

학과: 소프트웨어학부 컴퓨터과학전공

학번: 2312282

이름: 임다희

<실습 1> BankAccount 클래스 작성

-코드

```
10 /*작성자: 임다희(2312282)
2  *작성일: 2024-09-26
3  *Lab3-1.BankAccount 클래스 작성*/
4
5 import java.util.Scanner;
6
7 class BankAccount { //BankAccount 클래스 작성
8     private int balance; //필드 선언
9
10     BankAccount(int balance) {
11         this.balance = balance;
12     } //BankAccount 클래스의 생성자 정의
13
14     //각 클래스에 대한 접근자, 생성자 메소드 작성
15     public int getBalance() {
16         return balance;
17     }
18
19     public void setBalance(int balance) {
20         this.balance = balance;
21     }
22
23     //입금 메소드 생성.
24     public int deposit(int amount) {
25         //매개변수로 들어온 입금 금액 amount가 양수일 때 처리되도록 한다.
26         if (amount > 0) {
27             balance += amount;
28         }
29         return balance;
30     }
31
32     //출금 메소드 생성.
33     public int withdraw(int amount) {
34         //매개변수로 들어온 출금 금액 amount가 현재 잔액보다 클 때 처리되도록 한다.
35         if (amount <= balance) {
36             balance -= amount;
37         }
38         return balance;
39     }
40
41     //toString 메소드, 필드 값 정보를 String 값으로 반환한다.
42     public String toString() {
43         return String.format("잔액 : %d ", balance);
44     }
45
46     //계좌 이체 메소드 생성.
47     //이체할 금액(int 타입), 이체할 다른 계좌(BankAccount)를 매개변수로 받는다.
48     public void transfer(int amount, BankAccount otherAccount) {
49         if (amount <= this.balance) {
50             this.balance -= amount;
51             otherAccount.balance += amount;
52         }
53         else {
54             System.out.println("잔액이 부족합니다. ");
55         }
56         //이체할 금액이 현재 잔액보다 많으면 이체하지 않도록 처리한다.
57     }
58 }
59
60 //BankAccountTest 클래스.
61 public class BankAccountTest {
62
63     public static void main(String[] args) {
64         Scanner scan = new Scanner(System.in);
65         System.out.print("계좌 1 잔액 입력: ");
66         int a1 = scan.nextInt();
67         BankAccount myAccount1 = new BankAccount(a1);
68         System.out.print("계좌 2 잔액 입력: ");
69         int a2 = scan.nextInt();
70         //사용자에게 매개변수(계좌 잔액)를 입력받아 BankAccount 타입 객체를 2개 만든다.
71
72         BankAccount myAccount2 = new BankAccount(a2);
73         System.out.println("myAccount1: " + myAccount1);
74         System.out.println("myAccount2: " + myAccount2 + "\n");
75         //각 계좌 객체의 필드 정보를 출력한다.
76
77         System.out.print("계좌 1 --> 계좌 2 이체 금액 입력:");
78         int t = scan.nextInt();
79         myAccount1.transfer(t, myAccount2);
80         //사용자에게 이체할 금액을 입력받아, 계좌 1에서 2로 transfer 메소드를 이용해 이체한다.
81         System.out.println("transfer 호출 중");
82         System.out.println("myAccount1: " + myAccount1);
83         System.out.println("myAccount2: " + myAccount2);
84         //이체 후 변화된 각 계좌의 잔액 정보를 출력한다.
85     }
86 }
87 }
88 }
```

-실습 1 실행결과

```
80      //사용자에게 이체할 금액을 입력받아 계좌 1에서
81      System.out.println("transfer 호출
82      System.out.println("myAccount1: "
83      System.out.println("myAccount2: "
84      //이체 후 변화된 각 계좌의 잔액 정보를 출력함
85  }
86
87  }
88
```

Console ×

<terminated> BankAccountTest [Java Application] C:\WProgr

계좌 1 잔액 입력: 10000
계좌 2 잔액 입력 2000
myAccount1: 잔액 : 10000
myAccount2: 잔액 : 2000

계좌 1 --> 계좌 2 이체 금액 입력:3000
transfer 호출 중
myAccount1: 잔액 : 7000
myAccount2: 잔액 : 5000

<실습 2> Car 클래스 작성

-코드 및 실행결과

```
1  /* 작성자: 임다희 (2312282)
2   * 작성일: 2024-09-26
3   * Lab3-2.Car 클래스 작성 */
4
5  class Car { //Car 클래스 작성
6
7      // 생성된 Car 개체의 개수를 나타내는 static 변수 정의
8      // 필드 변수 정의
9      private static int numberOfCars=0;
10     private String model, make;
11
12     //Car 클래스의 생성자 정의.
13     //String 매개변수 2개를 입력받으며,
14     //생성될 때마다 static 변수 numberOfCars가 1씩 증가.
15     Car(String model, String make){
16         this.model=model;
17         this.make=make;
18         numberOfCars++;
19     }
20
21     //각 필드에 대한 접근자, 생성자 메소드 작성
22     public String getModel() {
23         return model;
24     }
25
26     public void setModel(String model) {
27         this.model = model;
28     }
29
30     public String getMake() {
31         return make;
32     }
33
34     public void setMake(String make) {
35         this.make = make;
36     }
37
38     //static 변수 getNumberOfCars에 접근해 값을 반환하는 메소드.(접근자)
39     public static int getNumberOfCars() {
40         return numberOfCars;
41     }
42     //getNumberOfCars의 생성자.
43     public static void setNumberOfCars(int numberOfCars) {
44         Car.numberOfCars = numberOfCars;
45     }
46
47     //toString 메소드. 필드 값 정보를 String 값으로 반환한다.
48     public String toString() {
49         return "Car [model=" + model + ", make=" + make + "]";
50     }
51 }
52
53 public class CarTest { //CarTest 클래스.
54     public static void main(String[] args) {
55         new Car("35SERIES","BENZ");
56         new Car("35SERIES","BENZ");
57         new Car("35SERIES","BENZ");
58         //새로운 Car 클래스의 객체를 3개 생성한다.
59         System.out.println("총 "+Car.getNumberOfCars()+"대의 자동차가 생산되었습니다.");
60         //static 변수에 접근해 Car 클래스의 객체가 몇개 인지를 출력한다.
61     }
62 }
63
```

```
56     //static 변수에 접근해 Car 클래스의 객체가 몇개 인지를
57 }
58 {
59
60 Console ×
<terminated> CarTest [Java Application] C:\Program Files\Java\jdk-
총 3대의 자동차가 생산되었습니다.
```

<실습 3> Circle 클래스 작성

-코드

```
1  *작성일: 2024-09-26
2  *Lab3-3.Circle 클래스 작성*/
3  import java.lang.Math;
4  class Circle { //Circle 클래스 작성
5      private int radius; //필드 변수 정의
6      public Circle(int d) {
7          radius=d;
8      }
9      //Circle 클래스의 생성자 정의.
10
11      public String toString() {
12          return "Circle [radius=" + radius + "]";
13      } //toString 메소드. 필드 값 정보를 String 값으로 반환한다.
14
15      //radius에 대한 생성자와 접근자.
16      public int getRadius() {
17          return radius;
18      }
19      public void setRadius(int radius) {
20          this.radius = radius;
21      }
22  }
23
24  public class CircleTest { //CircleTest 클래스 작성.
25      public static void main(String[] args) {
26          Circle[] list;
27          list=new Circle[3];
28          //크기가 3인 Circle 객체 배열을 생성한다.
29
30          for(int i=0; i<list.length; i++) {
31              int random=(int)(Math.random()*101);
32              //0~100 사이의 난수를 생성한다.
33              list[i]=new Circle(random);
34              //랜덤한 매개변수를 받는 Circle 객체를 생성해 배열에 저장한다.
35              System.out.println(list[i]);
36          }
37      }
38  }
```

-실행 결과

```
34      //랜덤한 매개변수를 받는 Circle 객체를 생성해 배열에 저장한다.
35      System.out.println(list[i]);
36  }
37  }
38  }
39
Console X
<terminated> CircleTest [Java Application] C:\Program Files\Java\jdk-17\bin\java.exe
Circle [radius=39]
Circle [radius=92]
Circle [radius=53]
```

<실습 4> Contacts 클래스 작성

-코드

```
1 1● /*작성자: 임다희(2312282)
2   *작성일: 2024-09-29
3   *Lab3-4.Contacts 클래스 작성*/
4
5 import java.util.Scanner;
6
7 class Contacts { // Contacts 클래스 작성.
8     private String name, tel, email;
9     private static int count = 0;
10
11     // 필드 변수 및 static 변수.
12 12● Contacts(String name, String tel, String email) {
13         this.name = name;
14         this.tel = tel;
15         this.email = email;
16         count++;
17     } // Contacts 클래스의 생성자 정의.
18         // 객체가 하나 생성될 때마다 static 변수 count가 1씩 증가한다.
19
20     // 각 필드 변수에 대한 접근자, 생성자 메소드 작성
21 21● public String getName() {
22         return name;
23     }
24
25 25● public void setName(String name) {
26         this.name = name;
27     }
28
29 29● public String getTel() {
30         return tel;
31     }
32
33 33● public void setTel(String tel) {
34         this.tel = tel;
35     }
36
37 37● public String getEmail() {
38         return email;
39     }
40
41 41● public void setEmail(String email) {
42         this.email = email;
43     }
44
45 45● public static void setCount(int count) {
46         Contacts.count = count;
47     }
48
49 49● public static int getCount() {
50         return count;
51     }
52
53     // toString 메소드. 필드 값 정보를 String 값으로 반환한다.
54 54● public String toString() {
55         return "Contacts [name=" + name + ", tel=" + tel + ", email=" + email + "];"
56     }
57 }
58
```



```

59 public class ContactsTest { // ContactsTest 클래스 생성.
60     public static void main(String[] args) {
61         Contacts[] list;
62         list = new Contacts[100]; // Contacts 객체를 저장하는 객체 배열(최대 100개까지 저장 가능)
63         Scanner scan = new Scanner(System.in);
64         System.out.println("연락처를 입력하시오(종료 -1)");
65         while (true) {
66             System.out.print("이름과 전화번호, 이메일을 입력하시오:");
67             String x = scan.next();
68             // 사용자에게 이름 정보를 입력받는다.
69             if (x.equals("-1")) {
70                 break;
71             }
72             // 사용자의 입력이 -1이 아니라면 계속 진행한다.
73             String y = scan.next();
74             String z = scan.next();
75             // 사용자에게 전화번호, 이메일 정보를 입력받는다.
76             list[Contacts.getCount()] = new Contacts(x, y, z);
77             // 객체 배열에 사용자가 입력한 정보를 이용해 새로운 Contacts 객체를 추가한다.
78         }
79         System.out.println("지인들의 수는 " + Contacts.getCount() + "입니다");
80         // static 변수에 접근해 Contacts 객체의 총 개수를 출력한다.
81         System.out.print("검색할 이름을 입력하시오: ");
82         String name = scan.next();
83         // 사용자에게 문자열을 입력받아 객체 배열에 대응하는 값이 있는지 검사한다.
84         for (int i = 0; i < list.length; i++) {
85             if (list[i] != null && name.equals(list[i].getName())) {
86                 System.out.print(name + "의 전화번호: " + list[i].getTel() + " 이메일: " + list[i].getEmail());
87                 // 사용자에게 입력받은 값과 배열 내 Contacts 객체의 Name 값이 같다면
88                 // 해당 객체의 Tel, Email 값을 함께 출력한다.
89             }
90         }
91     }
92 }

```

-실습 4 실행결과

```

85         if (list[i] != null && name.equals(list[i].getName())) {
86             System.out.print(name + "의 전화번호: " + list[i].getTel()
87                 // 사용자에게 입력받은 값과 배열 내 Contacts 객체의 Name 값이 같다면
88                 // 해당 객체의 Tel, Email 값을 함께 출력한다.

```

Console X

<terminated> ContactsTest [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (2024)

```

연락처를 입력하시오(종료 -1)
이름과 전화번호, 이메일을 입력하시오:Kim 010-1111-2222 kim@java.com
이름과 전화번호, 이메일을 입력하시오:Lee 010-2222-3333 lee@java.com
이름과 전화번호, 이메일을 입력하시오:Kang 010-5555-7777 kang@java.com
이름과 전화번호, 이메일을 입력하시오:Choe 010-1234-5678 choe@java.com
이름과 전화번호, 이메일을 입력하시오:-1
지인들의 수는 4입니다
검색할 이름을 입력하시오: Kang
Kang의 전화번호: 010-5555-7777 이메일: kang@java.com

```

<실습 5> Book 클래스 작성

```
10 /*작성자: 임다희(2312282)
11  *작성일: 2024-09-29
12  *Lab3-5.Book 클래스 작성*/
13 import java.util.Scanner;
14
15
16 class Book { //Book 클래스 작성
17
18     private String title;
19     private int score; //필드 변수 생성.
20
21     Book(String title, int score) {
22         this.title = title;
23         this.score = score;
24     } //Book 클래스의 생성자.
25
26     // 각 필드 변수에 대한 접근자, 생성자 메소드 작성
27     public String getTitle() {
28         return title;
29     }
30     public void setTitle(String title) {
31         this.title = title;
32     }
33     public int getScore() {
34         return score;
35     }
36     public void setScore(int score) {
37         this.score = score;
38     }
39
40     // toString 메소드. 필드 값 정보를 String 값으로 반환한다.
41     public String toString() {
42         return "Book [title=" + title + ", score=" + score + "]";
43     }
44 }
45
46 public class BookTest { //BookTest 클래스 작성
47
48     public static void main(String[] args) {
49         Scanner scan = new Scanner(System.in);
50         ArrayList<Book> list = new ArrayList<Book>();
51         //Book 타입의 원소를 저장하는 동적 객체 배열 생성.
52
53         while (true) {
54             System.out.println("=====");
55             System.out.println("1. 책 등록");
56             System.out.println("2. 책 검색");
57             System.out.println("3. 모든 책 출력");
58             System.out.println("4. 종료");
59             System.out.println("=====");
60             //메뉴 목록 출력. 사용자가 종료하기 전까지 계속 반복된다.
61             System.out.print("메뉴를 선택하십시오: ");
62             int menuNum = scan.nextInt();
63             //사용자로부터 메뉴 선택을 숫자로 입력받는다.
64
65             if (menuNum == 1) { //1번(책 등록)을 입력받은 경우
66                 scan.nextLine();
67                 System.out.print("책 제목: ");
68                 String title = scan.nextLine();
69
70                 System.out.print("책 평점: ");
71                 int score = scan.nextInt();
72
73                 list.add(new Book(title, score));
74                 //책 제목과 평점을 입력받아 새로운 Book 객체를 생성하고 객체 배열에 추가한다.
75             }
76             else if (menuNum == 2) { //2번(책 검색)을 입력받은 경우
77                 System.out.print("책 제목: ");
78                 scan.nextLine();
79                 String search = scan.nextLine(); //사용자로부터 검색할 책의 제목을 입력받는다.
80
81                 for (int i = 0; i < list.size(); i++) {
82                     if (list.get(i).getTitle().equals(search)) {
83                         System.out.println(
84                             "Book [title=" + list.get(i).getTitle() + ", score=" + list.get(i).getScore() + "]");
85                     }
86                 }
87                 //객체 배열 내 Book 객체가 사용자에게 입력받은 책 제목과 똑같은 Title 값을 가질 경우 title, score값을 함께 출력한다.
88             }
89             else if (menuNum == 3) { //3번(모든 책 출력)을 입력받은 경우
90                 for (int i = 0; i < list.size(); i++) {
91                     System.out.println(
92                         "Book [title=" + list.get(i).getTitle() + ", score=" + list.get(i).getScore() + "]");
93                 }
94                 //객체 배열 내에 있는 모든 Book 객체의 title, score 정보를 출력한다.
95             }
96             else if (menuNum == 4) { //4번(종료)을 입력받은 경우
97                 break; //while 루프문에서 빠져나가기 위해 과정을 종료한다.
98             }
99         }
100     }
101 }
```

-실습 5 코드 실행결과

```
<terminated> BookTest [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (2024. 9. 29. 오후 11:2)
=====
1. 책 등록
2. 책 검색
3. 모든 책 출력
4. 종료
=====
메뉴를 선택하시오: 1
책 제목: 자바 프로그래밍
책 평점: 8
=====
1. 책 등록
2. 책 검색
3. 모든 책 출력
4. 종료
=====
메뉴를 선택하시오: 1
책 제목: Python 기초
책 평점: 9
=====
1. 책 등록
2. 책 검색
3. 모든 책 출력
4. 종료
=====
메뉴를 선택하시오: 3
Book [title=자바 프로그래밍, score=8]
Book [title=Python 기초, score=9]
=====
1. 책 등록
2. 책 검색
3. 모든 책 출력
4. 종료
=====
메뉴를 선택하시오: 2
책 제목: 자바 프로그래밍
Book [title=자바 프로그래밍, score=8]
=====
1. 책 등록
2. 책 검색
3. 모든 책 출력
4. 종료
=====
메뉴를 선택하시오: 4
```


