# Bug 1 Test Investigation/Fix Log

## Hypothesis

That the method Game.playround() is the source of the bug as it calculates the winnings in each turn of play.

## Action

Place breakpoints into the Game.playround() method to observe the content of variables for winnings and balance.

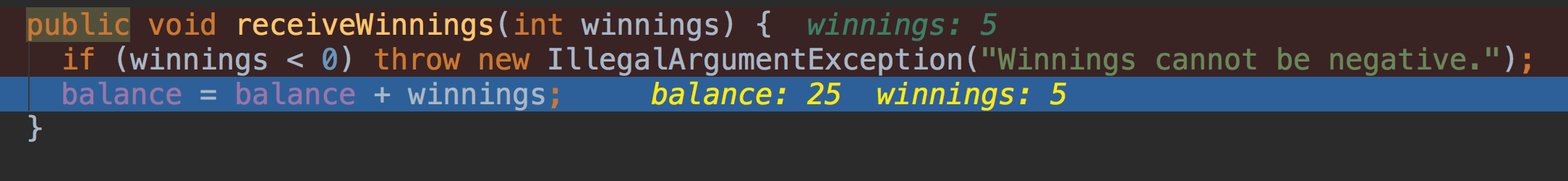
## Results

Investigation into Bug 1 shows that the balance is increased by the winnings when there is one match against a die. The balance is then reduced in the next turn when the value of the bet is subtracted from the player’s balance (which is correct). In other words, a match against only one die, although a win, is a zero-sum win and will simply keep the net balance as is.

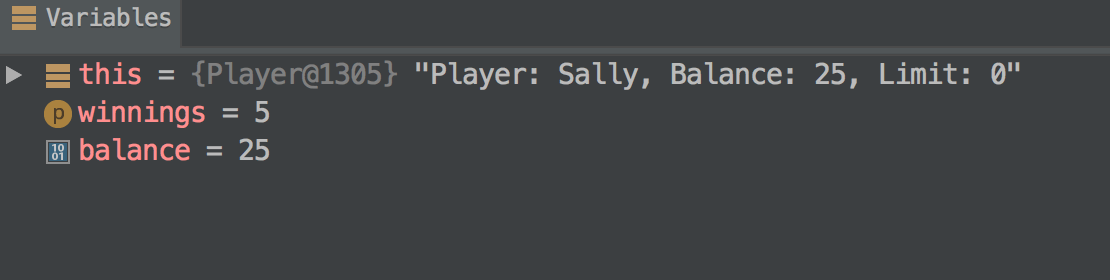
Note that this also resolves the issue of “Bug 5” as mentioned in UAT Scenario – Bug 1; Bug 5 was that when the player was observed as having rolled a matching die against two of the games rolled dice, only one win was recorded (the balance only increased by one bet). However, because the bet itself must be subtracted from the balance, a win of 2 to 1 can only gain a net increase of the value of one bet on the player’s balance (and a win of 3 to 1 can only gain a net increase of the value of two bets on the player’s balance).

This means that the program is behaving correctly. See screenshots below:

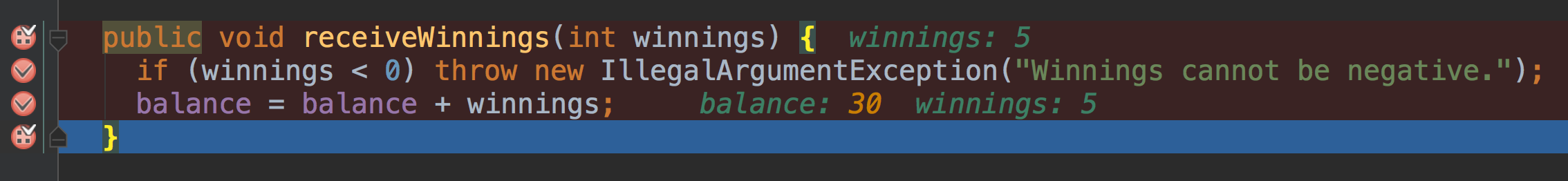
Screenshot 1 – code in receiveWinnings() method shows balance at $25, winnings at $5:



Screenshot 2 – variables reflect winnings:



Screenshot 3 – code in receiveWinnings() method shows balance at $30 after including previous winnings:



Screenshot 4 – variables reflect winnings increasing the player’s balance:

