

JAVA 웹 프로그래밍

실습 과제#6

학과명: 컴퓨터공학과(ss)

교수명: 김삼근 교수님

제출자: 이종수

학번: 2017250035

제출일: 2021년 3월 29일

1.

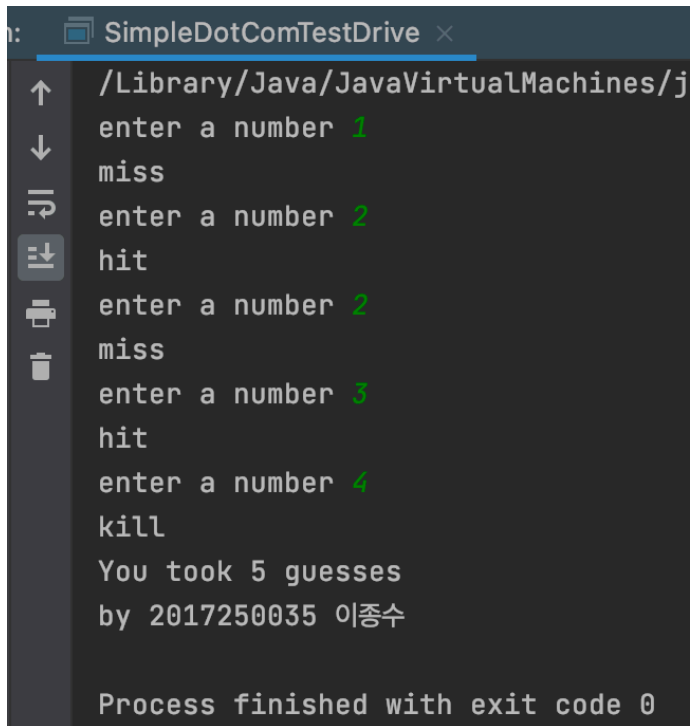
SimpleDotCom class

```
SimpleDotCom.java x SimpleDotComTestDrive.java x
1 package ch06_1;
2 import java.util.ArrayList;
3
4 public class SimpleDotCom {
5     private ArrayList<String> locationCells;
6
7     public void setLocationCells(ArrayList<String> loc) { locationCells = loc; }
8
9
10
11     public String checkYourself(String userInput) {
12         String result = "miss";
13         int index = locationCells.indexOf(userInput);
14         if (index >= 0) {
15             locationCells.remove(index);
16             if (locationCells.isEmpty()) {
17                 result = "kill";
18             } else result = "hit";
19         }
20         System.out.println(result);
21         return result;
22     }
23 }
```

SimpleDotComTestDrive class

```
SimpleDotCom.java x SimpleDotComTestDrive.java x
1 package ch06_1;
2 import java.util.ArrayList;
3
4 public class SimpleDotComTestDrive {
5     public static void main(String[] main) {
6         ArrayList<String> locations = new ArrayList<String>();
7         int numOfGuess = 0;
8         GameHelper helper = new GameHelper();
9
10        SimpleDotCom theDotCom = new SimpleDotCom();
11        int randomNum = (int)(Math.random()*5);
12
13        locations.add(Integer.toString(randomNum));
14        locations.add(Integer.toString(i: randomNum+1));
15        locations.add(Integer.toString(i: randomNum+2));
16        theDotCom.setLocationCells(locations);
17        boolean isAlive = true;
18
19        while(isAlive == true){
20            String guess = helper.getUserInput( prompt: "enter a number");
21            String result = theDotCom.checkYourself(guess);
22            numOfGuess++;
23            if (result.equals("kill")){
24                isAlive = false;
25                System.out.println("You took " + numOfGuess + " guesses");
26            }
27        }
28        System.out.println("by 2017250035 이종수");
29    }
30 }
```

실행 결과



```
SimpleDotComTestDrive x
/Library/Java/JavaVirtualMachines/j
enter a number 1
miss
enter a number 2
hit
enter a number 2
miss
enter a number 3
hit
enter a number 4
kill
You took 5 guesses
by 2017250035 이종수

Process finished with exit code 0
```

저번 과제에서 히트했다면 카운트하지 않게 코드를 작성하지 않아서 문제가 발생했습니다. 이를 해결하기 위해 셀이 이미 히트 되었는지 알아야 합니다. 이를 해결하기 위해 ArrayList를 사용하였습니다. ArrayList는 객체가 추가되고 제거됨에 따라 커지고 줄어들 수 있고, add를 사용하여 인덱스를 지정할 수 있습니다. 사용자가 입력한 숫자의 타입은 String이기 때문에 ArrayList<String>으로 하였습니다.

int index = locationCells.indexOf(userInput) -> 은 인덱스를 요청하여 사용자 추측이 ArrayList에 있는지 확인하고, 목록에 있으면 indexOf()는 그 index값을 반환하고, 없으면 -1을 반환합니다. 이후 if문을 통해 index>0이면 그 인덱스를 제거 -> ArrayList는 자동으로 하나 줄어듭니다. 이후 isEmpty()를 통해 다 제거 되었는지 확인합니다.

SimpleDotComTestDrive class에서 location을 int[]로 선언했기 때문에 이것또한 ArrayList<String>으로 변경해 줍니다. 이후 add를 통해 randomNum을 저장하는데 randomNumdms int형으로 선언되었기 때문에 add함수를 사용할 때 Integer.toString()함수를 사용해 String으로 변환해주는 코드를 작성합니다.

이제 실행을 시켜보면 위의 결과와 같이 처음 2를 hit하고 또 2를 입력해도 hit가 아닌 miss가 나오는 것을 확인할 수 있습니다.

2.

DotComBust class

```
DotCom.java x GameHelper.java x DotComBust.java x
1 package ch06_2;
2 import java.util.*;
3
4 public class DotComBust {
5     private GameHelper helper = new GameHelper();
6     private ArrayList<DotCom> dotComsList = new ArrayList<DotCom>();
7     private int numOfGuesses = 0;
8
9     private void setUpGame() {
10         DotCom one = new DotCom();
11         one.setName("Pets.com");
12         DotCom two = new DotCom();
13         two.setName("eToys.com");
14         DotCom three = new DotCom();
15         three.setName("Go2.com");
16         dotComsList.add(one);
17         dotComsList.add(two);
18         dotComsList.add(three);
19
20         System.out.println("Your goal is to sink three dot coms.");
21         System.out.println("Pets.com, eToys.com, Go2.com");
22         System.out.println("Try to sink them all in the fewest number of guesses");
23
24         for (DotCom dotComSet : dotComsList) {
25             ArrayList<String> newLocation = helper.placeDotCom( comSize: 3);
26             dotComSet.setLocationCells(newLocation);
27         }
28     }
29     private void startPlaying() {
30         while (!dotComsList.isEmpty()) {
31             String userGuess = helper.getUserInput( prompt: "Enter a guess");
32             checkUserGuess(userGuess);
33         }
34         finishGame();
35     }
36     private void checkUserGuess(String userGuess) {
37         numOfGuesses++;
38         String result = "miss";
39         for (DotCom dotComToTest : dotComsList) {
40             result = dotComToTest.checkYourself(userGuess);
41             if (result.equals("hit")) {
42                 break;
43             }
44         }
45     }
46 }
```

```

44         if (result.equals("kill")) {
45             dotComsList.remove(dotComToTest);
46             break;
47         }
48     }
49     System.out.println(result);
50 }
51
52 private void finishGame() {
53     System.out.println("All Dot Coms are dead! Your stock is now worthless");
54     if (numOfGuesses <= 18) {
55         System.out.println("It only took you " + numOfGuesses + " guesses");
56         System.out.println("You got out before your options sank.");
57         System.out.println("by 2017250035 이종수");
58     }
59     else {
60         System.out.println("Took you long enough. " + numOfGuesses + " guesses.");
61         System.out.println("Fish are dancing with your options.");
62         System.out.println("by 2017250035 이종수");
63     }
64 }
65
66 public static void main(String[] args) {
67     DotComBust game = new DotComBust();
68     game.setUpGame();
69     game.startPlaying();
70 }
71 }

```

GameHelper class

```

1  package ch06_2;
2  import java.io.*;
3  import java.util.*;
4  import javax.imageio.IOException;
5  import java.io.BufferedReader;
6  import java.io.InputStreamReader;
7
8  public class GameHelper {
9      private static final String alphabet = "abcdefg";
10     private int gridLength = 7;
11     private int gridSize = 49;
12     private int[] grid = new int[gridSize];
13     private int comCount = 0;
14
15     public String getUserInput(String prompt){
16         String inputLine = null;
17         System.out.print(prompt + " ");
18         try {
19             BufferedReader is = new BufferedReader(new InputStreamReader(System.in));
20             inputLine = is.readLine();
21             if (inputLine.length() == 0) return null;
22         } catch (IOException e){
23             System.out.println("IOException: " + e);
24         } catch (IOException e) {
25             e.printStackTrace();
26         }
27         return inputLine.toLowerCase();
28     }
29
30     public ArrayList<String> placeDotCom(int comSize) {
31         ArrayList<String> alphaCells = new ArrayList<>();
32
33         String temp = null;
34         int [] coords = new int[comSize];
35         int attempts = 0;
36         boolean success = false;
37         int location = 0;
38
39         comCount++;
40         int incr = 1;
41         if ((comCount % 2) == 1) {
42             incr = gridLength;
43         }

```

```

44     while ( !success & attempts++ < 200 ) {
45         location = (int) (Math.random() * gridSize);
46         int x = 0;
47         success = true;
48         while (success && x < comSize) {
49             if (grid[location] == 0) {
50                 coords[x++] = location;
51                 location += incr;
52                 if (location >= gridSize){
53                     success = false;
54                 }
55                 if (x>0 && (location % gridLength == 0)) {
56                     success = false;
57                 }
58             } else {
59                 success = false;
60             }
61         }
62     }
63     int x = 0;
64     int row = 0;
65     int column = 0;
66     while (x < comSize) {
67         grid[coords[x]] = 1;
68         row = (int) (coords[x] / gridLength);
69         column = coords[x] % gridLength;
70         temp = String.valueOf(alphabet.charAt(column));
71         alphaCells.add(temp.concat(Integer.toString(row)));
72         x++;
73     }
74     return alphaCells;
75 }
76 }

```

DotCom class

```

DotCom.java x GameHelper.java x DotComBust.java x
1  package ch06_2;
2  import java.util.ArrayList;
3
4  public class DotCom {
5      private ArrayList<String> locationCells;
6      private String name;
7
8      public void setLocationCells(ArrayList<String> loc) { locationCells = loc; }
12 public void setName(String n){ name = n; }
13
14 public String checkYourself(String userInput) {
15     String result = "miss";
16     int index = locationCells.indexOf(userInput);
17     if (index >= 0) {
18         locationCells.remove(index);
19         if (locationCells.isEmpty()) {
20             result = "kill";
21             System.out.println("Ouch! You sunk " + name + " : ( ");
22         } else result = "hit";
23     }
24     return result;
25 }
26 }

```

실행 결과

```

Enter a guess 02
hit
Enter a guess 03
miss
Enter a guess 01
hit
Enter a guess 02
miss
Enter a guess 00
Ouch! You sunk Pets.com : (
kill

Enter a guess 03
hit
Enter a guess 02
miss
Enter a guess 04
hit
Enter a guess 05
Ouch! You sunk Go2.com : (
kill
Enter a guess 06
hit
Enter a guess 04
Ouch! You sunk eToys.com : (
kill
All Dot Coms are dead! Your stock is now worthless
Took you long enough. 23 guesses.
Fish are dancing with your options.
by 2017250035 이종수

```

A	x						eToys.com
B							
C							
D	Pets.com			x	x		
E			x	Go2.com			
F		x		x	x		x
G		x			x	x	x
	0	1	2	3	4	5	6

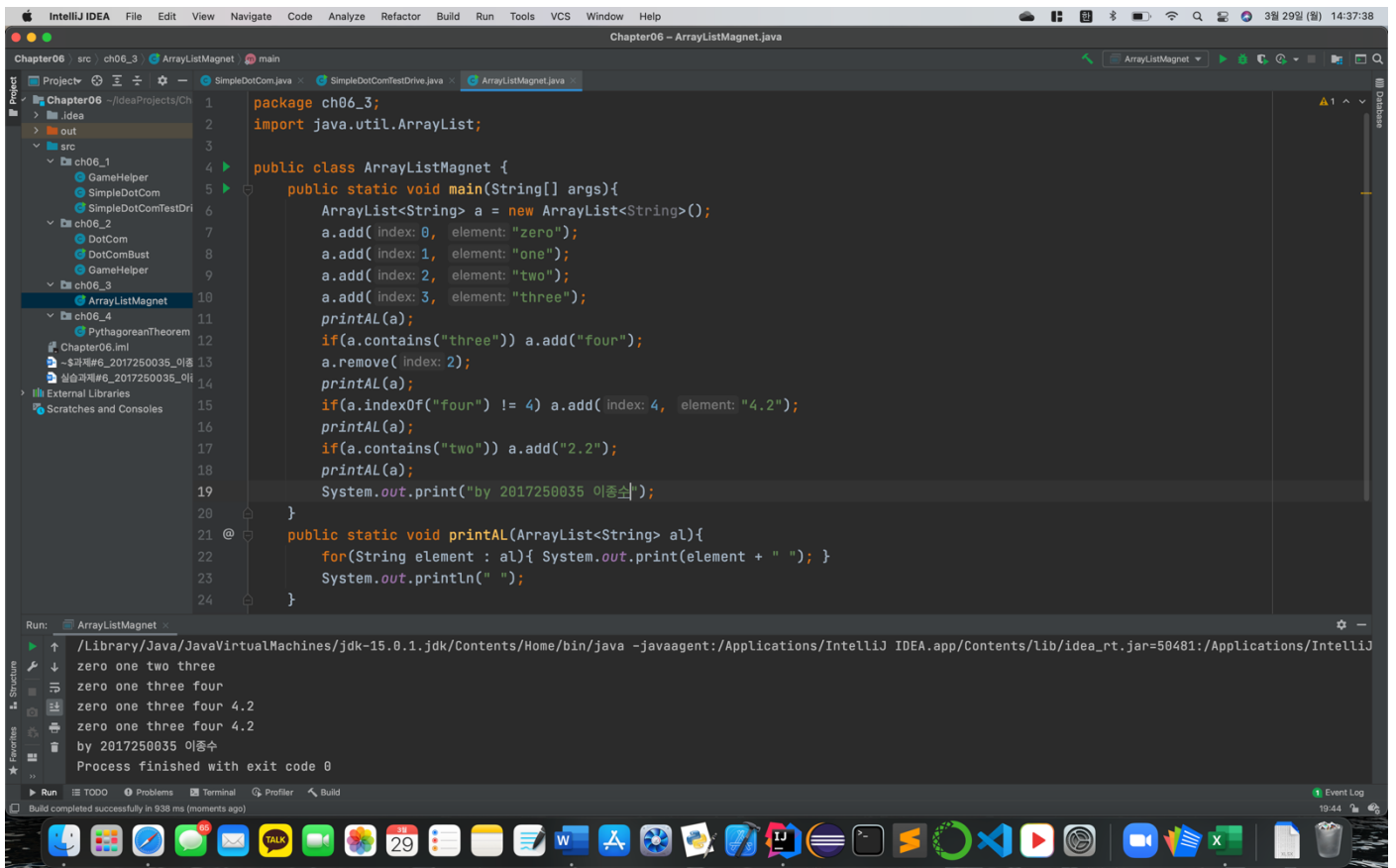
DotCom class는 ~.com이 맞을 때 마다 ~.com을 맞춘것을 출력해주는 부분을 제외하고 1번과 동일합니다.

GameHelper class에서는 사용자 명령 줄 입력을 받아들이고, DotCom 위치를 만드는 역할을 합니다.

DotComBust class에서는 세 개의 .com을 생성하고, 각각의 이름을 붙여줍니다. 이후 한 행이 아닌 그리드 상에 세 개의 닷컴을 모두 배치시킵니다. 이후 사용자가 값을 입력하면 사용자 추측을 모든 닷컴에 대해 체크를 하고 어떠한 닷컴도 남아있지 않을 때까지 게임을 계속합니다.

->main 메소드에서 DotComBust, GameHelper 객체를 생성합니다. -> DotComBust객체에서 3개의 닷컴 객체를 저장할 ArrayList 객체를 생성하고, 그들을 ArrayList에 저장합니다. -> Helper 객체에 닷컴의 위치를 요청하고, 닷컴 객체의 각각의 위치를 제공해줍니다. 각 닷컴 객체는 자신의 3개 셀 위치를 ArrayList에 저장합니다. -> Helper 객체에게 사용자 추측을 요청하고 닷컴 리스트에 대해 사용자 추측이 매치되었는지 반복해서 물오보고 DotCom 객체는 위치 ArrayList를 체크하여 결과(hit, miss 등)를 리턴합니다.

3.



String 객체 배열 a를 생성하고 add를 통해 값을 입력후 출력합니다. -> zero one two three 출력

첫 번째 if문 결과 객체 배열에 three라는 값이 저장되어있기 때문에 "four"를 추가하고 index=2인 "two"를 제거합니다. -> zero one three four 출력

두 번째 if문에서 a.indexOf("four")=3이기 때문에 4번째 자리에 4.2를 add합니다. -> zero one three four 4.4 출력

세 번째 if문에서는 객체 배열안에 "two"라는 값이 저장되어 있지 않으므로 조건문에 걸리지 않습니다. 따라서 -> zero one three four 4.4 출력합니다.

따라서 위와 같은 결과를 출력하게 됩니다.

4.

```

SimpleDotComTestDrive.java x ArrayListMagnet.java x PythagoreanTheorem.java x GameHelper.java x SimpleDotCom.java x
1 package ch06_4;
2
3 public class PythagoreanTheorem {
4     public static void main(String[] args) {
5         int result=0;
6         int count = 0;
7         for (int base = 1; base < 100; base++) {
8             for (int height = 1; height < 100; height++) {
9                 for(int hypotenuse = 1; hypotenuse < 100; hypotenuse++){
10                     if(Math.pow(hypotenuse, 2) == (Math.pow(base, 2) + Math.pow(height, 2))) {
11                         count++;
12                         System.out.println("a=" + base + " b=" + height + " c" + hypotenuse);
13                     }
14                 }
15             }
16         }
17         System.out.println("cnt = " + count);
18         System.out.println("by 2017250035 이종수");
19     }
20 }

```

실행 결과

/Library/Java/	a=35 b=12 c37	a=48 b=55 c73	
a=3 b=4 c5	a=35 b=84 c91	a=48 b=64 c80	
a=4 b=3 c5	a=36 b=15 c39	a=51 b=68 c85	
a=5 b=12 c13	a=36 b=27 c45	a=52 b=39 c65	
a=6 b=8 c10	a=36 b=48 c60	a=54 b=72 c90	
a=7 b=24 c25	a=36 b=77 c85	a=55 b=48 c73	
a=8 b=6 c10	a=39 b=52 c65	a=56 b=33 c65	
a=8 b=15 c17	a=39 b=80 c89	a=56 b=42 c70	
a=9 b=12 c15	a=40 b=9 c41	a=57 b=76 c95	
a=9 b=40 c41	a=40 b=30 c50	a=60 b=11 c61	
a=10 b=24 c26	a=40 b=42 c58	a=60 b=25 c65	
a=11 b=60 c61	a=40 b=75 c85	a=60 b=32 c68	
a=12 b=5 c13	a=42 b=40 c58	a=60 b=45 c75	
a=12 b=9 c15	a=42 b=56 c70	a=60 b=63 c87	
a=12 b=16 c20	a=44 b=33 c55	a=63 b=16 c65	
a=12 b=35 c37	a=45 b=24 c51	a=63 b=60 c87	
a=13 b=84 c85	a=45 b=28 c53	a=64 b=48 c80	
a=14 b=48 c50	a=45 b=60 c75	a=65 b=72 c97	
	a=48 b=14 c50	a=68 b=51 c85	
	a=48 b=20 c52	a=70 b=24 c74	
	a=48 b=36 c60	a=72 b=21 c75	
		a=72 b=30 c78	

PythagoreanTheorem x

a=72 b=30 c78

a=72 b=54 c90

a=72 b=65 c97

a=75 b=40 c85

a=76 b=57 c95

a=77 b=36 c85

a=80 b=18 c82

a=80 b=39 c89

a=84 b=13 c85

a=84 b=35 c91

cnt = 100

by 2017250035 이종수

Process finished with exit code 0

삼중 for문을 사용하였습니다. 첫 번째 for문에서는 밑변(base)가 1부터 100까지 커지고, 두 번째 for문에서는 높이(height)가 1부터 100까지 커지고, 세 번째 for문에서는 빗변(hypotenuse)가 1부터 100까지 1씩 커집니다. 이때 각각의 길이를 제공하기 위해서 Math.pow()함수를 사용하였고 if문을 사용해 피타고라스 정리가 성립하면 그 수를 출력하고 갯수(count)를 1씩 증가시켰습니다. 이후 count를 출력했습니다.

따라서 위와 같은 결과가 나오게 됩니다.