

컴프2과제 [3주차 보고서]

컴퓨터공학과
201802069 김승희

1.소스코드

```
ChocolateJar.java Jar.java
1 package chocolatejargame;
2
3 public interface Jar {
4
5     public int getItem();
6     public void takeItem(int hand);
7     public String status();
8     public boolean isEmpty();
9
10 }
11
```

- 5/ 항아리 내부 전체 아이템 수를 반환하는 메소드
- 6/ hand만큼 아이템을 항아리에서 뽑는 메소드
- 7/ 내부 아이템 상태를 []로 묶어서 문자열로 반환하는 메소드
- 8/ 항아리 내부에 아이템이 있는지 없는지 판단하는 메소드

ChocolateJar.java

```
1 package chocolatejargame;
2
3 public class ChocolateJar implements Jar {
4     private int chocolate;
5     private int chilli;
6
7     public ChocolateJar() {
8         chocolate = 13;
9         chilli = 1;
10    }
11
12    public ChocolateJar(int chocolate) {
13        this.chocolate = chocolate;
14        chilli = 1;
15    }
16
17    public int getItem() {
18        return chocolate + chilli;
19    }
20
21    public int getChocolate() {
22        return chocolate;
23    }
24
25    public int getChilli() {
26        return chilli;
27    }
28
29    public void takeItem(int hand) {
30        if (chocolate >= 0)
31            chocolate = chocolate - hand;
32        else {
33            chocolate = 0;
34            chilli = 0;
35        }
36    }
37
38    public String status() {
39        String jar = "[ ";
40        if (chilli > 0) {
41            jar = jar + "■ ";
42            for (int i = 0; i < chocolate; i++) {
43                jar = jar + "□ ";
44            }
45        }
46        jar = jar + "];";
47        return jar;
48    }
49
50    public boolean isEmpty() {
51        if (getItem() == 0)
52            return true;
53        else
54            return false;
55    }
56
57 }
58
59
```

ChocolateJar클래스

3/ Jar를 상속받은 ChocolateJar클래스

4,5/private로 변수 선언

7~/생성자를 생성. 인자값으로 아무것도 받지 않을 때는 chocolate의 개수를 13개 chilli의 개수를 1개로 설정

13~/인자값으로 chocolate의 값을 받는 생성자는 chocolate의 개수를 인자값으로 하고 chilli의 개수는 1개로 설정

18~/ 항아리 안의 모든 아이템의 수를 chocolate+chilli값으로 반환

22~/ 항아리 안의 chocolate수를 반환

26~/ 항아리 안의 chilli수를 반환

30~/ hand의 값만큼 항아리에서 아이템을 뺐는다.

chocolate의 수가 0보다 크거나 같으면 chocolate의 값에서 hand만큼 빼고

그렇지 않으면 chocolate와 chilli의 값 모두 0으로 만든다.

(chilli를 뺐으면 지는 게임임으로 항아리 안의 아이템보다 많은 수를 뺐는 경우는 생각하지 않았다.)

39~/항아리 내부 아이템 상태를 문자열로 출력해주기 위해 if문을 사용하여 chilli가 0보다 클 때 즉 하나있는 chilli가 있을 때는 문자열 jar에 ■를 추가하고 for문을 이용하여 chocolate의 수만큼 돌려서 문자열 jar에 □를 추가한다. for문과 if문을 나와서 jar에]를 추가한 후에 jar를 반환한다.

51~/ 항아리 내부에 아이템이 있는지 없는지 판단하는 메소드이므로 boolean을 사용하여 getItem()이 0일때는 true를 아닐 때는 false를 반환한다.

```

ChocolateJar.java  Jar.java  CJtest.java x
1 package chocolatejargame.test;
2
3 import static org.junit.Assert.assertEquals;
10
11 class CJtest {
12     ChocolateJar jar=new ChocolateJar();
13     @Test
14     void testChocolateJar() {
15         assertEquals( 13, jar.getChocolate());
16         assertEquals( 1, jar.getChilli());
17     }
18
19     @Test
20     void testChocolateJarInt() {
21         ChocolateJar cj=new ChocolateJar(11);
22         assertEquals( 11, cj.getChocolate());
23         assertEquals( 1, cj.getChilli());
24     }
25
26     @Test
27     void testGetItem() {
28         assertEquals( 14, jar.getItem());
29     }
30
31     @Test
32     void testGetChocolate() {
33         assertEquals( 13, jar.getChocolate());
34     }
35
36     @Test
37     void testGetChilli() {
38         assertEquals( 1, jar.getChilli());
39     }
40
41     @Test
42     void testTakeItem() {
43         assertEquals( 14, jar.getItem());
44         jar.takeItem(3);
45         assertEquals( 11, jar.getItem());
46         jar.takeItem(2);
47         assertEquals( 9, jar.getItem());
48         jar.takeItem(1);
49         assertEquals( 8, jar.getItem());
50     }
51
52     @Test
53     void testStatus() {
54         assertEquals( "[ ■ □ □ □ □ □ □ □ □ □ □ ]", jar.status());
55         jar.takeItem(3);
56         assertEquals( "[ ■ □ □ □ □ □ □ □ □ □ ]", jar.status());
57     }
58
59     @Test
60     void testIsEmpty() {
61         jar.takeItem(3);
62         jar.takeItem(3);
63         jar.takeItem(3);
64         jar.takeItem(3);
65         jar.takeItem(2);
66         assertTrue( jar.isEmpty());
67     }
68
69 }
70

```

Package Explorer JUnit x

Finished after 0.277 seconds

Runs: 8/8 Errors: 0 Failures: 0

> CJtest [Runner: JUnit 5] (0.054 s)

CJtest클래스

12/ jar라는 이름의 ChocolateJar 형의 객체 생성

14~/생성자 테스트

assertEquals함수를 사용하여 jar.getChocolate()와 13이 같은지 테스트

assertEquals함수를 사용하여 jar.getChilli()와 1과 같은지 테스트

20~/초콜릿수를 입력받는 생성자 테스트

cj라는 이름의 ChocolateJar 형의 객체 생성 인자값은 11

assertEquals함수를 사용하여 cj.getChocolate()와 11과 같은지 테스트

assertEquals함수를 사용하여 cj.getChilli()와 1과 같은지 테스트

27~/항아리 안의 모든 아이템 수 반환 테스트

assertEquals함수를 사용하여 jar.getItem()와 14와 같은지 테스트

32~/항아리 안의 초콜릿 수 반환 테스트

assertEquals함수를 사용하여 jar.getChocolate()와 13이 같은지 테스트

36~/항아리 안의 칠리 수 반환 테스트

assertEquals함수를 사용하여 jar.getChilli()와 1과 같은지 테스트

41~/항아리 안의 아이템 뽑기 테스트

assertEquals함수를 사용하여 jar.getItem()와 14와 같은지 테스트

jar.takeItem(3)를 한 후 11과 jar.getItem()이 같은지 테스트

(jar.takeItem(2), jar.takeItem(1)을 반복)

53~/항아리 안의 상태를 출력 테스트

assertEquals함수를 사용하여 jar.status()와 [■ □ □ □ □ □ □ □ □ □ □ □ □]과 같은지 테스트

jar.takeItem(3) 을 한 후

assertEquals함수를 사용하여 jar.status()와 [■ □ □ □ □ □ □ □ □ □ □]과 같은지 테스트

60~/항아리가 비어있는지 테스트

jar.takeItem(3)을 4번하고 jar.takeItem(2)을 1번한 후

assertTrue함수를 사용하여 jar.isEmpty()가 true인지 테스트

```

ChocolateJar.java  Jar.java  CJtest.java  ChocolateJarMain.java ✖
1  package chocolatejargame;
2
3  import java.util.Scanner;
4
5
6
7
8
9  public class ChocolateJarMain {
10
11     public static void main(String[] args) {
12         int num = 1;
13         int playernum, computernum, maxComputernum;
14
15         Scanner sc = new Scanner(System.in);
16         Random random = new Random();
17
18         ChocolateJar cj = new ChocolateJar();
19
20         System.out.println("게임을 시작합니다.");
21         System.out.println("항아리에는 " + cj.getItem() + "개의 아이템이 있습니다.");
22         System.out.println("게임을 시작합니다.");
23         System.out.println("[ ■ □ □ □ □ □ □ □ □ □ □ □ □ ]");
24         System.out.println("1~3개의 아이템을 꺼낼 수 있으며 칠리를 꺼내면 패배합니다. \n\n시작!!\n");
25
26         while (cj.getItem() > 1) {
27             System.out.println("+-----" + num + "번째 턴! -----+");
28             System.out.println("몇 개를 뽑을까요? (1~3)");
29             playernum = sc.nextInt();
30             if (playernum > 3 || playernum < 1) {
31                 System.out.println("1~3 사이의 숫자를 입력해주세요.");
32                 System.out.println("몇 개를 뽑을까요? (1~3)");
33                 playernum = sc.nextInt();
34             }
35             cj.takeItem(playernum);
36             System.out.println("플레이어는 " + playernum + "개의 초콜릿을 꺼냈습니다.");
37             System.out.println(cj.status());
38
39             if (cj.getItem() == 1) {
40                 System.out.println("플레이어 승리!");
41                 break;
42             } else if (cj.getItem() == 0) {
43                 System.out.println("플레이어 바보! 컴퓨터 승리!");
44                 break;
45             }
46
47             maxComputernum = Math.min(3, cj.getItem() - 1);
48             computernum = random.nextInt(1000) % maxComputernum % 3 + 1;
49             System.out.println("컴퓨터는 " + computernum + "개의 초콜릿을 꺼냈습니다.");
50             cj.takeItem(computernum);
51             System.out.println(cj.status());
52             if (cj.getItem() == 1) {
53                 System.out.println("컴퓨터 승리!");
54                 break;
55             } else if (cj.getItem() == 0) {
56                 System.out.println("플레이어 승리!");
57                 num++;
58             }
59         }
60
61     }
62

```

ChocolateJarMain클래스

12/ num=1이라는 변수 선언

13/ playernum, computernum, maxComputernum이라는 int형 변수 선언

15/ Scanner sc생성

16/ Random random생성

18/ cj라는 이름의 Chocolater형의 객체 생성

26~/ while문을 사용하여 반복 (getItem())>1보다 크면 돌아감)

27/num이라는 변수로 몇번째 턴인지 알려준다

29/playernum이라는 변수는 사용자에게 입력받는 int형의 숫자이다.

30~/if문을 사용하여 사용자에게 입력받은 playernum이라는 수가 3보다 크거나 1보다 작으면 1~3사이의 숫자를 입력해주세요. 라는 문장을 출력한 후에 다시 입력받은 숫자를 playernum으로 저장한 후 if문 탈출

35/takeItem()을 사용하여 playernum만큼 항아리에서 아이템을 꺼낸다.

37/status()를 사용하여 항아리 내부 아이템 상태를 출력한다.

39~/if문을 사용하여 항아리 내부의 아이템 수가 1일 경우(chilli만 남은 경우) 플레이어승리를 출력하고 while문 탈출

42~/칠리를 뽑으면 지는 게임인데 getItem()이 0이면 플레이어가 칠리를 뽑은 것이므로 컴퓨터 승리를 출력한다.

47/컴퓨터가 뽑을 개수의 최대값 설정

Math.min(A,B) A와 B중 작은 수

48/최대값이내의 랜덤값 생성

int(1000): 0에서 999사이의 랜덤값

52~/if문을 사용하여 getItem()이 1이면 컴퓨터 승리! 출력하고 if문 탈출

57/몇 번째 턴인지 알려주기 위해 num에 1을 더한다.

Console

<terminated> ChocolateJarMain [Java Application] C:WP

게임을 시작합니다.

항아리에는 14개의 아이템이 있습니다.

게임을 시작합니다.

[■ □ □ □ □ □ □ □ □ □ □ □ □ □ □]

1~3개의 아이템을 꺼낼 수 있으며 칠리를 꺼내면 패배합니다.

시작!!

+-----1번째 턴! -----+

몇 개를 뽑을까요? (1~3)

3

플레이어는 3개의 초콜릿을 꺼냈습니다.

[■ □ □ □ □ □ □ □ □ □ □ □ □]

컴퓨터는 2개의 초콜릿을 꺼냈습니다.

[■ □ □ □ □ □ □ □ □ □ □]

+-----2번째 턴! -----+

몇 개를 뽑을까요? (1~3)

4

1~3 사이의 숫자를 입력해주세요.

몇 개를 뽑을까요? (1~3)

2

플레이어는 2개의 초콜릿을 꺼냈습니다.

[■ □ □ □ □ □ □ □]

컴퓨터는 1개의 초콜릿을 꺼냈습니다.

[■ □ □ □ □ □ □]

+-----3번째 턴! -----+

몇 개를 뽑을까요? (1~3)

1

플레이어는 1개의 초콜릿을 꺼냈습니다.

[■ □ □ □ □]

컴퓨터는 2개의 초콜릿을 꺼냈습니다.

[■ □ □]

+-----4번째 턴! -----+

몇 개를 뽑을까요? (1~3)

2

플레이어는 2개의 초콜릿을 꺼냈습니다.

[■]

플레이어 승리!