

(디지털컨버전스) 스마트 콘텐츠와 웹 융합 응용SW개발자 양성과정

2021년 5월 17일
[7일차 복습]

- 수강생 : 김 민 규
- 강의장 : 강 남 C
- 수강 기간 : 2021. 05. 07 ~ 2021. 12. 08
- 수강 시간 : 15:30 ~ 22:00
- 이상훈 강사님 | 이은정 취업담임



▶ 내용 :

```
1  class PersonTest {
2      String name;
3      int age;
4
5      PersonTest() {
6          System.out.println("안녕 난 생성자야!");
7      }
8      public String getName() {
9          return name;
10     }
11     public void setName(String name) {
12         this.name = name;
13     }
14     public int getAge() {
15         return age;
16     }
17     public void setAge(int age) {
18         this.age = age;
19     }
20 }
21 ▶ public class a_ {
22 ▶     public static void main(String[] args) {
23         PersonTest pt = new PersonTest();
24
25         pt.setName("가오왕");
26         pt.setAge(10);
27
28         System.out.printf("나는 %s이고 %d살이다! 덤벼라!\n", pt.getName(), pt.getAge());
29     }
```

주요 내용

--

생성자의 특징

1. 리턴타입이 없다.
2. 클래스 이름과 매서드 이름이 같다.
3. New를 할 때 호출된다.

--

▶ 내용 : 생성자

```
1  class Constest {
2      int age;
3      String name;
4
5
6  }
7      Constest() {
8          System.out.println("안녕 나는 Constest() 이라고 해!");
9      }
10     Constest(int a) {
11         System.out.println("안녕 나는 Constest(int a) 이라고 해!");
12         age = a;
13     }
14     Constest(float f) {
15         System.out.println("안녕 나는 Constest(float f) 라고 해!");
16     }
17     Constest(int a, String n) {
18         System.out.println("안녕 나는 Constest(int a, String n)이라고 해!");
19         name = n;
20         age = a;
21     }
22
23     public int getAge() {
24         return age;
25     }
26     public String getName() {
27         return name;
28     }
29 }
```

데이터 저장영역

기능 설정 영역

```
29 ▶ public class b_생성자2 {
30 ▶     public static void main(String[] args) {
31         Constest ct1 = new Constest();
32         Constest ct2 = new Constest( a: 10);
33         Constest ct3 = new Constest( a: 20, n: "hi");
34         Constest ct4 = new Constest( a: 40);
35         Constest ct5 = new Constest( f: 3.3f);
36
37         System.out.printf("ct1 name = %s, age = %d\n", ct1.getName(), ct1.getAge());
38         System.out.printf("ct1 name = %s, age = %d\n", ct2.getName(), ct2.getAge());
39         System.out.printf("ct1 name = %s, age = %d\n", ct3.getName(), ct3.getAge());
40         System.out.printf("ct1 name = %s, age = %d\n", ct4.getName(), ct4.getAge());
41     }
42 }
```

Constest라는 class 타입이 아닌
데이터타입을 통해 판단을 함.
각각 독립되어있음.

주요 내용

--

데이터 저장영역

공간만 만들어둔 것.

값은 입력되지않아 int는 0 String은 null로
입력되어있음.

--

기능 설정 영역

생성자자는 여러 형식으로 구성할 수 있음.

이름이 같지만, 입력이 다른 케이스로
매서드를 사용하는 방식에 대하여
함수 오버로딩이라고 함.

함수 오버로딩 : Constest라는 명칭은
같아도 입력인자(int : a, String n 등)가
다르면 다르게 인식되어사용할 수 있음.

입력의 개수를 갖고 판단하지 않으며,
사용되는 입력에 데이터타입을 보고
판단한다는 점을 주의해야 한다.

--

생성자

객체를 처음 생성할 때 초기값을 설정해주는
역할을 하므로, Setter없이도 결과가 나옴
(초기생성에 한해 Setter의 역할을
대신해줄 수 있음)

▶ 내용 :

```
1  class Dicd {
2
3      int dice1;
4      int dice2;
5
6      int getDice1() {
7          return dice1;
8      }
9
10     void setDice1(int dice1) {
11         this.dice1 = dice1;
12     }
13
14     int getDice2() {
15         return dice2;
16     }
17     void setDice2(int dice2) {
18         this.dice2 = dice2;
19     }
20 }
21
22 public class c_Quiz34 {
23     public static void main(String[] args) {
24         Dicd D = new Dicd();
25
26         D.dice1 = (int) (Math.random()*6+1);
27         D.dice2 = (int) (Math.random()*6+1);
28
29         System.out.printf("1p값 : %d / 2p값 : %d", D.dice1, D.dice2);
30     }
31 }
```

주요 내용

--

D.dice1과 D.dice2의 값이

Int GetDice1() {
Return dice1로 반환

Int GetDice2() {
Return dice2로 반환

--

Dicd D = new Dicd();
생성자 호출
→ 초기값을 생성해주는 역할

--

▶ 내용 :

```
1 class TestDice2 {
2     int comDice;
3     int userDice;
4
5     TestDice2() {
6         comDice = getRandomDice();
7         userDice = getRandomDice();
8     }
9     int getRandomDice() {
10        return (int) (Math.random()*6+1);
11    }
12    void checkWinner() {
13        switch (whoWin()) {
14            case 1:
15                System.out.printf("페관수런입니다. %(컴퓨터 vs %d 사용자)\n", comDice, userDice);
16                break;
17
18            case 2:
19                System.out.printf("사용자가 이겼습니다. %d(컴퓨터) vs %d(사용자)\n", comDice, userDice);
20                break;
21
22            case 3:
23                System.out.printf("비겼으니 형은 면하겠습니다. %d (컴퓨터) vs %d(사용자)\n", comDice, userDice);
24                break;
25        }
26    }
27    int whoWin() {
28        if(comDice > userDice) {
29            return 1;
30        } else if (comDice < userDice) {
31            return 2;
32        } else {
33            return 3;
34        }
35    }
36 }
37 public class d_quiz35 {
38     public static void main(String[] args) {
39         TestDice2 td = new TestDice2();
40         td.checkWinner();
41     }
42 }
```

주요 내용

--

TestDice2 td = new TestDice2();
>> Testdice2 생성자를 호출하라

Td.checkWinner();
>> checkWinner에 진입(접근?)하라

이후 1~6까지 랜덤숫자 생성후
if문에 의해 반환값 1~3이 정해진후
출력

--

▶ 내용 :

```
34 int whoWin() {
35     if (comDice > userDice) {
36         return 1;
37     } else if (comDice < userDice) {
38         return 2;
39     } else {
40         return 3;
41     }
42 }
43 Boolean redoDicegame() {
44     System.out.println("게임을 계속 하시겠습니까? 0(아니오), 1(예) ");
45     int num = scan.nextInt();
46     Boolean isTrue = false;
47
48     switch(num) {
49         case 0:
50             isTrue = false;
51             break;
52         case 2:
53             comDice = getRandomDice();
54             userDice = getRandomDice();
55             isTrue = true;
56             break;
57     }
58     return isTrue;
59 }
60 }
61 ▶ public class e_Quiz35강화문제 {
62 ▶     public static void main(String[] args) {
63         TestDice3 td = new TestDice3();
64
65         do {
66             td.checkWinner();
67         } while (td.redoDicegame());
68
69     }
```

주요 내용

--

Q.

Boolean문이 들어간 부분에 대해
다시한번 설명해주시면
감사하겠습니다.

특히

Boolean isTrue=false와 true값이
어떻게 사용되었는지 알고 싶습니다.

--