BUG 1 Game doesn’t pay out at correct level

1.A hypothesis that will return the correct balance when the game is won.

public int getBalance() {

return balance++;

}

Result : Fail doesn’t change the output.

// A hypothesis that will increase the balance when the game is won.

public void receiveWinnings(int winnings) {

if (winnings < 0) throw new IllegalArgumentException("Winnings cannot be negative.");

balance = balance + winnings;

winnings++;

}

Result :Fail

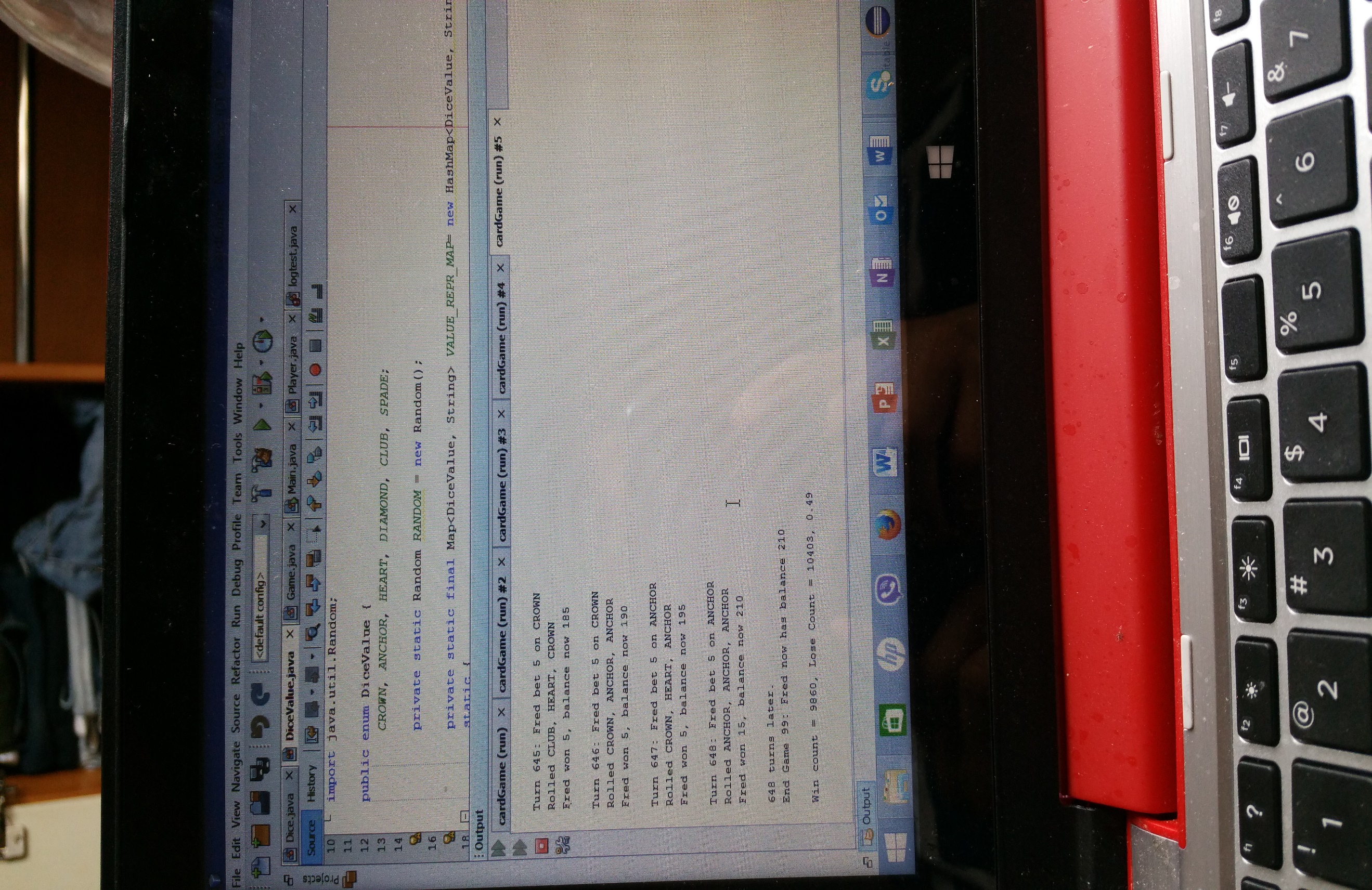
public void receiveWinnings(int winnings, int bet) {

if (winnings < 0) throw new IllegalArgumentException("Winnings cannot be negative.");

balance = balance + winnings +bet;

}

Result: Pass





Bug 2.

Player cannot reach betting limit

1. A hypothesis that player reach the betting limit

public boolean balanceExceedsLimit() {

return (balance > limit);

}

Result:failed

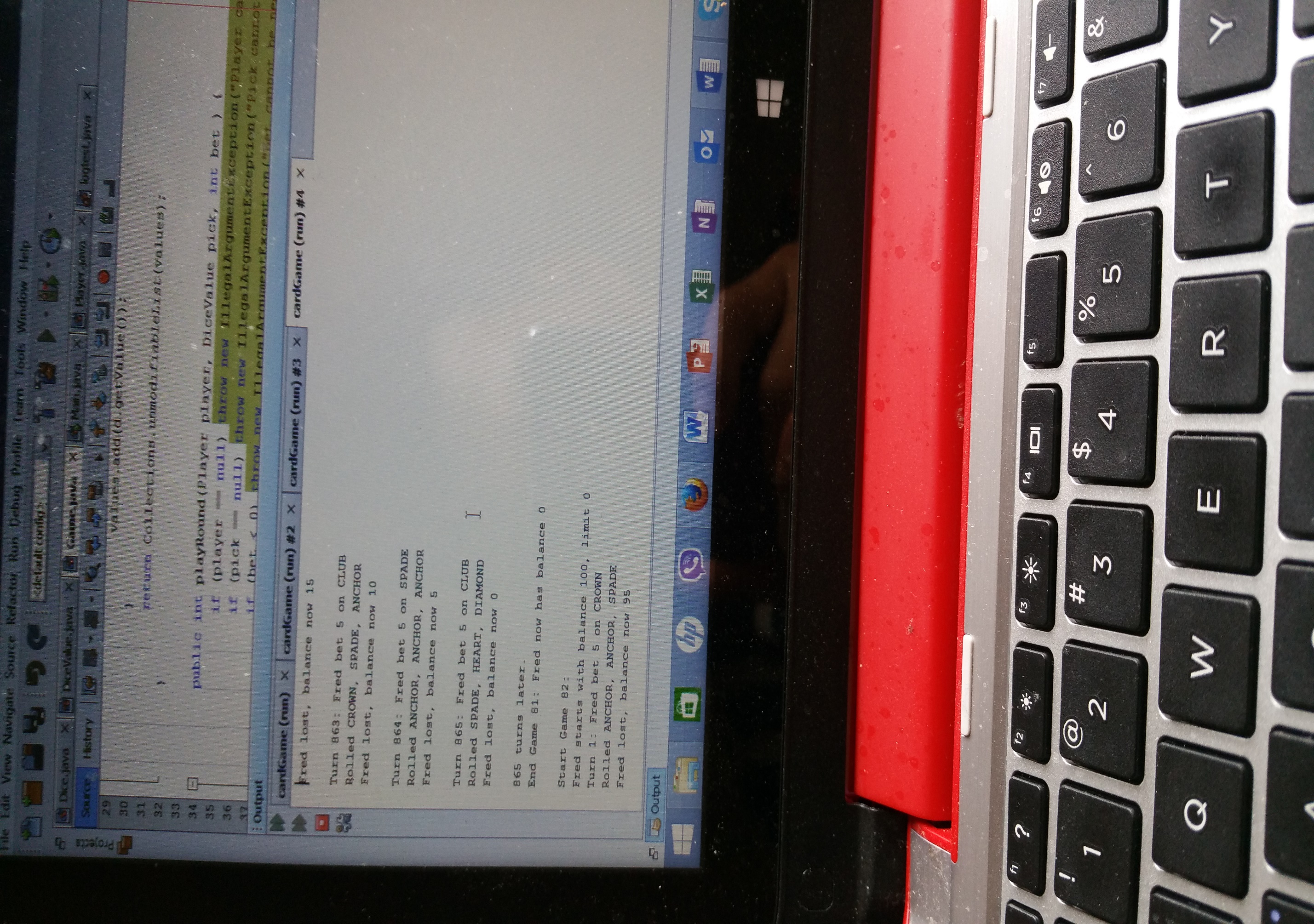
public boolean balanceExceedsLimitBy(int amount) {

//DEBUGGING ADDED = SIGN TO CHANGE THE VALUE TO 0

return (balance - amount >= limit);

}

Result: pass the value changes to “0” and has the betting limit balance.





Bug 3

Hypothesis that win/loss ratio of the game is equal to 0.42

for ( Dice d : dice) {

d.roll();

if (d.getValue().equals(pick)) {

matches += 1;

winnings += bet;

}

}

Result : pass

public static DiceValue getRandom() {

//DEBUGGING

//To pick up the spade value and to get the win/loss ratio around 42.0

//it must be increased by 1

int random = RANDOM.nextInt(DiceValue.SPADE.ordinal() +1);

return values()[random ];

}

Result: pass the win/loss ratio is equal to 0.42

