

[디지털 컨버전스] 스마트 콘텐츠와 웹 융합 응용SW 개발자 양성과정

강사 : 이상훈

학생 : 임초롱

ArrayList

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day15/src/ArrayListTest.java>

```
4 public class ArrayListTest {
5     public static void main(String[] args) {
6         String[] fruits = {"apple", "strawberry", "grape", "watermelon"};
7         // 원래 ArrayList의 기본형식은 ?
8         // ArrayList<String> 이름 = new ArrayList<String>();
9         // ArrayList<Integer> 이름 = new ArrayList<Integer>();
10        // asList()를 통해 배열등을 ArrayList로 변형할 수 있음
11
12        ArrayList<String> fruitsList = new ArrayList<>(①Arrays.asList(fruits));
13
14        for (int i = 0; i < 4; i++) {
15            System.out.printf("get(%d) = %s\n", i, fruitsList.②get(i));
16        }
17
18        System.out.println("복제 이후!");
19
20        // fruitsList의 내용을 clone 변수에 복제함
21        // ArrayList에 구현되어 있는 clone은 객체를 복제할 수 있게 서포트한다. ③
22        ArrayList<String> clone = (ArrayList<String>) fruitsList.clone();
23        for (int i = 0; i < 4; i++) {
24            // ArrayList에 있는 내용을 가져올때 get(index)를 사용합니다.
25            // 여기서 index는 | 데이터1 | ---> | 데이터2 | ---> | 데이터8 | --->
26            //                0                1                2
27            System.out.printf("get(%d) = %s\n", i, clone.get(i));
28        }
```

ArrayList의 압도적인 장점 :

데이터 변경 시, 대규모 데이터 관리시 arraylist가 유리하다
하지만 데이터 변경이 없다면 일반 배열이 훨씬 유리하다.

ArrayList의 형식 :

- ArrayList<String> 변수명 = new ArrayList<String>();
- ArrayList<Integer> 변수명 = new ArrayList<Integer>();

① asList()를 통해 ArrayList로 변형할 수 있다.

② get(index)를 통해 ArrayList에 있는 내용을 가져올 수 있다.

③ clone()를 통해 ArrayList 객체를 복제 할 수 있다.

```
↑ "C:\Program Files\Java\jdk-15.0.2\bin\java
↓
get(0) = apple
get(1) = strawberry
get(2) = grape
get(3) = watermelon
복제 이후!
get(0) = apple
get(1) = strawberry
get(2) = grape
get(3) = watermelon
```

기존 ArrayList를 복제한 것으로 값이 동일하다.

ArrayList

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day15/src/ArrayListTest.java>

```
30 fruitsList.remove(0, "grape"); // 원본에 대한 값 변화
31 System.out.println("fruitsList의 grape 삭제 이후 clone 다시 보기");
32
33 for (int i = 0; i < 4; i++) {
34     System.out.printf("get(%d) = %s\n", i, clone.get(i));
35 }
36
37 System.out.println("fruitsList는 지워졌을까 ?");
38
39 for (int i = 0; i < 3; i++) {
40     System.out.printf("get(%d) = %s\n", i, fruitsList.get(i));
41 }
42
43 System.out.println("그렇다면 새로운 과일 추가해보기");
44 fruitsList.add("cherry");
45
46 System.out.println("fruitsList에 cherry 추가 이후 clone 다시 보기");
47 for (int i = 0; i < 4; i++) {
48     System.out.printf("get(%d) = %s\n", i, clone.get(i));
49 }
50
51 System.out.println("fruitsList는 새 과일이 생겼을까 ?");
52 for (int i = 0; i < 4; i++) {
53     System.out.printf("get(%d) = %s\n", i, fruitsList.get(i));
54 }
55
```

fruitsList의 grape 삭제 이후 clone 다시 보기

```
get(0) = apple
get(1) = strawberry
get(2) = grape
get(3) = watermelon
```

fruitsList는 지워졌을까 ?

```
get(0) = apple
get(1) = strawberry
get(2) = watermelon
```

- ① remove()를 통해 값을 지울 수 있다.
- clone은 기존 ArrayList의 복사본임으로 기존 값을 유지한다.
 - fruitsList는 원본이므로, grape 값이 지워졌다.

그렇다면 새로운 과일 추가해보기

fruitsList에 cherry 추가 이후 clone 다시 보기

```
get(0) = apple
get(1) = strawberry
get(2) = grape
get(3) = watermelon
```

fruitsList는 새 과일이 생겼을까 ?

```
get(0) = apple
get(1) = strawberry
get(2) = watermelon
get(3) = cherry
```

- ② add()를 통해 값을 추가할 수 있다.
- 현재 리스트에서 가장 마지막에 추가된다.
- clone은 기존 ArrayList의 복사본임으로 기존 값을 유지한다.
 - fruitsList는 원본이므로, cherry값이 마지막에 추가되었다.

ArrayList

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day15/src/ArrayListTest.java>

```
56 // 완전 통채로 밀림 ①
57 fruitsList.clear();
58
59 System.out.println("after clear: " + fruitsList);
60
61 // 검색 - indexOf
62 // 검색 후, [2] 인덱스에 포도가 있기 때문에 2가 출력된다. ②
63 System.out.println("포도 어딴니 ? " + clone.indexOf("grape"));
64 // 검색 후, [1] 인덱스에 딸기가 있기 때문에 1이 출력된다.
65 System.out.println("딸기는 ? " + clone.indexOf("strawberry"));
66 // 존재하지 않는 것은 -1 이 나옴(오류)
67 System.out.println("드립을 치고 싶은데 ? " + clone.indexOf("드립"));
68
69 // 당첨자 명단 ~~~~
70 // 공모자 명단 ~~~~
71 // 시상식 참석자 명단 ~~~~
72 // for (참석자 명단 ~~~) {
73 //     if (indexOf("당첨자 명단") == -1) {
74 //         continue;
75 //     } else {
76 //         cnt++
77 //     }
78 // }
79 // indexOf에서 -1은 존재하지 않는 것을 뜻한다.
80 // 참석자 명단에서 검색했을때 어떠한 사람이 당첨자 명단에 존재하지 않는다면
81 // 계속 참석자 명단에 있는 이름을 가져와서 체크할 것이고,
82 // 당첨자 명단에 참석자 이름이 존재한다면, 그 수를 cnt++ 카운트할 것이다.
83
84 // 위와 같은 형태로 시상식에 참석하면 수상자 숫자를 집계할 수 있다.
85
```

after clear: []

- ① clear()를 통해 값을 전부 지울 수 있다.
- 값이 모두 지워져서 [], 즉 null인 상태가 출력된다.

포도 어딴니 ? 2

딸기는 ? 1

드립을 치고 싶은데 ? -1

- ② indexOf()를 통해 값을 검색할 수 있다.
- indexOf(), 괄호 안에 찾고자 하는 값을 넣으면, 이 예제의 경우, [2] 인덱스에 들어있는 포도를 찾고, 그 포도가 있는 위치, 인덱스의 번호를 출력해준다.

③ contains 또한 indexOf와 유사하게 사용이 가능하다.

// contains()는 참 혹은 거짓으로 결과가 나오는 것을 확인할 수 있다.

System.out.println("포도 어딴니 ? " + clone.contains("grape"));

System.out.println("딸기는 ? " + clone.contains("strawberry"));

System.out.println("드립을 치고 싶은데 ? " + clone.contains("드립"));

포도 어딴니 ? true

딸기는 ? true

드립을 치고 싶은데 ? false

- ③ contains()를 통해 값의 유무를 파악할 수 있다.
- indexOf()와 유사하다. indexOf()가 위치를 알려주었다면, contains()은 true와 false를 통해 유/무를 알려준다.

객체 전달과 값 전달

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day15/src/CanWeClone.java>

```
1  import java.util.Arrays;
2
3  class Test {
4      int[] arr;
5
6      public Test () {
7          arr = new int[3];
8
9          for (int i = 0; i < 3; i++) {
10             arr[i] = (int)(Math.random() * 6 + 1);
11             // [0] ~ [2]에 1 ~ 6까지의 랜덤 숫자를 3개를 각각 대입한다.
12         }
13     }
14     public int[] clone () {
15         int[] testArr = new int[arr.length];
16         // int [] testArr = new int[3];
17         // testArr의 인덱스 3개
18
19         for (int i = 0; i < arr.length; i++) {
20             testArr[i] = arr[i];
21             // testArr[0] = arr[0]
22             // testArr[0] = 1 ~ 6까지의 랜덤 숫자
23         }
24         return testArr;
25         // testArr[0] = 1 ~ 6
26         // testArr[1] = 1 ~ 6
27         // testArr[2] = 1 ~ 6를 리턴;
28     }
```

```
30     public void changeArr () {
31         arr[1] = 10000000;
32     }
33     @ public void vectorAdd (int[] vec) {
34         for (int i = 0; i < vec.length; i++) {
35             arr[i] += vec[i];
36         }
37     }
38
39     @Override
40     public String toString() {
41         return "Test{" +
42             "arr=" + Arrays.toString(arr) +
43             '}';
44     }
```

객체 전달과 값 전달

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day15/src/CanWeClone.java>

```
47 public class CanWeClone {
48     public static void main(String[] args) {
49         Test t = new Test();
50
51         // 원본
52         // [0] ~ [2]에 1 ~ 6 까지의 랜덤 숫자를 3개를 각각 대입한다.
53         // toString에 의해 출력된 그 값
54         System.out.println(t);
55         System.out.println("-----절취선-----");
56
57         int[] cloneArr = t.clone();
58         // t.clone()은 계산 후 리턴받은 testArr[]
59         // cloneArr[i] = testArr[i]
60
61
62         for (int i = 0; i < cloneArr.length; i++) {
63             System.out.printf("cloneArr[%d] = %d\n", i, cloneArr[i]);
64         }
65         System.out.println("-----절취선-----");
```

```
Test{arr=[2, 4, 6]}
-----절취선-----
cloneArr[0] = 2
cloneArr[1] = 4
cloneArr[2] = 6
-----절취선-----
```

```
67     t.changeArr();
68     // arr[1] = 10000000; 값을 바꿈
69
70     // 원본 = 값이 바뀐 것이 적용됨
71     System.out.println(t);
72     System.out.println("-----절취선-----");
73
74     // 말그대로 복사본, 초기값 유지되며 바뀐 값이 적용되지 않는다.
75     for (int i = 0; i < cloneArr.length; i++) {
76         System.out.printf("cloneArr[%d] = %d\n", i, cloneArr[i]);
77     }
78
79     System.out.println("-----절취선-----");
80     // clone을 변경하기 위해서는 ?
81     cloneArr[1] = 77777777;
82
83     for (int i = 0; i < cloneArr.length; i++) {
84         // 원본적용이 아닌, 복사본에 대해 변경값을 적용하면
85         // 복사본의 값도 변경된다.
86         System.out.printf("cloneArr[%d] = %d\n", i, cloneArr[i]);
87     }
```

```
Test{arr=[2, 10000000, 6]}
-----절취선-----
cloneArr[0] = 2
cloneArr[1] = 4
cloneArr[2] = 6
-----절취선-----
cloneArr[0] = 2
cloneArr[1] = 77777777
cloneArr[2] = 6
```

49번 문제 복습 : ArrayList

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day15/src/Quiz49Explain.java>

```
1 import java.util.ArrayList;
2
3 //ArrayList에 중복을 허용하여 랜덤 숫자를 10개 만든다.
4 //랜덤 숫자의 범위는 10 ~ 12 다.
5 //여기서 각각의 숫자들이 몇 개씩 중복 되었는지 카운트해보자!
6
7 class ArrayListCount {
8     ArrayList<Integer> intLists; //ArrayList 배열, 10개의 랜덤숫자
9     ArrayList<Integer> duplicateLists;
10
11     final int RAND_RANGE = 3;
12     final int RAND_START_OFFSET = 10;
13     final int DATA_LENGTH = 10;
14
15     final int FIRST_VALUE = 10;
16     final int SECOND_VALUE = 11;
17     final int THIRD_VALUE = 12;
18
19     final int FIRST_IDX = FIRST_VALUE - RAND_START_OFFSET;
20     final int SECOND_IDX = SECOND_VALUE - RAND_START_OFFSET;
21     final int THIRD_IDX = THIRD_VALUE - RAND_START_OFFSET;
22
23     // 생성자 - 초기화
24     public ArrayListCount () {
25         // 10 ~ 12의 숫자가 중복 허용된 상태로 10개 배치됨
26         intLists = new ArrayList<Integer>();
27         // 각각의 숫자가 몇 개씩 중복되었는지 체크에 활용함
28         duplicateLists = new ArrayList<Integer>();
29
30         // 10개 만들라 했으니 10개 만듦
31         for (int i = 0; i < DATA_LENGTH; i++) {
32             intLists.add((int)(Math.random() * RAND_RANGE + RAND_START_OFFSET));
33             //intLists.add(10);
34         }
```

```
36 // 10, 11, 12는 총 3개로 랜덤 범위만큼 생성
37 // 10은 인덱스 0번
38 // 11은 인덱스 1번
39 // 12는 인덱스 2번
40 // 위와 같이 취급하겠다는 전략
41 for (int i = 0; i < RAND_RANGE; i++) {
42     duplicateLists.add(0);
43     // duplicateLists.add(x): x를 현재 리스트에서 가장 마지막에 추가함
44     // ex) 0, 1, 4, 8
45     // ArrayList.add(77)
46     // ex) 0, 1, 4, 8, 77
47
48     // duplicateLists.add(0); 의 for문 3번이 반복될때,
49     // ex) 0(i=0), 0(i=1), 0(i=3)
50     // 각각 for문에 따른 인덱스 3개가 duplicateLists에 생긴다.
51
52 }
53
54
55 public void cntDuplicate () {
56     // foreach <<<
57     for (int num : intLists) {
58         // 빠른 값이 10이면
59         if (num == FIRST_VALUE) {
60             // ArrayList.add(x): x를 현재 리스트에서 가장 마지막에 추가함
61             // ex) 0, 1, 4, 8
62             // ArrayList.add(77)
63             // ex) 0, 1, 4, 8, 77
64
65             // ArrayList.add(idx, x): x를 idx 인덱스 위치에 저장하고
66             // 기존에 있던 정보를 뒤로 한칸씩 밀어버림
67             // ex) 0, 1, 4, 8
68             // ArrayList.add(2, 77)
69             // ex) 0, 1, 77, 4, 8
```


49번 문제 복습 : ArrayList

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day15/src/Quiz49Explain.java>

```
71 // 처음 시작시 [0, 0, 0]
72 duplicateLists.add(FIRST_IDX,
73     element: duplicateLists.get(FIRST_IDX) + 1);
74 // 이후 [1, 0, 0, 0]
75 // 그 다음은 [2, 1, 0, 0]
76 duplicateLists.remove(index: 1);
77 // 이후 [1, 0, 0]
78 // 그 다음은 [2, 0, 0]
79 // 빼온 값이 11이면
80 } else if (num == SECOND_VALUE) {
81     duplicateLists.add(SECOND_IDX,
82         element: duplicateLists.get(SECOND_IDX) + 1);
83     duplicateLists.remove(index: 2);
84     // 빼온 값이 12라면
85 } else if (num == THIRD_VALUE) {
86     duplicateLists.add(THIRD_IDX,
87         element: duplicateLists.get(THIRD_IDX) + 1);
88     duplicateLists.remove(index: 3);
89 }
90 }
91 }
92
93 @Override
94 public String toString() {
95     return "ArrayListCount{" +
96         "intLists=" + intLists +
97         ", duplicateLists=" + duplicateLists +
98         '}';
99 }
100
101 }
```

```
93 @Override
94 public String toString() {
95     return "ArrayListCount{" +
96         "intLists=" + intLists +
97         ", duplicateLists=" + duplicateLists +
98         '}';
99 }
100
101
102 public class Quiz49Explain {
103     public static void main(String[] args) {
104         ArrayListCount alc = new ArrayListCount();
105
106         System.out.println(alc);
107
108         alc.cntDuplicate();
109
110         System.out.println(alc);
111     }
112 }
```

Run: Quiz49Explain ×

```
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" -javaagent:C:\Users\user\AppData\Local\JetBra
ArrayListCount{intLists=[11, 12, 11, 12, 10, 12, 11, 10, 10, 10], duplicateLists=[0, 0, 0]}
ArrayListCount{intLists=[11, 12, 11, 12, 10, 12, 11, 10, 10, 10], duplicateLists=[4, 3, 3]}
```


51번 문제

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day15/src/Quiz51.java>

```
1 import java.util.ArrayList;
2 import java.util.Arrays;
3 import java.util.Scanner;
4
5 // ArrayList를 이용한 상점만들기.
6 // 1. 소지금을 정한다 (입력값)
7 // 2. 물건과 구매가를 정해두고 리스트를 만든다.(ArrayList)
8 // 3. 구매에 따라 소지금 변동을 만든다.
9 // 4. 구매리스트를 만들어 구매물품을 확인할 수 있도록 한다.
10 // 5. 구매한 물건을 재판매할 수 있도록 한다.
11 // 6. 판매에 따라 소지금 변동을 만든다.
12
13
14 class Shop{
15     ArrayList<String> storeList; // 상점 판매목록
16     ArrayList<String> myList; // 나의 장바구니
17     final String[] supermarket = {"고기","과자","음료수","라면","햇반","과일"};
18     final int[] price = {7000, 1000, 500, 3000, 1200, 5000};
19     int answer;
20     int wallet;
21     int money;
22     boolean isTrue;
23     int purchaseNum;
24     Scanner scan;
25
26     public Shop(){
27         scan = new Scanner(System.in);
28         isTrue = true;
29         storeList = new ArrayList<>(Arrays.asList(supermarket));
30         myList = new ArrayList<>();
31     }
32 }
```

```
public void shoppingStart(){
    Starwallet();
    shopping();
    buyObject();
    moreShopping();
    myShoppingList();
}

public void Starwallet(){
    System.out.print("현재 소지하고 계시는 금액을 입력해주세요 : ");
    money = scan.nextInt();
    wallet = money;
}

public void myShoppingList(){
    System.out.printf("현재 내 장바구니 = %s\n ", myList);
    System.out.println("가지고 계신 물품을 판매하시겠습니까? (Y:1/N:2) \n");
    answer = scan.nextInt();
    if(answer == 1) {
        sellObj();
    } else if(answer == 2) {
        System.out.printf("현재 내 장바구니 = %s, 소지 금액은 %d입니다. 쇼핑을 종료합니다.\n ",myList,wallet);
    } else {
        System.out.println("잘못된 입력입니다.\n");
    }
}
```

51번 문제

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day15/src/Quiz51.java>

```
public void plusSellObj(){
    System.out.println("추가적으로 물건을 더 판매하시겠습니까?(Y:1/N:2)\n ");
    answer = scan.nextInt();
    if(answer == 1){
        sellObj();
    } else if(answer == 2){
        System.out.printf("판매를 종료합니다. 현재 내 장바구니 = %s, 소지 금액은 %d입니다",myList,wallet);
    } else {
        System.out.println("잘못된 입력입니다.\n");
    }
}

public void sellObj(){
    System.out.print("가지고 계신 물품을 판매시 구매가격의 10%를 제외한 가격에 판매됩니다. " +
        "그래도 판매하시겠습니까?(Y:1/N:2)\n");
    answer = scan.nextInt();
    if(answer == 1){
        System.out.println("판매하고자 하는 물건을 입력하세요. : ");
        String obj = scan.nextLine();
        obj = scan.nextLine();

        int num = storeList.indexOf(obj);
        wallet += (price[num] * 0.9);

        myList.remove(obj);
        System.out.println(myList);

        plusSellObj();
    } else if(answer == 2){
        System.out.printf("판매를 종료합니다. 현재 내 장바구니 = %s, 소지 금액은 %d입니다",myList,wallet);
    } else {
        System.out.println("잘못된 입력입니다.\n");
    }
}
```

```
public void moreShopping(){
    while(isTrue){
        System.out.print("쇼핑을 더 진행하시겠습니까? (Y:1/N:2)\n");
        answer = scan.nextInt();

        if(answer ==1){
            shopping();
            buyObject();
            break;
        } else if (answer == 2){
            System.out.printf("쇼핑을 종료합니다. 현재 잔액 : %d\n", wallet);
            isTrue = false;
            break;
        } else {
            System.out.println("잘못된 입력입니다.\n");
        }
        break;
    }
}

public void buyObject() {
    System.out.print("구매할 물건의 번호를 입력하여 주세요.\n");
    purchaseNum = scan.nextInt();
    while(isTrue) {
        switch (purchaseNum) {
            case 1:
                System.out.printf("%s의 가격은 %s입니다. 구입하시겠습니까? (Y:1/N:2)\n", storeList.get(0), price[0]);
                answer = scan.nextInt();
                if (answer == 1) {
                    wallet -= price[0];
                    myList.add(storeList.get(0));
                    System.out.printf("%s를 구매하셨습니다. 현재 잔액 %d \n", storeList.get(0), wallet);
                    moreShopping();
                    break;
                }
            }
        }
    }
}
```

51번 문제

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day15/src/Quiz51.java>

```
    } else if(answer == 2) {
        moreShopping();
        break;
    } else {
        System.out.println("잘못된 입력입니다.\n");
    }
}

case 2:
System.out.printf("%s의 가격은 %s입니다. 구입하시겠습니까? (Y:1/N:2)\n", storeList.get(1), price[1]);
answer = scan.nextInt();
if (answer == 1) {
    wallet -= price[1];
    myList.add(storeList.get(1));
    System.out.printf("%s를 구매하셨습니다. 현재 잔액 %d \n", storeList.get(1), wallet);
    moreShopping();
    break;
} else if(answer == 2) {
    moreShopping();
    break;
} else {
    System.out.println("잘못된 입력입니다.\n");
}

case 3:
System.out.printf("%s의 가격은 %s입니다. 구입하시겠습니까? (Y:1/N:2)\n", storeList.get(2), price[2]);
answer = scan.nextInt();
if (answer == 1) {
    wallet -= price[2];
    myList.add(storeList.get(2));
    System.out.printf("%s를 구매하셨습니다. 현재 잔액 %d \n", storeList.get(2), wallet);
    moreShopping();
    break;
} else if(answer == 2) {
    moreShopping();
    break;
} else {
    System.out.println("잘못된 입력입니다.\n");
}
}
```

```
case 4:
System.out.printf("%s의 가격은 %s입니다. 구입하시겠습니까? (Y:1/N:2)\n", storeList.get(3), price[3]);
answer = scan.nextInt();
if (answer == 1) {
    wallet -= price[3];
    myList.add(storeList.get(3));
    System.out.printf("%s를 구매하셨습니다. 현재 잔액 %d \n", storeList.get(3), wallet);
    moreShopping();
    break;
} else if(answer == 2) {
    moreShopping();
    break;
} else {
    System.out.println("잘못된 입력입니다.\n");
}

case 5:
System.out.printf("%s의 가격은 %s입니다. 구입하시겠습니까? (Y:1/N:2)\n", storeList.get(4), price[4]);
answer = scan.nextInt();
if (answer == 1) {
    wallet -= price[4];
    myList.add(storeList.get(4));
    System.out.printf("%s를 구매하셨습니다. 현재 잔액 %d \n", storeList.get(4), wallet);
    moreShopping();
    break;
} else if(answer == 2) {
    moreShopping();
    break;
} else {
    System.out.println("잘못된 입력입니다.\n");
}

case 6:
System.out.printf("%s의 가격은 %s입니다. 구입하시겠습니까? (Y:1/N:2)\n", storeList.get(5), price[5]);
answer = scan.nextInt();
if (answer == 1) {
    wallet -= price[5];
    myList.add(storeList.get(5));
}
```

51번 문제

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day15/src/Quiz51.java>

```
201         System.out.printf("%s를 구매하셨습니다. 현재 잔액 %d \n", storeList.get(5), wallet);
202         moreShopping();
203         break;
204     } else if(answer == 2) {
205         moreShopping();
206         break;
207     } else {
208         System.out.println("잘못된 입력입니다.\n");
209     }
210
211     break;
212 }
213
214 }
215
216 }
217
218 public void shopping(){
219     System.out.print("<< 땡땡마트 판매 물품 목록입니다. >> \n");
220     for (int i = 0; i < supermarket.length; i++){
221         System.out.printf("%d ) %s(%s원)  ", i+1, storeList.get(i), price[i]);
222     }
223     System.out.println();
224 }
225
226 }
227
228 public class Quiz51 {
229     public static void main(String[] args) {
230         Shop s = new Shop();
231         s.shoppingStart();
232     }
233 }
```

현재 소지하고 계시는 금액을 입력해주세요 : 10000
<< 땡땡마트 판매 물품 목록입니다. >>
1) 고기(7000원) 2) 과자(1000원) 3) 음료수(500원) 4) 라면(3000원) 5) 햇반(1200원) 6) 과일(5000원)
구매할 물건의 번호를 입력하여 주세요.
1
고기의 가격은 7000입니다. 구입하시겠습니까? (Y:1/N:2)
1
고기를 구매하셨습니다. 현재 잔액 3000
쇼핑을 더 진행하시겠습니까? (Y:1/N:2)
1
<< 땡땡마트 판매 물품 목록입니다. >>
1) 고기(7000원) 2) 과자(1000원) 3) 음료수(500원) 4) 라면(3000원) 5) 햇반(1200원) 6) 과일(5000원)
구매할 물건의 번호를 입력하여 주세요.
5
햇반의 가격은 1200입니다. 구입하시겠습니까? (Y:1/N:2)
1
햇반을 구매하셨습니다. 현재 잔액 1800
쇼핑을 더 진행하시겠습니까? (Y:1/N:2)
2
쇼핑을 종료합니다. 현재 잔액 : 1800
현재 내 장바구니 = [고기, 햇반]
가지고 계신 물품을 판매하시겠습니까? (Y:1/N:2)
1
가지고 계신 물품을 판매시 구매가격의 10%를 제외한 가격에 판매됩니다. 그래도 판매하시겠습니까?(Y:1/N:2)
1
판매하고자 하는 물건을 입력하세요. :
고기
[햇반]
추가적으로 물건을 더 판매하시겠습니까?(Y:1/N:2)
2
판매를 종료합니다. 현재 내 장바구니 = [햇반], 소지 금액은 8100입니다