연습문제 이론 홀수 정답

1장 연습문제

이론문제

1. 자바 언어 소스 파일의 확장자는 .java이고 컴파일하면 .class 파일이 된다.

3. 자바는 플랫폼에 독립적이며 네트워크에 연결된 어느 클라이언트에서도 실행이 가능하다. 한 번 프로그램을 작성하면 어느 하드웨어에서나 또는 어느 운영체제에서나 자바 프로그램을 실행시킬 수 있는 것을 WORA(Write Once Run Anywhere)라고 한다.

5. JRE는 실행 환경으로 자바 가상 기계를 포함하고 있으며 자바 실행 환경만 필요한 경우에 사용되며, JDK는 자바 컴파일러, 도구, 라이브러리 등 자바 응용프로그램 개발에 필요한 모든 것을 포함하며 JRE도 JDK에 포함된다. 따라서 자바 응용프로그램 개발을 위해서는 JDK가 필요하다.

7. ③, ⑦이 잘못되었다. 하나의 클래스 파일에는 오직 하나의 클래스만 포함할 수 있다. 패지키의 목적은 서로 관련 있는 클래스를 묶어 관리하기 위함이다.

9. Add.java

2장 연습문제

이론문제

1. 자바에서 클래스를 선언할 때는 class 키워드를 사용한다.

3. 잘못된 규칙은 다음과 같다.

① '@', '#', '!', '_', '$'와 같은 특수문자, 공백(탭, 스페이스 등)은 식별자로 사용할 수 없다. → 특수문자(%, *, &, @, ^ 등), 공백(탭, space 등)은 식별자로 사용할 수 없으나 '_', '$'는 사용 가능하다.

② 식별자로 한글을 사용할 수 없다. → 식별자로 한글이 사용 가능하다.

⑦ 식별자의 길이는 128자를 넘을 수 없다. → 길이 제한이 없다.

5. 자바의 기본 데이터 타입과 크기는 다음과 같다.

- boolean - 1 byte
- char - 2 bytes
- byte - 1 byte
- short - 2 bytes
- int - 4 bytes
- long - 8 bytes
- float - 4 bytes
- double - 8 bytes

7. b = b + 100;
또는 b += 100;

9. 수식의 결과 값과 타입은 다음과 같다.

(1) 67 + 12.8의 결과 값과 타입은 double 타입의 79.8

(2) 'c' + 1의 결과 값과 타입은 int 타입의 100

(3) 10/3의 결과 값과 타입은 int 타입의 3

(4) 10.0/3의 결과 값과 타입은 double 타입의 3.3333333333333335

(5) 10==9의 결과 값과 타입은 boolean 타입의 false

11. 각 문장을 조건식으로 나타내면 다음과 같다.

(1) a는 b보다 크거나 같다. → a >= b 또는 a > b || a == b

(2) a는 b보다 작고 c보다 크다. → a < b && a > c

(3) a 더하기 3은 10과 같지 않다. → (a + 3) != 10
(4) a는 10보다 크거나 b와 같다. → a > 10 || a == b

13. if와 else 사이에 실행 문장이 2개 이상인 경우에는 {, }으로 실행 문장들을 둘러싸야 한다. 따라서 올바른 코드는 다음과 같다.

```
int j, k;
if (i >0 && i < 10) {
   j *= 2;
   k += 3;
}
else if (i >= 10 && i < 20) {
   j /= 2;
} else
   k -= 2;
```

15. 각 case 문에 break가 없어 모든 case 문과 default 문까지 실행되어 j 값은 -6이 된다.

3장 연습문제

이론문제

1. for 문에 ';'이 있으면 실행 문장이 없는 것으로 간주되어 제대로 실행되지 않는다. 따라서 다음과 같이 수정되어야 한다.

```
int i = 0;
for (int j = 0; j < 10; j++)
   i = i + 1;
```

3. do-while 문으로 바꾸면 다음과 같다.

```
int j = 0, k = 0;
do {
   System.out.println(k);
   j++;
   k += 3;
} while (j < 10);
```

5. for 문의 초기문에서 선언된 변수는 for 문 내에서만 유효한 지역 변수이며 for 문 밖으로 벗어나면 사라지므로 for 문 밖에 있는 출력문에서 변수 j를 사용할 수 없다. 변수 j는 for 문 밖에서 선언 및 초기화되어야 한다.

7. continue를 break로 바꾸면 결과는 48이다.

9. 출력 결과는 "5일째 4시까지 누적된 시간이 100시간 입니다."이다.

11. int i[] = {0,1,2,3,4,5};
또는 int[] i = {0,1,2,3,4,5};

13. 배열을 선언할 때 크기를 지정할 수 없으므로 int myArray[10]이 잘못되었다.
int myArray[]와 같이 선언해야 한다.

15. 그림과 같은 비정방형 배열의 선언은 다음과 같다.

```
float f[][] = new float[4][]; // 각 행의 레퍼런스 배열 생성
f[0] = new float[4]; // 첫째 행의 4개 실수 배열 생성
f[1] = new float[2]; // 둘째 행의 2개 실수 배열 생성
f[2] = new float[3]; // 셋째 행의 3개 실수 배열 생성
f[3] = new float[4]; // 넷째 행의 4개 실수 배열 생성
```

17. 프로그램을 실행하면 ArrayIndexOutOfBoundsException이 발생한다. 다음과 같이 try-catch 문으로 예외 처리를 해야 한다.

```
public class ExceptionTest {
   public static void main(String[] args){
      int[] intArray = new int[10];
      try {
         for (int i = 0; i <= intArray.length; i++) {
            intArray[i] = i;
            System.out.println("intArray["+i+"]"+"="+intArray[i]);
         }
      } catch (ArrayIndexOutOfBoundsException e) {
         System.out.println("배열의 범위를 벗어났습니다.");
      }
   }
}
```

4장 연습문제

이론문제

1. 클래스는 객체를 생성하기 위한 설계도 또는 틀이라고 볼 수 있고 객체는 설계도 또는 틀로 찍어낸 실체이다. 따라서 클래스는 객체들이 어떤 특성을 갖는다고 정의만 하고 값은 가질 수 없으나, 객체는 각각 자신만의 고유한 속성 값을 갖는다.

3. Wine 클래스는 다음과 같다.

```
class Wine {
   private String manufacturer;
   private String name;
   private String country;
   private String region;
   private String kind;
   private int year;
   private int grade;
}
```

5. 클래스 Car는 객체 지향의 캡슐화의 특성에 따라 선언되었다. 따라서 모든 멤버 필드는 private로 선언되어 클래스 외부에선 접근할 수 없으며, 오로지 공개된 멤버 함수를 통해서만 필드의 값을 읽거나 쓸 수 있다.

7. Person 클래스의 private 멤버 필드를 외부에서 직접 접근하면 접근 오류가 발생한다. 다음과 같이 Person 클래스의 멤버 필드는 public으로 선언하면 Example 클래스는 수정하지 않고 사용할 수 있다.

```
class Person {
   public String name;
   public int age;
}
```

그러나 클래스의 멤버 필드를 외부에 공개하는 것은 객체 지향의 캡슐화에 맞지 않으므로 멤버 필드를 접근할 수 있는 메소드를 추가하고 Example 클래스의 main() 메소드에서 이 공개 메소를 이용하여 멤버 필드를 접근하는 것이 객체 지향 프로그래밍에

적합하다. 수정된 소스는 다음과 같다.

```
class Person {
   private String name;
   private int age;
   public void setName(String s) {
      name = s;
   }
   public String getName() {
      return name;
   }
   public void setAge(int i) {
      age = i;
   }
   public int getAge() {
      return age;
   }
}
public class Example {
   public static void main (String args[]) {
      Person aPerson = new Person();
      aPerson.setName("홍길동“);
      a,Person.setAge(17);
   }
}
```

9. 두 개의 doAdd()의 인자의 수와 타입이 모두 같고 메소드 이름도 같지만 리턴 타입만 다르므로 메소드 오버로딩은 실패이다.

11. setNum() 메소드에서 메소드의 인자 num과 클래스 Example의 num 필드와 이름이 같아 모호하다. 메소드 setNum()을 다음과 같이 수정해야 한다.

```
public  void setNum(int num) {
   this.num = num;
}
```

13. main() 메소드의 a = b; 문장에서 가비지가 발생한다. 레퍼런스 변수 a는 원래 가리키던 객체 대신 b가 가리키던 객체를 가리키게 되어 원래 객체는 더 이상 참조

되지 않아 가비지가 된다.

15. Sample 클래스의 getId(), setId() 메소드는 static 함수이다. static 메소드에서는 오직 static 멤버만 접근할 수 있으므로 non-static 멤버인 id를 getId() 메소드에서 접근하는 것은 오류이다. 또한 static 메소드는 객체가 생성되지 않는 상황에서도 호출이 가능하기 때문에, 현재 실행 중인 객체를 가리키는 this 레퍼런스를 사용할 수 없다.

17. 가비지가 발생하지 않는다. a와 b에 null 값을 대입하여도 앞서 a가 참조하던 객체는 c가 여전히 참조하고 있으므로 가비지가 되지 않는다.

5장 연습문제

이론문제

1. ③. 자바에서 상속받은 클래스를 다시 상속받는 것은 가능하며 상속의 횟수에 제한을 두지 않는다.

3. 클래스 A가 슈퍼 클래스, 클래스 B가 서브 클래스이다.

5. ③. 모든 클래스의 최상위 슈퍼 클래스는 java.lang.Object이다.

7. 클래스 B의 생성자와 짝을 이룰 기본 생성자가 클래스 A에 없기 때문에 컴파일 오류가 발생된다. 해결 방법은 다음과 같이 클래스 A에 직접 기본 생성자를 작성해야 하거나

```
class A {
   private int a;
   public A() {
      a = 0;
   }
   public A(int i) {
      a = i;
   }
}
```

또는 클래스 B의 생성자에서 명시적으로 클래스 A의 생성자를 호출하도록 수정해야 한다.

```
class B extends  A {
   private int b;
   public B() {
      super(0);
      b = 0;
   }
}
```

9. ②. protected 멤버는 다른 패키지의 서브 클래스에서도 접근 가능하다.

11. 출력 결과는 다음과 같다.

```
false
false
true
false
true
true
true
true
```

13. ④. 메소드 오버라이딩은 메소드를 실행 시에 결정하는 동적 바인딩이 발생한다.

15. 클래스 B는 추상 클래스 A를 상속받아 추상 메소드 getB()만을 구현하고 getA()는 구현하지 않아 여전히 추상 메소드로 남아 있다. 따라서 클래스 B도 추상 클래스로 선언되어야 한다.

17. 추상 클래스나 인터페이스의 객체를 생성하려고 할 때 발생하는 오류 메시지이다.

6장 연습문제

이론문제

1. 자바에서 패키지란 서로 관련 있는 클래스나 인터페이스의 컴파일된 클래스(.class) 파일들을 한곳에 묶어 놓은 것을 말한다.

3. java.lang 패키지는 import하지 않고 사용할 수 있다.

5. import 문 없이 동작하려면 본문에 클래스의 전체 경로를 지정해야 한다. 소스는 다음과 같다.

```
public class Example {
   public static void main(String[] args) {
      java.util.StringTokenizer st = new java.util.StringTokenizer("라디오/카메라/
                                                                  오디오", "/");
      while (st.hasMoreTokens())
         System.out.println(st.nextToken());
   }
}
```

7. 결과는 false이다. == 연산자는 주어진 두 객체의 레퍼런스를 단순히 같은지 비교하고 내용에 대해서는 비교하지 않는다. 따라서 str1과 str2는 서로 다른 객체이므로 false이다. 내용에 대해 비교를 하려면 String 클래스의 equals() 메소드를 이용한다.

9. 스트링 버퍼에서 "bad"를 삭제하므로 출력 결과는 "programming"이다.

11. (1) int i = Integer.parseInt("20");
(2) double d = Double.parseDouble("35.9");
(3) boolean b = Boolean.parseBoolean("true");
(4) String s = Integer.toBinaryString(30);
(5) String s = Integer.toHexString(50);

13. (1) b
(2) c,e

7장 연습문제

이론문제

1. 컬렉션

3. ④. Map은 Collection 인터페이스를 상속 받지 않는다.

5.

```
Vector<String> v = new Vector<String>();
```

7.

```
Vector<Integer> v = new Vector<Integer>();
v.add(100); // auto boxing
int k = v.get(0); // auto unboxing
```

9. 최종적으로 출력되는 벡터의 용량은 12이다. 벡터의 초기 용량은 3이며 루프가 4번째 돌 때 벡터의 공간이 부족하므로 현재 벡터의 용량의 2배로 용량을 증가시킨다. 그러므로 용량이 6이 된다. 다시 7번째 루프를 돌 때 용량이 12가 되며, 10번 루프를 돌았을 때의 용량은 여전이 12이다. 코드를 아래와 같이 고쳐서 실행해 보면 더욱 분명해진다.

```
Vector<Integer> v = new Vector<Integer>(3);
for(int i=0; i<10; i++) {
   v.add(i);
   System.out.println(v.capacity());
}
```

11. new T();를 할 수 없다.

8장 연습문제

이론문제

1. ②. 스트림은 데이터 처리를 위하여 몇 개라도 연결이 가능하다.

3. 문자 스트림 클래스는 문자로 인식되는 데이터, 즉 텍스트 파일만 처리할 수 있다. 그러므로 음악, 이미지 등과 같은 바이너리 파일을 문자 스트림으로 읽어서 다시 문자 스트림으로 출력하여 저장하면 원래의 데이터가 그대로 보존되지 않는다.

5. File 클래스는 파일 삭제, 디렉터리 생성 등과 같은 중요한 파일 관리 작업을 지원한다.

7. 빈칸을 채운 코드는 다음과 같다.

```
BufferedOutputStream bout = new BufferedOutputStream(System.out, 50);
FileReader fin = new FileReader("c:\\tmp\\sample.txt");
int c;
while ((c = fin.read()) != -1) {
   bout.write((char)c);
}
fin.close();
bout.close();
```

9장 연습문제

이론문제

1. 자바는 AWT 패키지와 스윙 패키지 두 가지를 지원한다.

3. AWT에서 지원하는 컴포넌트는 몇 개 되지 않고 모양이 그다지 예쁘지 않다. 그러나 스윙은 AWT에서 지원하는 컴포넌트를 모두 지원하고 추가적으로 많은 새로운 컴포넌트를 제공한다. 이들 컴포넌트들은 예쁘고 화려하며 기능적 완성도가 높아서 개발자가 응용프로그램을 작성하기 매우 용이하다.

5. 정답은 다음과 같다.

```
import javax.swing.*;
import java.awt.*;
public class MyFrame extends JFrame {
   MyFrame() {
      Container c = getContentPane(); // 컨텐트팬을 알아낸다.
      c.setBackground(Color.BLUE); // 컨텐트팬의 바탕색을 파란색으로 설정한다.
      setSize(300,300);
      setVisible(true);
   }
}
```

7. 틀린 보기는 다음과 같다.
   ① 배치관리자는 4개 이상의 여러 개가 있으며 사용자가 새로운 배치관리자를 만들 수 있다.
   ④ 배치관리자는 자신이 소속된 컨테이너의 크기를 조절할 수 없다. 다만 컨테이너의 자식 컴포넌트들의 크기를 조절한다.
   ⑦ 배치관리자는 한 컨테이너에 오직 하나만 설정될 수 있다.
   ⑧ 컨테이너가 배치관리자를 가지지 않도록 할 수 있다.

9. 컨텐트팬은 JFrame, JDialog 등과 같이 스윙의 최상위 컨테이너들만 가지는 것으로 최상위 컨테이너에 스윙 컴포넌트들이 특별히 부착되는 전용 공간 또는 전용 컨테이너이다.

10장 연습문제

이론문제

1. 이벤트 소스란 이벤트를 발생시킨 컴포넌트이다.

3. MyActionListener를 작성하면 다음과 같다.

```
class MyActionListener implements ActionListener {
   public void actionPerformed(ActionEvent e) {
      System.exit(0);
   }
}
```

5. 익명 클래스를 이용하여 다시 작성하면 다음과 같다.

```
JButton btn = new JButton("Hello");
btn.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
         System.out.println("Click");
      }
   }
};
```

7. 틀린 부분을 수정하면 다음과 같다.

(1) ActionAdapter는 존재하지 않으며 틀린 부분을 implements ActionListener로 수정하여야 한다.

```
class MyActionListener ~~extends ActionAdapter~~ { // implements ActionListener로 수정
   public void actionPerformed(ActionEvent e) {
      System.out.println("Click");
   }
```

(2) MouseListener를 implements하면 mouseReleased(), mouseClicked(), mouseEntered(), mouseExited()를 모두 구현하여야 한다. 그러므로 틀린 부분을

extends MouseAdapter로 수정하여야 한다.

```
class MyMouseListener implements MouseListener { // extends MouseAdapter로 수정
   public void mousePressed(MouseEvent e) {
      System.out.println("Mouse Pressed");
   }
}
```

(3) Key 이벤트가 발생하면 KeyEvent 객체가 생성되고 메소드에 인자로 넘어온다. 그러므로 ActionEvent가 아니라 KeyEvent로 수정하여야 한다.

```
class MyKeyListener extends KeyAdapter {
   public void keyTyped(ActionEvent e) { // KeyEvent로 수정
      System.out.println("Key Pressed");
   }
}
```

9. ① ItemListener

11. 〈Alt〉, 〈Tab〉, 〈Delete〉, 〈Shift〉, 〈Help〉

13. a 키는 유니코드 키이므로 keyPressed(), keyTyped(), keyReleased() 메소드순으로 호출된다.

15. 눌러진 키가 k 키인 경우 현재 응용프로그램을 종료시키는 keyPressed() 메소드를 작성하면 다음과 같다.

```
public void keyPressed(KeyEvent e) {
   if(e.getKeyChar() == 'k')
      System.exit(0);
```

11장 연습문제

이론문제

1. ② Frame

3. "java.jpg"를 가진 JLabel 컴포넌트는 다음과 같이 생성한다.

```
ImageIcon image = new ImageIcon("java.jpg");
JLabel label = new JLabel(image);
```

5. 3개의 이미지 아이콘은 다음과 같다.
   - normalIcon
     버튼이 보통 상태에 있을 때 출력되는 이미지
   - rolloverIcon
     버튼 위에 마우스가 올라가면 출력되는 이미지
   - pressedIcon
     마우스 버튼이 눌러져 있는 동안 출력되는 이미지

7. JCheckBox와 JRadioButton의 가장 큰 차이점은 JCheckBox 컴포넌트는 그룹을 형성하지 않고 개별적으로 체크 가능하지만, JRadioButton 컴포넌트들은 여러 개가 모여 하나의 그룹을 형성하고 이들 중 오직 하나만 선택 가능하다.

9. ② 선택 상태 체크박스를 클릭한 경우
   선택 상태 체크박스를 클릭하여 다시 선택하면 선택 상태의 변화가 발생하지 않았기 때문에 Item 이벤트가 발생하지 않는다.

12장 연습문제

이론문제

1. 자바에서 지원하는 컴포넌트를 사용하여 구성할 수 없는 UI가 필요한 경우로서, 대표적으로 게임 등 매우 자유로운 모양의 GUI를 사용하여야 하는 경우이다.

3. JPanel

5. AWT 컴포넌트와 스윙 컴포넌트가 화면에 그려지는 과정은 매우 다르다. AWT 컴포넌트의 그리기는 Component 클래스의 paint(), update() 등의 메소드에 의해 지배받지만, 스윙 컴포넌트의 경우 JComponent에 구현된 paint(), paintComponent(), paintChildren() 등의 메소드에 의해 지배받는다. 그러므로 한 컨테이너에 AWT 컴포넌트와 스윙 컴포넌트가 존재하게 되면 AWT 컴포넌트가 화면에 그려지지 않거나 이상하게 그려질 가능성이 있다.

7.

(1) 이미지를 원본 크기로 (10, 20) 위치에 그리는 코드는 비교적 간단하다.

```
g.drawImage(img, 10, 20, this);
```

(2) 패널에서 상, 하, 좌, 우 10픽셀씩 간격을 두고 그 안에 이미지가 모두 보이도록 그리기 위해서는 시작 좌표는 (10, 10)이며, 이미지의 크기는 폭이 10*2만큼 작게, 높이도 10*2만큼 작게 그려야 한다. 그러므로 다음과 같다.

```
g.drawImage(img, 10, 20, this.getWidth()-10*2, this.getHeight()-10*2, this);
```

13장 연습문제

이론문제

1. 영화 보면서 팝콘 먹기, 전화하면서 문서 작성하기

3. `public void run()`

5. 우선순위 값이 높을수록 스레드는 높은 스케쥴링 우선순위를 가진다. 그러므로 스레드 A가 먼저 실행되며 일정 시간이 지난 후 JVM이 다시 스케쥴링을 시행한다. 그러나 여전히 A가 우선순위가 높기 때문에 다시 실행된다. 이런 식으로 A가 종료할 때까지 계속된다. 그러고 나면 B, C의 우선순위는 동일하므로 어떤 것이 먼저 선택될지는 모른다. 만일 B가 선택된다면 B가 실행되며 일정 시간 후 B의 실행이 중단되고 JVM이 다시 스케쥴링을 실시한다. 여전히 B, C의 우선순위가 동일하므로 round-robin 전략에 의해 C가 선택된다. C의 실행 도중 일정 시간이 지나 다시 스케쥴링되면 B가 선택되며, B와 C는 번갈아 실행된다.

7. JVM이 실행되는 동안에 스레드는 두 가지 종류가 존재한다. 데몬 스레드와 일반 스레드이다. 가비지 컬렉터는 데몬 스레드이며, main 스레드는 일반 스레드이다. 아무 조건 없이 응용프로그램에서 만들어진 스레드는 일반 스레드이다. JVM은 일반 스레드가 모두 종료하고 데몬 스레드만 남은 경우 스스로 종료한다.

9. 스레드는 입출력을 실행하는 순간 BLOCK 상태가 되며 다른 스레드로 스케쥴링된다.

14장 연습문제

이론문제

1. ④ Separator

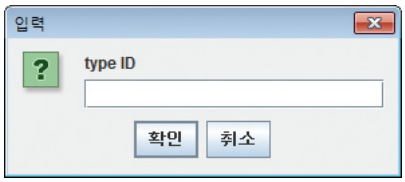
3. ⑤ 툴바의 핸들을 마우스 드래깅할 수 없게 만드는 메소드는 JToolBar.setEnabled(false)이다.

5.

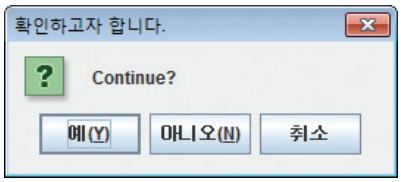
```
JButton b = new JButton("Hello");
b.setToolTipText("안녕하세요");
```

7. 모달 다이얼로그란 다이얼로그에 입력을 마치지 않고 다른 작업을 할 수 없도록 다이얼로그가 키 입력을 독점하는 것을 말한다. 일반적으로 파일을 읽고 읽은 파일을 처리하는 응용프로그램은 파일 열기 다이얼로그를 이용하여 사용자로부터 읽고자 하는 파일을 선택한다. 그러고 나서 응용프로그램은 사용자가 선택한 파일을 읽고 읽은 데이터를 가지고 계속 작업을 실행한다. 그러므로 파일 다이얼로그가 화면에 출력된 상태에서 사용자가 파일을 선택하지 않고 다른 작업을 시도하면 파일을 읽지 않았기 때문에 다른 작업을 하는 것이 무의미하다. 혹은 사용자가 파일 다이얼로그가 파일을 선택하는 과정을 생략한 채 다른 작업을 진행하도록 어설프게 응용프로그램을 작성하여서는 안 된다. 그러므로 파일 다이얼로그는 모달 다이얼로그로 작성되어야 한다.

9. (1) JOptionPane.showInputDialog("type ID");



⑵ JOptionPane.showConfirmDialog(null, "Continue?", "확인하고자 합니다.",
JOptionPane.YES_NO_CANCEL_OPTION);



15장 연습문제

이론문제

1. ③ 애플릿은 자바 가상 기계의 도움 없이도 웹 브라우저만 있으면 실행 가능하다.

3. 애플릿의 생명 주기는 웹 브라우저에 의해 제어된다.

5. Applet을 상속받아 작성되는 애플릿의 경우 애플릿을 그리는 그래픽 코드는 Applet을 상속받는 클래스의 paint() 메소드에 두어야 한다.

7. 서버 컴퓨터의 HTML 파일이 있는 디렉터리 \bin\etc\sample\test.class

9. 애플릿 응용프로그램은 웹 페이지에 내장되어 있기 때문에 웹 페이지를 접근하는 원격 사용자가 이미 인지하지 못하는 경우가 대부분이다. 그러므로 임의의 웹 페이지를 방문하였을 때 인지하지 못한 애플릿 프로그램이 사용자의 컴퓨터에서 실행되는 것은 매우 위험한 상황에 노출되는 것이다. 인터넷의 웹 페이지가 악의적인 목적을 가지고 있을 때, 애플릿이 사용자 몰래 사용자의 컴퓨터에서 실행되어 정보를 빼가거나 정보를 지우는 등 악의적인 경우가 발생할 수 있다. 그러므로 자바에서는 애플릿이 사용자의 컴퓨터에서 활동을 제한하는 보안 정책을 마련하고 있다. 즉, 사용자의 컴퓨터의 파일 시스템에 접근할 수 없고, 설치된 프로그램을 실행할 수 없으며, USB 등 연결된 장치들에 대한 접근을 제한하고, 서버 외 다른 컴퓨터로의 네트워크 접근을 막고 있다.

16장 연습문제

이론문제

1. ④. UDP는 TCP와 마찬가지로 transport 계층의 프로토콜이다.

3. ④. FTP, TELNET, GOPHER 등과 같은 프로토콜도 URL의 프로토콜 식별자로 사용될 수 있다.

5. 빈칸에 들어갈 문장은 aURL.openStream()이다.

7. URLConnection 클래스

9. 서버 소켓은 accept() 메소드를 이용하여 클라이언트로부터의 연결 요청을 기다린다. accept() 메소드는 연결 요청이 오면 새로운 Socket 객체를 반환한다.

17장 연습문제

이론문제

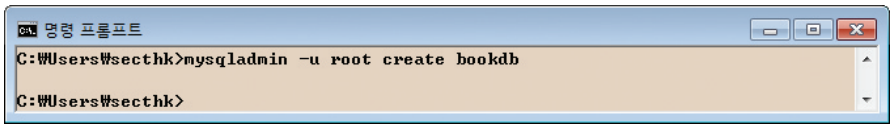
1. ④. JDBC는 관계형 데이터베이스에 대한 API를 제공한다.

3. SQL

5. 열의 이름을 인자로 하거나 또는 열의 인덱스를 인자로 하여 각각 `getInt("id")` 또는 `getInt(1)`를 빈칸에 삽입할 수 있다.

실습문제

1.



3.