**Professional Programming Practice – Assignment 4**

**HOANG, Van Cuong**

**11613599**

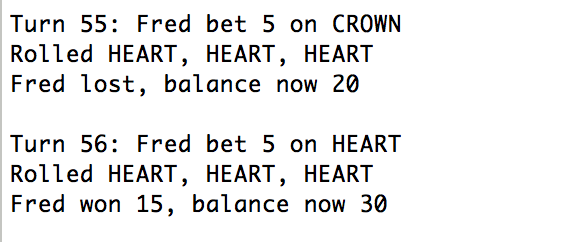
**BUG 1 – Incorrect balance increase on winning**

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

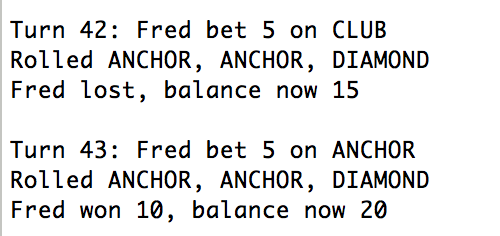
1. **Replication.**

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

Turn 55, player lost and the balance is 20. In turn 56 he wins 15, the balance should be 20 + 15 = 35 but it is actually only 30.

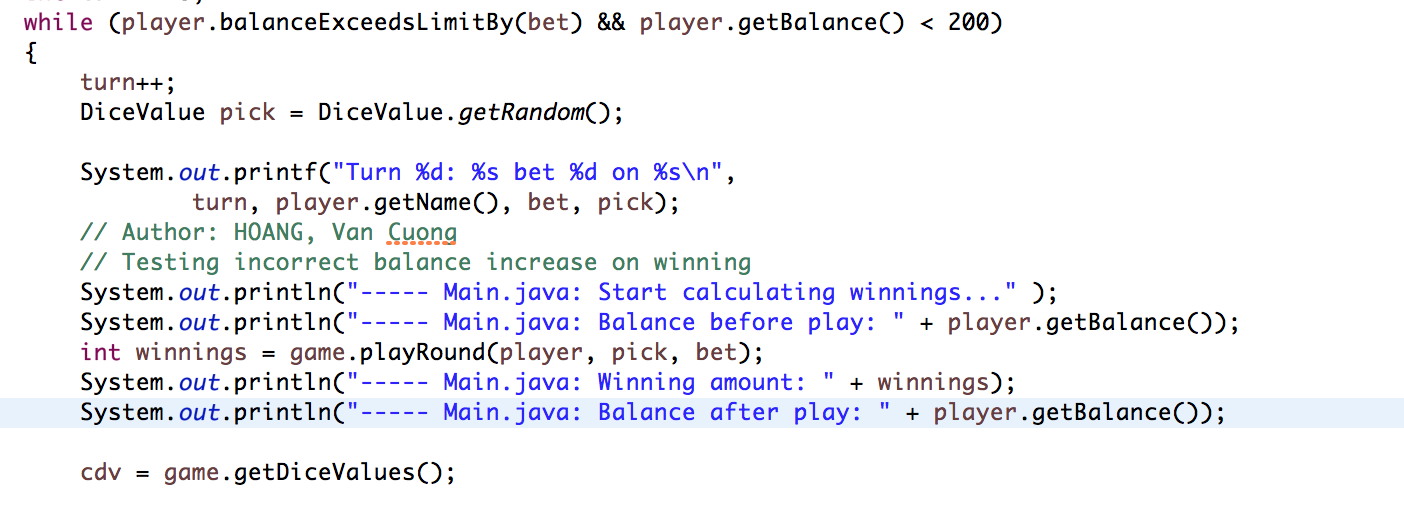
****

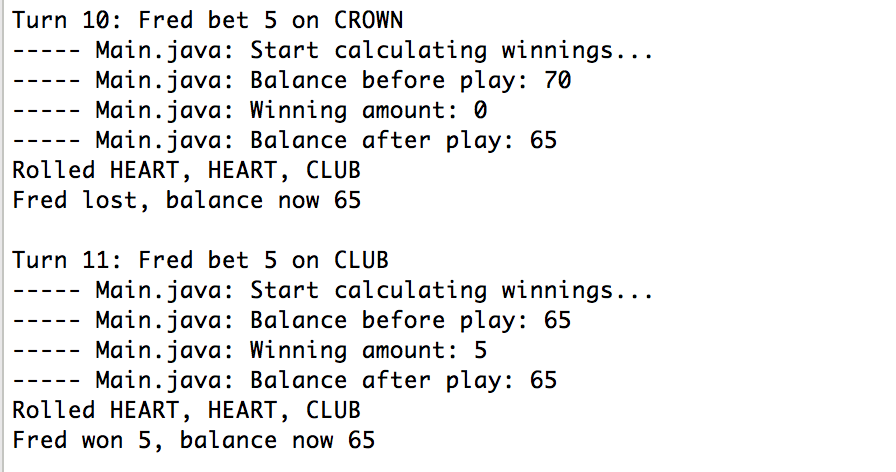
Or from 15, after winning of 10, the balance is only 20 (should be 25 to be correct)

****

**2. Simplification.**

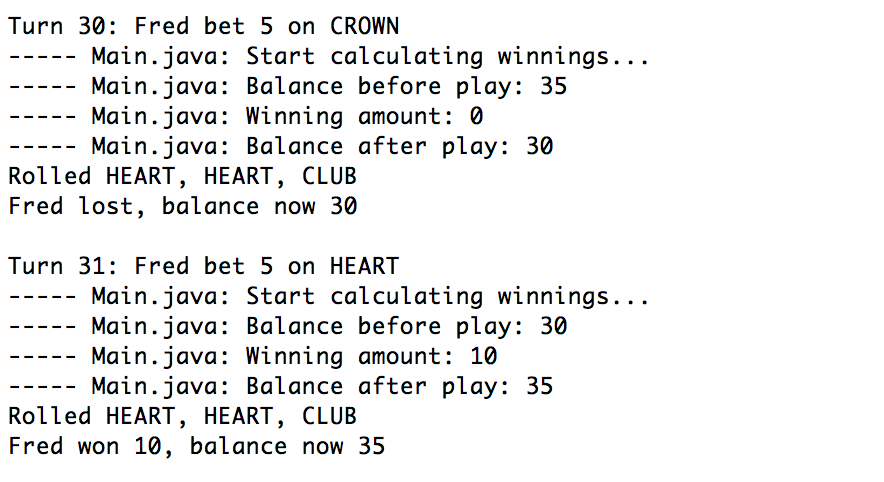
The bug can be detected by the bellow debugging messages. In the Main.java class we print out the balance before and after the game is played as well as the winnings amount. In case winning is greater than 0 and the balance after game played less than the old balance plus the winning, the bug must occurs inside the playground() method:

And here is the result when we run the program:

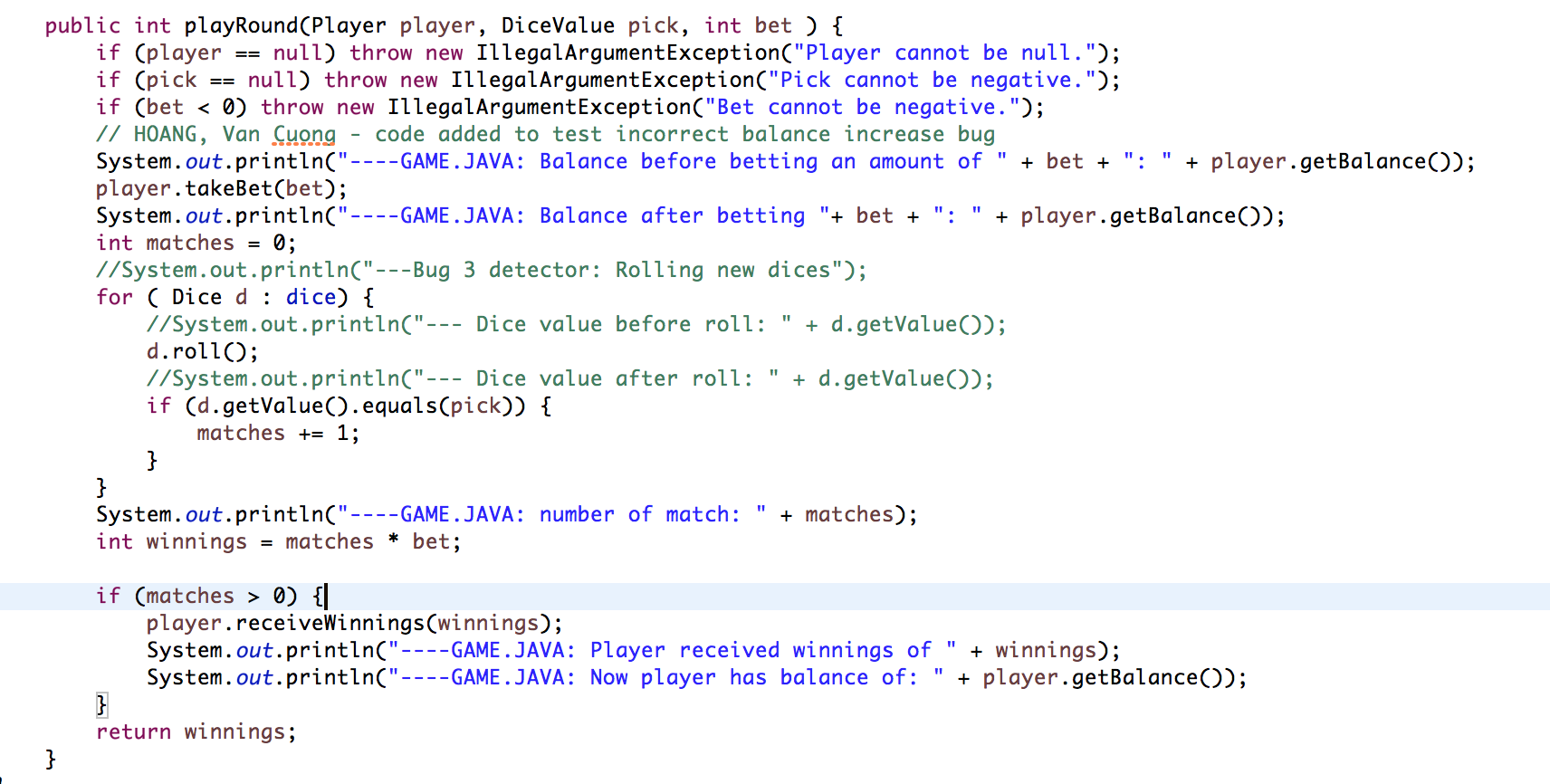


Clearly we can see, at the turn 11 balance before play is 65, Fred wins 5 but the balance after play still remains 65.

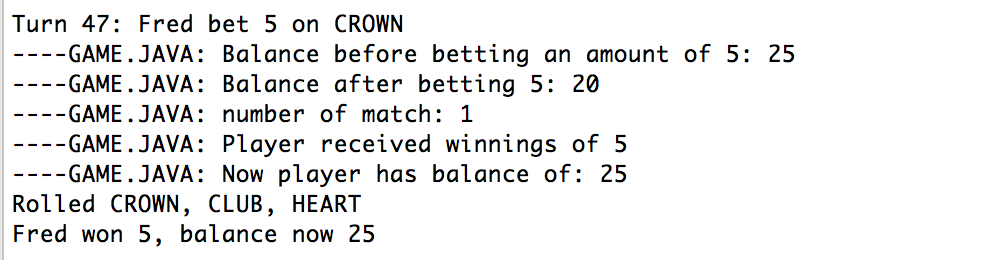
In case player wins 10 the balance increase only 5:



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

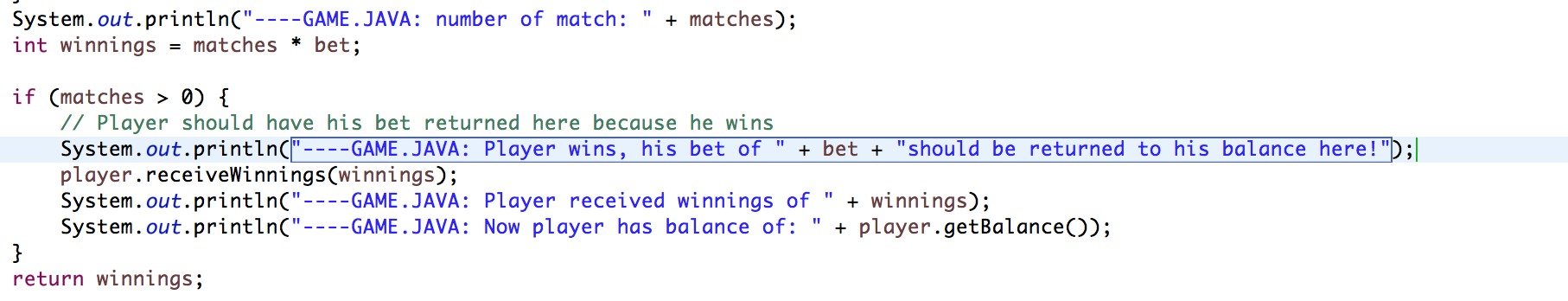
In the playGround method, we also add some debugging messages to print out the winnings:

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.

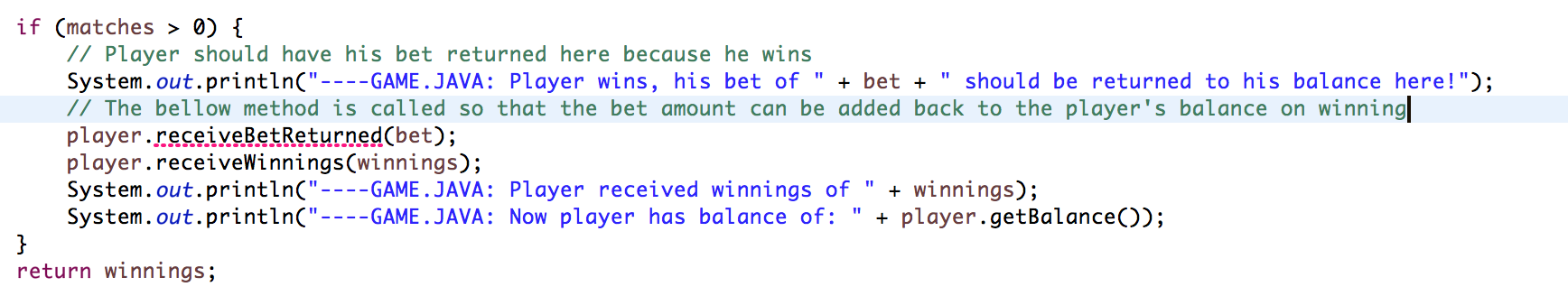
**3. Tracing.**

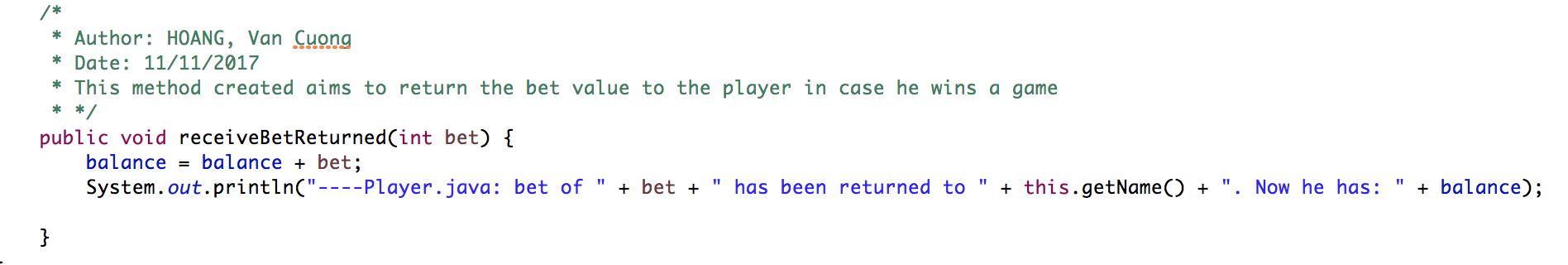
Following all debugging messages, we can clearly see the point of bug. It occurs only when player wins a game, in other words, in case matches variable is greater than 0. At there, player should be returned his bet together with his winnings:

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

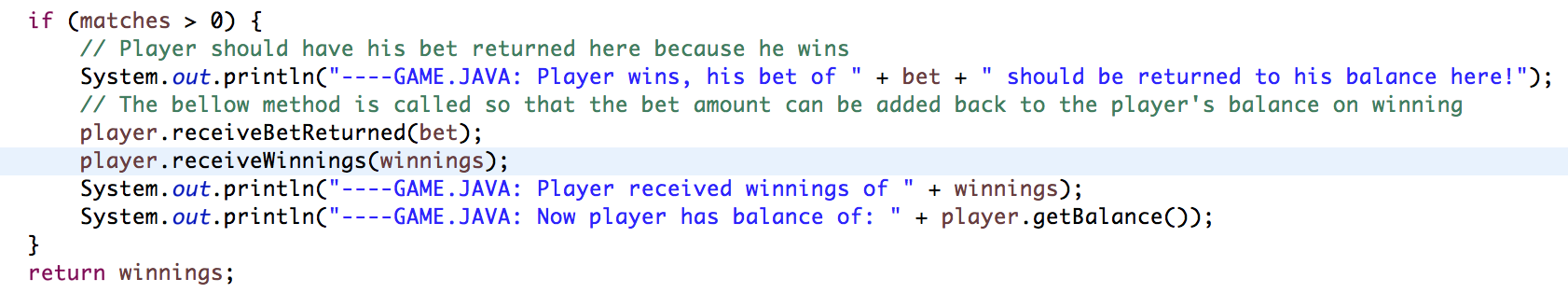
**4. Resolution.**

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.

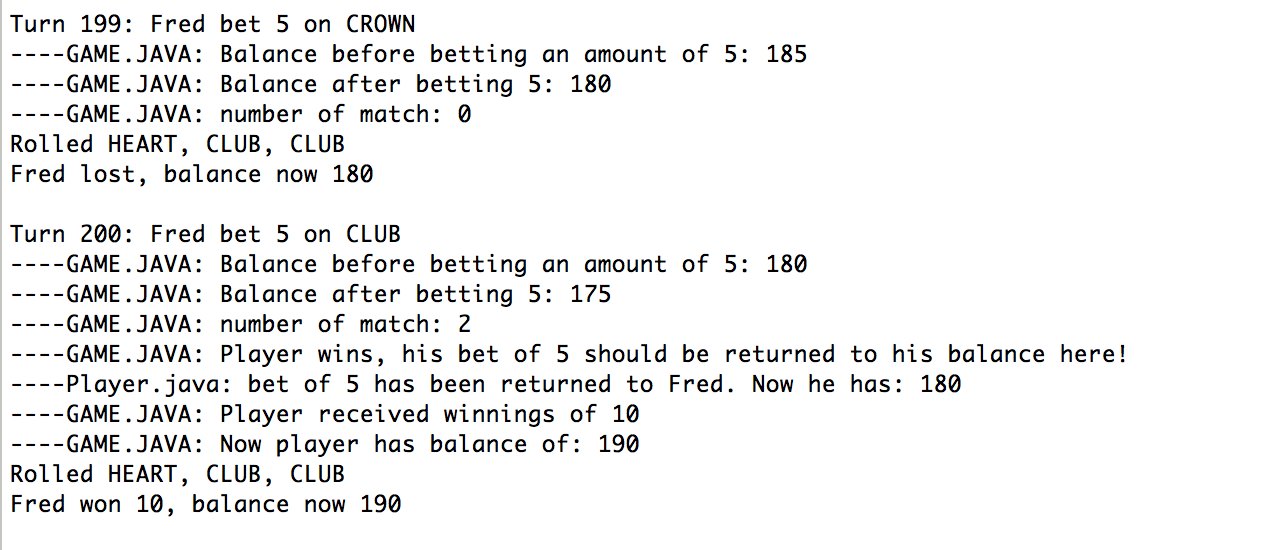


The method simply will add the pass-in bet value to the current balance:

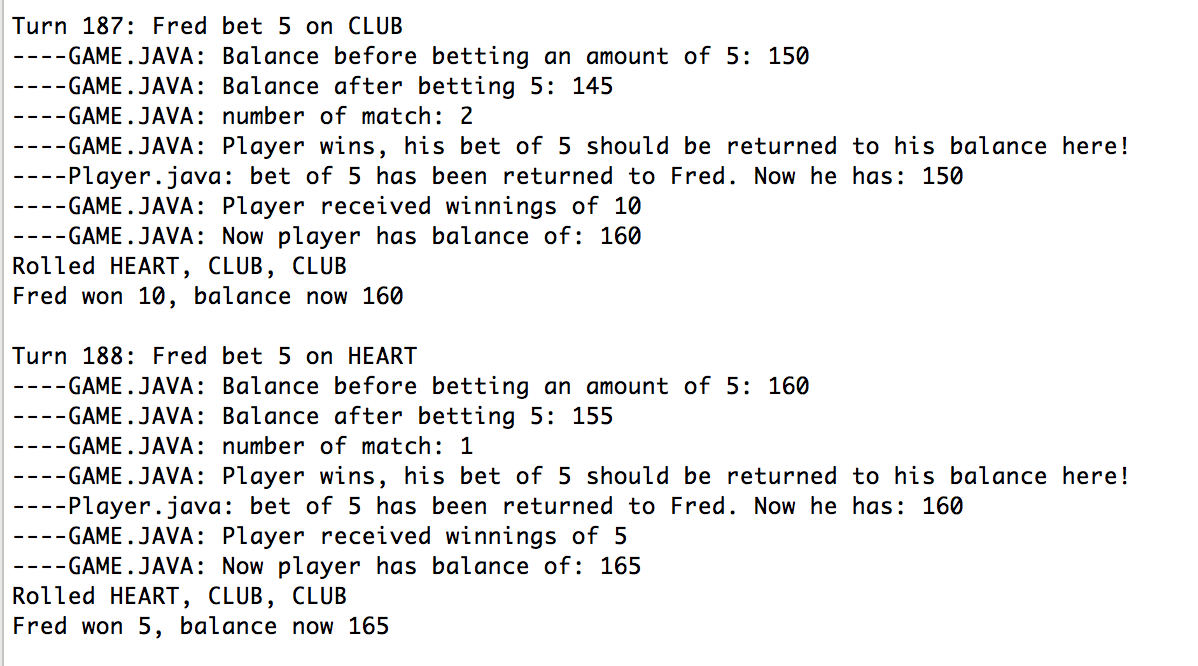
Error gone as the method implemented:



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