**Professional Programming Practice – Assignment 4**

**HOANG, Van Cuong**

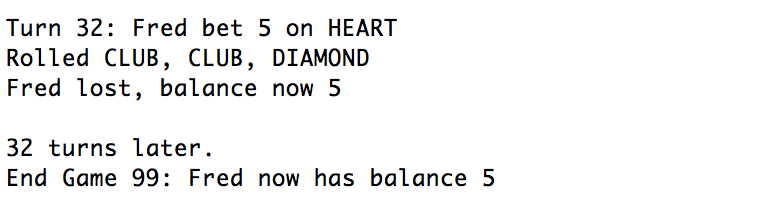
**11613599**

**BUG 2 - Game stops when balance greater than 0**

**Description:** Game ends when player’s balance remains greater than 0 and equals bet amount

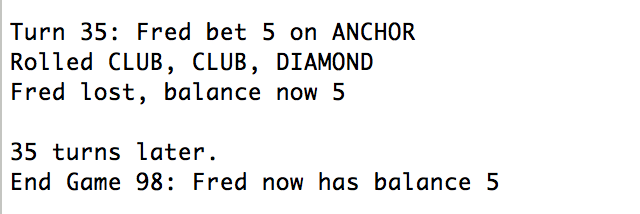
1. **Replication.**

Bellows are screenshots which best describing the bug. According to them, the game stops even when player still have his balance greater than 0:



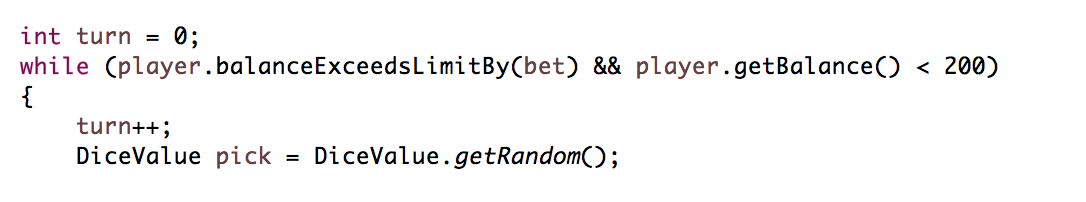
After 32 turns, game end even when Fren has balance of 5

Or in game 98, Fred forced to stop at turn 35th when he still has 5 (equals bet amount):

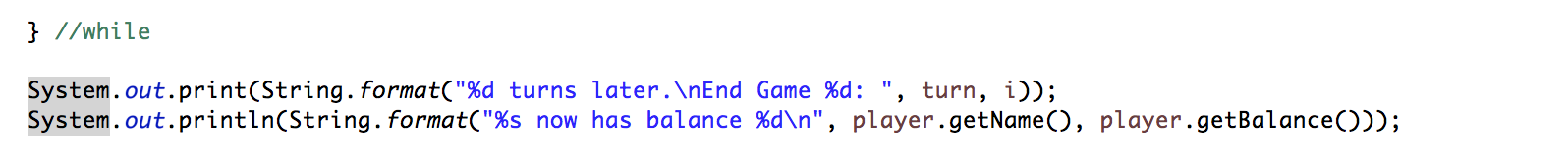


**2. Simplification.**

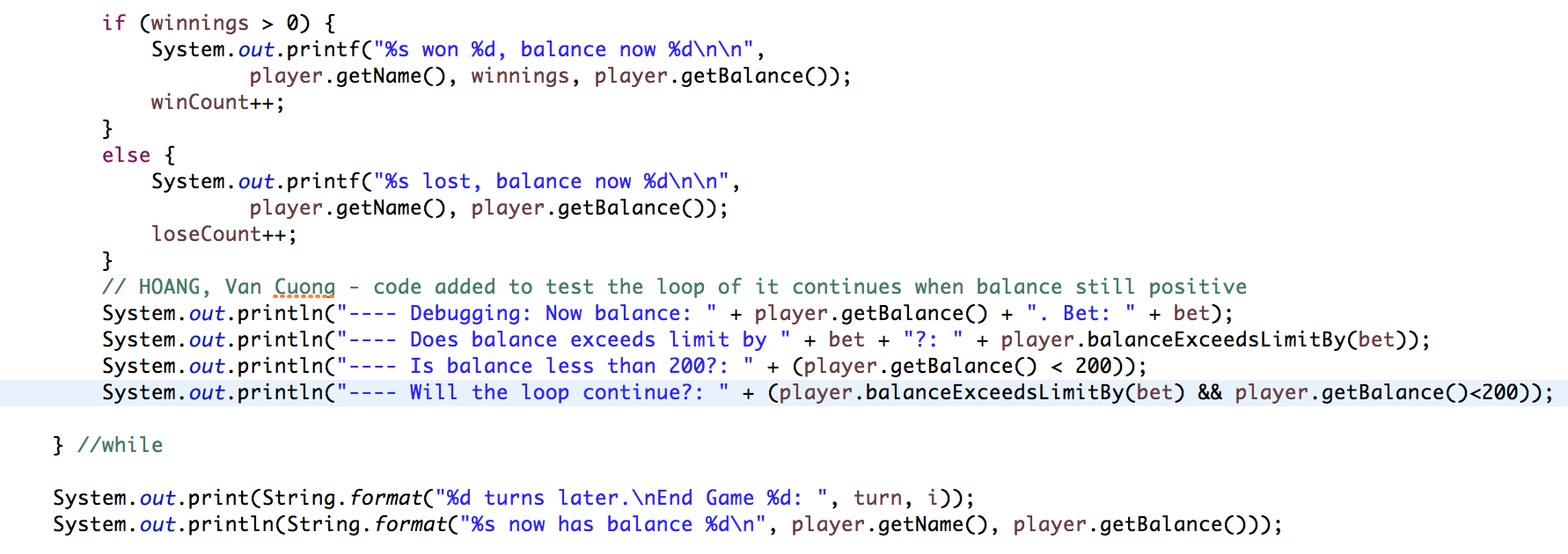
After studying the code in Main class, we can see that the main loop which replay the game over and over again is the while loop:



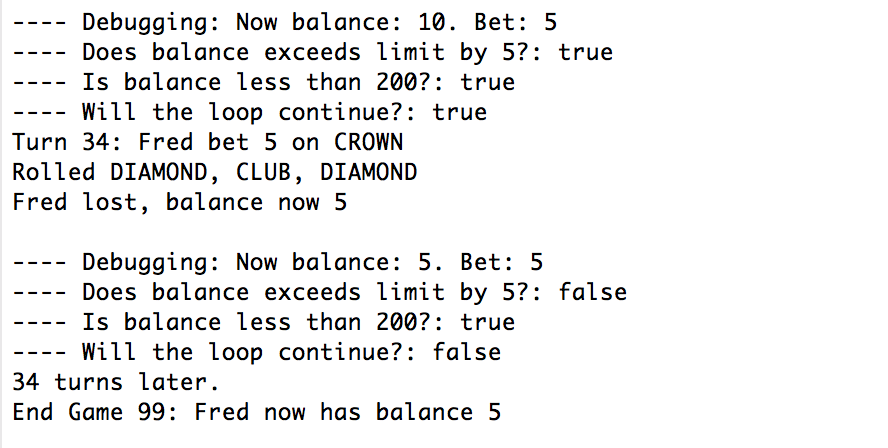
Which ends with:



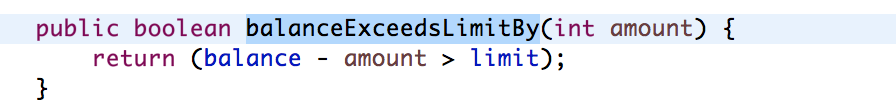
Now we add some messages at the end of each iteration to check the condition of the while:



And then run the program:

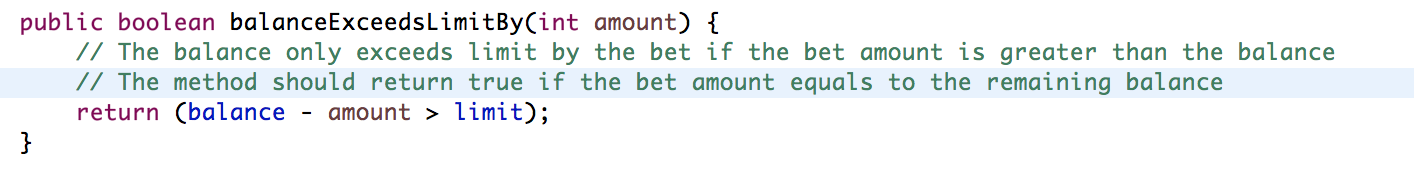


With this, we can see that the loop terminates because the balance exceeds limit by 5. (Even when the balance equals the bet). It should be something happens to the balanceExceedsLitmitBy(Int) in the Player class:



**3. Tracing.**

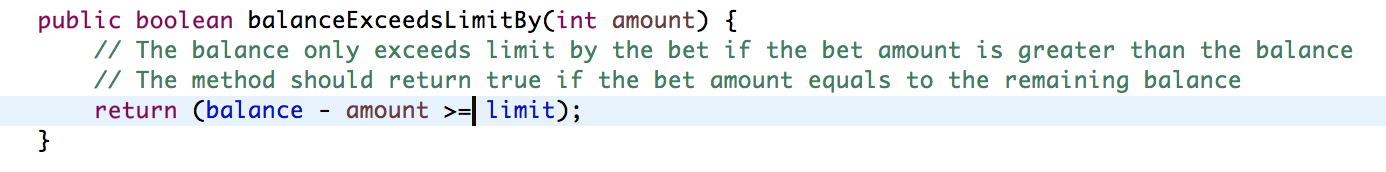
Following all debugging messages, we can clearly see the point of bug. It occurs inside the balanceExceedsLimitBy() of Player class. The return value is incorrect in the case of bug. Because:



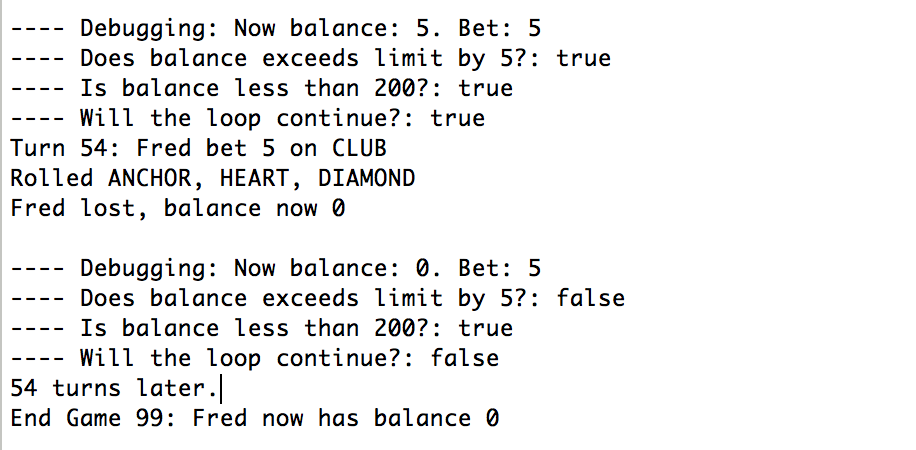
The bug produced because the return value is affected which must be true instead of false (to decide whether to continue the loop or not).

**4. Resolution.**

Simply we just check the condition to return true of the method. If the balance remains same the amount of bet, this method returns true. Or:

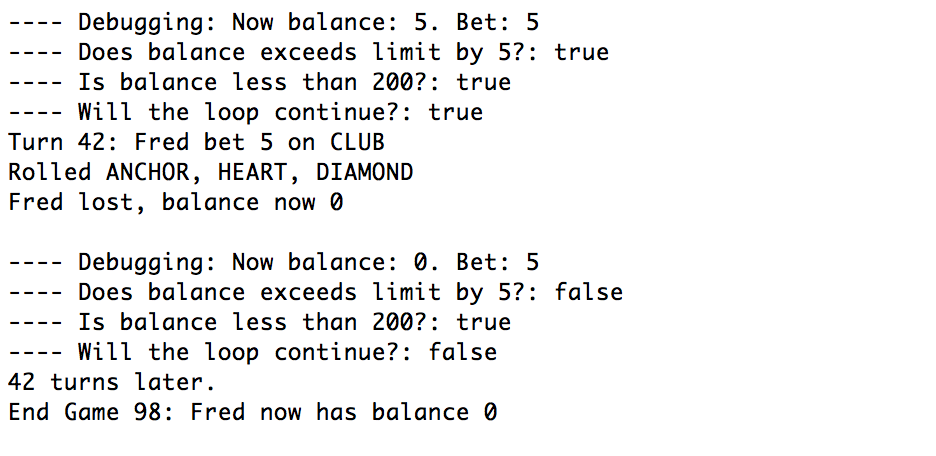


And now we can check if the bug is fixed:



Bug fixed, game ends when balance equals 0.

In an other run:



Clearly, the bug is completely fixed!