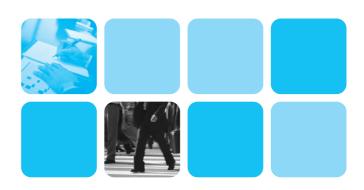


Ⅲ. 패키지의 활용

- 1. 패키지의 개요
- 2. java.lang 패키지
- 3. java.io 패키지

학습목표

- 자바 API 구조를 이해할 수 있다.
- 자바에서 제공하는 클래스를 효율적으로 사용할 수 있다.
- java.lang 패키지에 포함된 클래스의 메소드들을 사용할 수 있다.
- 자바에서 데이터를 입 · 출력하는 원리를 알 수 있다.
- java,io 패키지에서 제공하는 $\mathbf{u} \cdot \mathbf{\dot{z}}$ 력 관련 클래스를 이용하여 $\mathbf{u} \cdot \mathbf{\dot{z}}$ 력할 수 있다.



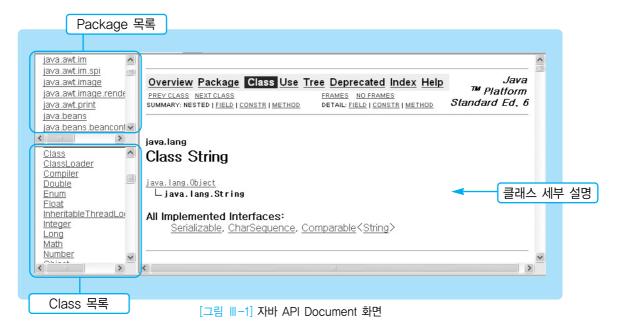
1

패키지의 개요

[01] 패키지의 개념과 분류

① 패키지의 개념

패키지(package)는 자바 개발 환경에서 제공되는 거대한 클래스 라이브러리로 관련된 클래스와 인터페이스의 모임을 말한다. 자바에서 하나의 패키지는 하나의 디렉토리와 대응되며, 각 디렉토리에는 관련된 클래스와 인터페이스가 저장되어 있다. 자바 패키지의 구조를 보기 위해 자바에서는 자바 API(Application Programming Interface)를 제공한다.



왼쪽 상단에는 자바가 지원하는 패키지 명이 있고, 하단에는 선택된 패키지에 해당하는 클래스와 인터페이스를 제공하고, 클래스 명을 클릭하면 오른쪽에 해당되는 클래스의 정보를 제공한다.

자바 API 문서는 http://java.sun.com/javase/downloads/index.jsp 사이트에서 Java SE 6(버전) Documentation 파일을 다운로드 받아서 압축을 푼 후 index.html 파일을 실행하면 오프라인에서도 사용할 수 있다.

② 패키지의 종류

다양한 패키지 중에서 많이 사용하는 주요 패키지를 살펴보면 다음과 같다.

② java.lang

자바 언어의 기초적인 사항을 정의한 클래스와 관련된 패키지로 Object 클래스, 문자열 관련 클래스, 시스템 관련 클래스, 멀티 스레드 클래스 등과 같이 자바 프로그램 작성에 필요한 기본적인 기능을 제공한다.

⊕ java.io

자바 입 · 출력 기능과 관련된 패키지로 파일이나 버퍼 등의 입 · 출력 관련 기능을 제공한다.

java.awt

GUI 작성과 관련된 패키지로 javax.swing 패키지와 함께 자바 GUI 어플리케이션 작성 시 필수적으로 사용되는 버튼, 텍스트 필드, 메뉴 등 관련 컴포넌트와 이벤트 기능을 제공하다.

@ java.applet

애플릿 작성에 필요한 기능을 모아 놓은 패키지이다.

java.net

자바의 네트워크에 관련된 패키지로 소켓과 관련된 기능을 제공한다.

javax.swing

java.awt 패키지와 더불어 자바 GUI 애플리케이션 작성과 관련된 패키지로 java.awt 보다 다양하고 융통성 있는 컴포넌트를 제공한다.

⊕ java.util

유틸리티성 기능과 관련된 패키지로 날짜 표현이나 여러 자료형을 하나로 취급하는 컬렉션과 관련된 기능을 제공한다.

iava.sql

데이터베이스 기능과 관련된 패키지로 데이터베이스 연결과 그 연결을 관리하며, 데이터베이스 명령 처리에 필요한 기능을 제공한다.

02 패키지의 이용

자바가 제공하는 클래스를 이용하려면 그 클래스의 패키지를 프로그램에서 명시해야 하는데 그 형식은 다음과 같다.

형식

```
import 패키지명.*;
import 패키지명.클래스명;
```

import 문은 프로그램의 맨 앞에 기술해야 한다. 만약 String 클래스를 사용하기 위해서는 import java.lang.* 또는 import java.lang.String과 같이 기술해야 한다.

```
예제 III-1 PckageExam.java
1
      import java.lang.*;
                                                                              \bigcirc
2
      class PckageExam{
3
          public static void main(String[] args){
4
             String strData ="I Love You";
5
             System.out.println("strData = " + strData);
6
          }
7
      }
```

:: 실행 결과

strData = I Love You

:: 설명

위 프로그램에서 ① 부분을 사용하지 않아도 정상적으로 실행된다. 이는 java.lang 패키지에 속하는 클래스는 디폴트로 이미 정의되어 있기 때문에 import 문을 이용하지 않아도 되기 때문이다. String 클래스는 java.lang 패키지에 속하는 클래스이다.

import를 사용하지 않아도 자바에서 제공하는 클래스를 사용할 수 있는데 이때는 클래스의 이름을 단독으로 사용하지 않고 패키지 이름을 포함한 전체 이름을 사용해야 한다.

예를 들어 java.util 패키지의 Date 클래스를 사용하고 싶을 경우 다음과 같이 사용한다.

java.util.Date date = new java.util.Date();

하지만 이는 클래스 이름이 너무 길어 사용하기가 불편하므로 import문을 사용하여 미리 사용하고자 하는 클래스를 포함시키는 것이 좋다.

2

java lang 패키지

java.lang 패키지에 속한 클래스들은 특별히 import하지 않아도 사용할 수 있으며, java.lang.Object 클래스는 모든 자바 클래스의 최상위 클래스로 아무런 클래스를 상속받지 않는 클래스를 정의해도 모든 클래스는 Object 클래스를 자동으로 상속받는다. Object 클래스의 중요 메소드는 equals() 메소드와 toString() 메소드로 대부분의 클래스에서는 재정의하여 사용한다.

01 String 클래스

java, lang. String 클래스는 문자열 처리를 위한 클래스로 String 클래스로 생성된 문자열은 한번 생성되면 변경할 수 없는 문자열이다.

① 객체 생성 방법

String 클래스 객체를 생성하는 방법에는 문자열 상수를 이용하는 방법과 String 클래스 생성자를 이용하는 방법이 있다. 두 가지 방법에는 분명한 차이가 있다. 문자열 상수를 이용하여 생성할 경우에는 자바 가상 머신은 내부 문자열 저장소에 같은 내용의 String 객체가 있는지 확인하여 있을 경우 해당 객체의 레퍼런스만을 리턴한다. String 클래스 생성자를 이용하여 생성할 경우에는 매번 새로운 객체가 생성된다.



[그림 Ⅲ-2] 문자열 객체 생성 방법

পুৰা III-2 EqualExam.java

```
class EqualExam{
1
2
              public static void main(String[] args){
                     String str1 = "sunrin";
3
                     String str2 = "sunrin";
4
                     String str3 = new String("highschool");
5
                     String str4 = new String("highschool");
6
7
                     if(str1 == str2 ) System.out.println("주소 같다");
8
                      else System.out.println("주소 다르다");
9
10
                     if(str3 == str4 ) System.out.println("주소 같다");
                      else System.out.println("주소 다르다");
11
                     if(str1.equals(str2) ) System.out.println("내용 같다");
12
13
                      else System.out.println("내용 다르다");
                     if(str3.equals(str4)) System.out.println("내용 같다");
14
                      else System.out.println("내용 다르다");
15
             }
16
17
      }
```

∷ 실행 결과

주소 같다 주소 다르다 내용 같다 내용 같다

:: 설명

== 연산자는 객체의 주소를 비교하는 연산자이므로 strl과 str2를 비교시 "주소 같다"를 출력하고, str3과 str4를 비교시 "주소 다르다"를 출력한다. equal() 메소드는 문자열의 내용이 같은지를 비교하는 연산자이므로 모두 "내용 같다"라고 출력한다.

② 대표적인 메소드

② length() 메소드

String 클래스의 문자열의 길이를 반환하는 메소드이다.

StringBuffer 클래스의 length() 메소드와 같이 클래스 객체에 저장된 문자열의 길이를 반환하며, 이때 공백도 하나의 문자로 인식한다.

형식

public int length()

예제 III-3 LengthExam.java

```
class LengthExam{
public static void main(String[] args){
String strData1 ="I Love You";
System.out.println("strData1.length()= " + strData1.length());
}

| System.out.println("strData1.length()= " + strData1.length());
| System.out.println("strData1.length()= " + strData1.length()= " + strD
```

∷ 실행 결과

strData1.length()=10

:: 설명

공백도 하나의 문자로 인식한다.

(나) charAt() 메소드

문자열의 특정 위치의 문자를 반환하는 메소드이다.

형식

public char charAt(int index) // index의 범위: 0 ~ length() - 1

예제 III-4 CharAtExam.java

```
class CharAtExam {
1
2
      public static void main(String[] args) {
3
          String str = "I love Korea";
          for( int i=0; i(str.length(); i++){
4
5
              System.out.print(str.charAt(i));
6
7
          System.out.println();
8
      }
9
```

∷ 실행 결과

I Love Korea

© equals() 메소드

문자열의 내용이 같은지를 비교하는 메소드로 객체 내의 문자열이 같으면 true를 반환하고 거짓이면 false를 반환하는 메소드이다.

형식

public boolean equals(Object obj)

পামা III-5 StringEqualExam.java

```
class StringEqualExam{
1
2
             public static void main(String[] args){
                     String R1 = new String("JAVA");
                     String R2 = new String("JAVA");
4
5
                     String R3 = new String("java");
                     System.out.println("R1 == R2 \rightarrow" + (R1==R2));
6
                     System.out.println("R1.equals(R2) -->" + (R1.equals(R2)));
                     System.out.println("R1 == R3 \rightarrow" + (R1==R3));
8
                     System.out.println("R1.equals(R3) -->" + (R1.equals(R3)));
9
             }
10
      }
11
```

∷ 실행 결과

```
R1 == R2 \longrightarrow false
R1.equals(R2) \longrightarrow true
R1 == R3 \longrightarrow false
R1.equals(R3) \longrightarrow false
```

:: 설명

R1, R2, R3 모두 주소가 다르기 때문에 "==" 연산자 결과는 모두 false이고, 자바는 대소문자를 구별하기 때문에 equals() 비교시 "JAVA"와 "Java"는 다른 문자열로 인식한다.

পামা III-6 EqualsExam.java

```
1
      class EqualsExam {
2
       public static void main(String[] args){
3
          String str1 = "Sunrin";
          String str2 = "sunrin";
4
5
          String str3 = new String("highschool");
6
          String str4 = new String("highschool");
7
8
          if(str1.equals(str2))System.out.println("같다");
9
             else System.out.println("다르다");
10
          if(str3.equals(str4))System.out.println("같다");
             else System.out.println("다르다");
11
12
          }
13
      }
```

∷ 실행 결과

다르다 같다

라 valueOf() 메소드

매개 변수의 값을 String 형으로 변환하는 메소드이다.

형식

```
public static String valueOf(매개 변수)
```

매개 변수에는 boolean, char, char[], double, float, int, long, Object 등이 올 수 있다.

পুৰা III–7 ValueofExam.java class ValueofExam { 1 2 public static void main(String[] args){ 3 char ch = 'A'; 4 String strData1 ="A"; 5 String strData2; strData2 = String.valueOf(ch); // 🗇 6 7 if(strData1.equals(strData2)) 8 System.out.println("같음"); 9 else System.out.println("다름"); 10 } 11 } 12 13

∷ 실행 결과

같음

:: 설명

⊙에 의해 StrData2에는 "A"가 저장된다.

@ trim() 메소드

문자열의 앞뒤 공백을 제거하는 메소드이다.

형식

public String trim()

পামা III−8 TrimExam.java

```
class TrimExam{
1
      public static void main(String[] args){
2
3
          String strData1 =" I Love You ";
          String strData2;
4
          strData2 = strData1.trim();
5
6
          System.out.println("strData2.trim() -> "
                                                                strData2);
7
          if( strData1.equals( strData2 ))
8
             System.out.println("같음");
9
          else System.out.println("다름");
10
          }
      }
11
```

:: 실행 결과

StrData2.trim() -> I LoveYou 다름

🕕 substring() 메소드

문자열의 일부를 반환하는 메소드이다.

형식 1

public String substring(int start)

start부터 끝까지 문자열을 반환하는 메소드이다. 문자열의 위치는 0부터 시작한다.

형식 2

public String substring(int start, int end)

start부터 end-1까지 문자열을 리턴하는 메소드이다.

পাম।।।-9 SubstringExam.java

```
class SubstringExam{
1
2
       public static void main(String[] args){
3
           String strData1 ="I Love You";
4
           String strData3;
5
           strData3 = strData1.substring(7);
           System.out.println("strData3.substring(7) \rightarrow " + strData3);
6
           strData3 = strData1.substring(2, 5);
7
          System.out.println("strData3.substring(2, 5) \rightarrow" + strData3);
8
        }
9
       }
10
```

∷ 실행 결과

StrData3.substring(7) \rightarrow You StrData3.substring(2, 5) \rightarrow Lov

참고사항

- + 연산자
- ① 문자 + 정수 -> 정수
- ② 스트링 + (정수, 실수, 문자) -> 스트링

[02] StringBuffer 클래스

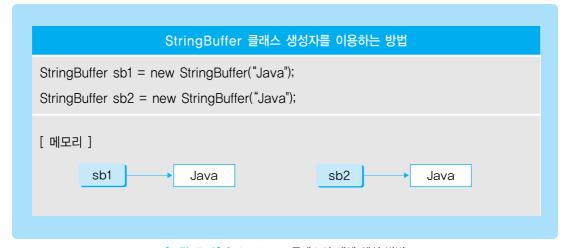
Java.lang.StringBuffer 클래스는 문자열 처리를 위한 클래스로 StringBuffer 클래스로 생성한 문자열은 내용 변경이 가능하다. 따라서 동적으로 문자열의 내용을 바꾸거나 위치를 조정하는 등의 일을 할 때 사용하는 클래스이다.

① 객체 생성 방법

StringBuffer 클래스 생성자를 이용하여 객체를 생성하는 것으로 형식은 다음과 같다.

형식

StringBuffer 객체 변수 = new StringBuffer(String str);



[그림 Ⅲ-3] StringBuffer 클래스의 객체 생성 방법

② 대표적인 메소드

② append() 메소드

문자열을 추가하는 메소드이다.

형식

```
public StringBuffer append( 매개 변수 )
```

매개 변수에는 String, StringBuffer, boolean, char, char[], double, float, int, long, Object 등이 올 수 있다.

পাম III–10 AppendExam.java

```
class AppendExam {
1
2
      public static void main(String[] args) {
          StringBuffer sb1 = new StringBuffer(" Java Programming");
3
          StringBuffer sb2 = new StringBuffer(" Project");
4
5
          StringBuffer str1 = null;
          StringBuffer str2 = null;
6
7
          str1 = sb1.append(" Project");
                                                               //①
          str2 = sb1.append(sb2);
                                                               //©
8
9
          System.out.println("sb1 = " + sb1);
          System.out.println("sb2 = " + sb2);
10
          System.out.println("str1 = " + str1);
11
          System.out.println("str2 = " + str2);
12
13
          System.out.println();
```

∷ 실행 결과

```
sb1 = Java Programming Project Project
sb2 = Project
str1 = Java Programming Project Project
str2 = Java Programming Project Project
```

:: 설명

③과 ⓒ 명령에서 객체 변수 sb1 에 "Project"가 추가된 후 str1과 str2에 저장된다.

마 replace() 메소드

문자열을 치환하는 메소드이다.

형식

public StringBuffer replace(int start, int end, String str)

start ~ end-1의 문자열을 str로 치환하는 메소드이다.

পামা III–11 ReplaceExam.java

```
class ReplaceExam {
public static void main(String[] args){
StringBuffer sb1 = new StringBuffer("java Programming");
StringBuffer sb2 = null;
sb2 = sb1.replace(0,4,"JAVA");
System.out.println("sb2 = " + sb2);
}

8
```

:: 실행 결과

sb2 = JAVA programming

© reverse() 메소드

문자열을 역으로(거꾸로) 반환하는 메소드이다.

형식

public StringBuffer reverse()

পামা III–12 ReverseExam.java

```
class ReverseExam{
1
      public static void main(String[] args){
2
3
         StringBuffer sb1 = new StringBuffer("java Programming");
         StringBuffer sb2 = null;
4
         sb2 = sb1.reverse();
5
         System.out.println("sb2 = " + sb2);
6
7
         }
8
      }
```

:: 실행 결과

sb2 = gnimmargorP avaj

@ toString() 메소드

String형으로 반환하는 메소드이다.

형식

public String toString()

পুৰু III–13 ToStringExam.java

```
class ToStringExam{
public static void main(String[] args){
StringBuffer sb1 = new StringBuffer("java Programming");
String str = null;
str = sb1.toString();
System.out.println("str.toUpperCase() = " + str.toUpperCase());
}

}
```

∷ 실행 결과

str.toUpperCase() = JAVA PROGRAMMING

:: 설명

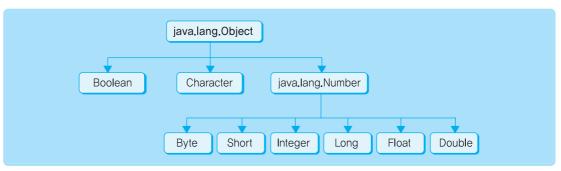
toUpperCase() 메소드는 String 클래스 객체의 모든 문자를 대문자로 바꾼다. 그런데 sb1은 StringBuffer 클래스 객체이므로 toUpperCase() 메소드를 사용할 수 없다. 따라서 toSting() 메소드로 StringBuffer 클래스 객체를 String 클래스 객체로 변환하여 처리하였다.

03 Wrapper 클래스

Wrapper 클래스는 기본 데이터 타입을 클래스 타입으로 포장하는 클래스로 각각의 기본 데이터 타입에 해당하는 Wrapper 형식의 클래스가 하나씩 존재한다. 기본형을 위해 Byte, Short, Integer, Long, Float, Double, Character, Boolean을 제공하고 문자열을 다루기 위해 String, StringBuffer를 제공하고 큰 숫자를 다루기 위해 BigInteger와 BigDecimal을 제공한다.

기본 데이터형	Wrapper 클래스	기본 데이터형	Wrapper 클래스
boolean	Boolean	int	Integer
byte	Byte	long	Long
char	Character	float	Float
short	Short	double	Double

[표 Ⅲ-1] Wrapper 클래스



[그림 Ⅲ-4] Wrapper 클래스 구조

① 대표적인 메소드

① equals() 메소드

Integer, Long, Double, Character 등의 Wrapper 클래스 객체가 가지고 있는 기본 적인 데이터형의 값이 같으면 true를 그렇지 않으면 false를 반환하는 메소드이다.

পামা III–14 IntegerEqualExam.java 1 class IntegerEqualExam{ 2 public static void main(String[] args){ 3 Integer in1= new Integer(100); 4 Integer in2= new Integer(100); 5 Integer in3= new Integer(200); 6 System.out.println("in1.equals(in2) -->" + (in1.equals(in2))); System.out.println("in1.equals(in3) -->" + (in1.equals(in3))); 7 } 8 9 }

:: 실행 결과

```
in1.equals(in2) --> true in1.equals(in3) --> false
```


기본 데이터를 String형으로 반환하는 메소드이다.

형식

public static String toString(기본 데이터형)

পামা III–15 WrapExam.java

```
1
      class WrapExam {
          public static void main(String[] args) {
2
             Integer wrapi = new Integer(100);
3
             Double wrapd = new Double(55.7);
4
             int i = 100;
5
             double d = 55.7;
6
             double dl;
7
             dl = d+i;
8
             System.out.println(i + d);
9
             System.out.println(Integer.toString(i) + Double.toString(d));
10
             System.out.println(wrapi.toString() + wrapd.toString());
11
          }
12
      }
```

프로그래밍 (JAVA)

:: 실행 결과

155.7 10055.7 10055.7

⑤ valueOf() 메소드

String형 데이터를 반환형으로 선언된 Wrapper형 데이터로 반환하는 메소드이다.

형식

public static Wrapper형 valueOf(String s)

Wrapper형에는 Byte, Integer, Long, Float, Double 등이 올 수 있다.

পামা III–16 ValueofExam.java

```
class ValueofExam{
1
2
          public static void main(String[] args) {
             String stri="123456";
3
             String strd="123.56";
4
             System.out.println(Integer.valueOf(stri));
5
6
             System.out.println(Double.valueOf(strd));
             System.out.println(Integer.valueOf(stri)+Double.valueOf(strd));
7
         }
8
9
      }
```

∷ 실행 결과

123456 123.56 123579.56

e parseInt(), parseFloat(), parseDouble() 메소드

String형 데이터를 기본 데이터형으로 반환하는 메소드이다. parseInt()는 int형으로 변환하고, ParseFloat()는 float형으로 변환하고, parseDouble()는 double형으로 변환한다. 이들은 숫자 형태의 문자열 데이터를 연산에 이용하고자 할 때 주로 사용하는 메소드이다.

형식

```
public static int parseInt(String s)
public static float parseFloat(String s)
public static double parseDouble(String s)
```

পামা III–17 parseDataExam.java

```
1
      public class parseDataExam{
2
          public static void main(String args[]){
              String a="10", b="20";
3
                                                              //(¬)
             int sum=0;
4
5
             float average = 0;
              System.out.println("input data: "+a+" "+b);
6
              int c = Integer.parseInt(a);
                                                              //(L)
             int d = Integer.parseInt(b);
                                                              //©
8
              sum = c+d;
9
              average = sum/2;
10
              System.out.println(a+" + "+b+ " = " + sum);
11
              System.out.println("평균 = "+average);
12
13
          }
14
      }
```

:: 실행 결과

input data : 10 20 10 + 20 = 30 평균 = 15.0

:: 설명

つ에서 a와 b변수에 저장된 값은 String형이므로 연산에 이용될 수 없다. 숫자 형태의 String형의 값을 연산에 이용하려면 정수형이나 실수형으로 변환해야 하는데 ▷과 ⓒ에 의해 String형의 값을 int형으로 변환하여 c와 d변수에 저장하였다.

@ intValue(), floatValue(), doubleValue() 메소드

Wrapper형 데이터를 기본 데이터형으로 반환하는 메소드이다. 예를 들어 public int intValue()일 경우 Integer 형의 값을 int형으로 반환해 준다. intValue()은 Integer형의 값을 int형으로 반환하고, floatValue()는 Float형의 값을 float형으로 반환하고, doubleValue()는 Double형의 값을 double형으로 반환한다.

형식

```
public int intValue( )
public float floatValue( )
public double doubleValue( )
```

পুমা III–18 dataValueExam.java

```
1
      import java.util.*;
2
      class dataValueExam{
3
          public static void main(String[] args){
              String line;
4
              String tmpStr = "hong 용산구 10.5 20.5";
5
6
              StringTokenizer parse = new StringTokenizer(tmpStr);
7
              String name;
              String address;
8
9
              double math;
10
              double enalish;
11
              double total;
12
              double avg;
13
              name= parse.nextToken();
              address= parse.nextToken();
14
              math= Double.valueOf( parse.nextToken() ).doubleValue();
15
              english= Double.valueOf( parse.nextToken() ).doubleValue();
16
              total= math + english;
17
18
              avg= total /2.0;
19
              System.out.println( name + "\t" + address + "\t" + math + "\t" +
                                 english+"\t" +total+ "\t" + avg);
         }
21
      }
```

:: 실행 결과

hong 용산구 10.5 20.5 31.0 15.5

참고사항

StringTokenizer 클래스

- java.util 패키지에 포함된 클래스
- 파싱 : 하나의 문자열을 여러 개의 문자열로 나누는 것
- 토큰 : 나누는 단위가 되는 문자열
- 생성자 : StringTokenizer st = new StringTokenizer(파싱하려는 문자열, 토큰)
- 주요 메소드
 - ① public String nextToken(): 파싱한 문자열을 차례대로 넘겨줌
 - © public boolean hasMoreTokens(): nextToken() 메소드를 실행한 뒤에 토큰이 더 남아있는지를 확인
 - © public int countTokens(): 파싱된 문자열이 모두 몇 개인지 넘겨줌

문제 III-1

String 형의 데이터 10과 20을 정수와 실수형으로 변환하여 덧셈을 하는 프로그램을 작성하여라.

[처리 조건]

- 1. parseInt(), parseDouble() 메소드 이용
- 2. intValue(). doubleValue() 메소드 이용

:: 실행 결과

data1 = 10, data2 = 20

>> parseInt()와 parseDouble()메소드 이용

data1 + data2 = 30

data1 + data2 = 30.0

》〉intValue()와 doubleValue()메소드 이용

data1 + data2 = 30

data1 + data2 = 30.0

[**04**] Math 클래스

Math 클래스는 수학 함수와 이에 필요한 상수들을 제공하는 클래스이다. Math 클래스의 모든 메소드는 static 형으로 정의되어 있어 상속이나 오버라이딩 할 수 없는 메소드들이며, Math 클래스는 생성자를 제공하지 않아 객체를 생성할 수 없다. 따라서, 객체 생성 없이 클래스 이름으로 멤버 변수나 메소드를 접근한다.

Math 클래스의 멤버 변수는 Math.E 와 Math.PI가 있으며 다음과 같이 정의되어 있다.

- public static final double E는 자연수 E를 나타내는 상수이다.
- public static final double PI는 원주율 π를 나타내는 상수이다.

① 주요 메소드

⑦ abs() 메소드

절대값을 반환하는 메소드이다.

형식

public static double abs(숫자 데이터)

পাম III–19 AbsExam.java

```
class AbsExam {
1
2
         public static void main(String[] args){
             System.out.println(Math.abs(25));
3
             System.out.println(Math.abs(-25));
4
5
             System.out.println(Math.abs(25.5));
             System.out.println(Math.abs(-25.5));
6
7
         }
      }
8
```

∷ 실행 결과

25 25 25.5 25.5

① round() 메소드

매개 변수 값과 같거나 가장 가까운 정수를 반환하는 메소드이다.

형식

public static int round(float a)

পামা।।–20 RoundExam.java

```
class RoundExam{
1
2
          public static void main(String[] args){
              System.out.println(Math.round(25));
3
              System.out.println(Math.round(25.3));
4
5
              System.out.println(Math.round(25.5));
             System.out.println(Math.round(-25.4));
6
7
              System.out.println(Math.round(-25.5));
             System.out.println(Math.round(-25.6));
8
         }
9
      }
10
```

∷ 실행 결과

25 25 26 -25 -25 -26

⑤ pow() 메소드

거듭제곱 값을 반환하는 메소드이다.

형식

public static double pow(double a, double b)

PowerExam.java 예제 III-21 class PowerExam { 1 2 public static void main(String[] args){ 3 System.out.println(Math.pow(3, 2)); 4 System.out.println(Math.pow(3.0f, 2.0f)); 5 System.out.println(Math.pow(5.0, 2)); System.out.println(Math.pow(-5.0, 3.0)); 6 7 }

∷ 실행 결과

}

9.0 9.0

25.0

-125.0

8

② sqrt() 메소드

제곱근을 반환하는 메소드이다.

형식

7

8

}

public static double sqrt(double a)

음수일 경우 NaN(Not a Number)를 반환한다.

o||X|| ||-22SquareExam.java1class SquareExam{2public static void main(String[] args){3System.out.println(Math.sqrt(9));4System.out.println(Math.sqrt(9.0f));5System.out.println(Math.sqrt(9.0));6System.out.println(Math.sqrt(-9.0));

프로그래밍 (JAVA)

∷ 실행 결과

3.0

3.0

3.0 NaN

® random() 메소드

0.0보다 크거나 같고 1.0보다 작은 난수를 반환하는 메소드이다.

형식

public static double random()

int 값을 얻기 위해서는 int 형으로 형 변환해야 한다.

পাম III–23 RandExam.java

```
class RandExam {
1
2
          public static void main(String[] args){
3
               System.out.println(Math.random());
4
               System.out.println(Math.random());
               System.out.println((int)Math.random()* 5);
6
               System.out.println((int)(Math.random()* 5));
               System.out.println((int)(Math.random()* 10));
7
               System.out.println((int)(Math.random()*100));
8
9
          }
10
      }
```

∷ 실행 결과

0.6252399581098143

0.38896352535722756

0

4

6

80

:: 설명

0~9까지의 난수를 발생시키고자 할 경우 수식은 (int)(Marh, random()*10과 같다. 1~10까지의 난수를 발생시켜려면 (int)(Math,random() * 10)+1과 같은 수식을 작성한다.

참고사항 Random 클래스

임의의 값을 발생시켜 주는 클래스로 Random 클래스를 한 번 생성하면 여러 번에 걸쳐서 임의 값을 얻을 수 있다.

문제 III-2

피타고라스 정리를 이용하여 두 점((0, 0), (3, 4)) 사이의 거리를 구하는 프로그램을 작성하 여라.

∷ 실행 결과

거리 = 5

문제 III-3

랜덤하게 발생하는 0에서 100 사이의 임의의 정수를 알아 맞추는 프로그램을 작성하여라. (기회는 5번)

∷ 실행 결과

숫자 입력(0~99)? 50 >>> 너무 작습니다.(1회) 숫자 입력(0~99)? 70 >>> 너무 큽니다.(2회) 숫자 입력(0~99)? 60 >>> 너무 작습니다.(3회) 숫자 입력(0~99)? 65 ***** 축하합니다. *****

3

iava, io 패키지

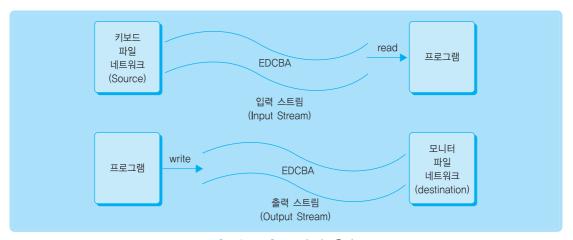
[**01**] 입·출력의 개요

자바 프로그램에서 데이터를 입·출력하기 위해서는 스트림을 이용한다. 스트림(stream)이 우리나라 말로 "시내, 개울, 흐름"을 나타내는 것과 같이 스트림은 데이터의 흐름을 의미한다. 시냇물은 한 방향으로만 흐른다. 만약 위 아래로 흐르는 시냇물을 원한다면 두 개의 시내가 있어야 한다. 같은 원리로 입력 장치로부터 데이터를 읽기 위한 스트림이 있어야 하고, 출력 장치에 데이터를 출력하기 위한 스트림이 있어야 한다. 이와 같이 입·출력 장치로 데이터들이 들어오고 나가는 통로를 스트림이라 하고, 이 스트림은 단방향이다.

자바 언어에서 데이터를 입·출력할 때 생각해야 할 것은 입·출력 단위와 입·출력 장치이다. 입·출력 단위는 크게 바이트 단위와 문자 단위로 구분할 수 있고, 입·출력 장치는 키보드, 모니터, 메모리, 파일, 네트워크 등이다. java.io 패키지를 통해서 입·출력 클래스를 제공하며 기본적인 입·출력 방식은 스트림을 기반으로 한다. 각 장치마다 연결할 수 있는 각각의 스트림이 존재하고 자바에서는 해당 스트림을 처리하기 위한 클래스들이 준비되어 있다.

스트림을 이용하여 데이터를 입·출력하기 위해서는 첫째, 스트림을 생성하고, 둘째, 스트림 으로부터 데이터를 읽고 쓴다.

스트림 생성은 해당 클래스를 객체 생성함으로써 가능하고, 각 클래스가 제공하는 메소드를 이용하여 데이터를 읽고 쓰면 된다. 스트림은 단방향이므로 입력용 스트림과 출력용 스트림을 모두 생성해야 한다.



[그림 Ⅲ-5] 스트림 입ㆍ출력

① 표준 입·출력

자바에서는 키보드와 모니터를 이용한 표준 입·출력 방식을 제공한다.

표준 입·출력은 java.lang.System 클래스에서 제공하는 in, out, err 멤버 변수에 의해 표준 입력, 표준 출력, 표준 에러를 제공한다. 표준 입·출력 멤버 변수의 선언 형식이 [표 Ⅲ-2]와 같으므로 in은 InputStream 클래스가 가진 메소드를 사용할 수 있고, err와 out은 PrintStream 클래스가 가진 메소드를 사용할 수 있다.

```
public static InputStream in // The "standard" input stream.

public static PrintStream out // The "standard" output stream.

public static PrintStream err // The "standard" error output stream.
```

표준 입·출력 스트림은 미리 생성된 스트림이기 때문에 사용자가 직접 생성할 필요 없이 다음과 같이 사용하다.

⑦ 표준 입력

키보드로 데이터를 입력하는 것으로 방법은 다음과 같다.

System.in.read();

(l) 표준 출력

화면으로 메시지를 출력하는 것으로 방법은 다음과 같다.

```
• System.out.println( ); • System.out.print( );
```

따 표준 에러

화면으로 오류를 출력하는 것으로 방법은 다음과 같다.

```
• System.err.println( ); • System.err.print( );
```

পামা III-24 Echo1Exam.java

```
import java.io.*;
public class Echo1Exam{
  public static void main(String[] args){
  int bt;
  try{
    bt = System.in.read();
}
```

프로그래밍 (JAVA)

:: 실행 결과

T

참고사항 try~ catch문(p151 참고)

InputStream의 read() 메소드의 정의를 보면 public int read(byte[] b) throws IOException 와 같이 선언되어 있다. 정의된 메소드에 throws IOException가 있을 경우 예외 처리를 이를 호출한 쪽에서 처리해 주어야 하기 때문에 try ~ catch 문을 사용하여야 한다.

[예제 Ⅲ-24] 프로그램은 한 문자를 입력하고 종료한다. 위 프로그램을 수정하여 출력 결과와 같이 여러 문자를 입력하여 출력하도록 수정하여 보자.

```
পামা III-25 Echo2Exam.java
1
      import java.io.*;
2
      public class Echo2Exam{
3
          public static void main(String[] args){
4
             int bt;
5
             try{
                 while(( bt = System.in.read()) != −1){ // Ctrl + Z 입력하면 종료
6
7
                    System.out.print((char)bt);
8
9
               }catch(IOException e){
             System.err.println(e);
10
11
           }
         }
12
      }
13
```

∷ 실행 결과

Test Test ^Z

② 자바의 입 · 출력 방식

자바 입·출력 방식은 크게 바이트 스트림을 이용한 입·출력과 문자 스트림을 이용한 입·출력으로 분류할 수 있다.

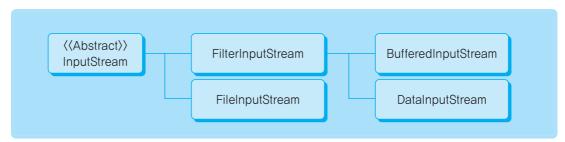
바이트 스트림은 8bit의 바이트를 읽고 쓰기 위한 스트림이고, 문자 스트림은 16bit의 문자 나 문자열을 읽고 쓰기 위한 스트림이다.

바이트 단위 입·출력 스트림의 경우 상위 클래스가 InputStream과 OutputStream이고, 문자 단위 입·출력 스트림의 경우 상위 클래스가 Reader와 Writer이다.

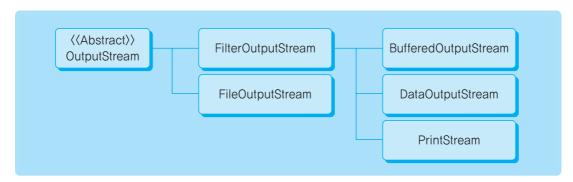
InputStream과 OutputStream, Reader와 Writer 클래스는 모두 추상 클래스이기 때문에 이 클래스의 주요 메소드들은 하위 클래스에서 오버라이딩하여 사용하여야 한다.

② 바이트 단위 스트림 계층 구조

바이트 단위 스트림의 상위 클래스는 InputStream, OutputStream이고, 주요 클래스는 다음과 같다.



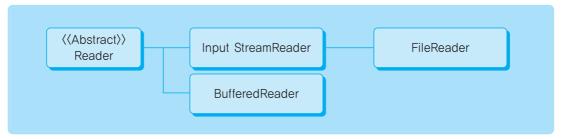
[그림 Ⅲ-6] InputStream 클래스의 계층 구조



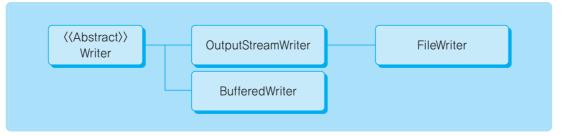
[그림 Ⅲ-7] OutputStream 클래스의 계층 구조

(l) 문자 단위 스트림 구조

문자 단위 스트림의 상위 클래스는 Reader, Writer이고, Reader 클래스와 Writer 클래스는 문자 입·출력을 담당하는 추상 클래스로 다른 문자 입·출력 클래스들에 대한 상위 클래스이다.



[그림 Ⅲ-8] Reader 클래스 계층 구조



[그림 Ⅲ-9] Writer 클래스 계층 구조

참고사항 바이트 단위 스트림 클래스와 문자 단위 스트림 클래스의 비교

각 바이트 단위 $\mathbf{U} \cdot \mathbf{\check{z}}$ 력 클래스에는 대응하는 문자 단위 $\mathbf{U} \cdot \mathbf{\check{z}}$ 력 클래스들이 있다. 두 클래스에 서 사용하는 메소드는 거의 동일하고 단지 처리하는 데이터를 바이트 단위로 처리하느냐 문자 단위로 처리하느냐의 차이이다.

[# I	11-21	인려	큭래人	НΙШ

바이트 입력 클래스	특징	문자 입력 클래스
InputStream	기본 입력	Reader
FileInputStream	파일 입력	FileReader
BufferedInputStream	버퍼를 이용한 입력	BufferedReader
DataInputStream	자바 기본형 데이터 입력	없음

[표 Ⅲ-3] 출력 클래스 비교

바이트 입력 클래스	특징	문자 입력 클래스
OutputStream	기본 출력	Writer
FileOutputStream	파일 출력	FileWriter
BufferedOutputStream	버퍼를 이용한 출력	BufferedWriter
DataOutputStream	자바 기본형 데이터 출력	없음
PrintStream	표준 출력 장치로 출력	PrintWriter

[02] 바이트 스트림 입·출력

입·출력 스트림은 기본적으로 바이트를 기반으로 순차적으로 입력하거나 순차적으로 출력하는 것으로 입·출력을 처리한다. 바이트 스트림(Byte Stream)을 이용한 입·출력을 하기 위해서는 InputStream과 OutputStream 클래스의 하위 클래스들을 이용하고, 각 하위 클래스들은 InputStream과 OutputStream의 주요 메소드를 오버라이딩하여 byte 단위의 읽기와 쓰기가가능하도록 read() 메소드와 write() 메소드를 지원한다.

[표 Ⅲ-4] InputStream 주요 메소드

메소드	설명
abstract int read()	• 스트림으로부터 1바이트를 읽는 메소드이다. • 스트림의 끝은 -1을 반환한다.
int read(byte[] b)	바이트 배열에 데이터를 읽어들이는 메소드이다.한 번에 여러 데이터를 읽어들인다.
void close()	• 해당 스트림과 관련된 모든 자원을 해제한다.

[표 Ⅲ-5] OutputStream 주요 메소드

메소드	설명
abstract void write(int b)	• int형 정수를 인자로 받아 출력한다. • byte형으로 자동 캐스팅된 후 출력한다.
void write(byte[] b)	• 버퍼에 들어있는 바이트 배열만큼 출력한다.
void flush()	• 버퍼에 쌓여있는 데이터를 출력시키고자 할 때 사용한다.
void close()	• 출력한 후 해당 스트림과 관련된 모든 자원을 해제한다.

① 1바이트 문자 입ㆍ출력

다음은 한 바이트씩 키보드로 문자를 입력하여 출력하는 프로그램의 예이다.

পামা III–26 InputExam.java 1 import java.io.*; 2 public class InputExam{ public static void streamTest(InputStream is){ 3 trv { 4 while (true) { int i = is.read(); 6 // InputStream에서 한 바이트를 읽는다.(int 형태임) if (i == -1) break; 7 // 마지막에 Ctrl + Z 입력 char c = (char) i;8

프로그래밍 (JAVA)

```
9
                  System.out.print(c);
               }
10
             } catch (IOException e) {
11
12
               System.err.println(e);
13
14
             System.out.println();
15
16
          public static void main(String args[])
17
                    streamTest(System.in);
18
                                                            //InputExam.streamTest(System.in);
19
          }
      }
```

::실행 결과

Test Test 홍길동 ??±æ?¿

② 한글 한 문자 입ㆍ출력

পামা III–27 ReadFromSysExam.java

}

[예제 Ⅲ-26]은 한글 처리가 불가능하다. 이를 해결하기 위해서는 다음과 같이 byte[] 형태의 read() 메소드를 사용해야 한다.

import java.io.*; 1 2 class ReadFromSysExam{ public static void main(String[] args){ 3 byte[] b = new byte[1024];//b : byte형 배열 객체 변수 4 int len = 0; 5 try{ 6 7 len = System.in.read(b); 8 System.out.write(b, 0, len); 9 }catch(IOException ioe){ System.out.println("입력 실패"); 10 11 }

12 13

}



홍길동

홍길동

③ BufferedInputStream 클래스 / BufferedOutputStream 클래스

BufferedInputStream 클래스와 BufferedOutputStream 클래스는 입·출력 스트림에 버퍼를 추가한 필터 스트림 클래스이다. 즉, 데이터 입·출력 시 버퍼를 이용하여 처리하는 것이다. 버퍼가 추가될 경우 스트림에서 1바이트씩 처리하지 않고 한번에 많은 데이터를 미리 버퍼에 놓고 바이트 블록 단위로 입·출력을 하기 때문에 데이터 처리 속도를 향상시켜 준다. 버퍼 입·출력시 중요한 메소드는 flush() 메소드이다. flush() 메소드는 버퍼에 저장된데이터를 읽어서 처리하도록 하는 메소드로 close() 메소드 전에 반드시 호출해야만 버퍼에 저장된데이터를 손실 없이 처리할 수 있다.

[표 베-6] BufferedInputStream와 BufferedOutputStream

	BufferedInputStream	BufferedOutputStream
의미	바이트 스트림으로부터 데이터를 읽어 들여 버퍼링하는 기능을 제공	버퍼링된 바이트 스트림을 출력하는 기능을 제공
생성자	BufferedInputStream(InputStream in)	BufferedOutputStream(OutputStream out)

BufferedInputStream과 BufferedOutputStream 클래스는 필터 입·출력 스트림이기 때문에 입·출력 스트림을 인자로 받는다.

```
পামা III–28 BufferedExam.java
1
      import java.io.*;
2
      class BufferedExam{
3
            public static void main(String[] args){
4
                    BufferedInputStream bi = new BufferedInputStream(System.in);
5
                    BufferedOutputStream bo = new BufferedOutputStream(System.out);
                   int aa;
6
7
                   try{
                            while((aa = bi,read()) != -1){
8
                                   bo.write(aa);
9
10
                            }
                            bo.flush();
11
                   }catch(IOException ie){}
12
            }
13
      }
14
```

프로그래밍(JAVA)

∷ 실행 결과

홍길동

۸Z

홍길동

④ FileInputStream 클래스 / FileOutputStream 클래스

FileInputStream 클래스와 FileOutputStream 클래스는 파일로부터 데이터를 바이트 단위로 읽고 출력하기 위한 입·출력 스트림을 생성하는 클래스이다.

[표 Ⅲ-7] FileInputStream 클래스와 FileOutputStream 클래스

	FileInputStream	FileOutputStream
의미	파일 이름을 인자로 받아 해당 파일에서 읽는 스트림을 생성	파일 이름을 인자로 받아 해당 파일에서 쓰 는 스트림을 생성
생성자	FileInputStream(String name)	FileOutputStream(String name)
메소드	read(), read(byte[] b) read(byte[] b, int off, int len)	<pre>write(b), write(byte[] b) write(byte[] b, int off, int len)</pre>

만약 파일이 없으면 FileNotFoundException이 발생하기 때문에 Exception 처리를 해주어야 한다.

```
পামা III-29 FileStreamExam.java
1
      import java.io.*;
2
      class FileStreamExam{
3
             public static void main(String[] args) throws IOException{
                    byte[] b = new byte[1024];
4
                    FileInputStream fis = new FileInputStream("tin.txt");
5
6
                    FileOutputStream fos = new FileOutputStream("tout.txt");
7
                    fis.read(b);
8
                    fos.write(b);
                    fis.close();
9
10
                    fos.close();
            }
11
12
      }
```

:: 설명

tin.txt의 내용이 "Java Programming"이라면 tout.txt의 내용도 "Java Programming"이다.

[03] 문자 스트림 입·출력

16비트 문자나 문자열을 읽고 쓰기 위한 스트림으로 모두 Reader와 Writer의 하위 클래스이다. Reader 클래스와 Writer 클래스를 이용하여 문자 단위(16bit)의 읽기와 쓰기를 지원하는 read() 메소드와 write() 메소드를 제공한다.

[표 Ⅲ-8] Reader 주요 메소드

메소드	설명
abstract int read()	• 스트림으로부터 1바이트를 읽는 메소드이다. • 스트림의 끝은 -1을 반환한다.
int read(char[] b)	문자 배열에 데이터를 읽어 들이는 메소드이다.한 번에 여러 데이터를 읽어 들인다.
void close()	• 해당 스트림과 관련된 모든 자원을 해제한다.

[표 Ⅲ-9] Writer 주요 메소드

메소드	설명
abstract void write(int b)	• int형 정수를 인자로 받아 출력한다. • byte형으로 자동 캐스팅된 후 출력한다.
void write(char[] b)	• 버퍼에 들어있는 문자 배열만큼 출력한다.
void flush()	• 버퍼에 쌓여있는 데이터를 출력시키고자 할 때 사용한다.
void close()	• 출력한 후 해당 스트림과 관련된 모든 자원을 해제한다.

① InputStreamReader 클래스 / OutputStreamWriter 클래스

InputStreamReader 클래스와 OutputStreamWriter 클래스는 바이트 스트림에서 문자 스트림으로, 문자 스트림에서 바이트 스트림으로의 변환을 제공하는 입·출력 스트림 클래스이다.

[표 Ⅲ-10] InputStreamReader와 OutputStreamReader의 관계

	InputStreamReader	OutputStreamWriter
의미	바이트 스트림으로부터 문자 스트림으로 의 변환을 제공	문자 스트림에서 바이트 스트림으로의 변환 을 제공
생성자	InputStreamReader(InputStream in)	OutputStreamWriter(OutputStream out)
메소드	read() read(char[] cbuf, int off, int len)	<pre>write (int c) write(char[] cbuf, int off, int len) write(String s, int off, int len) flush()</pre>

다음은 한 문자씩 키보드로 문자를 입력하여 출력하는 프로그램의 예이다.

```
পামা III–30 InputReaderExam.java
1
      import java.io.*;
2
      public class InputReaderExam{
      public static void main(String args[]){
3
           Reader reader = new InputStreamReader(System,in); // Reader는 추상 클래스
4
5
          try {
            while(true) {
6
7
               int i = reader.read();
               if(i == -1) break;
8
                                                                   // 마지막에 Ctrl + Z 입력
9
               char c = (char)i;
10
               System.out.print(c);
             }
11
          } catch(IOException e) {
12
13
            System.err.println(e);
          }
14
15
          System.out.println();
        }
16
      }
17
```

∷ 실행 결과

Test

Test

홍길동

홍길동

۸Z

InputStream 클래스를 이용하여 문자를 입·출력할 경우 byte 배열을 이용해야 하지만, Reader 클래스를 이용할 경우 배열을 사용하지 않아도 한글 처리가 가능하다.

② BufferedReader 클래스 / BufferedWriter 클래스

BufferedReader 클래스와 BufferedWriter 클래스는 문자 입력 스트림으로부터 문자를 읽어 들이거나 문자 출력 스트림으로 문자를 내보낼 때 버퍼링함으로써 문자, 문자배열, 문자열 라인 등을 보다 효율적으로 처리할 수 있도록 하는 클래스이다.

[표 Ⅲ-11] BufferedReader 클래스와 BufferedWriter 클래스의 관계

	BufferedReader	BufferedWriter
의미	문자 스트림으로부터 문자를 읽어들여 버퍼링하는 기능을 제공	버퍼링된 문자 스트림을 출력하는 기능을 제공
생성자	BufferedReader(Reader in)	BufferedWriter(Writer out)
메소드	read() read(char[] cbuf, int off, int len) readLine()	write(int c) write(char[] cbuf, int off, int len) write(String s, int off, int len) flush()

자바에서는 2개 이상의 클래스를 연결한 입·출력 스트림을 생성해서 입·출력하는 경우가 일반적이다. 다음은 InputStreamReader와 BufferedReader 클래스를 이용하여 한 라인씩 입·출력하고자 할 때 사용하는 일반적인 형태이다.

```
InputStreamReader isr = new InputStreamReader(System.in));
BufferedReader dos = new BufferedReader (isr);
dos.readLine();
```

পাসা III–31 BufferedReaderExam.java import java.io.*; 1 2 class BufferedReaderExam{ 3 public static void main(String[] args){ BufferedReader br = new BufferedReader(new InputStreamReader(System.in)); 4 try{ 5 String aa = br.readLine(); 6 7 System.out.println(aa); }catch(IOException ie){ } 8 } 9 10 }

:: 실행 결과

한 줄 내용을 입력하고 Enter 키를 누른다. 한 줄 내용을 입력하고 Enter 키를 누른다.

③ FileReger 클래스 / FileWriter 클래스

FileReaer 클래스와 FileWriter 클래스는 파일로부터 데이터를 문자 단위로 읽고 출력하기 위한 입·출력 스트림을 생성하는 클래스이다.

[표 Ⅲ-12] FileReader 클래스와 FileWriter 클래스

	FileReader	FileWriter	
의미	파일 이름을 인자로 받아 해당 파일에서 읽는 스트림을 생성	파일 이름을 인자로 받아 해당 파일에서 쓰 는 스트림을 생성	
생성지	FileReader(String name)	FileWriter(String name)	
메소드	InputStreamReader 클래스로부터 메 소드를 상속받아 사용한다.	OutputStreamWriter 클래스로부터 메소드 를 상속받아 사용한다.	

만약 파일이 없으면 FileNotFoundException이 발생하기 때문에 Exception 처리를 해주어야 한다.

예제	III-32 FileExam.java	
1	import java.io.*;	
2	class FlleExam{	
3	public static void	d main(String[] args){
4	char buffe	er[] = new char[100];
5	try{	
6	F	FileReader fr = new FileReader("datar.txt");
7	F	FileWriter fw = new FileWriter("dataw.txt");
8	f	r.read(buffer, 0, 100);
9	(String str = new String(buffer);
10		System.out.println(str);
11	f	w.write(buffer);
12	f	w.flush();
13	}catch(Ex	ception e){
14		System.out.println(e.toString());
15	}	
16	}	
17	}	

∷실행 결과

전문계 고등학교

Java Programming

::설명

datar.txt 파일의 내용

전문계 고등학교 Java Programming

dataw.txt 파일의 내용은 datar.txt의 내용과 같다.

위 프로그램은 입력 파일 datar.txt의 내용을 화면과 dataw.txt 파일에 그대로 출력하는 프로그램을 작성한 것이다.

버퍼를 이용하지 않고 다음과 같이 한 바이트씩 읽어서 파일을 복사할 수 있다. [예제 Ⅲ-32]의 8~11 라인을 수정해 보자.

[04] 데이터 입·출력 실습

편리한 입·출력을 위하여 입·출력 스트림들을 결합하여 사용한다. 이와 같이 필터링 기능을 제공하는 클래스를 필터링 스트림 클래스라고 한다.

대표적 필터링 스트림 클래스로는 BufferedInputStream, BufferedOuputStream, BufferedReader, BufferedWriter, DataInputStream, DataOutputStream 등이 있다. 필터링 스트림을 이용하여 다양한 형태의 데이터를 입 · 출력하여 보자.

① 키보드로부터 줄 단위 문자 입력

InputStreamReader로 한 문자씩 읽은 것을 BufferedReader로 필터링한다.

```
পামা III–33 BufferInputExam.java
1
      import java.io.*;
      public class BufferInputExam{
2
3
          public static void main(String args[]){
4
               BufferedReader in = new BufferedReader( new InputStreamReader(System.in));
5
               String s=" ";
6
               try{
7
                      s = in.readLine();
               }catch(Exception e){
8
                      System.out.println("Error: " + e.toString());
9
                }
10
               System.out.println(s);
11
          }
12
13
      }
```

프로그래밍 (JAVA)

```
∷ 실행 결과
```

홍길동

홍길동

② 파일로부터 한 줄단위 문자 입력

파일로부터 문자열을 읽어 BufferedReader로 필터링한다. 다음 프로그램은 파일에 저장된 내용을 읽어 화면에 출력하는 프로그램이다.

```
BufferInput.txt 파일의 내용
인터넷 고등학교
IT 특성화 학교
서울시 용산구 청파동
```

পামা III–34 ReadLineFileExam.java

```
import java.io.*;
1
2
      class ReadLineFileExam{
3
             public static void main(String[] args){
                    String fname = "BufferInput.txt";
4
5
                    try{
6
                           FileReader fr = new FileReader(fname);
7
                           BufferedReader br = new BufferedReader(fr);
                           String s=" ";
8
                           while((s = br.readLine()) != null){
9
                           System.out.println(s);
10
11
                           br.close();
12
                           fr.close();
13
                    }catch(Exception e){
14
                           System.out.println(e.toString());
15
16
                    }
17
             }
18
      }
```

∷ 실행 결과

인터넷 고등학교 IT 특성화 학교 서울시 용산구 청파동

:: 설명

BufferInput,txt에 저장된 파일을 한 라인씩 읽어 들여 String형 객체 변수 s에 저장한 후 저장된 값을 출력한다.

③ 키보드로부터 숫자 데이터 입력

키보드로 BufferedReader 클래스를 이용하여 라인 단위로 데이터를 읽은 후 Integer. parseInt()를 이용하여 처리한다.

পামা III–35 InExam.java

```
1
      import java.io.*;
2
      public class InExam{
3
         public static void main(String args[]) {
              String a="", b="";
4
              int sum=0;
              float average = 0;
6
7
              BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
8
              try{
9
                     System.out.print("Enter the string(정수) >");
                     a = in.readLine();
10
                     System.out.print("Enter the string(정수) >");
11
                     b = in.readLine();
12
13
              }catch(IOException e) {
14
                     System.out.println(e);
15
              System.out.println("Your input is "+a+", "+b);
16
              int c = Integer.parseInt(a);
17
18
              int d = Integer.parseInt(b);
              sum = c+d;
19
              average = (float)sum/2;
20
              System.out.println(a+" + "+b+ " = " + sum);
21
22
              System.out.println("평균= "+average);
23
              }
24
      }
```

∷ 실행 결과

```
Enter the String(정수) 〉1
Enter the String(정수) 〉2
Your input is 1, 2
1 + 2 = 3
평균 = 1.5
```

:: 설명

키보드로부터 숫자 데이터를 한 라인씩 읽어 정수로 변환 한 후 합과 평균을 출력하는 프로그램으로 데이터 입력 시 하나씩 입력(1 Enter) 하야 한다.

④ 파일로부터 성적 데이터 입력 후 처리

파일로부터 데이터를 읽어 BufferedReader 클래스를 이용하여 라인 단위로 읽은 후 StringTokenizer를 이용하여 처리한다. 다음은 writed.txt 파일로부터 데이터를 읽어 총점과 평균을 구하여 실행 결과처럼 출력하는 프로그램이다.

```
writed.txt 파일의 내용
Kim
        용산구
               100
                       100
                        90
Lee
        구로구
               90
Choi
        강남구
               100
                        100
Hong
        강서구
               90
                        90
```

পামা III–36 sungDatasMain.java

```
1
      import java.io.*;
2
      import java.util.*;
3
      class sungDatasExam{
            public void reading Data (String fname) throws IOException {
4
5
                try{
6
                   FileReader fr = new FileReader(new File(fname));
7
                   BufferedReader br = new BufferedReader(fr);
                   String csvStr = "";
8
                   String tmpStr = "";
9
10
                   do{
                       tmpStr=br,readLine();
11
12
                       if(tmpStr != null){
13
                         csvStr += tmpStr + "\t";
                       }
14
                   }while(tmpStr != null);
15
16
                   StringTokenizer parse = new StringTokenizer(csvStr, "\t");
17
                   int length = parse.countTokens()/ 4;
18
                   String[] name = new String[length];
                   String[] address = new String[length];
19
20
                   double[] math = new double[length];
21
                   double[] english = new double[length];
22
                   double[] total = new double[length];
23
                   double[] avg = new double[length];
24
                   for( int j = 0; j < length; j++){
25
                       name[i] = parse.nextToken();
```

```
26
                       address[i] = parse.nextToken();
                       math[i] = Double.valueOf(parse.nextToken()).doubleValue();
27
                       english[i] = Double.valueOf(parse.nextToken()).doubleValue();
28
29
                       total[i] = math[i] + english[i];
                       avg[i] = total[i] /2.0;
30
31
                   }
32
                   for( int j = 0; j < length; j++){
                       System.out.println(name[i] +"\t" +address[i] +"\t" + math[i] + "\t" +
33
                       english[i]+"\t" +total[i]+ "\t" + avg[i]);
34
35
                   }
            }catch(FileNotFoundException e){
36
                       System.out.println("잘못된 파일 이름을 입력했습니다");
37
38
            }
         }
39
      }
40
41
      class sungDatasMain{
42
         public static void main(String[] args){
43
            sungDatasExam wd = new sungDatasExam();
44
            trv{
45
                wd.readingData("writed.txt");
46
            }catch(IOException e){
47
                   e.printStackTrace();
           }
48
         }
49
      }
50
```

∷ 실행 결과

```
Kim
      용산구
             100.0 100.0
                          200.0 100.0
Lee
      구로구
             90.0
                   90.0
                          180.0 90.0
Choi
      강남구
             100.0 100.0
                          200.0 100.0
Hona
      강서구
             90.0
                   90.0
                          180.0 90.0
계속하려면 아무 키나 누르십시오...
```

:: 설명

writed.txt 파일로부터 한 줄씩 데이터를 읽어 들여 토큰 "\t"로 데이터를 분할한 후 각 항목을 name[], address[], math[], english[] 배열에 저장하여 합과 평균을 구하여 total[], avg[]에 저장한 후 출력하는 프로그램이다. 이때 writed.txt 파일에 데이터 입력 시 각 항목을 탭(tab) 키로 구분하여야 한다.

⑤ 파일로부터 성적 데이터 입력후 처리 - 파일에서 읽고 파일에 쓰기

입력 데이터 writed.txt로부터 데이터를 읽어 실행 결과처럼 writing.txt 파일에 출력시키는 프로그램을 작성하여라.

〈〈 입력 데이터 〉〉 writed.txt의 내용 Kim 용산구 100 100 Lee 구로구 90 90 Choi 강남구 100 100 강서구 90 90 Hong

```
〈〈 실행 결과 〉〉 writing.txt의 내용
Kim
        용산구
               100.0
                       100.0
                                200.0
                                       100.0
Lee
        구로구
               90.0
                       90.0
                                180.0
                                       90.0
                                200.0
Choi
        강남구
               100.0
                       100.0
                                       100.0
        강서구
               90.0
                       90.0
                                180.0
                                       90.0
Hong
```

পামা III–37 Writing Datas Main. java

```
import java.io.*;
1
      import java.util.*;
2
3
      class sungDataRWExam{
          int length;
4
          String[] name = new String[10];
5
          String[] address = new String[10];
6
7
          double[] math = new double[10];
          double[] english = new double[10];
8
          double[] total = new double[10];
9
          double[] avg = new double[10];
10
          public void writingData(String fname, boolean append) throws IOException(
11
            FileWriter fw = null;
12
            BufferedWriter bw = null:
13
14
            try{
               fw = new FileWriter( new File(fname));
15
               bw = new BufferedWriter(fw);
16
               for( int j = 0; j < length; j++){
17
                  bw.write(name[i] +"\t" +address[i] +"\t" + math[i] + "\t" + english[i]+"\t"
18
                          +total[i]+ "\t" + avg[i]);
19
                  bw.newLine();
20
```

```
}
21
               bw.close();
22
23
               fw.close();
24
               }catch(FileNotFoundException e){
                  System.out.println("잘못된 파일 이름을 입력했습니다");
25
               }
26
27
          public void readingData(String fname) throws IOException{
28
29
            try{
               String csvStr = "";
30
               String tmpStr = "";
31
32
               FileReader fr = new FileReader(new File(fname));
               BufferedReader br = new BufferedReader(fr);
33
34
               do{
35
                 tmpStr=br.readLine();
                 if(tmpStr != null){
36
                      csvStr += tmpStr + "\t";
37
38
                 }
               }while(tmpStr != null);
40
                  StringTokenizer parse = new StringTokenizer(csvStr, "\t");
41
                 length = parse.countTokens()/ 4;
                 for( int i = 0; i \in length; i++){}
42
                      name[i] = parse.nextToken();
43
                      address[i] = parse.nextToken();
44
                      math[i] = Double.valueOf(parse.nextToken()).doubleValue();
45
                      english[i] = Double.valueOf(parse.nextToken()).doubleValue();
46
                      total[i] = math[i] + english[i];
47
                      avg[i] = total[i] /2.0;
48
49
            }catch(FileNotFoundException e){
50
                 System.out.println("잘못된 파일 이름을 입력했습니다");
51
            }
52
          }
53
      }
54
      class WritingDatasMain{
55
          public static void main(String[] args){
56
            sungDataRWExam wd = new sungDataRWExam();
57
            try{
58
```

프로그래밍 (JAVA)

```
wd.readingData("writed.txt");
wd.writingData("writing.txt", false);
for a large of the second o
```

읽을거리

위피(WIPI)

위피는 'Wireless Internet Platform for Interoperability'의 머리글자를 딴 것이다. 이동통신 업체들이 같은 플랫폼을 사용하도록 함으로써 국가적 낭비를 줄이자는 목적으로 2001년부터 국책사업으로 추진되기 시작하였다. 무선인터넷 플랫폼이란 이동전화 단말기에서 퍼스널컴퓨터의 운영체제(OS)와 같은 역할을 하는 기본 소프트웨어를 말한다.

한국의 이동통신 업체들은 그동안 회사마다 각기 다른 방식으로 무선인터넷 플랫폼을 만들어 사용하였기 때문에 콘텐츠 제공업체들도 같은 콘텐츠를 여러 개의 플랫폼으로 만들 수밖에 없었다. 따라서 콘텐츠 제작과서비스에 따르는 여러 가지 불필요한 낭비 요소가 발생하였는데, 위피는 국가적인 차원에서 이러한 낭비 요소를 줄일 목적으로 탄생한 것이다.

2003년 6월 LG전자(주)에서 처음으로 위피를 적용한 휴대폰이 출시되었고, 이보다 앞서 정보통신부와 이동통 신업계는 위피를 국제 표준으로 만들기 위해 2002년 6월 국제무선인터넷표준화기구(OMA)에 국제 표준으로 제안하였다. 2,0 버전은 위의 기관들 외에 지어소프트(주)·이노에이스(주)·IBM·아로마소프트(주)·모토로라 ·베텔시스템 등 국내외 60여 개의 기업이 플랫폼 엔진 및 단말기 제조, 콘텐츠의 연구 개발에 참여하였다.

2004년 2월 자바표준화단체(JCP)의 표준 규격인 CLDC/MIDP와 완전한 호환성을 갖춘 2.0 버전을 완성하였으며, 2005년 4월 1일부터 전기통신설비의 상호접속기준고시에 따라 신규 출시되는 모든 단말기에 위피가의무 탑재된다.

[출처 네이버 백과사전]



- 패키지는 자바 개발 환경에서 제공되는 거대한 클래스 라이브러리로 관련된 클래스와 인터페이스의 모임을 말한다.
- import 명령으로 패키지의 클래스들을 사용할 수 있다.
- java.lang 패키지의 클래스들은 자바 언어의 기초적인 사항을 정의한 클래스로 import 시키지 않아도 사용할 수 있다.
- 문자열 처리를 위한 클래스는 String 클래스와 StringBuffer 클래스이다.
- String 클래스로 생성된 문자열은 한 번 생성되면 변경할 수 없다.
- StringBuffer 클래스로 생성된 문자열은 변경할 수 있는 문자열로, 동적으로 문자열의 내용을 바꾸거나 위치를 지정할 수 있다.
- Wrapper 클래스는 기본 데이터 타입을 클래스 타입으로 포장한 클래스 이다.
- Math 클래스는 수학 함수와 이에 필요한 상수들을 제공하는 클래스이다.
- java.jo 패키지는 입·출력 관련 클래스를 포함한다.
- 입·출력 스트림은 단방향이기 때문에 입력용 스트림과 출력용 스트림을 모두 생성해 주어야 한다.
- 자바의 입·출력 방식은 바이트 스트림을 이용한 입·출력 방법과 문자 스트림을 이용한 입·출력 방법이 있다.
- 바이트 단위 스트림의 상위 클래스는 InputStream과 OutputStream 이고, 버퍼 입·출력을 위해 BufferedInputStream, BufferedOutput Stream 클래스를 사용하고, 파일 입·출력을 위해 FileInputStream, FileOutputStream 클래스를 사용한다.
- 문자 단위 스트림의 상위 클래스는 Reader와 Writer이고, 한 문자 입· 출력을 위해 InputStreamReader, OutputStreamWriter 클래스를 사용하고, 여러 문자열을 입·출력하기 위해 BufferedReader, BufferedWriter 클래스를 사용하고, 파일 입·출력을 위해 FileReader, FileWriter 클래스를 사용한다.



* 종합문제



String 클래스와 StringBuffer 클래스의 차이점을 설명하여라.



이름, 전화번호, 주소를 가진 주소록 클래스를 만들어 키보드로부터 이름을 입력하여 전화번호와 주소를 검색하는 프로그램을 작성해 보자.



1에서 45까지의 난수를 발생시켜 임의의 숫자 6개를 출력하는 프로그램을 작성해 보자.



밑변과 높이를 키보드로 입력하여 삼각형의 넓이를 구하는 프로그램을 작성해 보자.



Tab 간격으로 데이터들이 구분되어 있는 파일로부터 데이터를 입력하여 화면에 출력하는 프로그램을 작성해 보자.

〈파일 구조〉

학교 이름 주소 전화번호 교사 수 학생 수

〈파일 내용〉

선린	용산구	02-1111-2222	78	900
서울	강서구	02-234-1234	80	1000
컴퓨터	강동구	02-123-4321	100	1200