

4 주차 과제(연습과제 2)_ 1930022 최하원

#1

<코드>

```
public class Question1 {  
  
    public static void main(String[] args) {  
        String s1 = "Java Korea";  
        String s2 = new String("Java Korea");  
        String s3 = s1;  
  
        // #1.  
        System.out.println("#1. s1.equals(s2): " + s1.equals(s2));  
        // #2.  
        System.out.println("#2. s1.replace('a', 'A'): " +  
s1.replace('a', 'A'));  
        // #3.  
        System.out.println("#3. s2.toUpperCase(): " + s2.toUpperCase());  
        // #4.  
        System.out.println("#4. s2.length(): " + s2.length());  
        // #5.  
        System.out.println("#5. s1.lastIndexOf('a'): " +  
s1.lastIndexOf('a'));  
        // #6.  
        System.out.println("#6. s2.substring(5): " + s2.substring(5));  
  
        // #7.  
        System.out.println("#7. s1.substring(0,4): " +  
s1.substring(0,4));  
        // #8.  
        System.out.println("#8. s1.compareTo(s2): " + s2.compareTo(s2));  
  
        // #9.  
        System.out.println("#9. s1.compareTo(s3): " + s1.compareTo(s3));  
        // #10.  
        System.out.println("#10. s1.startsWith(\"java\") " +  
s1.startsWith("java") );  
    }  
}
```

<결과>

```
*Question1.java ×
1 public class Question1 {
2
3     public static void main(String[] args) {
4         String s1 = "Java Korea";
5         String s2 = new String("Java Korea");
6         String s3 = s1;
7
8         // #1.
9         System.out.println("#1. s1.equals(s2): " + s1.equals(s2));
10        // #2.
11        System.out.println("#2. s1.replace('a', 'A'): " + s1.replace('a', 'A'));
12        // #3.
13        System.out.println("#3. s2.toUpperCase(): " + s2.toUpperCase());
14        // #4.
15        System.out.println("#4. s2.length(): " + s2.length());
16        // #5.
17        System.out.println("#5. s1.lastIndexOf('a'): " + s1.lastIndexOf('a'));
18        // #6.
19        System.out.println("#6. s2.substring(5): " + s2.substring(5));
20        // #7.
21        System.out.println("#7. s1.substring(0,4): " + s1.substring(0,4));
22        // #8.
23        System.out.println("#8. s1.compareTo(s2): " + s1.compareTo(s2));
24        // #9.
25        System.out.println("#9. s1.compareTo(s3): " + s1.compareTo(s3));
26        // #10.
27        System.out.println("#10. s1.startsWith(\"java\") " + s1.startsWith("java") );
28    }
29 }
```

Problems @ Javadoc Declaration Console ×

<terminated> Question1 [Java Application] C:\Users\chw10\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.v

```
#1. s1.equals(s2): true
#2. s1.replace('a', 'A'): JAvA KoreA
#3. s2.toUpperCase(): JAVA KOREA
#4. s2.length(): 10
#5. s1.lastIndexOf('a'): 9
#6. s2.substring(5): Korea
#7. s1.substring(0,4): Java
#8. s1.compareTo(s2): 0
#9. s1.compareTo(s3): 0
#10. s1.startsWith("java") false
```

#2

<코드>

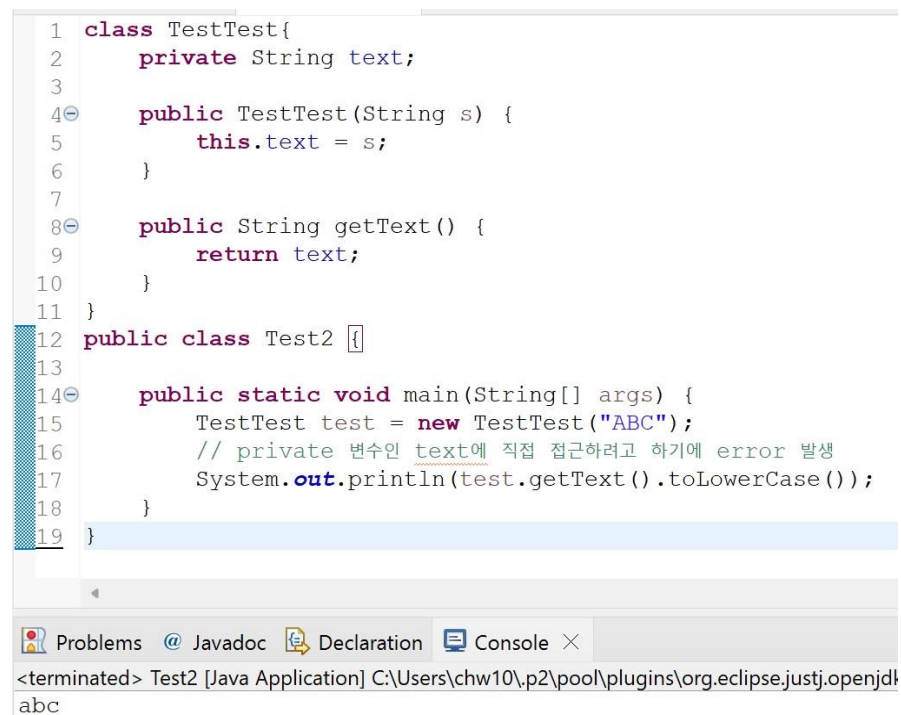
```
class TestTest{
    private String text;

    public TestTest(String s) {
        this.text = s;
    }

    public String getText() {
        return text;
    }
}
public class Test2 {

    public static void main(String[] args) {
        TestTest test = new TestTest("ABC");
        // private 변수인 text에 직접 접근을 시도해서 error 발생 -> public
        변수로 수정
        System.out.println(test.getText().toLowerCase());
    }
}
```

<결과>



The screenshot shows the Eclipse IDE with a Java project. The code editor displays the same code as in the previous block. The console window at the bottom shows the output of the program.

```
1 class TestTest{
2     private String text;
3
4     public TestTest(String s) {
5         this.text = s;
6     }
7
8     public String getText() {
9         return text;
10    }
11 }
12 public class Test2 {
13
14     public static void main(String[] args) {
15         TestTest test = new TestTest("ABC");
16         // private 변수인 text에 직접 접근하려고 하기에 error 발생
17         System.out.println(test.getText().toLowerCase());
18     }
19 }
```

Problems @ Javadoc Declaration Console ×

<terminated> Test2 [Java Application] C:\Users\chw10\p2\pool\plugins\org.eclipse.justj.openjdk
abc

#3

<코드>

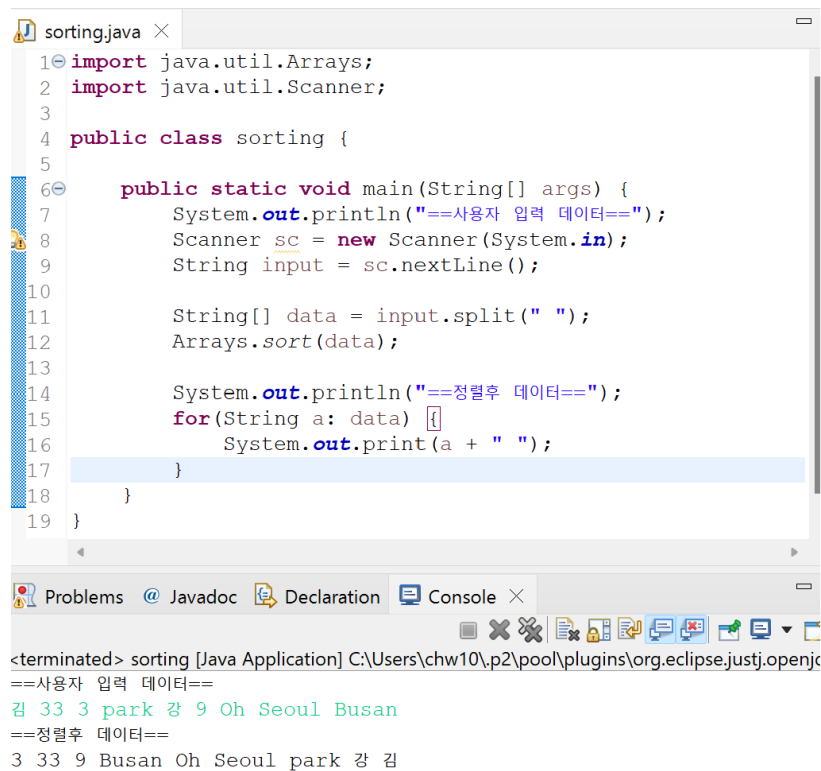
```
import java.util.Arrays;
import java.util.Scanner;

public class sorting {
    public static void main(String[] args) {
        System.out.println("==사용자 입력 데이터==");
        Scanner sc = new Scanner(System.in);
        String input = sc.nextLine();

        String[] data = input.split(" ");
        Arrays.sort(data);

        System.out.println("==정렬후 데이터==");
        for(String a: data) {
            System.out.print(a + " ");
        }
    }
}
```

<결과>



The screenshot shows the Eclipse IDE with the file 'sorting.java' open. The code is identical to the one in the previous block. The console at the bottom shows the output of the program:

```
<terminated> sorting [Java Application] C:\Users\chw10\p2\pool\plugins\org.eclipse.justj.openj
==사용자 입력 데이터==
김 33 3 park 강 9 Oh Seoul Busan
==정렬후 데이터==
3 33 9 Busan Oh Seoul park 강 김
```

#4

<코드>

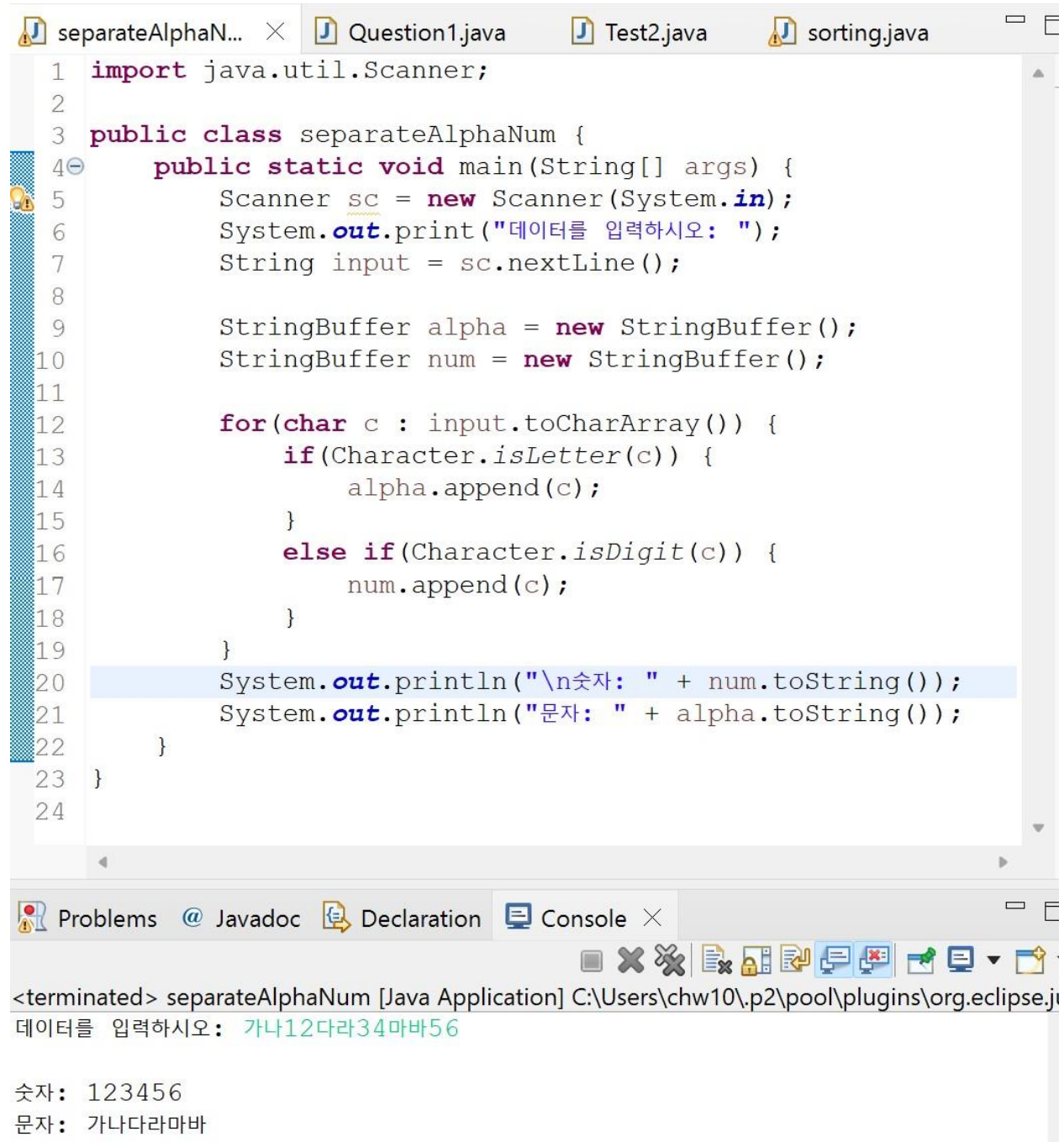
```
import java.util.Scanner;

public class separateAlphaNum {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("데이터를 입력하시오: ");
        String input = sc.nextLine();

        StringBuffer alpha = new StringBuffer();
        StringBuffer num = new StringBuffer();

        for(char c : input.toCharArray()) {
            if(Character.isLetter(c)) {
                alpha.append(c);
            }
            else if(Character.isDigit(c)) {
                num.append(c);
            }
        }
        System.out.println("\n 숫자: " + num.toString());
        System.out.println("문자: " + alpha.toString());
    }
}
```

<결과>



The screenshot shows the Eclipse IDE with a Java project. The editor displays a file named `separateAlphaNum.java` with the following code:

```
1 import java.util.Scanner;
2
3 public class separateAlphaNum {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         System.out.print("데이터를 입력하시오: ");
7         String input = sc.nextLine();
8
9         StringBuffer alpha = new StringBuffer();
10        StringBuffer num = new StringBuffer();
11
12        for(char c : input.toCharArray()) {
13            if(Character.isLetter(c)) {
14                alpha.append(c);
15            }
16            else if(Character.isDigit(c)) {
17                num.append(c);
18            }
19        }
20        System.out.println("\n숫자: " + num.toString());
21        System.out.println("문자: " + alpha.toString());
22    }
23 }
24
```

The console output shows the program's execution:

```
<terminated> separateAlphaNum [Java Application] C:\Users\chw10\p2\pool\plugins\org.eclipse.j
데이터를 입력하시오: 가나12다라34마바56

숫자: 123456
문자: 가나다라마바
```

#5

<코드>

```
public class MarcRecordParser {
    public static void main(String[] args) {
        String marcRecord = "0001 123456\n" +
                            "100 1 $aAuthor Name\n"
                            + "245 10 $aBook Title : $bSubtitle /
$cAuthor Name.\n"
                            + "260 $aPublisher Name $bYear\n"
                            + "300 $a200 pages\n";

        // 변수 선언
        int idNum = -1;
        String bookTitle = "";
        String subtitle = "";
        String authorName = "";
        String publisherName = "";
        String year = "";
        String totalPages = "";

        // 줄 단위로 문자열을 분할해서 입력처리
        String[] lines = marcRecord.split("\n");
        for (String line : lines) {
            // 파싱 후 변수 저장

            if (line.startsWith("0001")) { // 책 번호
                String[] parts = line.split(" ");
                if (parts.length >= 2) {
                    idNum = Integer.parseInt(parts[1]);
                }
            } else if (line.startsWith("100")) { // 작가 이름
                String[] parts = line.split("\\$");
                for (String part : parts) {
                    if (part.startsWith("a")) {
                        authorName = part.substring(1).trim();
                    }
                }
            } else if (line.startsWith("245")) { // 책 제목, 부제목, 작가 이름
                String[] parts = line.split("\\$");
                for (String part : parts) {
                    if (part.startsWith("a")) {
                        bookTitle = part.substring(1, part.length() -
2).trim();
                    } else if (part.startsWith("b")) {
                        subtitle = part.substring(1, part.length() -
2).trim();
                    } else if (part.startsWith("c")) {
                        authorName = part.substring(1, part.length() -
1).trim();
                    }
                }
            }
        }
    }
}
```

```

    }
    } else if (line.startsWith("260")) { // 출판사 이름, 출간연도
        String[] parts = line.split("\\$");
        for (String part : parts) {
            if (part.startsWith("a")) {
                publisherName = part.substring(1).trim();
            } else if (part.startsWith("b")) {
                year = part.substring(1).trim();
            }
        }
    } else if (line.startsWith("300")) { // 총 페이지 수
        String[] parts = line.split("\\$");
        for (String part : parts) {
            if (part.startsWith("a")) {
                totalPages = part.substring(1).trim();
            }
        }
    }
}

// 출력
//System.out.println("번호: " + idNum);
System.out.println("책명 : " + bookTitle + " , " + subtitle);
System.out.println("저자 : " + authorName);
System.out.println("출판사 : " + publisherName);
System.out.println("출판년도 : " + year);
System.out.println("총 페이지 : " + totalPages);
}
}

```


<결과>

```
MarcRecordParser.java × Question1.java Test2.java sorting.java
1 public class MarcRecordParser {
2     public static void main(String[] args) {
3         String marcRecord = "0001 123456\n" +
4                               "100 1 $aAuthor Name\n"
5                               + "245 10 $aBook Title : $bSubtitle / $cAuthor Name.\n"
6                               + "260 $aPublisher Name $bYear\n"
7                               + "300 $a200 pages\n";
8
9         // 변수 선언
10        int idNum = -1;
11        String bookTitle = "";
12        String subtitle = "";
13        String authorName = "";
14        String publisherName = "";
15        String year = "";
16        String totalPages = "";
17
18        // 줄 단위로 문자열을 분할해서 입력처리
19        String[] lines = marcRecord.split("\n");
20        for (String line : lines) {
21            // 파싱 후 변수 저장
22            if (line.startsWith("0001")) { // 책 번호
23                String[] parts = line.split(" ");
24                if (parts.length >= 2) {
25                    idNum = Integer.parseInt(parts[1]);
26                }
27            } else if (line.startsWith("100")) { // 작가 이름
28                String[] parts = line.split("\\$");
29                for (String part : parts) {
30                    if (part.startsWith("a")) {
31                        authorName = part.substring(1).trim();
32                    }
33                }
34            } else if (line.startsWith("245")) { // 책 제목, 부제목, 작가 이름
35                String[] parts = line.split("\\$");
36            }
37        }
38    }
39}
```

Problems @ Javadoc Declaration Console ×

<terminated> MarcRecordParser [Java Application] C:\Users\chw10\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.
책명 : Book Title , Subtitle
저자 : Author Name
출판사 : Publisher Name
출판년도 : Year
총 페이지 : 200 pages