BUG 1 – Player cannot reach betting limit.

Limit set to be 0, but game ends with player still with $ 5 remaining.

1. Replication

Test Name	Set Balance Limit 0
Use Case Tested:	Player.java
Test Description:	Game will over when the player balance goes to 0.
Pre-conditions	Player must have some money for play game.
Post-conditions	Player will lose all the money at the end.
Notes:	According to rule while playing the game if player lose all the money and his balance become 0 then game will stop but in this scenario game is over when player remains balance $5.
Result (Pass/Fail/Warning/Incomplete)	

	TEST STEP	**EXPECTED TEST RESULTS**	**P**	**F**
1.	Set bet on ‘ Anchor’	Set bit of $ 5	**P**	
2.	Dice rolled	Lose the bet	**P**	
3.	Result	Player balance 0 and Game over.		**F**

Test Data Table

	1	**2**	**3**	**4**	**5**
Player Name	Fred	Fred	Fred	Fred	Fred
Balance	25(after win)	$ 20 (after lose)	$ 20(after win)	$10 (after lose)	$5 (after lose)

2. Simplification

Error in player.java class

Method	Parameters	Mutates	Returns
Constructor	String name, Int age, Int balance	Check the name, age and balance of player should not be null.	Object reference.
Set Limit	Int limit	Limit cannot be 0	Void
blanceExceedsLimit	----	If balance greater than limit then show the balance.	Boolean
blanceExceedsLimitBy	Int amount	Game ends when balance greater than limit.	Boolean
takeBet	Int bet	Cheks the conditions for bet.	int
receiveWinnings	Int winnings	If winnings less than 0 throw exception.	int
tostring			Name, balance, limit

3. Tracing

Code with buggy behaviour

Origin of code is in line no 59. Below is the screen shot which shows the code with origin of bug and output where game ends when player balance is $ 5.

ScreenShot of Code and Output which shows buggy behaviour.

Origin of code

```
44
45      public boolean balanceExceedsLimit() {
46          return (balance >limit);
47      }
48
49      public boolean balanceExceedsLimitBy(int amount)
50      {
51          //Eradicated bug 2 now player reach to betting limit, game ends when balance goes 0
52          return (balance - amount >limit);
53      }
54
55      public void takeBet(int bet) {
56          if (bet < 0) throw new IllegalArgumentException("Bet cannot be negative.");
57          if (!balanceExceedsLimitBy(bet)) throw new IllegalArgumentException("Placing bet would go below limit.");
58          balance = balance - bet;
```

Problems @ Javadoc Declaration Console Git Staging History

```
Main (7) [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (6Oct.,2016, 11:00:24 am)
Rolled CLUB, ANCHOR, CLUB
Fred won 5, balance now 25

Turn 51: Fred bet 5 on CROWN
Rolled CLUB, ANCHOR, CLUB
Fred lost, balance now 10

Turn 52: Fred bet 5 on HEART
Rolled CLUB, ANCHOR, CLUB
Fred lost, balance now 5

52 turns later.
End Game 99: Fred now has balance 5

Win count = 1985, Lose Count = 2921, 0.40
```

4. Resolution

Code and output after eradicated bug.

```
44
45      public boolean balanceExceedsLimit() {
46          return (balance >limit);
47      }
48
49      public boolean balanceExceedsLimitBy(int amount)
50      {
51          //Eradicated bug 2 now player reach to betting limit, game ends when balance goes 0
52          return (balance - amount >= limit);
53      }
54
55      public void takeBet(int bet) {
56          if (bet < 0) throw new IllegalArgumentException("Bet cannot be negative.");
57          if (!balanceExceedsLimitBy(bet)) throw new IllegalArgumentException("Placing bet would go below limit.");
58          balance = balance - bet;
```

Problems @ Javadoc Declaration Console Git Staging History

```
Main (7) [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (6Oct.,2016, 11:05:01 am)
Rolled ANCHOR, CLUB, ANCHOR
Fred lost, balance now 10

Turn 44: Fred bet 5 on CROWN
Rolled ANCHOR, CLUB, ANCHOR
Fred lost, balance now 5

Turn 45: Fred bet 5 on DIAMOND
Rolled ANCHOR, CLUB, ANCHOR
Fred lost, balance now 0

45 turns later.
End Game 99: Fred now has balance 0

Win count = 2098, Lose Count = 3061, 0.41
```

Test Name	Set Balance Limit 0
Use Case Tested:	Player.java
Test Description:	Game will over when the player balance goes to 0.
Pre-conditions	Player must have some money for play game.
Post-conditions	Player will lose all the money at the end.
Notes:	While playing the game if player lose all the money and his balance become 0 then game will stop.
Result (Pass/Fail/Warning/Incomplete)	

	TEST STEP	**EXPECTED TEST RESULTS**	**P**	**F**
4.	Set bet on ‘ Anchor’	Set bit of $ 5	**P**	
5.	Dice rolled	Lose the bet	**P**	
6.	Result	Player balance 0 and Game over.	**P**	

Test Data Table

	1	**2**	**3**	**4**	**5**
Player Name	Fred	Fred	Fred	Fred	Fred
Balance	25(after win)	$ 20 (after lose)	$ 20(after win)	$10 (after lose)	$5 (after lose)

Bug 2.Game does not pay out at correct level.

When player wins on 1 match, balance does not increase.

1. Replication

Test Name	Increment balance.
Use Case Tested:	Main.java

Test Description:	Increase player's balance when wins the match.
Pre-conditions	Player must win the match.
Post-conditions	Player balance increases.
Notes:	According to rule while playing the game if player win the match then his balance increases but in this scenario when player wins his balance does not increase.
Result (Pass/Fail/Warning/Incomplete)	

	TEST STEP	**EXPECTED TEST RESULTS**	**P**	**F**
1	Set bet on ' Anchor'	Set bit of $ 5	**P**	
2	Dice rolled	Win the bet	**P**	
3	Result	Player balance increase by $ 10.		**F**

Test Data Table

	1	**2**	**3**	**4**	**5**
Player Name	Fred	Fred	Fred	Fred	Fred
Balance	25(after win)	$ 20 (after lose)	$ 20(after win)	$20 (after Win)	$10 (after lose)

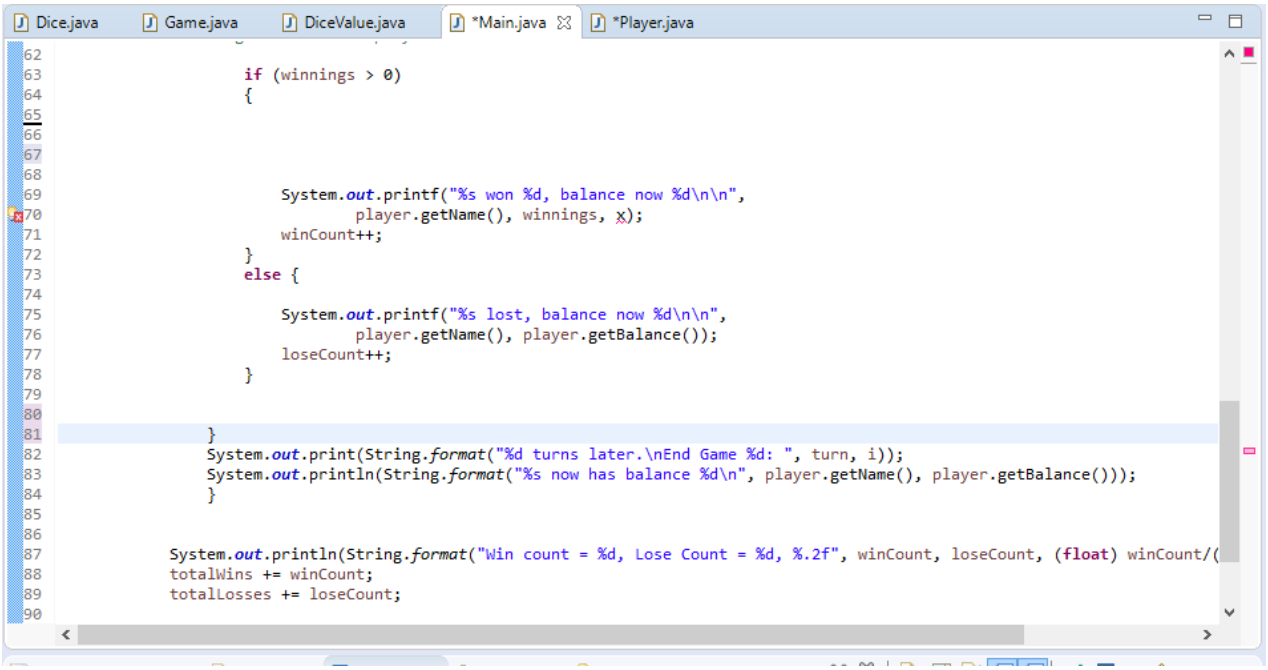
2. Simplification

Method	**Parameters**	**Mutates**	**Returns**
Constructor	Int player	Sets player's name, age and balance.	Objects reference.
Main	Int arguments	Init main program	int
Winnings	Int win	Calculate the total winnings of palyer.	int

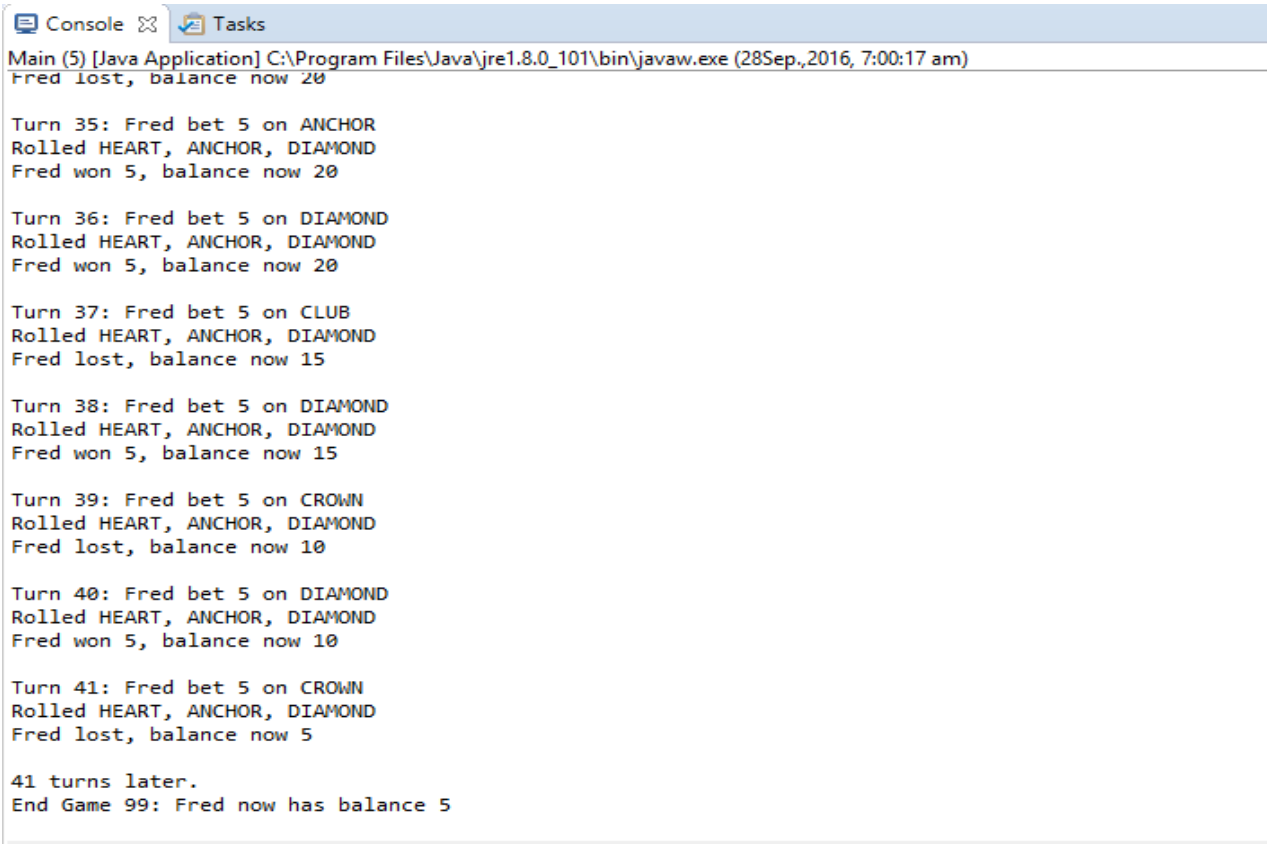
3. Tracing

Buggy Code screen shot

Origin



Buggy Output



4. Resolution

Code after eradicated bug

Dice.java | Game.java | DiceValue.java | Main.java | *Player.java

```
51                      System.out.printf("Turn %d: %s bet %d on %s\n",
52                              turn, player.getName(), bet, pick);
53
54                      int winnings = game.playRound(player, pick, bet);
55                      cdv = game.getDiceValues();
56
57                      System.out.printf("Rolled %s, %s, %s\n",
58                              cdv.get(0), cdv.get(1), cdv.get(2));
59
60
61          //increasing balance when player wins
62
63                      if (winnings > 0)
64                      {
65
66                       int x = bet + player.getBalance(); //eradicated bug, now player balance increases on win
67
68
69                          System.out.printf("%s won %d, balance now %d\n\n",
70                                  player.getName(), winnings, x);
71                          winCount++;
72                      }
73                      else {
74
75                          System.out.printf("%s lost, balance now %d\n\n",
76                                  player.getName(), player.getBalance());
77                          loseCount++;
78                      }
79
```

OutPut after eradicated bug

Console | Tasks

```
Main (5) [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (28Sep.,2016, 7:06:06 am)
Turn 31: Fred bet 5 on CLUB
Rolled CLUB, HEART, ANCHOR
Fred won 5, balance now 25

Turn 32: Fred bet 5 on DIAMOND
Rolled CLUB, HEART, ANCHOR
Fred lost, balance now 15

Turn 33: Fred bet 5 on CROWN
Rolled CLUB, HEART, ANCHOR
Fred lost, balance now 10

Turn 34: Fred bet 5 on ANCHOR
Rolled CLUB, HEART, ANCHOR
Fred won 5, balance now 15
```

Test Name	Increment balance.
Use Case Tested:	Main.java
Test Description:	Increase player's balance when wins the match.
Pre-conditions	Player must win the match.
Post-conditions	Player balance increases.
Notes:	According to rule while playing the game if player win the match then his balance increases but in this scenario when player wins his balance does not increase.
Result (Pass/Fail/Warning/Incomplete)	

	TEST STEP	**EXPECTED TEST RESULTS**	**P**	**F**
7.	Set bet on ' Anchor'	Set bit of $ 5	**P**	
8.	Dice rolled	Win the bet	**P**	
9.	Result	Player balance increase by $ 10.	**P**	

Test Data Table

	1	**2**	**3**	**4**	**5**
Player Name	Fred	Fred	Fred	Fred	Fred
Balance	20(after lose)	$ 25 (after win)	$ 15(after lose)	$10 (after lose)	$15 (after win)

Bug 3. Odds in the game do not appear to be correct.

1. Replication

Test Name	Sets Odds.
Use Case Tested:	Main.java
Test Description:	Remove the bias from game.
Pre-conditions	Player must play the match.

Post-conditions	The ratio should be approximately 0.42.
Notes:	In this game 8% bias is given to house.
Result (Pass/Fail/Warning/Incomplete)	

	TEST STEP	EXPECTED TEST RESULTS	P	F
1	Set bet on ‘ Anchor’	Set bit of $ 5	**P**	
2	Dice rolled	Win the bet	**P**	
3	Result	Player balance increase by $ 10.	**P**	
4	Game End	Ratio is less then or equals to 0.42		**F**

Test Data Table					
	1	**2**	**3**	**4**	**5**
Player Name	Fred	Fred	Fred	Fred	Fred
Balance	25(after win)	$ 20 (after lose)	$ 25 (after win)	$30 (after Win)	$25 (after lose)

2. Simplification

Method	Parameters	Mutates	Returns
Constructor	Int player	Sets player’s name, age and balance.	Objects reference.
Main	Int arguments	Init main program	int
Winnings	Int win	Calculate the total winnings of palyer.	int

3. Tracing

Buggy Code

Dice.java | Game.java | DiceValue.java | Main.java | Player.java

```
64  {
65
66   int x = bet + player.getBalance(); //eradicated bug, now player balance increases on win
67
68
69      System.out.printf("%s won %d, balance now %d\n\n",
70              player.getName(), winnings, x);
71      winCount++;
72  }
73  else {
74
75      System.out.printf("%s lost, balance now %d\n\n",
76              player.getName(), player.getBalance());
77      loseCount++;
78  }
79
80  'while checking bugs
81  'Eradicate bug 3, Getting right winCounts
82  :em.out.print(String.format("%d turns later.\nEnd Game %d: ", turn, i));
83  :em.out.println(String.format("%s now has balance %d\n", player.getName(), player.getBalance()));
84
85
86
87  ut.println(String.format("Win count = %d, Lose Count = %d, %.2f", winCount, loseCount, (float) loseCount/(winCount+loseCount
88  s += winCount;
89  ses += loseCount;
90
91  ns = console.readLine();
92  equals("q")) break;
93
94  println(String.format("Overall win rate = %.1f%%", (float)((totalLosses  / (totalWins + totalLosses)*100))));
95      }
```

Buggy Output

Problems | Javadoc | Declaration | Console | Git Staging | History

Main (7) [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (7Oct.,2016, 8:30:02 am)

```
Rolled DIAMOND, CROWN, ANCHOR
Fred won 5, balance now 15

Turn 53: Fred bet 5 on CLUB
Rolled DIAMOND, CROWN, ANCHOR
Fred lost, balance now 5

Turn 54: Fred bet 5 on ANCHOR
Rolled DIAMOND, CROWN, ANCHOR
Fred won 5, balance now 10

Turn 55: Fred bet 5 on CLUB
Rolled DIAMOND, CROWN, ANCHOR
Fred lost, balance now 0

55 turns later.
End Game 99: Fred now has balance 0

Win count = 3018, Lose Count = 2000, 0.60
```

Ratio is 0.60

4. Resolution

Code after bug fixing

Dice.java | Game.java | DiceValue.java | Main.java | Player.java

```
64  {
65
66   int x = bet + player.getBalance(); //eradicated bug, now player balance increases on win
67
68
69      System.out.printf("%s won %d, balance now %d\n\n",
70              player.getName(), winnings, x);
71      winCount++;
72  }
73  else {
74
75      System.out.printf("%s lost, balance now %d\n\n",
76              player.getName(), player.getBalance());
77      loseCount++;
78  }
79
80  'while checking bugs
81  'Eradicate bug 3, Getting right winCounts
82  :em.out.print(String.format("%d turns later.\nEnd Game %d: ", turn, i));
83  :em.out.println(String.format("%s now has balance %d\n", player.getName(), player.getBalance()));
84
85
86
87  ut.println(String.format("Win count = %d, Lose Count = %d, %.2f", winCount, loseCount, (float) loseCount/(winCount+loseCount
88  is += winCount;
89  ses += loseCount;
90
91  ns = console.readLine();
92  equals("q")) break;
93
94  println(String.format("Overall win rate = %.1f%%", (float)((totalLosses  / (totalWins + totalLosses)*100))));
95      }
```

Output after bug fixing

Problems | Javadoc | Declaration | Console | Git Staging | History

Main (7) [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (7Oct.,2016, 8:31:49 am)

```
Rolled CROWN, DIAMOND, DIAMOND
Rolled CROWN, DIAMOND, DIAMOND
Fred lost, balance now 10

Turn 92: Fred bet 5 on HEART
Rolled CROWN, DIAMOND, DIAMOND
Fred lost, balance now 5

Turn 93: Fred bet 5 on CROWN
Rolled CROWN, DIAMOND, DIAMOND
Fred won 5, balance now 10

Turn 94: Fred bet 5 on HEART
Rolled CROWN, DIAMOND, DIAMOND
Fred lost, balance now 0

94 turns later.
End Game 99: Fred now has balance 0

Win count = 494, Lose Count = 807, 0.38
```

Ratio is less than 0.42

BUG – 4

1. Replication

Test Name	Sets Balance limit
Use Case Tested:	Main.java
Test Description:	In while condition compare player balance with 200 instead of 100
Pre-conditions	Player must have balance $100
Post-conditions	Sets the limit 100
Notes:	According to game player has balance $ 100 but in main class checks the condition when playerbalnce< 200.
Result (Pass/Fail/Warning/Incomplete)	

	TEST STEP	**EXPECTED TEST RESULTS**	**P**	**F**
1	Set bet on ‘ Anchor’	Set bit of $ 5	**P**	
2	Dice rolled	Win the bet	**P**	
3	Result	Player balance increase by $ 10.	**P**	

Test Data Table

	1	**2**	**3**	**4**	**5**
Player Name	Fred	Fred	Fred	Fred	Fred
Balance	20(after lose)	$ 25 (after win)	$ 15(after lose)	$10 (after lose)	$15 (after win)

2. Simplification

Method	**Parameters**	**Mutates**	**Returns**
Constructor	Int player	Sets player’s name, age and balance.	Objects reference.

Main	Int arguments	Init main program	int
Winnings	Int win	Calculate the total winnings of palyer.	int

3. Tracing

Buggy Code

Dice.java | Game.java | DiceValue.java | *Main.java | *Player.java

```
30
31              for (int i = 0; i < 100; i++)
32              {
33                  String name = "Fred";
34                  int age = 19;
35                  int balance = 100;
36                  int limit = 0;
37                  player = new Player(name, age, balance);
38                  player.setLimit(limit);
39                  int bet = 5;
40
41                  System.out.println(String.format("Start Game %d: ", i));
42                  System.out.println(String.format("%s starts with balance %d, limit %d",
43                          player.getName(), player.getBalance(), player.getLimit()));
44
45                  int turn = 0;
46                  while (player.balanceExceedsLimitBy(bet) && player.getBalance() < 200)
47                  {
48                      turn++;
49                      DiceValue pick = DiceValue.getRandom();
50
51                      System.out.printf("Turn %d: %s bet %d on %s\n",
52                              turn, player.getName(), bet, pick);
53
54                      int winnings = game.playRound(player, pick, bet);
55                      cdv = game.getDiceValues();
56
57                      System.out.printf("Rolled %s, %s, %s\n",
58                              cdv.get(0), cdv.get(1), cdv.get(2));
```

Buggy output

Dice.java | DiceValue.java | Game.java | *Main.java | Player.java

```
33              int limit = 0;
34              player = new Player(name, age, balance);
35              player.setLimit(limit);
36              int bet = 5;
37
38              System.out.println(String.format("Start Game %d: ", i));
39              System.out.println(String.format("%s starts with balance %d, limit %d",
40                      player.getName(), player.getBalance(), player.getLimit()));
41
42              int turn = 0;
43                                              // Eradicated bug sets the limit 200
44              while (player.balanceExceedsLimitBy(bet) && player.getBalance() < 200)
45              {
46               //   turn++;
47                  DiceValue pick = DiceValue.getRandom();
48
49                  System.out.printf("Turn %d: %s bet %d on %s\n",
50                          turn, player.getName(), bet, pick);
51                  turn++;
52
53                  //DiceValue pick = DiceValue.getRandom();
54
55                  int winnings = game.playRound(player, pick, bet);
56                  cdv = game.getDiceValues();
57
```

Problems | Javadoc | Declaration | Console | Git Staging | History

4. Resolution

Code after bug fixing

Dice.java | DiceValue.java | Game.java | *Main.java | Player.java

```
33              int limit = 0;
34              player = new Player(name, age, balance);
35              player.setLimit(limit);
36              int bet = 5;
37
38              System.out.println(String.format("Start Game %d: ", i));
39              System.out.println(String.format("%s starts with balance %d, limit %d",
40                      player.getName(), player.getBalance(), player.getLimit()));
41
42              int turn = 0;
43                                              // Eradicated bug sets the limit less than and equals to 100
44              while (player.balanceExceedsLimitBy(bet) && player.getBalance() <= 100)
45              {
46               //   turn++;
47                  DiceValue pick = DiceValue.getRandom();
48
49                  System.out.printf("Turn %d: %s bet %d on %s\n",
50                          turn, player.getName(), bet, pick);
51                  turn++;
52
53                  //DiceValue pick = DiceValue.getRandom();
54
55                  int winnings = game.playRound(player, pick, bet);
56                  cdv = game.getDiceValues();
```

5. Bug

If player age is less than 18 then throw exception message.

1. Replication

Test Name	Set AGE of player
Use Case Tested:	Player.java
Test Description:	Program will show error message if age of player is less than 18 yeras.
Pre-conditions	Player must try to play game.
Post-conditions	Player age is greater than 18 years
Notes:	According to rule only player greater than age of 18 can play this game.
Result (Pass/Fail/Warning/Incomplete)	

	TEST STEP	EXPECTED TEST RESULTS	P	F
1	Set bet on ‘ Anchor’	Set bit of $ 5	**P**	
1	Get name	Accept name of player	**P**	
2	Get Balance	Set balance $100	**P**	
3	Get Age	Accept age of player.		**F**

Test Data Table					
	1	**2**	**3**	**4**	**5**
Player Name	Fred	Fred	Fred	Fred	Fred
Age	19	19	19	19	19

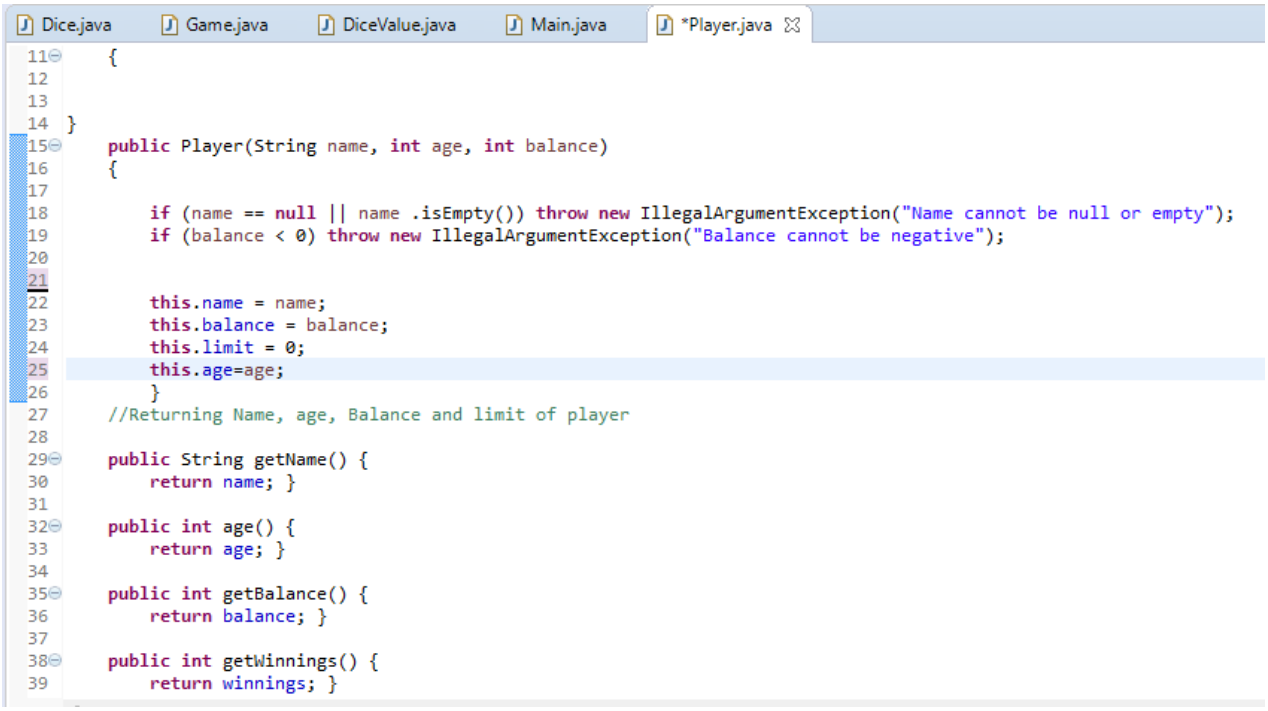
2. Simplification

Method	Parameters	Mutates	Returns
Constructor	Int player	Sets player’s name, age and balance.	Objects reference.

Main	Int arguments	Init main program	int
Winnings	Int win	Calculate the total winnings of player.	int

3. Tracing

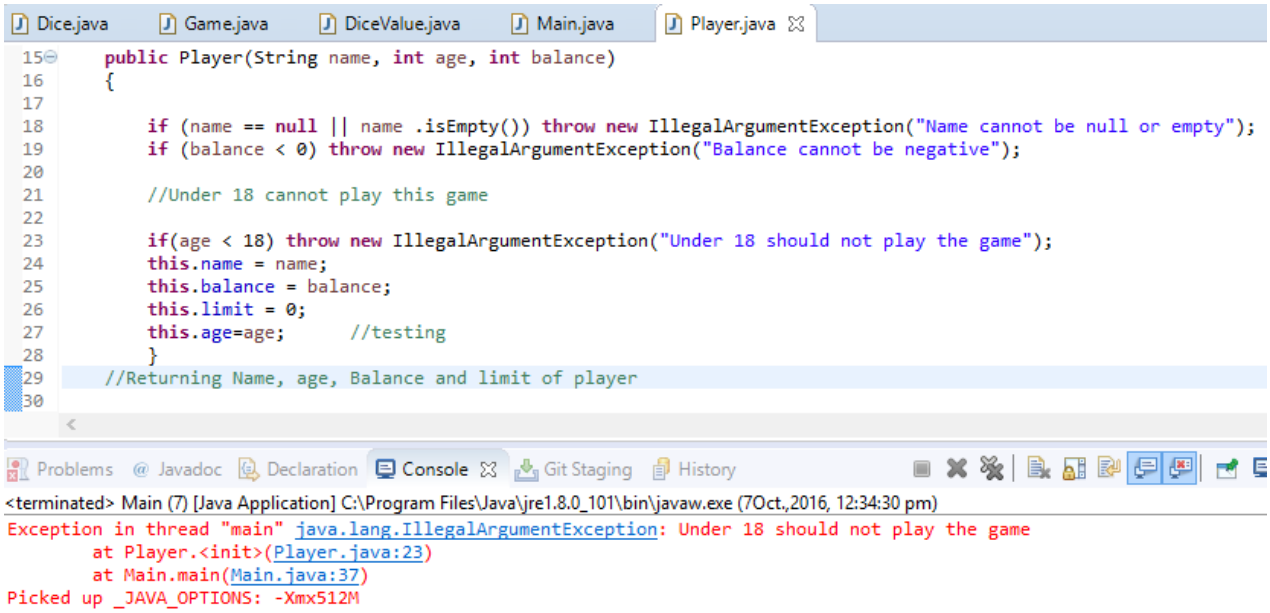
Buggy Code



```
Dice.java  Game.java  DiceValue.java  Main.java  *Player.java
11      {
12
13
14  }
15      public Player(String name, int age, int balance)
16      {
17
18          if (name == null || name .isEmpty()) throw new IllegalArgumentException("Name cannot be null or empty");
19          if (balance < 0) throw new IllegalArgumentException("Balance cannot be negative");
20
21
22          this.name = name;
23          this.balance = balance;
24          this.limit = 0;
25          this.age=age;
26          }
27      //Returning Name, age, Balance and limit of player
28
29      public String getName() {
30          return name; }
31
32      public int age() {
33          return age; }
34
35      public int getBalance() {
36          return balance; }
37
38      public int getWinnings() {
39          return winnings; }
```

4. Resolution

Code after bug fixing



```
Dice.java  Game.java  DiceValue.java  Main.java  Player.java
15      public Player(String name, int age, int balance)
16      {
17
18          if (name == null || name .isEmpty()) throw new IllegalArgumentException("Name cannot be null or empty");
19          if (balance < 0) throw new IllegalArgumentException("Balance cannot be negative");
20
21          //Under 18 cannot play this game
22
23          if(age < 18) throw new IllegalArgumentException("Under 18 should not play the game");
24          this.name = name;
25          this.balance = balance;
26          this.limit = 0;
27          this.age=age;       //testing
28          }
29      //Returning Name, age, Balance and limit of player
30
Problems  Javadoc  Declaration  Console  Git Staging  History
<terminated> Main (7) [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (7Oct.,2016, 12:34:30 pm)
Exception in thread "main" java.lang.IllegalArgumentException: Under 18 should not play the game
        at Player.<init>(Player.java:23)
        at Main.main(Main.java:37)
Picked up _JAVA_OPTIONS: -Xmx512M
```

Test Name	Set AGE of player
Use Case Tested:	Player.java
Test Description:	Program will show error message if age of player is less than 18 yeras.
Pre-conditions	Player must try to play game.
Post-conditions	Player age is greater than 18 years
Notes:	According to rule only player greater than age of 18 can play this game.
Result (Pass/Fail/Warning/Incomplete)	

	TEST STEP	**EXPECTED TEST RESULTS**	**P**	**F**
1	Set bet on ‘ Anchor’	Set bit of $ 5	**P**	
1	Get name	Accept name of player	**P**	
2	Get Balance	Set balance $100	**P**	
3	Get Age	Accept age of player.	**F**	

Test Data Table					
	1	**2**	**3**	**4**	**5**
Player Name	Fred	Fred	Fred	Fred	Fred
Age	17	17	17	17	17