

연습문제 정답

1-1 한글 지원 점검 테스트

[소스 코드]

```
class HelloJava {
    public static void main(String args[]) {
        System.out.println("안녕하세요, 자바");
    }
}
```

[실행 결과]

안녕하세요, 자바

2-1 2차원 배열의 항목 수

[소스 코드]

```
class Answer {
    public static void main(String args[]) {
        int table[][] = { { 1, 2, 3, 4 }, { 5, 6, 7, 8 }, { 9, 10, 11, 12 } };
        System.out.println(table.length);
    }
}
```

[실행 결과]

3

2-2 switch 문을 if 문으로

[소스 코드]

```
class Answer {
    public static void main(String args[]) {
        int num = 3;
        if (num == 1)
            System.out.println("Good Morning, Java");
        else if (num == 2)
            System.out.println("Good Afternoon, Java");
        else if (num == 3)
            System.out.println("Good Evening, Java");
        else
            System.out.println("Hello, Java");
        System.out.println("Done.");
    }
}
```

[실행 결과]

```
Good Evening, Java
Done.
```

2-3 for 문의 연습

배열의 3번째 항목 인덱스는 2이고, 7번째 항목 인덱스는 6입니다. 그러므로 다음과 같이 프로그램을 작성하면 됩니다.

[소스 코드]

```
class Answer {
    public static void main(String args[]) {
        int arr[] = { 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 };
        int total = 0;
        for (int cnt = 2; cnt < 7; cnt++)
            total += arr[cnt];
        System.out.println(total);
    }
}
```

[실행 결과]

```
50
```

2-4 break 문의 연습

[소스 코드]

```
class Answer {
    public static void main(String args[]) {
        int arr[] = { 435, 88, 67, 32, 88, -1, 6, 12, 7, 8, 45, 11 };
        for (int cnt = 2; cnt < 7; cnt++) {
            if (arr[cnt] == -1)
                break;
            System.out.println(arr[cnt]);
        }
    }
}
```

[실행 결과]

```
67
32
88
```

2-5 메소드 호출문의 연습

[소스 코드]

```
class Answer {
    public static void main(String args[]) {
```

```

        int total = 0;
        for (String str : args) {
            int num = Integer.parseInt(str);
            total += num;
        }
        System.out.println(total);
    }
}

```

[실행 결과]

```

명령 프롬프트
E:\연습문제정답\2-5>java Answer
0
E:\연습문제정답\2-5>java Answer 10 20
30
E:\연습문제정답\2-5>java Answer -12 5
-7
E:\연습문제정답\2-5>

```

2-6 익셉션 처리 문제

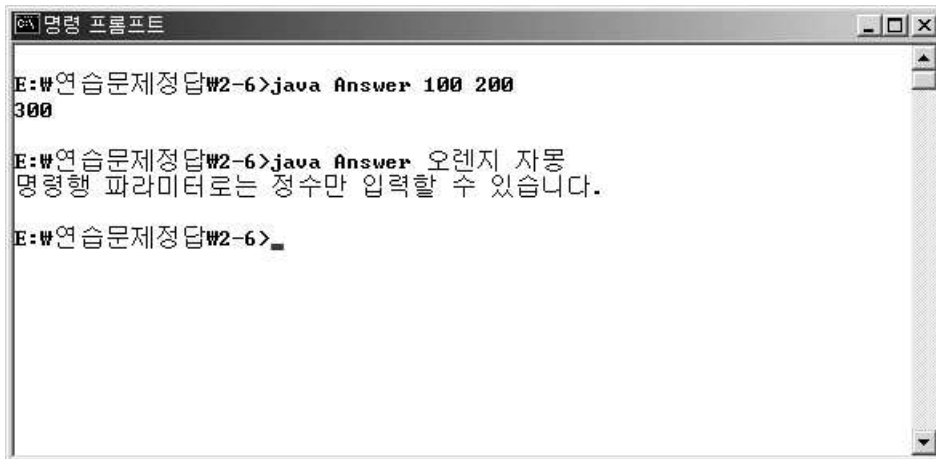
[소스 코드]

```

class Answer {
    public static void main(String args[]) {
        int total = 0;
        try {
            for (String str : args) {
                int num = Integer.parseInt(str);
                total += num;
            }
            System.out.println(total);
        }
        catch (java.lang.NumberFormatException e) {
            System.out.println("명령행 파라미터로는 정수만 입력할 수 있습니다.");
        }
    }
}

```

[실행 결과]



3-1 정수 내부 표현의 확인

[소스 코드] -37의 내부 표현을 출력하는 프로그램

```
class Answer1 {
    public static void main (String args[]) {
        String str;
        str = Integer.toBinaryString(-37);
        System.out.println(str);
    }
}
```

[실행 결과]

```
11111111111111111111111111111111011011
```

[소스 코드] -5의 내부 표현을 출력하는 프로그램

```
class Answer2 {
    public static void main (String args[]) {
        String str;
        str = Integer.toBinaryString(-5);
        System.out.println(str);
    }
}
```

[실행 결과] -5의 내부 표현을 출력하는 프로그램

```
11111111111111111111111111111111011
```

3-2 부동소수점수 내부 표현의 확인

[소스 코드]

```
class Answer {
    public static void main (String args[]) {
        double num1 = -12.375, num2 = 12.375;
        long num3 = Double.doubleToRawLongBits(num1);
    }
}
```

```

        long num4 = Double.doubleToRawLongBits(num2);
        String str1 = Long.toBinaryString(num3);
        String str2 = Long.toBinaryString(num3);
        System.out.println(str1);
        System.out.println(str2);
    }
}

```

[실행 결과]

```
11000000001010001100000000000000000000000000000000000000000000000000  
11000000001010001100000000000000000000000000000000000000000000000000
```

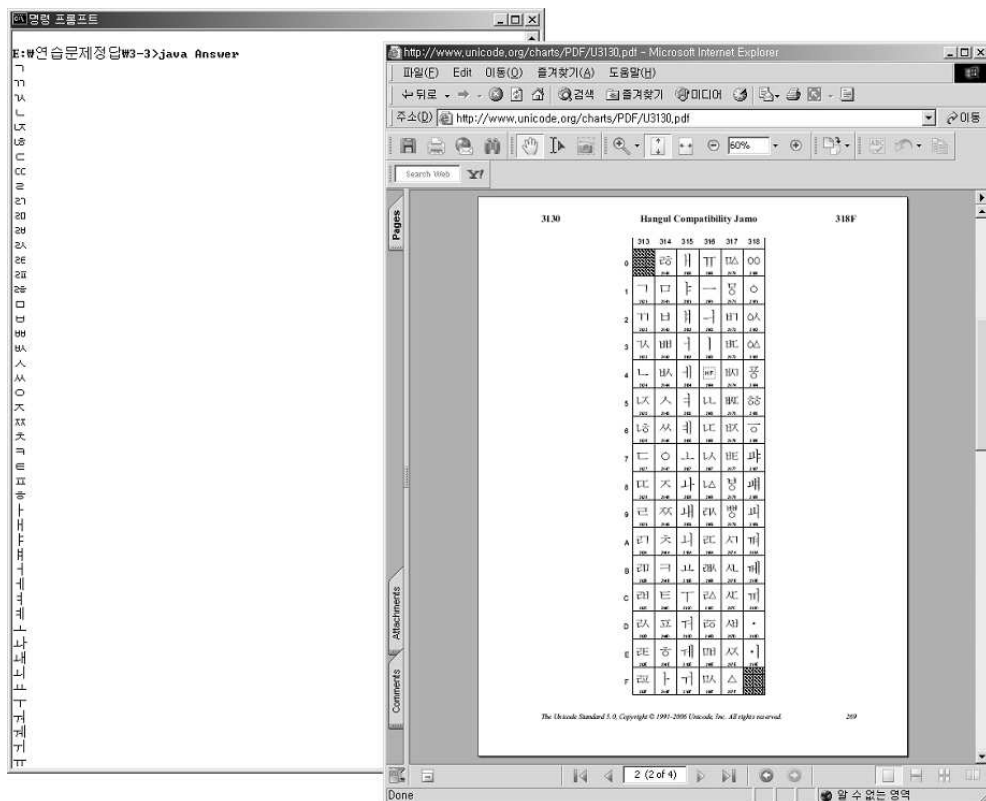
3-3 Unicode 문자 출력하기

[소스 코드]

```
class Answer {
    public static void main (String args[]) {
        for (char ch = 12593; ch < 12687; ch++)
            System.out.println(ch);
    }
}
```

[실행 결과]

ㄱ부터¹까지의 한글 자음, 모음이 출력되는데, 이 문자들을 Unicode 홈페이지의 12593(3131₁₆)부터 12686(318E₁₆)에 해당하는 문자들과 비교하면 해당 문자들과 일치함을 확인할 수 있습니다.

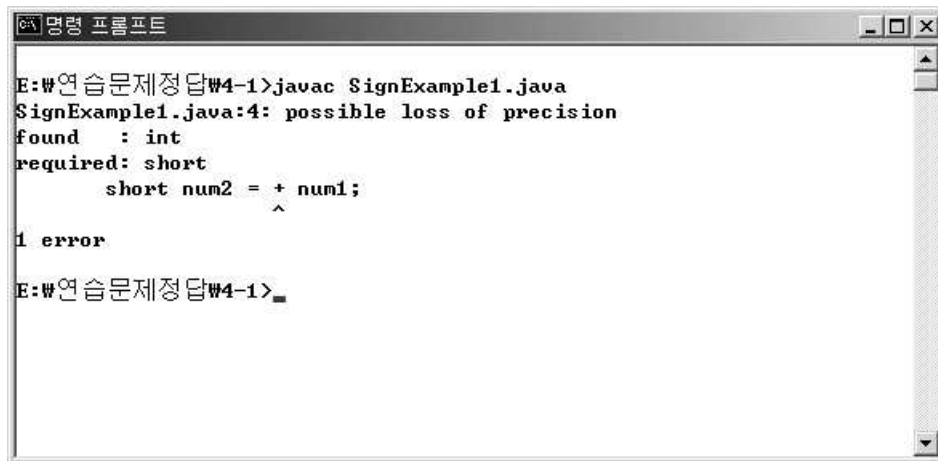


4-1 부호 연산자

[소스 코드]

```
class SignExample1 {
    public static void main(String args[]) {
        short num1 = 100;
        short num2 = + num1;
        System.out.println(num2);
    }
}
```

[컴파일 결과]



4-2 //과 /의 차이

[소스 코드]

```
class ConditionalOrExample1 {
    public static void main(String args[]) {
        int num1 = 0, num2 = 0;
        if (++num1 > 0 | ++num2 > 0)
            System.out.println("num1이 0보다 크거나 num2가 0보다 큼니다.");
        System.out.println("num1 = " + num1);
        System.out.println("num2 = " + num2);
    }
}
```

[실행 결과]

```
num1이 0보다 크거나 num2가 0보다 큼니다.
num1 = 1
num2 = 1
```

5-1 메소드 호출 연습 (1)

[소스 코드]

```
class Answer {
    public static void main(String args[]) {
```

```

        StringBuffer obj;                // 객체를 담을 변수 선언
        obj = new StringBuffer("Hey, Java"); // 객체를 생성해서 변수에 대입
        obj.replace(1, 3, "i");           // 객체의 메서드 호출
        System.out.println(obj);
    }
}

```

[실행 결과]

Hi, Java

5-2 메소드 호출 연습 (2)

[소스 코드]

```

class Answer {
    public static void main(String args[]) {
        GoodsStock obj;
        obj = new GoodsStock();
        obj.goodsCode = "52135";
        obj.stockNum = 200;
        System.out.println("상품코드:" + obj.goodsCode);
        System.out.println("재고수량:" + obj.stockNum);
        obj.addStock(1000);
        System.out.println("상품코드:" + obj.goodsCode);
        System.out.println("재고수량:" + obj.stockNum);
        obj.subtractStock(1);
        System.out.println("상품코드:" + obj.goodsCode);
        System.out.println("재고수량:" + obj.stockNum);
    }
}

```

[실행 결과]

상품코드:52135
 재고수량:200
 상품코드:52135
 재고수량:1200
 상품코드:52135
 재고수량:1199

5-3 생성자 선언 연습

[소스 코드]

```

// 재고 정보 클래스
class GoodsStock {
    String goodsCode;
    int stockNum;
    GoodsStock(String code, int num) {

```

```

        goodsCode = code;
        if (num < 0)
            stockNum = 0;
        else
            stockNum = num;
    }
    void addStock(int amount) {
        stockNum += amount;
    }
    int subtractStock(int amount) {
        if (stockNum < amount)
            return 0;
        stockNum -= amount;
        return amount;
    }
}
// 재고 정보 클래스의 객체를 생성해서 필드 값을 출력하는 프로그램
class Answer {
    public static void main(String args[]) {
        GoodsStock obj;
        obj = new GoodsStock("12345", -50);
        System.out.println("상품코드:" + obj.goodsCode);
        System.out.println("재고수량:" + obj.stockNum);
    }
}

```

[실행 결과]

```

상품코드:12345
재고수량:0

```

5-4 생성자가 발생하는 의외선

[소스 코드]

```

class Answer {
    public static void main(String args[]) {
        try {
            Account obj1 = new Account("777-777-7777", "연놀부", 10000000);
            System.out.println(obj1.balance);
            Account obj2 = new Account("000-000-000000", "연흥부", -100000);
            System.out.println(obj2.balance);
        }
        catch (Exception e) {
            String msg = e.getMessage();
            System.out.println(msg);
        }
    }
}

```



```
}
}
}
```

* Account 객체를 생성하는 명령문을 try 문만 두기만 하면 프로그램은 위와 다른 방식으로 작성해도 됩니다.

[실행 결과]

```
10000000
객체를 생성할 수 없습니다.
```

6-1 생성자가 있는 슈퍼클래스의 상속 테스트

[실행 결과]

Account 클래스([예제 6-1]), CheckingAccount 클래스([예제 6-4])와 함께 실행했을 때와 똑같은 결과가 나옵니다.

```
지불액:47000
잔액:53000
```

6-2 추상 메소드와 throws 절의 관계 테스트

[컴파일 결과]

다음과 같이 정상적으로 컴파일 됩니다.

```
명령 프롬프트
E:\연습문제정답\W6-2>javac Lendable.java
E:\연습문제정답\W6-2>javac SeparateVolume.java
E:\연습문제정답\W6-2>
```

7-1 배열의 레퍼런스 타입 특성 테스트

[실행 결과]

하나의 배열을 두 개의 배열 변수에 대입해도 배열 자체는 하나이기 때문에 arr2 변수를 이용해서 바꾼 두 번째 배열 항목의 값을 다음과 같이 arr1 변수를 가지고 출력해서 볼 수 있습니다.

```
1
2
3
1
7
3
```

7-2 배열 변수와 null

[소스 코드]

```
class ArrayVarTest2 {
    public static void main(String args[]) {
        int arr[] = { 1, 2, 3, 4, 5 };
        printArray(arr);
        arr = null;
        printArray(arr);
    }
    static void printArray(int arr[]) {
        if (arr == null)
            return;
        for (int num : arr)
            System.out.println(num);
    }
}
```

[실행 결과]

```
1
2
3
4
5
```

7-3 열거 타입 연습

[소스 코드]

```
enum Day {
    MONDAY("월"), TUESDAY("화"), WEDNESDAY("수"), THURSDAY("목"), FRIDAY("금"), SATURDAY("토"), SUNDAY("일");
    final private String name;
    Day(String name) {
        this.name = name;
    }
    String value() {
        return name;
    }
}
```

8-1 접근 제어 연습문제

[소스 코드]

```
import geometry.shape.Square;
class Answer {
    public static void main(String args[]) {
        Square obj = new Square(100, 200, 15);
        System.out.println("(" + obj.getX(0) + ", " + obj.getY(0) + ")");
        System.out.println("(" + obj.getX(1) + ", " + obj.getY(1) + ")");
    }
}
```

```

        System.out.println("(" + obj.getX(2) + ", " + obj.getY(2) + ")");
        System.out.println("(" + obj.getX(3) + ", " + obj.getY(3) + ")");
    }
}

```

[실행 결과]

```

(100, 200)
(115, 200)
(115, 215)
(100, 215)

```

9-1 String 클래스의 특성 확인하기

[실행 결과]

```

뇌를 자극하는 C
뇌를 자극하는 하드웨어
뇌를 자극하는 하드웨어 입문

```

이 결과를 통해 알 수 있는 것은 replace, concat 메소드를 이용하여 문자열을 조작해도 원래 String 객체가 가지고 있던 문자열의 내용은 바뀌지 않는다는 것입니다. 382 페이지의 본문에서 이야기하고 있듯이 이 두 메소드는 String 객체가 가지고 있는 문자열의 내용을 바꾸는 것이 아니라, 바뀐 내용을 갖는 새로운 String 객체를 생성하기 때문입니다.

9-2 문자열에서 문자/부분 문자열 검색하기

[실행 결과]

```

0
13
0
13
37
13
13
13

```

9-3 StringBuilder의 메소드

[실행 결과]

```

어떤 돌이 내 얼굴을 물끄러미 쳐다보는 것만
어떤 꽃이 내 얼굴을 물끄러미 치어다보는 것만
만것 는보다어치 미러끄물 을굴얼 내 이꽃 떼어

```

9-4 토큰 추출 프로그램

[소스 코드]

```

import java.util.StringTokenizer;
class Answer {

```

```

public static void main(String args[]) {
    StringTokenizer stok = new StringTokenizer("고슴도치, 앵무새 | 토끼", ",| ");
    while (stok.hasMoreTokens()) {
        String str = stok.nextToken();
        System.out.println(str);
    }
}

```

[실행 결과]

```

고슴도치
앵무새
토끼

```

9-5 세계 시간 프로그램

[소스 코드]

```

import java.util.*;
import java.text.*;
class Answer {
    public static void main(String args[]) {
        GregorianCalendar calendar = new GregorianCalendar();
        printDateTime(calendar, "뉴욕", "America/New_York");
        printDateTime(calendar, "홍콩", "Asia/Hong_Kong");
        printDateTime(calendar, "파리", "Europe/Paris");
    }
    static void printDateTime(GregorianCalendar calendar, String location, String timeZone) {
        SimpleDateFormat dateFormat = new SimpleDateFormat(" yyyy/MM/dd (E) aa hh:mm");
        dateFormat.setTimeZone(TimeZone.getTimeZone(timeZone));
        String str = dateFormat.format(calendar.getTime());
        System.out.println(location + str);
    }
}

```

[실행 결과]

```

뉴욕 2006/09/14 (목) 오전 12:45
홍콩 2006/09/14 (목) 오후 12:45
파리 2006/09/14 (목) 오전 06:45

```

* 위 결과는 우리나라 시간으로 2006년 9월 14일 오후 1시 45분에 이 프로그램을 실행했을 때의 결과입니다.

9-6 Random 클래스

[소스 코드]

```

import java.util.*;
class Answer {
    public static void main(String args[]) {

```

```

Random random = new Random();
int headNum = 0;
for (int cnt = 0; cnt < 100; cnt++) {
    boolean isHead = random.nextBoolean();
    if (isHead)
        headNum++;
}
System.out.println("앞면: " + headNum);
System.out.println("뒷면: " + (100 - headNum));
}
}

```

[실행 결과]

```

E:\연습문제정답\9-6>java Answer
앞면: 48
뒷면: 52

E:\연습문제정답\9-6>java Answer
앞면: 51
뒷면: 49

E:\연습문제정답\9-6>java Answer
앞면: 47
뒷면: 53

E:\연습문제정답\9-6>

```

이 프로그램을 실행할 때마다 다른 결과가 나옵니다. 그러니까 독자 여러분의 컴퓨터에서 이 프로그램을 실행했을 때는 위와 다른 결과가 나올 수 있습니다.

10-1 배열 파라미터를 받는 read 메소드

[소스 코드]

```

import java.io.*;

class Answer {
    public static void main(String args[]) {
        FileReader reader = null;
        try {
            reader = new FileReader("poem.txt");
            while (true) {
                char arr[] = new char[80];
                int num = reader.read(arr);
                if (num < 0)
                    break;
                System.out.print(new String(arr));
            }
        }
        catch (FileNotFoundException fnfe) {
            System.out.println("파일이 존재하지 않습니다.");
        }
    }
}

```

```

    }
    catch (IOException ioe) {
        System.out.println("파일을 읽을 수 없습니다.");
    }
    finally {
        try {
            reader.close();
        }
        catch (Exception e) {
        }
    }
}
}

```

[실행 결과]

* 이 프로그램의 실행 결과는 [예제 10-3]과 동일합니다. (p.436 [그림 10-5] 참조)

10-2 프리미티브 타입 데이터는 파일에 어떻게 저장될까?

[실행 결과]

FileDump 프로그램([예제 10-6])을 작성했던 디렉토리에서 다음과 같이 실행하면 됩니다.

또는 [예제 10-7]의 결과물인 output.dat 파일이 있는 디렉토리에서 다음과 같이 실행해도 됩니다. (부록 참조)

10-3 직렬화 가능 클래스와 직렬화 불가능 클래스의 차이점 비교

[소스 코드]

```
import java.io.*;
```

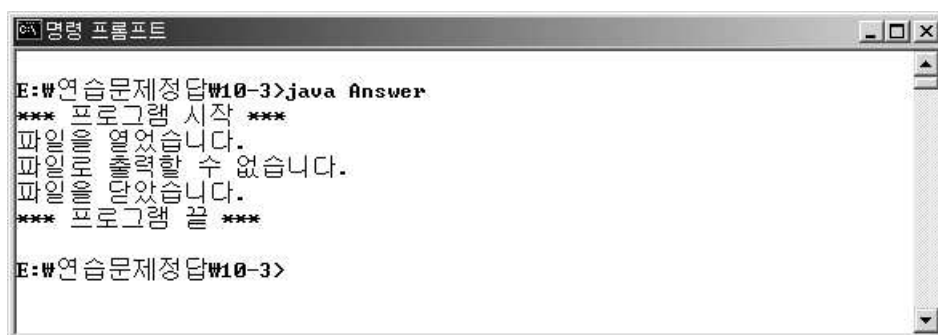
```

import java.util.StringTokenizer;
class Answer {
    public static void main(String args[]) {
        System.out.println("*** 프로그램 시작 ***");
        ObjectOutputStream out = null;
        try {
            out = new ObjectOutputStream(new FileOutputStream("output.dat"));
            System.out.println("파일을 열었습니다.");
            out.writeObject(new StringTokenizer("사과|딸기|배", "|"));
            System.out.println("파일로 StringTokenizer 객체를 출력했습니다.");
        }
        catch (IOException ioe) {
            System.out.println("파일로 출력할 수 없습니다.");
        }
        finally {
            try {
                out.close();
            }
            catch (Exception e) {
            }
        }
        System.out.println("파일을 닫았습니다.");
    }
}

```

* 프로그램의 어느 곳에서 에러가 발생하는지 알기 위해 일시적으로 써놓은 명령문은 위에서처럼 들여쓰기를 하지 않고 첫 번째 컬럼부터 써두면, 나중에 찾아서 제거할 때 편리합니다.

[실행 결과]



이 실행 결과를 통해 알 수 있는 것은 “파일을 열었습니다.” 라는 메시지를 출력하는 명령문과 “파일로 StringTokenizer 객체를 출력했습니다.”라는 메시지를 출력하는 명령문 사이에서 에러가 발생하여 catch 블록으로 프로그램 실행 흐름의 제어가 넘어갔다는 것입니다. 두 메시지 출력 명령문 사이에 있는 명령문은 writeObject 메소드 호출문이므로 그 부분에서 에러가 발생했다는 것을 알 수 있습니다.

10-4 객체의 직렬화 연습

[소스 코드]

```
// 주간 주가 추이를 파일에 저장하는 프로그램
import java.io.*;
import java.util.GregorianCalendar;

class Answer1 {
    public static void main(String args[]) {
        ObjectOutputStream out = null;
        try {
            out = new ObjectOutputStream(new FileOutputStream("output.dat"));
            out.writeObject(new GregorianCalendar(2006, 3, 10));
            out.writeFloat(1398.29f);
            out.writeObject(new GregorianCalendar(2006, 3, 11));
            out.writeFloat(1386.08f);
            out.writeObject(new GregorianCalendar(2006, 3, 12));
            out.writeFloat(1383.59f);
            out.writeObject(new GregorianCalendar(2006, 3, 13));
            out.writeFloat(1405.72f);
            out.writeObject(new GregorianCalendar(2006, 3, 14));
            out.writeFloat(1432.72f);
        }
        catch (IOException ioe) {
            System.out.println("파일로 출력할 수 없습니다.");
        }
        finally {
            try {
                out.close();
            }
            catch (Exception e) {
            }
        }
    }
}

// 주간 주가 추이를 파일로부터 읽어오는 프로그램
import java.io.*;
import java.util.GregorianCalendar;
import java.util.Calendar;

class Answer2 {
    public static void main(String args[]) {
        ObjectInputStream in = null;
        try {
            in = new ObjectInputStream(new FileInputStream("output.dat"));
            while (true) {
                GregorianCalendar calendar = (GregorianCalendar) in.readObject();
                int year = calendar.get(Calendar.YEAR);
                int month = calendar.get(Calendar.MONTH) + 1;
                int date = calendar.get(Calendar.DATE);
            }
        }
        catch (Exception e) {
        }
    }
}
```



```

        float kospi = in.readFloat();
        System.out.println(year + "/" + month + "/" + date + " " + kospi);
    }
}
catch (FileNotFoundException fnfe) {
    System.out.println("파일이 존재하지 않습니다.");
}
catch (EOFException ioe) {
    System.out.println("끝");
}
catch (IOException ioe) {
    System.out.println("파일을 읽을 수 없습니다.");
}
catch (ClassNotFoundException cnfe) {
    System.out.println("해당 클래스가 존재하지 않습니다.");
}
finally {
    try {
        in.close();
    }
    catch (Exception e) {
    }
}
}
}

```

[실행 결과]

주간 주가 추이를 파일에 저장하는 프로그램(Answer1)을 실행하면 output.dat 파일이 생성되고, 주간 주가 추이를 파일로부터 읽어오는 프로그램(Answer2)을 실행하면 그 파일에 있는 데이터가 다음과 같이 출력됩니다.

```

명령 프롬프트
E:\#연습문제정답\10-4>java Answer1
E:\#연습문제정답\10-4>java Answer2
2006/4/10 1398.29
2006/4/11 1386.08
2006/4/12 1383.59
2006/4/13 1405.72
2006/4/14 1432.72
끝
E:\#연습문제정답\10-4>

```

10-5 *BufferedReader* 클래스 사용 연습

[소스 코드]

```

import java.io.*;
class ReaderExample1 {

```

```

public static void main(String args[]) {
    BufferedReader reader = null;
    try {
        reader = new BufferedReader(new FileReader("poem.txt"));
        while (true) {
            int data = reader.read();
            if (data == -1)
                break;
            char ch = (char) data;
            System.out.print(ch);
        }
    }
    catch (FileNotFoundException fnfe) {
        System.out.println("파일이 존재하지 않습니다.");
    }
    catch (IOException ioe) {
        System.out.println("파일을 읽을 수 없습니다.");
    }
    finally {
        try {
            reader.close();
        }
        catch (Exception e) {
        }
    }
}

```

[실행 결과]

* 이 프로그램의 실행 방법과 실행 결과는 [예제 10-3]과 동일합니다. (p.436 [그림 10-5] 참조)

10-6 BufferedOutputStream 클래스의 활용

[소스 코드]

```

import java.io.*;

class OutputStreamExample1 {
    public static void main(String args[]) {
        BufferedOutputStream out = null;
        try {
            out = new BufferedOutputStream(new FileOutputStream("output.dat"));
            byte arr[] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
                           10, 11, 12, 13, 14, 15, 16, 17, 18, 19 };
            for (int cnt = 0; cnt < arr.length; cnt++)
                out.write(arr[cnt]);
        }
        catch (IOException ioe) {
            System.out.println("파일로 출력할 수 없습니다.");
        }
    }
}

```

```

    }
    finally {
        try {
            out.close();
        }
        catch (Exception e) {
        }
    }
}
}
}

```

[실행 결과]

* 이 프로그램의 실행 방법과 실행 결과는 [예제 10-5]와 동일합니다. (p.441 [그림 10-7] 참조)

11-1 복제 가능 클래스의 구분

복제 가능 클래스	복제 불가능 클래스	
Calendar	String	ObjectInputStream
GregorianCalendar	StringBuilder	ObjectOutputStream
TimeZone	StringBuffer	BufferedReader
DataFormat	StringTokenizer	BufferedWriter
SimpleDateFormat	Math	BufferedInputStream
Date	Random	BufferedOutputStream
	FileReader	LineNumberReader
	FileWriter	PrintWriter
	FileInputStream	PrintStream
	FileOutputStream	File
	DataInputStream	
	DataOutputStream	

12-1 래퍼 클래스 관련 예외 처리

[소스 코드]

```

class WrapperExample2 {
    public static void main(String args[]) {
        try {
            int total = 0;
            for (int cnt = 0; cnt < args.length; cnt++) {
                Integer obj = new Integer(args[cnt]);
                total += obj.intValue();
            }
            System.out.println(total);
        }
        catch (NumberFormatException e) {
            System.out.println("잘못된 숫자 포맷입니다.");
        }
    }
}

```

```
}
}
```

[실행 결과]

```
E:\w\연습문제정답\w12-1>java WrapperExample2 10 20
30

E:\w\연습문제정답\w12-1>java WrapperExample2 a b
잘못된 숫자 포맷입니다.

E:\w\연습문제정답\w12-1>
```

13-1 LinkedList의 검색

[소스 코드]

```
import java.util.*;

class Answer {
    public static void main(String args[]) {
        LinkedList<String> list = new LinkedList<String>();
        list.add("머루");
        list.add("사과");
        list.add("앵두");
        list.add("자두");
        list.add("사과");
        int index1 = list.indexOf("사과");
        int index2 = list.lastIndexOf("사과");
        System.out.println("첫번째 사과: " + index1);
        System.out.println("마지막 사과: " + index2);
    }
}
```

[실행 결과]

```
첫번째 사과: 1
마지막 사과: 4
```

13-2 장바구니를 리스트로 표현하기

[소스 코드]

먼저 장바구니에 들어가는 상품 항목 클래스를 다음과 같이 선언해야 합니다.

```
// 장바구니 상품 항목 클래스
class CartItem {
    String code;    // 상품코드
    int num;        // 수량
    int price;      // 단가
}
```

```

    CartItem(String code, int num, int price) {
        this.code = code;
        this.num = num;
        this.price = price;
    }
}

```

위와 같이 선언된 클래스를 타입 파라미터로 삼아서 리스트를 만들면 그 리스트를 장바구니로 사용할 수 있습니다. 그런데 인터넷 쇼핑몰의 장바구니에는 추가와 삭제가 빈번히 일어날 수 있으므로 리스트 클래스로는 ArrayList 클래스보다 LinkedList 클래스를 사용하는 것이 더 적합합니다. 다음은 그런 식으로 장바구니를 만들어서 항목을 추가하고 삭제하는 예를 보여주는 프로그램입니다.

// 장바구니를 리스트 자료 구조로 표현하는 예를 보여주는 프로그램

```

import java.util.*;
class ShoppingProgram {
    public static void main(String args[]) {
        LinkedList<CartItem> list = new LinkedList<CartItem>();
        list.add(new CartItem("50001", 2, 2000));    // 장바구니에
        list.add(new CartItem("73505", 1, 7000));    // 세 개의 항목을
        list.add(new CartItem("88012", 3, 25000));    // 추가합니다.
        list.remove(1);                                // 장바구니에서 두 번째 항목을 제거합니다.
        System.out.println("상품코드  수량  가격");
        System.out.println("-----");
        for (CartItem item : list)
            System.out.printf("%5s %8d %8d %n", item.code, item.num, item.price);
    }
}

```

* CartItem 클래스를 이용하여 리스트를 만드는 것을 보여주지만 하면 프로그램은 위와 다르게 작성해도 됩니다.

[실행 결과]

상품코드	수량	가격
50001	2	2000
88012	3	25000

13-3 LinkedList를 스택으로 사용하기

[소스 코드]

```

import java.util.*;
class Answer {
    public static void main(String args[]) {
        LinkedList<Integer> stack = new LinkedList<Integer>();
        stack.addFirst(new Integer(12));
        stack.addFirst(new Integer(59));
    }
}

```

```
stack.addFirst(new Integer(7));
while(!stack.isEmpty()) {
    Integer num = stack.removeFirst();
    System.out.println(num);
}
}
```

[실행 결과]

```
7
59
12
```

13-4 해쉬 테이블 연습

[소스 코드]

먼저 전화번호와 주소 필드를 갖는 다음과 같은 클래스를 선언해야 합니다.

```
// 연락처 클래스
class ContactInfo {
    String phoneNo;    // 전화번호
    String address;    // 주소
    ContactInfo(String phoneNo, String address) {
        this.phoneNo = phoneNo;
        this.address = address;
    }
}
```

그리고 나서 이름(String 타입)을 키로 삼고, 위 클래스의 객체를 데이터로 삼는 해쉬 테이블을 만들어서 사용하면 됩니다. 다음은 그런 예를 보여주는 프로그램입니다.

```
// 이름을 검색하는 프로그램
import java.util.HashMap;
class Answer {
    public static void main(String args[]) {
        HashMap<String, ContactInfo> hashtable = new HashMap<String, ContactInfo>();
        hashtable.put("홍길동", new ContactInfo("산속 오두막", "010-0000-0000"));
        hashtable.put("연흥부", new ContactInfo("강남", "02-300-0000"));
        hashtable.put("연놀부", new ContactInfo("청담동", "02-500-0000"));
        ContactInfo obj = hashtable.get("연놀부");
        System.out.println("연놀부의 연락처");
        System.out.println("전화번호: " + obj.phoneNo);
        System.out.println("주소: " + obj.address);
    }
}
```

[실행 결과]

연놀부의 연락처
전화번호: 청담동
주소: 02-500-0000

16-1 로컬 이너 클래스 파일의 이름

[컴파일 결과]

ContactInfo라는 이름의 로컬 이너 클래스를 포함한 [예제 16-13]을 컴파일해보면 다음과 같은 두 개의 클래스 파일이 생성됩니다.

```

E:\work\chap16\16-2-3\example2>javac NestedClassExample7.java

E:\work\chap16\16-2-3\example2>dir
E 드라이브의 볼륨: 로컬 디스크
볼륨 일련 번호: 305B-17D4

E:\work\chap16\16-2-3\example2 디렉터리

2006-09-09  06:43p    <DIR>          .
2006-09-09  06:43p    <DIR>          ..
2006-09-09  07:14p                1,083 NestedClassExample7.java
2006-09-15  08:02p                501 NestedClassExample7$1ContactInfo.class
2006-09-15  08:02p                1,317 NestedClassExample7.class
                3개 파일                2,901 바이트
                2 디렉터리            50,266,112 바이트 남음

E:\work\chap16\16-2-3\example2>
  
```

이 중에서 NestedClassExample7\$1ContactInfo.class라는 이름의 파일이 로컬 이너 클래스인 ContactInfo에 해당하는 클래스 파일인데, 이 이름은 인클로징 클래스의 이름 뒤에 \$와 1이라는 숫자를 붙이고, 그 뒤에 로컬 이너 클래스의 이름을 붙이고, .class라는 확장자를 붙인 이름입니다. 로컬 이너 클래스의 이름이 이렇게 인클로징 클래스와 이너 클래스 이름 외에 숫자를 포함하는 이유는 하나의 인클로징 클래스 안에 똑같은 이름의 로컬 이너 클래스가 여러 개 있을 수 있기 때문입니다. 예를 들어 NestedClassExample7라는 클래스 안에 ContactInfo라는 이름의 로컬 이너 클래스가 하나 더 있다면, 그 클래스의 클래스 파일은 NestedClassExample7\$2ContactInfo.class라는 이름으로 생길 것입니다.

16-2 이름 없는 이너 클래스의 클래스 파일 이름

[컴파일 결과]

[예제 16-7]을 컴파일하면 다음과 같이 NestedClassExample10.class라는 이름의 클래스 파일과 NestedClassExample10\$1.class라는 이름의 클래스 파일들이 생성되는데 이 중에서 후자가 로컬 이너 클래스의 클래스 파일입니다.

```

명령 프롬프트
E:\work\chap16\16-2-3\example5>dir
E 드라이브의 볼륨: 로컬 디스크
볼륨 일련 번호: 305B-17D4

E:\work\chap16\16-2-3\example5 디렉터리

2006-09-09  06:43p    <DIR>          .
2006-09-09  06:43p    <DIR>          ..
2006-09-09  07:18p                72 Player.java
2006-09-09  07:20p            416 NestedClassExample10.java
2006-09-15  08:10p            810 NestedClassExample10$1.class
2006-09-15  08:10p            473 NestedClassExample10.class
2006-09-15  08:10p            153 Player.class
                5개 파일            1,924 바이트
                2 디렉터리        50,241,536 바이트 남음

E:\work\chap16\16-2-3\example5>cls
  
```

이름 없는 이너 클래스의 경우에는 이렇게 인클로징 클래스 이름 뒤에 \$와 숫자를 붙이고, 그 뒤에 .class라는 확장자를 붙인 이름으로 클래스 파일이 생성됩니다. 이 숫자는 같은 클래스 안에 이름 없는 이너 클래스가 여러 개 있을 경우 이를 구분하기 위해 붙여지는 숫자입니다.

17-1 예제 실행

[실행 결과]

[예제 17-22]의 두 Rectangle 클래스는 getArea 메소드가 있고 없고의 차이는 있지만 버전번호가 같기 때문에 어느 Rectangle 클래스를 가지고 [예제 17-22]의 직렬화 프로그램을 실행하든, 역직렬화 프로그램을 실행하든 에러 없이 의도했던 결과가 출력됩니다.

18-1 멀티스레드 프로그램 연습

[소스 코드]

// main 메소드를 포함하는 클래스

```

class MultithreadExample3 {
    public static void main(String args[]) {
        Thread thread = new Thread(new SmallLetters()); // 쓰레드를 생성
        thread.start(); // 쓰레드를 시작
        char arr[] = { 'ㄱ', 'ㄴ', 'ㄷ', 'ㄹ', 'ㅁ', 'ㅂ', 'ㅅ',
                       'ㅇ', 'ㅈ', 'ㅊ', 'ㅋ', 'ㅌ', 'ㅍ', 'ㅎ' };
        for (char ch : arr) {
            System.out.print(ch);
            try {
                Thread.sleep(500);
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
    }
}
// 영문 소문자를 출력하는 클래스
  
```



```
class SmallLetters implements Runnable {
    public void run() {
        for (char ch = 'a'; ch <= 'z'; ch++) {
            System.out.print(ch);
            try {
                Thread.sleep(700);
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
    }
}
```

[실행 결과]

위와 같이 수정된 프로그램을 실행하면 Thread.sleep 메소드 호출로 벌어진 시간 간격으로 인해 한글 자음과 영문 소문자가 좀더 고르게 섞여서 출력됩니다.



* 이 프로그램은 멀티스레드 프로그램이기 때문에 독자 여러분의 컴퓨터에서 이 프로그램을 실행했을 때는 위와 똑같은 결과가 나오지 않을 수도 있습니다.

19-1 한글 지원 테스트

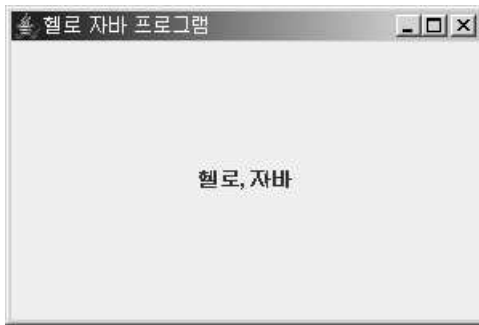
[소스 코드]

```
import java.awt.*;
import javax.swing.*;

class WindowExample1 {
    public static void main(String[] args) {
        JFrame frame = new JFrame("헬로 자바 프로그램");
        frame.setLocation(500, 400);
        frame.setPreferredSize(new Dimension(300, 200));
        Container contentPane = frame.getContentPane();
        JLabel label = new JLabel("헬로, 자바", SwingConstants.CENTER);
        contentPane.add(label);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);
    }
}
```

[실행 결과]

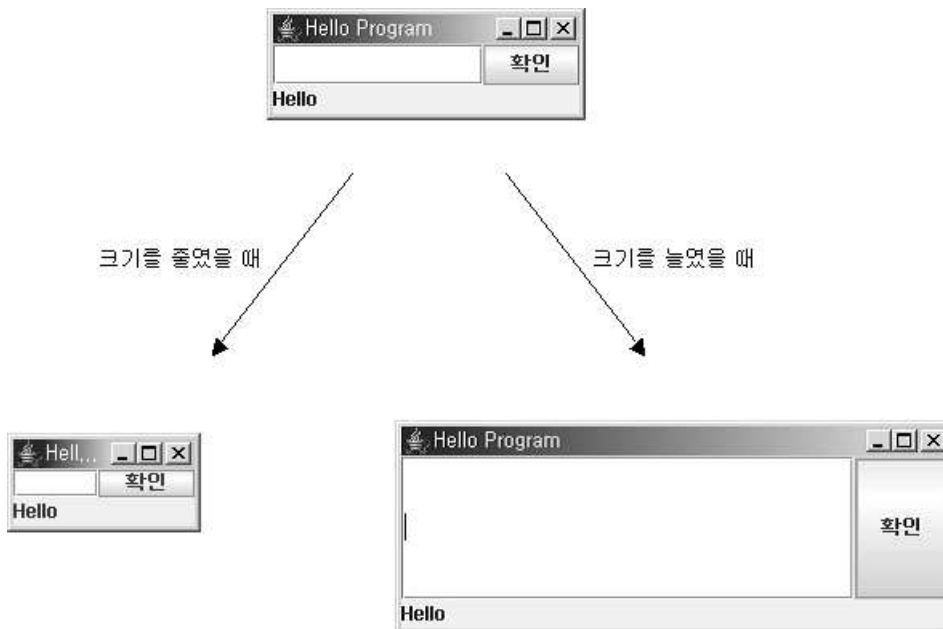
다음과 같은 윈도우가 모니터 화면 중앙에 나타납니다.



19-2 border layout의 특성 테스트

[실행 결과]

윈도우의 크기를 늘이거나 줄이면 오른쪽 버튼의 폭과 아래쪽 라벨의 높이를 바뀌지 않고, 중앙에 있는 텍스트 상자의 크기는 윈도우 크기에 따라서 늘어되거나 줄어듭니다.



19-3 윈도우로 작성하는 덧셈 프로그램

[소스 코드]

```
// 윈도우로 작동하는 덧셈 프로그램
import javax.swing.*;
import java.awt.*;

class Answer {
    public static void main(String[] args) {
        JFrame frame = new JFrame("덧셈 프로그램");
        frame.setLocation(500, 400);
        Container contentPane = frame.getContentPane();
        JPanel pane1 = new JPanel();
```

```

        JPanel pane2 = new JPanel();
        contentPane.add(pane1, BorderLayout.CENTER);
        contentPane.add(pane2, BorderLayout.SOUTH);
        pane1.setLayout(new FlowLayout());
        JTextField text1 = new JTextField(5);
        JTextField text2 = new JTextField(5);
        JTextField text3 = new JTextField(5);
        pane1.add(text1);
        pane1.add(new JLabel("+"));
        pane1.add(text2);
        pane1.add(new JLabel("="));
        pane1.add(text3);
        pane2.setLayout(new FlowLayout());
        JButton button1 = new JButton("확인");
        JButton button2 = new JButton("취소");
        pane2.add(button1);
        pane2.add(button2);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        ConfirmButtonActionListener actionListener1 =
            new ConfirmButtonActionListener(text1, text2, text3);
        CancelButtonActionListener actionListener2 =
            new CancelButtonActionListener(text1, text2, text3);
        button1.addActionListener(actionListener1);
        button2.addActionListener(actionListener2);
        frame.pack();
        frame.setVisible(true);
    }
}

// 확인 버튼을 처리하는 이벤트 리스너 클래스
import javax.swing.*;
import java.awt.event.*;

class ConfirmButtonActionListener implements ActionListener {
    JTextField text1, text2, text3;
    ConfirmButtonActionListener(JTextField text1, JTextField text2, JTextField text3) {
        this.text1 = text1;
        this.text2 = text2;
        this.text3 = text3;
    }
    public void actionPerformed(ActionEvent e) {
        int num1 = Integer.parseInt(text1.getText());
        int num2 = Integer.parseInt(text2.getText());
        int sum = num1 + num2;
        text3.setText(sum+"");
    }
}
}

```

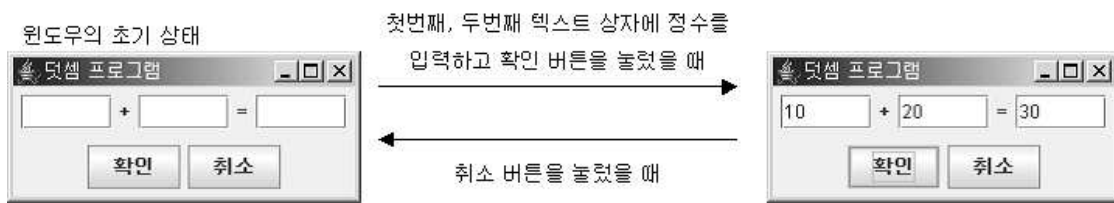
```
// 취소 버튼을 처리하는 이벤트 리스너 클래스
import javax.swing.*;
import java.awt.event.*;

class CancelButtonActionListener implements ActionListener {
    JTextField text1, text2, text3;

    CancelButtonActionListener(JTextField text1, JTextField text2, JTextField text3) {
        this.text1 = text1;
        this.text2 = text2;
        this.text3 = text3;
    }

    public void actionPerformed(ActionEvent e) {
        text1.setText("");
        text2.setText("");
        text3.setText("");
    }
}
```

[실행 결과]



* 이 프로그램을 좀 더 완벽하게 만들려면 세 번째 텍스트 상자에 키보드 입력이 안되도록 만들어야 합니다. 그런 일은 이 프로그램의 main 메소드에서 윈도우를 띄우는 명령문 전에 다음과 같은 명령문을 써넣음으로써 할 수 있습니다.

```
text3.setEditable(false);
```

19-4 여러 가지 색으로 그림 그리기

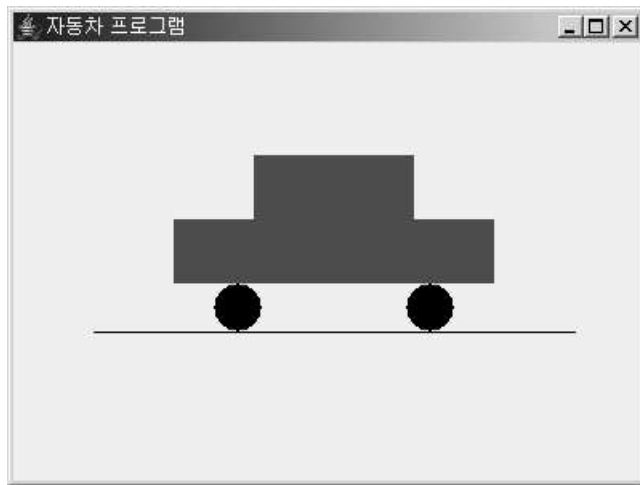
[소스 코드]

자동차 그림이 있는 패널 클래스만 다음과 같이 수정하면 됩니다.

```
import javax.swing.*;
import java.awt.*;

class CarDrawingPanel extends JPanel {
    public void paint(Graphics g) {
        g.setColor(Color.RED);
        g.fillRect(100, 110, 200, 40);
        g.fillRect(150, 70, 100, 40);
        g.setColor(Color.BLACK);
        g.fillRect(125, 150, 30, 30);
        g.fillRect(245, 150, 30, 30);
        g.drawLine(50, 180, 350, 180);
    }
}
```

[실행 결과]



19-5 paint 메소드는 언제 호출되나?

[실행 결과]

[예제 19-11]의 DrawingPanel의 paint 메소드 마지막 부분에 `System.out.println("paint 메서드가 호출되었습니다.");`라는 명령문을 넣으면 다음과 같은 일이 일어날 때마다 "paint 메서드가 호출되었습니다."라는 메시지가 출력됩니다.

- 윈도우가 처음 뜰 때
- 그래프 그리기 버튼을 눌렀을 때 (repaint 메소드에 의해 간접적으로 호출되는 경우임)
- 윈도우(정확히 말하면 DrawingPanel 부분)의 일부 또는 전부가 가려졌다가 다시 나타날 때
- 윈도우가 아이콘화되었다가 다시 나타날 때
- 윈도우가 최대화될 때
- 윈도우가 최대화되었다가 다시 원래의 크기로 돌아올 때
- 윈도우의 크기를 크게 또는 작게 변경할 때

21-1 테이블 생성하기와 데이터 저장

mysql 프로그램을 이용하여 `mailldb` 데이터베이스로 들어간 후 다음과 같은 `create` 문과 `insert` 문을 이용하여 테이블을 생성하고 데이터를 입력하면 됩니다.

```

mysql> create table stockinfo (
  -> code char(5) not null,
  -> num int(5) not null,
  -> primary key(code)
  -> );
Query OK, 0 rows affected (0.08 sec)

mysql> insert into stockinfo (code, num) values('10001', 100);
Query OK, 1 row affected (0.04 sec)

mysql> insert into stockinfo (code, num) values('10002', 200);
Query OK, 1 row affected (0.02 sec)

mysql> insert into stockinfo (code, num) values('10003', 50);
Query OK, 1 row affected (0.03 sec)

mysql> insert into stockinfo (code, num) values('10004', 100);
Query OK, 1 row affected (0.03 sec)

mysql> insert into stockinfo (code, num) values('10005', 150);
Query OK, 1 row affected (0.03 sec)

mysql>

```

데이터가 올바르게 입력되었는지는 다음과 같은 select 문을 이용하여 확인할 수 있습니다.

```

mysql> select * from stockinfo;
+-----+-----+
| code | num |
+-----+-----+
| 10001 | 100 |
| 10002 | 200 |
| 10003 | 50  |
| 10004 | 100 |
| 10005 | 150 |
+-----+-----+
5 rows in set (0.02 sec)

mysql>

```

21-2 select 문 연습

[실행 결과]

실행 결과는 다음과 같습니다.

```

mysql> select * from goodsinfo where maker = 'LG';
+-----+-----+-----+-----+
| code | name       | price | maker |
+-----+-----+-----+-----+
| 10001 | 디지털 TV  | 350000 | LG    |
| 10002 | DVD 플레이어 | 250000 | LG    |
+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> select * from goodsinfo where name like '디지털%';
+-----+-----+-----+-----+
| code | name       | price | maker |
+-----+-----+-----+-----+
| 10001 | 디지털 TV  | 350000 | LG    |
| 10003 | 디지털 카메라 | 210000 | 삼성  |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from goodsinfo where maker = '삼성' and price < 250000;
+-----+-----+-----+-----+
| code | name       | price | maker |
+-----+-----+-----+-----+
| 10003 | 디지털 카메라 | 210000 | 삼성  |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>

```

21-3 select 문의 결과를 출력하기

mysql 프로그램을 이용하여 문제의 select 문을 실행하면 다음과 같은 결과가 출력됩니다.

```

mysql> select goodsinfo.code, name, num from goodsinfo, stockinfo where goodsinfo.code = stockinfo.code;
+-----+-----+-----+
| code | name       | num |
+-----+-----+-----+
| 10001 | 디지털 TV  | 100 |
| 10002 | DVD 플레이어 | 200 |
| 10003 | 디지털 카메라 | 50  |
| 10004 | 컬러 전자사전 | 100 |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>

```

[소스 코드]

```

import java.sql.*;

class Answer {
    public static void main(String args[]) {
        Connection conn = null;
        Statement stmt = null;
        try {
            Class.forName("com.mysql.jdbc.Driver");

```

```

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/malldb",
                                   "root", "");

stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery(
    "select goodsinfo.code, name, num from goodsinfo, stockinfo " +
    "where goodsinfo.code = stockinfo.code;");
System.out.println("상품코드 상품명 Wt재고수량");
System.out.println("-----");
while (rs.next()) {
    String code = rs.getString("goodsinfo.code");
    String name = rs.getString("name");
    int num = rs.getInt("num");
    System.out.printf("%8s %s Wt%8d %n", code, toUnicode(name), num);
}
}
catch (ClassNotFoundException cnfe) {
    System.out.println("해당 클래스를 찾을 수 없습니다." + cnfe.getMessage());
}
catch (SQLException se) {
    System.out.println(se.getMessage());
}
finally {
    try {
        stmt.close();
    }
    catch (Exception ignored) {
    }
    try {
        conn.close();
    }
    catch (Exception ignored) {
    }
}
}

private static String toUnicode(String str) {    // ISO-8859-1 문자열 -> Unicode 문자열
    try {
        byte[] b = str.getBytes("ISO-8859-1");
        return new String(b);
    }
    catch (java.io.UnsupportedEncodingException uee) {
        System.out.println(uee.getMessage());
        return null;
    }
}
}
}
[실행 결과]

```



```

E:\연습문제정답\21-3>java Answer
상품코드 상품명 재고수량
-----
10001 디지털 TV 100
10002 DVD 플레이어 200
10003 디지털 카메라 50
10004 컬러 전자사전 100
E:\연습문제정답\21-3>

```

21-4 데이터 검색 프로그램

[소스 코드]

[예제 21-3] 14행의 executeQuery 문을 다음과 같이 수정하면 됩니다.

```

ResultSet rs = stmt.executeQuery(
    "select code, name, price, maker " +
    "from goodsinfo where name like '" +
    toLatin1(args[0]) + "%';");

```

[실행 결과]

```

E:\연습문제정답\21-4>java JDBCExample3 디지털
상품코드 상품명 가격 제조사
-----
10001 디지털 TV 350000 LG
10003 디지털 카메라 210000 삼성
E:\연습문제정답\21-4>

```

21-5 데이터 입력 예러 제거하기

[소스 코드]

```

import java.sql.*;
class JDBCExample4 {
    public static void main(String args[]) {
        if (args.length != 4) {
            System.out.println("Usage: java JDBCExample4 상품코드 상품명 가격 제조사");
            return;
        }
        Connection conn = null;
        Statement stmt1 = null;
        Statement stmt2 = null;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/malldb",

```

```

        "root", "");

stmt1 = conn.createStatement();
ResultSet rs = stmt1.executeQuery("select code from goodsinfo where code='"
        + toLatin1(args[0]) + "';");

stmt2 = conn.createStatement();
if (rs.next()) {
    int rowNum = stmt2.executeUpdate("update goodsinfo set " +
        "name='" + toLatin1(args[1]) + "', " +
        "price='" + toLatin1(args[2]) + "', " +
        "maker='" + toLatin1(args[3]) + "' " +
        "where code = '" + toLatin1(args[0]) + "';");
    System.out.println(rowNum + "행이 수정되었습니다.");
}
else {
    int rowNum = stmt2.executeUpdate(
        "insert into goodsinfo (code, name, price, maker) values('" +
        toLatin1(args[0]) + "', '" +
        toLatin1(args[1]) + "', " +
        toLatin1(args[2]) + "', '" +
        toLatin1(args[3]) + "')");
    System.out.println(rowNum + "행이 추가되었습니다.");
}
}

catch (ClassNotFoundException cnfe) {
    System.out.println("해당 클래스를 찾을 수 없습니다." + cnfe.getMessage());
}

catch (SQLException se) {
    System.out.println(se.getMessage());
}

finally {
    try {
        stmt1.close();
    }
    catch (Exception ignored) {
    }
    try {
        stmt2.close();
    }
    catch (Exception ignored) {
    }
    try {
        conn.close();
    }
    catch (Exception ignored) {
    }
}
}

```

```

    }
    private static String toLatin1(String str) {    // Unicode 문자열 -> ISO-8859-1 문자열
        try {
            byte[] b = str.getBytes();
            return new String(b, "ISO-8859-1");
        }
        catch (java.io.UnsupportedEncodingException uee) {
            System.out.println(uee.getMessage());
            return null;
        }
    }
}

```

[실행 결과]

이 프로그램으로 기존에 있던 데이터와 똑같은 상품 코드의 데이터를 입력했을 때의 결과는 다음과 같습니다.

