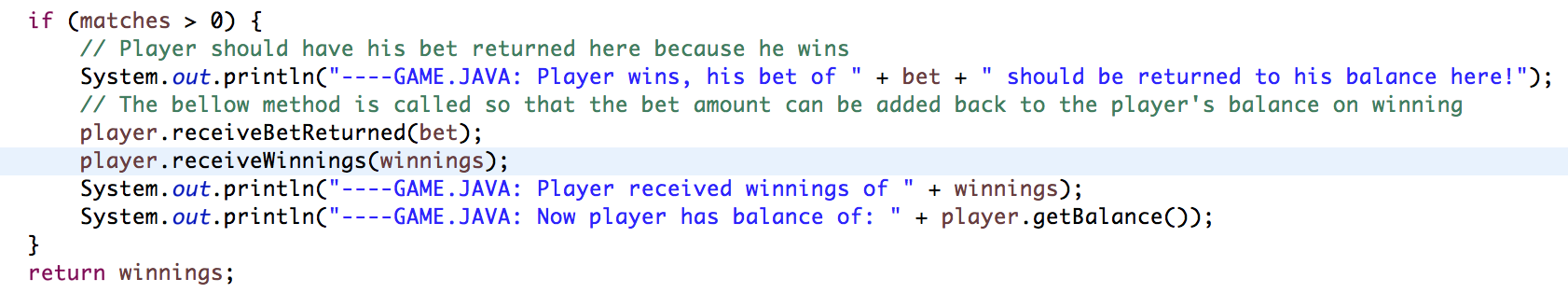
**4. Resolution.**

Once the bug is identified and the bug point detected, the solution is that we can create 1 more method in Player class, say receiveBetReturn(int bet) which can be called inside the block when matches is greater than 0.

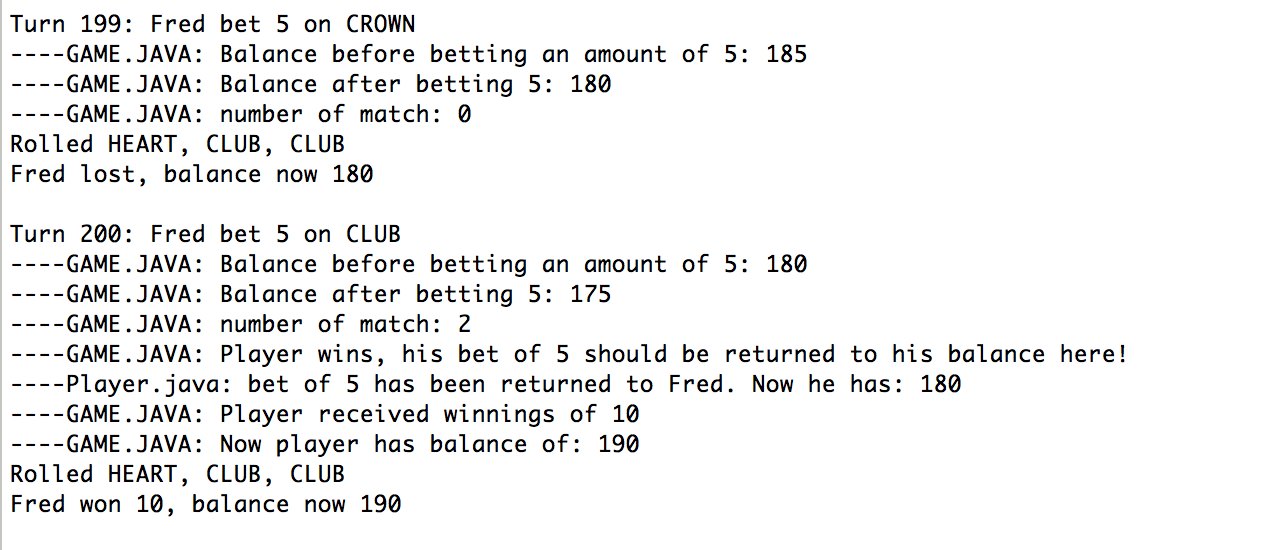


The method simply will add the pass-in bet value to the current balance:

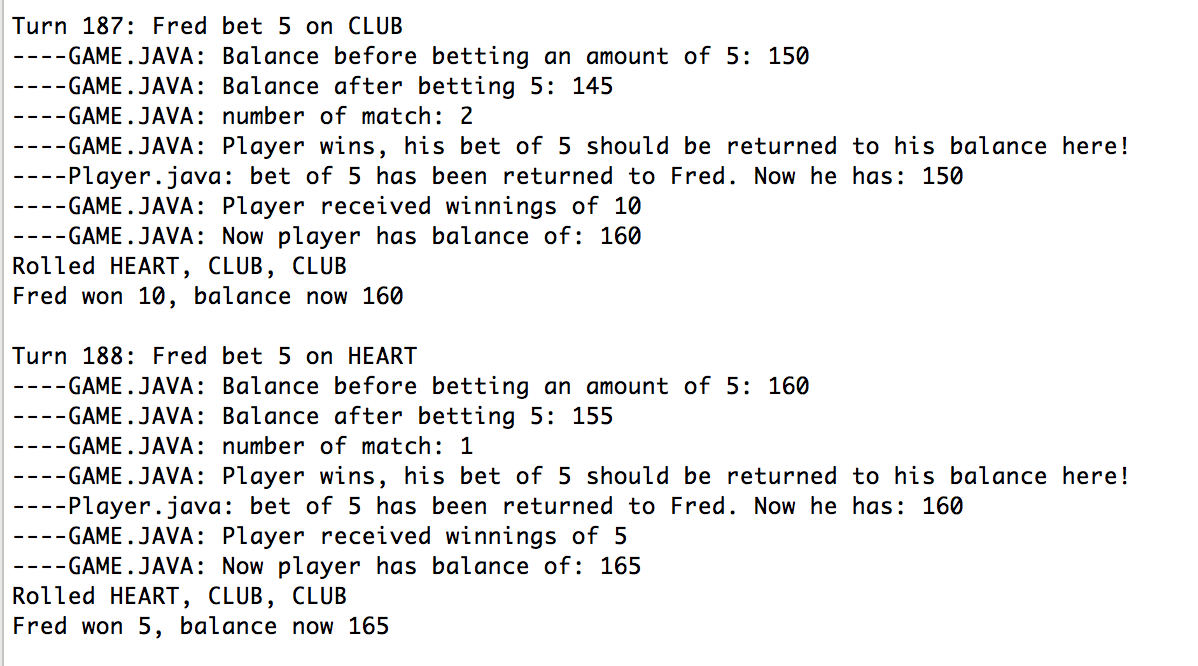
Error gone as the method implemented:



Now if we run the program, the bug fixed! In turn 200, balance was 180, after win 10 balances is 190 - correct



Try another run:



In turn 187, Fred win 10 from balance of 150, balance after win is 160 - correct