

20200514 Java과제

인스턴스 기본조작

B반 신석환

#toString 재정의

Object라는 가장 큰 클래스 안에 있는
메소드 toString() 을 오버라이드 하여
재정의 할 수 있다.

때에 따라 내가 원하는 출력문이 있다면
바꿔줄 수 있다.

```
1 package TEST;
2
3 public class instancePractice {
4     private String name;
5     private int hp;
6
7
8     @Override
9     public String toString() {
10         // TODO Auto-generated method stub
11         return "My name is " + name + " and Hp is " + hp;
12     }
13
14     public static void main(String[] args) {
15         instancePractice IP = new instancePractice();
16
17         IP.name = "shin";
18         IP.hp = 40;
19
20         System.out.println(IP);
21     }
22 }
23
```

Console X

<terminated> instancePractice [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (20

My name is shin and Hp is 40

#equals 재정의

Equals 메소드를 재정의 하여 두개의 생성자가 같은지 아닌지 판별할 때 일정 값만 같아도 두개가 같은 것으로 만들 수 있다.

#IP는 ("shin", 40, 30)이고, IP2는 ("shin", 40, 20)이기 때문에 다르다고 판별하여 "false"를 출력한다.

```
1 package TEST;
2
3 public class instancePractice {
4     private String name;
5     private int hp;
6     private int mp;
7
8
9     @Override
10    public String toString() {
11        // TODO Auto-generated method stub
12        return "My name is " + name + " and Hp is " + hp;
13    }
14
15    public static void main(String[] args) {
16        instancePractice IP = new instancePractice();
17        instancePractice IP2 = new instancePractice();
18
19        IP.name = "shin";
20        IP.hp = 40;
21        IP.mp = 30;
22
23        IP2.name = "shin";
24        IP2.hp = 40;
25        IP2.mp = 20;
26        System.out.println(IP.equals(IP2));
27    }
28 }
29
```

Console x

<terminated> instancePractice [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (20

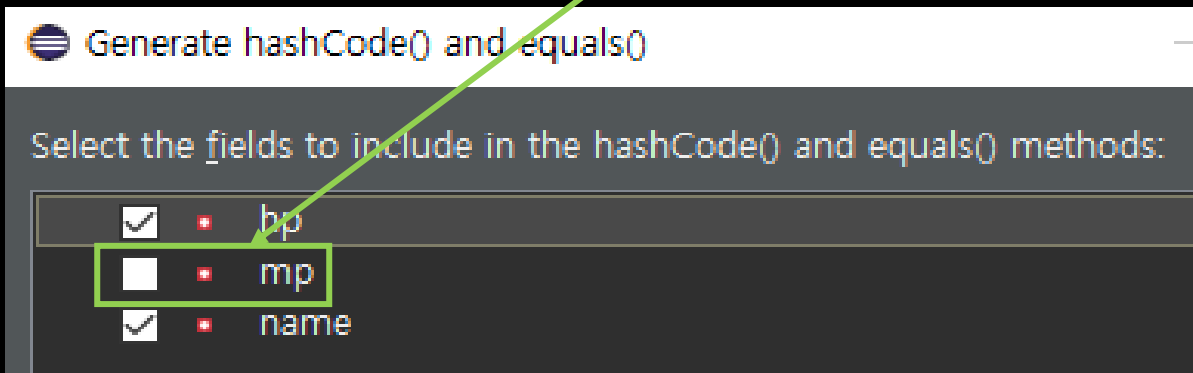
false

#equals 재정의


Equals 메소드의 기본 정의를 오버로드 한 것이다.
여기서 내가 비교하지 않을 것을 빼 줄 수 있다.

만약 내가 mp를 비교하지 않고 싶다면 mp부분을 빼주면 된다.

(간편한 방법은 오버라이드한 메소드를 지우고 다시 오버라이드 하는 과정에서 mp를 체크 안하면 된다. 하나하나 찾아서 지우는 번거로움을 줄이자)



```
@Override
public boolean equals(Object obj) {
    if (this == obj)
        return true;
    if (obj == null)
        return false;
    if (getClass() != obj.getClass())
        return false;
    InstancePractice other = (InstancePractice) obj;
    if (hp != other.hp)
        return false;
    if (mp != other.mp)
        return false;
    if (name == null) {
        if (other.name != null)
            return false;
    } else if (!name.equals(other.name))
        return false;
    return true;
}
```



```
@Override
public boolean equals(Object obj) {
    if (this == obj)
        return true;
    if (obj == null)
        return false;
    if (getClass() != obj.getClass())
        return false;
    InstancePractice other = (InstancePractice) obj;
    if (hp != other.hp)
        return false;
    if (name == null) {
        if (other.name != null)
            return false;
    } else if (!name.equals(other.name))
        return false;
    return true;
}
```

#hashCode

hashCode인 IPSet안에 IP를 넣을 수 있다.

IPSet의 size를 출력해 보면 1인 것을 확인할 수 있다

IP에 hp값을 추가 한다고 해서 size는 변함없다.

hashCode에 IP라는 생성자가 들어간 것이고. 그 안에서 값을 변경하는 것이기 때문에 결국 사이즈 1개이다.

```
49 public static void main(String[] args) {  
50     Set<instancePractice> IPSet = new HashSet<>();  
51     instancePractice IP = new instancePractice();  
52     instancePractice IP2 = new instancePractice();  
53     IP.name = "shin";  
54     IPSet.add(IP);  
55     System.out.println(IPSet.size());  
56  
57     // System.out.println(IP.equals(IP2));  
58 }  
59  
60  
61  
62
```

Console x
<terminated> instancePractice [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe

```
1  
49 public static void main(String[] args) {  
50     Set<instancePractice> IPSet = new HashSet<>();  
51     instancePractice IP = new instancePractice();  
52     instancePractice IP2 = new instancePractice();  
53     IP.name = "shin";  
54     IP.hp = 40;  
55     IPSet.add(IP);  
56     System.out.println(IPSet.size());  
57  
58     // System.out.println(IP.equals(IP2));  
59 }  
60  
61  
62
```

Console x
<terminated> instancePractice [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe

#hashCode()의 오버라이드

- Hashcode란 객체를 식별할 하나의 정수값을 말한다.

객체의 메모리 번지를 이용해서 해시코드를 만들어 리턴하기 때문에 객체 마다 다른 값을 가지고 있다.

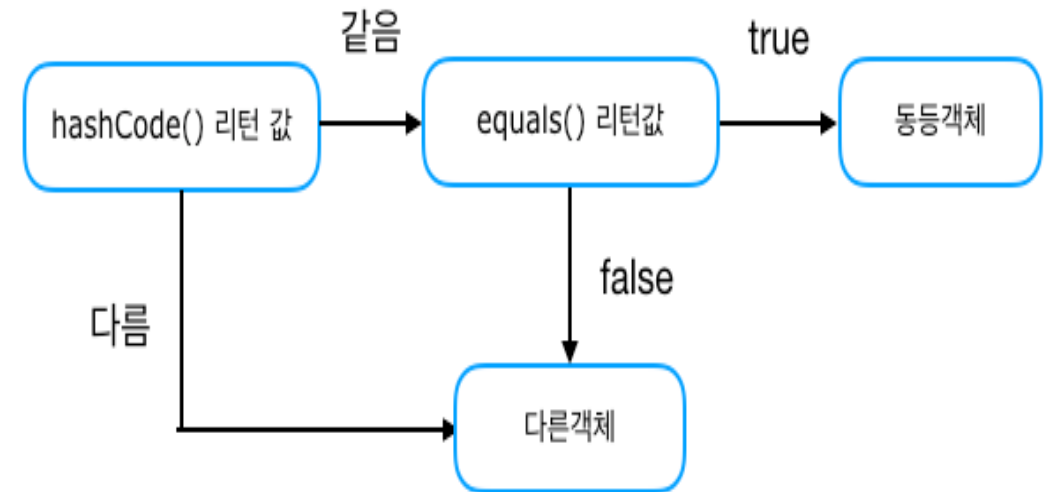
- 해시코드값이 같으면 equals()메소드로 다시 비교한다.
이 두개가 모두 맞아야 동등객체로 판단한다.

```
49 public static void main(String[] args) {
50     Set<instancePractice> IPSet = new HashSet<>();
51     instancePractice IP = new instancePractice();
52     instancePractice IP2 = new instancePractice();
53     IP.name = "shin";
54     IP2.name = "shin";
55     IPSet.add(IP);
56     IPSet.add(IP2);
57
58     System.out.println(IPSet.size());
59
60     System.out.println(IP.hashCode());
61     System.out.println(IP2.hashCode());
62 }
63 }
64
```

Console x

<terminated> instancePractice [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe

1
3530235
3530235



#인스턴스의 순서 결정

- Collections.sort() 메소드는 컬렉션 내부를 정렬해 준다. (hasSet은 사용불가)
- 하지만 sort()메소드를 사용하기 위해서는 다음과 같은 제약이 따른다.

```
public static <T extends Comparable<? super T>> void sort(List<T> list)
```

- 제약) 정렬 대상은 반드시 Comparable 인터페이스를 구현해야 한다.

```
@Override
public int compareTo(InstancePractice o) {
    if(this.mp == o.mp) {
        return 0;
    } else if(this.mp > o.mp) {
        return 1;
    } else {
        return -1;
    }
}
```

#clone() 메소드

생성자를 복제하여 사용할 수 있다.

Clone()메소드를 사용 하면서 내가 어떤 속성을 복제하고 싶은지 재정의해 줄 수 있다.

Name 과 Hp만 같게 clone을 재정의 하고 Clone을 사용해서 출력을 해본 결과

```
60 @Override
61 protected instancePractice clone() throws CloneNotSupportedException {
62     instancePractice result = new instancePractice();
63     result.name = this.name;
64     result.hp = this.hp;
65
66     return result;
67 }
68
69
70 public static void main(String[] args) throws CloneNotSupportedException {
71     Set<instancePractice> IPSet = new HashSet<>();
72     instancePractice IP = new instancePractice();
73
74     IP.name = "shin";
75     IP.hp = 40;
76     IP.mp = 30;
77     instancePractice IP2 = IP.clone();
78
79     System.out.println(IPSet.size());
80
81     System.out.println(IP);
82     System.out.println(IP2);
83 }
84 }
85
```

Console x

<terminated> instancePractice [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (2020. 5. 15. 오후 5:29:14 - 오후 5:29:14)

0

My name is shin and Hp is 40 and Mp is 30

My name is shin and Hp is 40 and Mp is 0

다음과 같은 Book 클래스가 있습니다.

```
1 public class Book {  
2     private String title;  
3     private Date publishDate;  
4     private String comment;  
5  
6     // getter/setter  
7 }
```

다음 동작을 할 수 있도록 Book 클래스를 수정하시오.

- 1) 제목과 출간일이 같으면 같은 책으로 판단한다. 또한 HashSet 등의 컬렉션에 넣어도 동일 객체로 판단한다.
- 2) Book 인스턴스를 담고 있는 컬렉션이 대해 Collections.sort() 를 사용하면 출간일이 오래된 순서대로 정렬된다.
- 3) clone() 메서드를 호출하면 복제된다.

1, 2번

List Book을 만든다.
Calendar cal을 만든다.
Cal에 날짜를 set해준다. (2015, 5, 2)
Book1을 만든다("shin", date, "asdf")

List Book을 만든다.
Calendar cal2을 만든다.
Cal2에 날짜를 set해준다. (2015, 5, 2)
Book1을 만든다("shin", date, "asd")

비교하면 같다.

Sort로 정렬한다. 정렬이 된다.

이유는 뒷장

```
2 import java.text.ParseException;
3 import java.util.ArrayList;
4 import java.util.Calendar;
5 import java.util.Collections;
6 import java.util.Date;
7 import java.util.List;
8 import java.util.List;
9
10 public class MainForBook {
11     public static void main(String[] args) throws ParseException {
12         List<Book> Book = new ArrayList<>();
13         Calendar cal = Calendar.getInstance();
14         cal.set(2015, Calendar.MAY, 2);
15         Date date = new Date(cal.getTimeInMillis());
16         Book book1 = new Book("shin", date, "asdf");
17         Book.add(book1);
18
19         Calendar cal2 = Calendar.getInstance();
20         cal2.set(2015, Calendar.MAY, 2);
21         Date date2 = new Date(cal2.getTimeInMillis());
22         Book book2 = new Book("shin", date2, "asf");
23         Book.add(book2);
24
25         Calendar cal3 = Calendar.getInstance();
26         cal2.set(2015, Calendar.JUNE, 2);
27         Date date3 = new Date(cal3.getTimeInMillis());
28         Book book3 = new Book("shin", date3, "asf");
29         Book.add(book3);
30
31         Collections.sort(Book);
32
33         System.out.println(book1.equals(book2));
34         for (int i = 0; i < 3; i++) {
35             System.out.println(Book.get(i));
36         }
37     }
38 }
39
```

```
Console x
<terminated> MainForBook [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (2020. 5. 14. 오후 1:14)
true
Book [title= shin, PublishDate= 2015/ 05/ 02, comment= asdf]
Book [title= shin, PublishDate= 2015/ 05/ 02, comment= asf]
Book [title= shin, PublishDate= 2020/ 05/ 14, comment= asf]
```

#이유

toString값만 비교하도록 equals를 바꿔준다.

SimpleDateFormat 으로 날짜가 출력되는 방식을 정해주고,
그걸 String s는 PublishDate를 String 형태로 바꾼다.

그다음 출력을 해준다.

String형태로 바꿔주는것은 출력할때 이뤄지는 것이기 때문에 처음에 Date로 들어갈때는 무관하다.

```
@Override
public String toString() {
    SimpleDateFormat SDF = new SimpleDateFormat("yyyy/ MM/ dd");
    String s = SDF.format(PublishDate);
    return "Book [title= " + title + ", PublishDate= " + s + ", comment= " + comment + "];"
}
```

```
@Override
public int hashCode() {
    final int prime = 31;
    int result = 1;
    result = prime * result + ((PublishDate == null) ? 0 : PublishDate.hashCode());
    result = prime * result + ((title == null) ? 0 : title.hashCode());
    return result;
}
```

```
@Override
public boolean equals(Object obj) {
    if (this == obj)
        return true;
    if (obj == null)
        return false;
    if (getClass() != obj.getClass())
        return false;
    Book other = (Book) obj;
    if (PublishDate.toString() == null) {
        if (other.PublishDate.toString() != null)
            return false;
    } else if (!PublishDate.toString().equals(other.PublishDate.toString()))
        return false;
    if (title == null) {
        if (other.title != null)
            return false;
    } else if (!title.equals(other.title))
        return false;
    return true;
}
```

3번

오버라이드로 클론을 만들어서

Book4가 book1의 클론이 되게 하고

Sout를 해보면 boo1과 같은 값이 나온다.

```
@Override
protected Book clone() {
    Book result = new Book(title, PublishDate, comment);
    result.title = this.title;
    result.PublishDate = this.PublishDate;
    result.comment = this.comment;
    return result;
}
```

```
36
37     Book book4 = book1.clone();
38
39     System.out.println(book4);
40
41 }
42 }
43
```

Console x

<terminated> MainForBook [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (2020. 5. 14)
Book [title= shin, PublishDate= 2015/ 05/ 02, comment= asdf]