

## 23-여름학기 자바프로그래밍(001) Lab0628

소프트웨어학부 컴퓨터과학전공 2016133 이유진

### 1. 실습 1 PizzaTest.java

```
1 //이유진 (2016133)
2 //2023-06-28
3 //Lab0628-1: Circle클래스를 상속받는 Pizza 클래스를 이용해, 피자에 대한 정보를 출력함
4
5 class Circle {
6     protected int radius;
7
8     public Circle(int r) {
9         radius = r;
10    }
11 }
12
13 class Pizza extends Circle {
14     String pizzaName;
15
16     //생성자
17     public Pizza(String pizzaName, int r) {
18         super(r); //부모의 생성자
19         this.pizzaName = pizzaName;
20     }
21
22     //getter, setter
23     public String getPizzaName() {
24         return pizzaName;
25     }
26
27     public void setPizzaName(String pizzaName) {
28         this.pizzaName = pizzaName;
29     }
30
31     #Override
32     public String toString() {
33         //Circle로부터 상속받은 radius를 super 키워드로 사용
34         return "피자의 종류: " + pizzaName + ", 피자의 크기: " + super.radius;
35     }
36 }
37 }
```

```
38
39 public class PizzaTest {
40
41     public static void main(String[] args) {
42
43         Pizza obj = new Pizza("Pepperoni", 20);
44         System.out.println(obj);
45     }
46 }
47 }
```

```
Console X
<terminated> PizzaTest [Java Application] C:\Program Files\Java\jdk-17\bin
피자의 종류: Pepperoni, 피자의 크기: 20
```

## 2. 실습 2 Mybook.java

```
1 //이유진 (2016133)
2 //2023-06-28
3 //Lab0628-2: MyBook 클래스를 상속받는 Magazine 클래스를 구현함
4
5 class MyBook {
6     // 속성 정보
7     private String title; // 제목
8     private int pages; // 페이지 수
9     private String author; // 저자
10    private static int count = 0; // 책의 개수
11
12    // 생성자 (하나 이상 정의)
13    public MyBook(String title, int pages, String author) {
14        super();
15        this.title = title;
16        this.pages = pages;
17        this.author = author;
18        count++; // 책이 생성될 때마다 책의 수 증가
19    }
20
21    // 실행자, 접근자
22    public String getTitle() {
23        return title;
24    }
25
26    public void setTitle(String title) {
27        this.title = title;
28    }
29
30    public int getPages() {
31        return pages;
32    }
33
34    public void setPages(int pages) {
35        this.pages = pages;
36    }
37
38    public String getAuthor() {
39        return author;
40    }
41
42    public void setAuthor(String author) {
43        this.author = author;
44    }
45
46    // count에 대해서는 접근자 메소드만 클래스 메소드로 작성 -> set 없음
47    public static int getCount() {
48        return count;
49    }
50 }
51
52 class Magazine extends MyBook { // MyBook에 대한 하위 클래스
53     // 추가 속성: 발매일
54     private String date; // 추가 속성: 발매일
55
56     // 생성자 (한 개 이상)
57     public Magazine(String title, int pages, String author, String date) {
58         super(title, pages, author); // 생성자 내에서 부모 생성자 호출
59         this.date = date;
60     }
61
62     // 실행자, 접근자
63     // 부모 클래스에서 받은 필드들은 별도의 메소드 작성 없이 사용 가능하므로 작성하지 않음
64     public String getDate() {
65         return date;
66     }
67
68     public void setDate(String date) {
69         this.date = date;
70     }
71 }
72
73 }
```

```
27
28    public String getAuthor() {
29        return author;
30    }
31
32    public void setAuthor(String author) {
33        this.author = author;
34    }
35
36    // count에 대해서는 접근자 메소드만 클래스 메소드로 작성 -> set 없음
37    public static int getCount() {
38        return count;
39    }
40 }
41
42 class Magazine extends MyBook { // MyBook에 대한 하위 클래스
43     // 추가 속성: 발매일
44     private String date; // 추가 속성: 발매일
45
46     // 생성자 (한 개 이상)
47     public Magazine(String title, int pages, String author, String date) {
48         super(title, pages, author); // 생성자 내에서 부모 생성자 호출
49         this.date = date;
50     }
51
52     // 실행자, 접근자
53     // 부모 클래스에서 받은 필드들은 별도의 메소드 작성 없이 사용 가능하므로 작성하지 않음
54     public String getDate() {
55         return date;
56     }
57
58     public void setDate(String date) {
59         this.date = date;
60     }
61 }
62
63 }
```

### 3. 실습 3 ColorPointTest.java

```
1 //이유진 (2016133)
2 //2023-06-28
3 //Lab0628-3: Point 클래스를 상속받는 ColorPoint 클래스를 이용해 색깔있는 점의 정보를 출력함
4
5 class Point {
6     private int x, y;
7
8     // 생성자
9     public Point(int x, int y) {
10         super();
11         this.x = x;
12         this.y = y;
13     }
14
15     // 읽기자, 접근자
16     public int getX() {
17         return x;
18     }
19
20     public int getY() {
21         return y;
22     }
23
24     public void setXY(int x, int y) { // x, y의 set 메서드를 한 번에 작성
25         this.x = x;
26         this.y = y;
27     }
28 }
29
30 class ColorPoint extends Point {
31     private String color;
32
33     public ColorPoint(int x, int y, String color) {
34         super(x, y);
35         this.color = color;
36     }
37
38     // 읽기, 설정 메서드는 부모 클래스에게 받아 사용할
39     // 색깔 설정 메서드
```

```
38
39     // 색깔 설정 메서드
40     public void setColor(String color) {
41         this.color = color;
42     }
43
44     @Override
45     public String toString() {
46         return color + "색의 (" + super.getX() + ", " + super.getY() + ")의 점";
47     }
48
49 }
50
51 public class ColorPointTest {
52
53     public static void main(String[] args) {
54
55         ColorPoint cp = new ColorPoint(5, 5, "YELLOW");
56         cp.setXY(10, 20);
57         cp.setColor("RED");
58         System.out.println(cp.toString() + "입니다.");
59     }
60
61 }
62
```

```
Console X
<terminated> ColorPointTest [Java Application] C:\Pro
RED색의 (20, 20)의 점입니다.
```

#### 4. 실습 4 StudentTest.java

```
1 //이유진 (2016113)
2 //2023-06-28
3 //Lab0628-4: Human 클래스와 Human 클래스를 상속받는 Student 클래스로
4 //사람과 학생의 정보를 객체배열에 저장 후 출력
5
6 import java.util.Scanner;
7
8 class Human {
9     // 속성
10     private String name;
11     private int age;
12
13     // 생성자: 매개변수로 이름, 나이를 받아와서 멤버 변수 값 초기화
14     public Human(String name, int age) {
15         super();
16         this.name = name;
17         this.age = age;
18     }
19
20     // 접근자, 변경자
21     public String getName() {
22         return name;
23     }
24
25     public void setName(String name) {
26         this.name = name;
27     }
28
29     public int getAge() {
30         return age;
31     }
32
33     public void setAge(int age) {
34         this.age = age;
35     }
36
37     @Override
38     public String toString() { // 객체의 현재 값을 문자열로 반환
39         return "이름: " + name + ", 나이: " + age;
40     }
41 }
```

```
37 @Override
38 public String toString() { // 객체의 현재 값을 문자열로 반환
39     return "이름: " + name + ", 나이: " + age;
40 }
41
42 }
43
44 class Student extends Human {
45     // 추가 속성 정의
46     private String major;
47     private int sID;
48
49     // 생성자
50     public Student(String name, int age, String major, int sID) {
51         super(name, age); // 부모 클래스의 생성자 호출
52
53         // 매개변수로 받은 값으로 멤버 변수의 전공과 학번을 초기화
54         this.major = major;
55         this.sID = sID;
56     }
57
58     // 별도(전공)에 대한 접근자, 설정자
59     public String getMajor() {
60         return major;
61     }
62
63     public void setMajor(String major) {
64         this.major = major;
65     }
66
67     public int getSID() {
68         return sID;
69     }
70
71     public void setSID(int sID) {
72         this.sID = sID;
73     }
74
75     @Override
```

```
75 @Override
76 public String toString() { // 부모의 toString()을 호출해 받은 문자열 끝에 추가로 전공과 학번
77     return "[학생 정보]" + super.toString() + ", 전공: " + major + ", 학번: " + sID;
78 }
79
80 }
81
82 public class StudentTest {
83
84     public static void main(String[] args) {
85         Scanner scan = new Scanner(System.in);
86
87         // 크기 3의 Human 객체와 Student 객체 배열 생성
88         Human[] human = new Human[3];
89         Student[] student = new Student[3];
90
91         // 사용자로부터 이름과 나이를 입력받아서 3명의 Human 객체를 생성
92         for (int i = 0; i < human.length; i++) {
93             System.out.print("[ " + i + " ] Human 입력: ");
94             String name = scan.next();
95             int age = scan.nextInt();
96             human[i] = new Human(name, age);
97         }
98
99         // 사용자로부터 이름, 나이, 전공, 학번을 입력받아서 3명의 Student 객체 생성
100         for (int i = 0; i < student.length; i++) {
101             System.out.print("[ " + i + " ] Student 입력: ");
102             String name = scan.next();
103             int age = scan.nextInt();
104             String major = scan.next();
105             int sID = scan.nextInt();
106             student[i] = new Student(name, age, major, sID);
107         }
108
109         // 각 객체 참조변수 이름으로 출력문 작성
110         for (int i = 0; i < human.length; i++)
111             System.out.println(human[i]);
112     }
113 }
```

```

110      // 각 객체 참조변수 이름으로 출력은 가능
111      for (int i = 0; i < human.length; i++)
112          System.out.println(human[i]);
113      for (int i = 0; i < student.length; i++)
114          System.out.println(student[i]);
115      scan.close();
116  }
117  }
118  }
119  }

```

Console X

<terminated> StudentTest (1) [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe

```

[0] Human 입력: 춘향 18
[1] Human 입력: 몽룡 21
[2] Human 입력: 사또 50
[0] Student 입력: 미진 21 컴퓨터 112345
[1] Student 입력: 가현 22 경영 123456
[2] Student 입력: 용준 24 경제 143215
이름: 춘향, 나이: 18
이름: 몽룡, 나이: 21
이름: 사또, 나이: 50
[학생 정보] 이름: 미진, 나이: 21, 전공: 컴퓨터, 학번: 112345
[학생 정보] 이름: 가현, 나이: 22, 전공: 경영, 학번: 123456
[학생 정보] 이름: 용준, 나이: 24, 전공: 경제, 학번: 143215

```

## 5. 실습 5 BuyerTest.java

```
1 //이유진 (2016133)
2 //2023-06-28
3 //Lab0628-5: Item 클래스를 상속받는 클래스들과 Buyer클래스를 이용해 구매 프로그램 구현
4
5 class Item {
6     private String name;
7     private int price;
8
9     // 생성자
10    public Item(String name, int price) {
11        super();
12        this.name = name;
13        this.price = price;
14    }
15
16    // 접근자, 설정자
17    public String getName() {
18        return name;
19    }
20
21    public void setName(String name) {
22        this.name = name;
23    }
24
25    public int getPrice() {
26        return price;
27    }
28
29    public void setPrice(int price) {
30        this.price = price;
31    }
32 }
33
34 class Food extends Item { // Item의 하위 클래스
35     // 생성자: 이름과 가격을 매개변수로 받아서 설정
36     public Food(String name, int price) {
37         super(name, price); // 부모 클래스의 설정자를 받음
38     }
39 }
```

```
39 }
40
41 @Override
42 public String toString() { // toString 메소드 재정의
43     return "[Food]" + this.getName();
44 }
45
46 class Book extends Item { // Item의 하위 클래스
47     // 추가 속성
48     private String author;
49
50     // 생성자: 이름, 가격, 저자를 매개변수로 받아서 설정
51     public Book(String name, int price, String author) {
52         super(name, price);
53         this.author = author;
54     }
55
56     // 접근자, 설정자
57     public String getAuthor() {
58         return author;
59     }
60
61     public void setAuthor(String author) {
62         this.author = author;
63     }
64
65 @Override
66 public String toString() { // toString 메소드 재정의
67     return "[Book]" + this.getName() + ", 저자: " + this.getAuthor();
68 }
69
70 class Movie extends Item { // Item의 하위 클래스
71     // 추가 속성
72     private String director;
73
74     // 생성자: 이름, 가격, 감독을 매개변수로 받아서 설정
75     public Movie(String name, int price, String director) {
76         super(name, price);
77         this.director = director;
78     }
79
80     // 접근자, 설정자
81     public String getDirector() {
82         return director;
83     }
84
85     public void setDirector(String director) {
86         this.director = director;
87     }
88
89 @Override
90 public String toString() { // toString 메소드 재정의
91     return "[Movie]" + this.getName() + ", 감독: " + this.getDirector();
92 }
93
94 class Buyer {
95     private int money;
96
97     // 생성자: money를 매개변수로 받아서 멤버 변수 설정
98     public Buyer(int money) {
99         super();
100        this.money = money;
101    }
102
103     // 접근자
104     public int getMoney() {
105         return money;
106     }
107
108     public void setMoney(int money) {
109        this.money = money;
110    }
111 }
```

```
75
76 // 생성자: 이름, 가격, 감독을 매개변수로 받아서 설정
77 public Movie(String name, int price, String director) {
78     super(name, price);
79     this.director = director;
80 }
81
82 // 접근자, 설정자
83 public String getDirector() {
84     return director;
85 }
86
87 public void setDirector(String director) {
88     this.director = director;
89 }
90
91 @Override
92 public String toString() { // toString 메소드 재정의
93     return "[Movie]" + this.getName() + ", 감독: " + this.getDirector();
94 }
95
96 class Buyer {
97     private int money;
98
99     // 생성자: money를 매개변수로 받아서 멤버 변수 설정
100    public Buyer(int money) {
101        super();
102        this.money = money;
103    }
104
105    // 접근자
106    public int getMoney() {
107        return money;
108    }
109
110    public void setMoney(int money) {
111        this.money = money;
112    }
113 }
```

```
109 }
110
111 // public void setMoney(int money) {
112 //     this.money = money;
113 // }
114
115 // buy 메소드: Item과 개수를 매개변수로 받아서 출력 결과와 같은 형태로 남은 금액
116 public void buy(Item t, int n) {
117     System.out.print(t.toString());
118     System.out.println("==>" + n + "개 구매");
119
120     // 구매 가능 여부 체크하는 코드 작성 후 money에서 차감
121     if (money >= t.getPrice() * n) {
122         money -= t.getPrice() * n;
123         System.out.println("구매 완료 후 잔액: " + money);
124     } else {
125         System.out.println("** 잔액 부족으로 구매 불가능 **");
126         System.out.println("잔액: " + money);
127     }
128 }
129 }
130
131 public class BuyerTest {
132
133     public static void main(String[] args) {
134         // 초기 자금을 50000원 가진 Buyer 생성
135         Buyer m = new Buyer(50000);
136
137         // 총 5번의 구매 실행
138         m.buy(new Food("비빔밥", 7000), 3);
139         m.buy(new Food("라면", 5000), 2);
140         m.buy(new Book("Java Programming", 12000, "이설희"), 1);
141         m.buy(new Movie("부산행", 9000, "연상호"), 1);
142         m.buy(new Food("김밥", 2000), 3);
143     }
144 }
145
146 }
147
```

```
Console X
<terminated> BuyerTest [Java Application] C:\Program Files\Java\jdk-17\bin\javaw
[Food] 비빔밥==>3개 구매
구매 완료 후 잔액: 29000
[Food] 라면==>2개 구매
구매 완료 후 잔액: 19000
[Book] Java Programming, 저자: 이설희==>1개 구매
구매 완료 후 잔액: 7000
[Movie] 부산행, 감독: 연상호==>1개 구매
** 잔액 부족으로 구매 불가능 **
잔액: 7000
[Food] 김밥==>3개 구매
구매 완료 후 잔액: 1000
```