72 Java의 정석定石 3판 - 연습문제 풀이

# Chapter

java.lang패키지와 유용한 클래스

java.lang package & util classes

## [ 연습문제 ]

[9-1] 다음과 같은 실행결과를 얻도록

SutdaCard클래스의

equals()를 멤버변수인

num,

isKwang의 값을 비교하도록 오버라이딩하고 테스트 하시오.

class Exercise9\_1 {

public static void main(String[] args) { SutdaCard c1 = new SutdaCard(3,true); SutdaCard c2 = new SutdaCard(3,true);

System.out.println("c1="+c1); System.out.println("c2="+c2); System.out.println("c1.equals(c2):"+c1.equals(c2));

}

}

class SutdaCard { int num;

boolean isKwang;

SutdaCard() {

this(1, true);

}

SutdaCard(int num, boolean isKwang) { this.num = num;

this.isKwang = isKwang;

}

public boolean equals(Object obj) {

**/\***

**(1)** 매개변수로 넘겨진 객체의 **num, isKwang**과

멤버변수 **num, isKwang**을 비교하도록 오버라이딩 하시오**.**

**\*/**

}

public String toString() {

return num + ( isKwang ? "K":"");

}

}

[연습문제]**/ch9/Exercise9\_1.java**

c1=3K c2=3K

c1.equals(c2):true

[실행결과]

[9-2] 다음과 같은 실행결과를 얻도록 Point3D클래스의 equals()를 멤버변수인 x, y, z 의 값을 비교하도록 오버라이딩하고, toString()은 실행결과를 참고해서 적절히 오버라이 딩하시오.

class Exercise9\_2 {

public static void main(String[] args) { Point3D p1 = new Point3D(1,2,3); Point3D p2 = new Point3D(1,2,3);

System.out.println(p1); System.out.println(p2); System.out.println("p1==p2?"+(p1==p2));

System.out.println("p1.equals(p2)?"+(p1.equals(p2)));

}

}

class Point3D { int x,y,z;

Point3D(int x, int y, int z) { this.x=x;

this.y=y; this.z=z;

}

Point3D() {

this(0,0,0);

}

public boolean equals(Object obj) {

**/\***

1. 인스턴스변수 **x, y, z**를 비교하도록 오버라이딩하시오**.**

**\*/**

}

public String toString() {

**/\***

1. 인스턴스변수 **x, y, z**의 내용을 출력하도록 오버라이딩하시오**.**

**\*/**

}

}

[연습문제]**/ch9/Exercise9\_2.java**

[1,2,3]

[1,2,3]

p1==p2?false p1.equals(p2)?true

[실행결과]

[9-3] 다음과 같은 실행결과가 나오도록 코드를 완성하시오.

class Exercise9\_3 {

public static void main(String[] args) {

String fullPath = "c:\\jdk1.8\\work\\PathSeparateTest.java"; String path = "";

String fileName = "";

**/\***

**(1)** 알맞은 코드를 넣어 완성하시오**.**

**\*/**

System.out.println("fullPath:"+fullPath); System.out.println("path:"+path); System.out.println("fileName:"+fileName);

}

}

[연습문제]**/ch9/Exercise9\_3.java**

fullPath:c:\jdk1.8\work\PathSeparateTest.java path:c:\jdk1.8\work fileName:PathSeparateTest.java

[실행결과]

[9-4] 다음과 같이 정의된 메서드를 작성하고 테스트하시오.

메서드명 : printGraph

기 능 : 주어진 배열에 담긴 값만큼 주어진 문자를 가로로 출력한 후, 값을 출력한다. 반환타입 : 없음

매개변수 : int[] dataArr - 출력할 그래프의 데이터 char ch - 그래프로 출력할 문자.

class Exercise9\_4 {

static void printGraph(int[] dataArr, char ch) {

**/\***

**(1) printGraph**메서드를 작성하시오**.**

**\*/**

}

public static void main(String[] args) { printGraph(new int[]{3,7,1,4},'\*');

}

}

[연습문제]**/ch9/Exercise9\_4.java**

\*\*\*3

\*\*\*\*\*\*\*7

\*1

\*\*\*\*4

[실행결과]

메서드명 : count

기 능 : 주어진 문자열(src)에 찾으려는 문자열(target)이 몇 번 나오는지 세어서 반환한다. 반환타입 : int

매개변수 : String src

String target

[Hint] String클래스의 indexOf(String str, int fromIndex)를 사용할 것

class Exercise9\_5 {

public static int count(String src, String target) { int count = 0; // 찾은 횟수

int pos = 0; // 찾기 시작할 위치

**/\***

**(1)** 반복문을 사용해서 아래의 과정을 반복한다**.**

1. **src**에서 **target**을 **pos**의 위치부터 찾는다**.**
2. 찾으면 **count**의 값을 **1** 증가 시키고**,**

**pos**의 값을 **target.length**만큼 증가시킨다**.**

1. **indexOf**의 결과가 **-1**이면 반복문을 빠져나가서 **count**를 반환한다**.**

**\*/**

}

public static void main(String[] args) { System.out.println(count("12345AB12AB345AB","AB")); System.out.println(count("12345","AB"));

}

}

[연습문제]**/ch9/Exercise9\_5.java**

3

0

[실행결과]

[9-6] 다음과 같이 정의된 메서드를 작성하고 테스트하시오.

메서드명 : fillZero

기 능 : 주어진 문자열(숫자)로 주어진 길이의 문자열로 만들고, 왼쪽 빈 공간은 '0'으로 채운다.

만일 주어진 문자열이 null이거나 문자열의 길이가 length의 값과 같으면 그대로 반환한다.

만일 주어진 length의 값이 0보다 같거나 작은 값이면, 빈 문자열("")을 반환한다. 반환타입 : String

매개변수 : String src - 변환할 문자열

int length - 변환한 문자열의 길이

class Exercise9\_6 {

public static String fillZero(String src, int length) {

**/\* (1) fillZero**메서드를 작성하시오**.**

1. **src**가 널이거나 **src.length()**가 **length**와 같으면 **src**를 그대로 반환한다**.**
2. **length**의 값이 **0**보다 같거나 작으면 빈 문자열**("")**을 반환한다**.**
3. **src**의 길이가 **length**의 값보다 크면 **src**를 **length**만큼 잘라서 반환한다**.**
4. 길이가 **length**인 **char**배열을 생성한다**.**
5. **4**에서 생성한 **char**배열을 **'0'**으로 채운다**.**
6. **src**에서 문자배열을 뽑아내서 **4**에서 생성한 배열에 복사한다**.**
7. **4**에서 생성한 배열로 **String**을 생성해서 반환한다**.**

**\*/**

}

public static void main(String[] args) { String src = "12345"; System.out.println(fillZero(src,10)); System.out.println(fillZero(src,-1)); System.out.println(fillZero(src,3));

}

}

[연습문제]**/ch9/Exercise9\_6.java**

0000012345

123

[실행결과]

[9-7] 다음과 같이 정의된 메서드를 작성하고 테스트하시오.

메서드명 : contains

기 능 : 첫 번째 문자열(src)에 두 번째 문자열(target)이 포함되어 있는지 확인한다. 포함되어 있으면 true, 그렇지 않으면 false를 반환한다.

반환타입 : boolean 매개변수 : String src

String target

[Hint] String클래스의 indexOf()를 사용할 것

class Exercise9\_7 {

**/\***

**(1) contains**메서드를 작성하시오**.**

**\*/**

public static void main(String[] args) { System.out.println(contains("12345","23")); System.out.println(contains("12345","67"));

}

}

[연습문제]**/ch9/Exercise9\_7.java**

true false

[실행결과]

메서드명 : round

기 능 : 주어진 값을 반올림하여, 소수점 이하 n자리의 값을 반환한다.

예를 들어 n의 값이 3이면, 소수점 4째 자리에서 반올림하여 소수점 이하 3자리의 수를 반환한다.

반환타입 : double

매개변수 : double d - 변환할 값

int n - 반올림한 결과의 소수점 자리

[Hint] Math.round()와 Math.pow()를 이용하라.

class Exercise9\_8 {

**/\***

**(1) round**메서드를 작성하시오**.**

**\*/**

public static void main(String[] args) { System.out.println(round(3.1415,1)); System.out.println(round(3.1415,2)); System.out.println(round(3.1415,3)); System.out.println(round(3.1415,4)); System.out.println(round(3.1415,5));

}

}

[연습문제]**/ch9/Exercise9\_8.java**

3.1

3.14

3.142

3.1415

3.1415

[실행결과]

[9-9] 다음과 같이 정의된 메서드를 작성하고 테스트하시오.

메서드명 : delChar

기 능 : 주어진 문자열에서 금지된 문자들을 제거하여 반환한다. 반환타입 : String

매개변수 : String src - 변환할 문자열

String delCh - 제거할 문자들로 구성된 문자열

[힌트] StringBuffer와 String클래스의 charAt(int i)과 indexOf(int ch)를 사용하라.

}

}

System.out.println("(1!2@3^4~5)"+" -> "

+ delChar("(1!2@3^4~5)","~!@#$%^&\*()")); System.out.println("(1 2 3 4\t5)"+" -> "

+ delChar("(1 2 3 4\t5)"," \t"));

{

class Exercise9\_9 {

**/\***

**(1) delChar**메서드를 작성하시오**.**

**\*/**

public static void main(String[] args)

[연습문제]**/ch9/Exercise9\_9.java**

(1!2@3^4~5) -> 12345

(1 2 3 4 5) -> (12345)

[실행결과]

메서드명 : format

기 능 : 주어진 문자열을 지정된 크기의 문자열로 변환한다. 나머지 공간은 공백으로 채운다. 반환타입 : String

매개변수 : String str - 변환할 문자열

int length - 변환된 문자열의 길이

int alignment - 변환된 문자열의 정렬조건 (0:왼쪽 정렬, 1: 가운데 정렬, 2:오른쪽 정렬)

class Exercise9\_10

{

**/\***

**(1) format**메서드를 작성하시오**.**

1. **length**의 값이 **str**의 길이보다 작으면 **length**만큼만 잘라서 반환한다**.**
2. **1**의 경우가 아니면**, length**크기의 **char**배열을 생성하고 공백으로 채운다**.**
3. 정렬조건**(alignment)**의 값에 따라 문자열**(str)**을 복사할 위치를 결정한다**. (System.arraycopy()**사용**)**
4. **2**에서 생성한 **char**배열을 문자열로 만들어서 반환한다**.**

**\*/**

public static void main(String[] args) { String str = "가나다";

System.out.println(format(str,7,0)); // 왼쪽 정렬 System.out.println(format(str,7,1)); // 가운데 정렬 System.out.println(format(str,7,2)); // 오른쪽 정렬

}

}

[연습문제]**/ch9/Exercise9\_10.java**

가나다

가나다

가나다

[실행결과]

[9-11] 커맨드라인으로 2~9사이의 두 개의 숫자를 받아서 두 숫자사이의 구구단을 출력 하는 프로그램을 작성하시오.

예를 들어 3과 5를 입력하면 3단부터 5단까지 출력한다.

C:\jdk1.8\work\ch9>java Exercise9\_11 2 시작 단과 끝 단, 두 개의 정수를 입력해주세요. USAGE : GugudanTest 3 5

C:\jdk1.8\work\ch9>java Exercise9\_11 1 5

단의 범위는 2와 9사이의 값이어야 합니다. USAGE : GugudanTest 3 5

C:\jdk1.8\work\ch9>java Exercise9\_11 3 5 3\*1=3

3\*2=6

3\*3=9

3\*4=12

3\*5=15

3\*6=18

3\*7=21

3\*8=24

3\*9=27

4\*1=4

4\*2=8

4\*3=12

4\*4=16

4\*5=20

4\*6=24

4\*7=28

4\*8=32

4\*9=36

5\*1=5

5\*2=10

5\*3=15

5\*4=20

5\*5=25

5\*6=30

5\*7=35

5\*8=40

5\*9=45

[실행결과]

[주의] Math.random()을 사용하는 경우 실행결과와 다를 수 있음.

메서드명 : getRand

기 능 : 주어진 범위(from~to)에 속한 임의의 정수값을 반환한다. (양쪽 경계값 모두 범위에 포함)

from의 값이 to의 값보다 클 경우도 처리되어야 한다. 반환타입 : int

매개변수 : int from - 범위의 시작값

int to - 범위의 끝값

[Hint] Math.random()과 절대값을 반환하는 Math.abs(int a), 그리고 둘 중에 작은 값을 반환하는 Math.min(int a, int b)를 사용하라.

class Exercise9\_12

{

**/\***

**(1) getRand**메서드를 작성하시오**.**

**\*/**

public static void main(String[] args)

{

for(int i=0; i< 20; i++) System.out.print(getRand(1,-3)+",");

}

}

[연습문제]**/ch9/Exercise9\_12.java**

0,-1,1,0,-2,-2,1,1,-3,0,-1,1,1,1,0,-1,1,0,-1,-3,

[실행결과]

[9-13] 다음은 하나의 긴 문자열(source) 중에서 특정 문자열과 일치하는 문자열의 개수 를 구하는 예제이다. 빈 곳을 채워 예제를 완성하시오.

}

System.out.println("aa를 " + stringCount(src, "aa") +"개 찾았습니다.");

}

static int stringCount(String src, String key) { return stringCount(src, key, 0);

}

static int stringCount(String src, String key, int pos) { int count = 0;

int index = 0;

if (key == null || key.length() == 0) return 0;

**/\***

**(1)** 알맞은 코드를 넣어 완성하시오**.**

**\*/**

return count;

}

{

public class Exercise9\_13 {

public static void main(String[] args) String src = "aabbccAABBCCaa"; System.out.println(src);

[연습문제]**/ch9/Exercise9\_13.java**

aabbccAABBCCaa aa를 2개 찾았습니다.

[실행결과]

[9-14] 다음은 화면으로부터 전화번호의 일부를 입력받아 일치하는 전화번호를 주어진

문자열 배열에서 찾아서 출력하는 프로그램이다. 시오.

[Hint] Pattern, Matcher클래스를 사용할 것

알맞은 코드를 넣어 프로그램을 완성하

import java.util.\*; import java.util.regex.\*;

class Exercise9\_14 {

public static void main(String[] args) { String[] phoneNumArr = {

"012-3456-7890",

"099-2456-7980",

"088-2346-9870",

"013-3456-7890"

};

[연습문제]**/ch9/Exercise9\_14.java**

ArrayList list = new ArrayList(); Scanner s = new Scanner(System.in);

while(true) {

System.out.print(">>");

String input = s.nextLine().trim();

if(input.equals("")) { continue;

} else if(input.equalsIgnoreCase("Q")) { System.exit(0);

}

**/\***

**(1)** 알맞은 코드를 넣어 완성하시오**.**

**\*/**

if(list.size()>0) { System.out.println(list); list.clear();

} else {

System.out.println("일치하는 번호가 없습니다.");

}

}

} // main

}

>>

>>

>>asdf

일치하는 번호가 없습니다.

>>

>>

>>0

[012-3456-7890, 099-2456-7980, 088-2346-9870, 013-3456-7890]

>>234

[012-3456-7890, 088-2346-9870]

>>7890

[012-3456-7890, 013-3456-7890]

>>q

[실행결과]