연습문제 이론 홀수 정답

1장 연습문제

이론 문제

1. 자바 소스의 확장자는 .java이고 컴파일된 클래스 파일의 확장자는 .class이다.

3. WORA(Write Once Run Anywhere)

5. JRE는 자바 프로그램의 실행 환경으로 자바 가상 기계를 포함하고 있으며 자바 실행 환경만 필요한 경우에 사용되며, JDK는 자바 컴파일러, 도구, 라이브러리 등 자바 프로그램 개발에 필요한 모든 것과 JRE를 포함한다. 따라서 자바 응용프로그램 개발을 위해서는 JDK가 필요하다.

7. ③

9. (1) W.java에 저장되어야 한다.
   (2) 총 4개의 클래스 파일(W.class, W$X.class, Y.class, Z.class)이 생성된다.

2장 연습문제

이론 문제

1. 자바에서 클래스를 선언할 때는 class 키워드를 사용한다.

3. 올바른 변수 선언은 다음과 같다.
   (1) int age;
   (2) float f = 0.25F;
   (3) double d = age + f; 또는 double d = (double)age + (double)f
   (4) char c = 'a';
   (5) String name = "황기태";

5. (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)

7. (1) SampleProgram.java
   (2) SampleProgram 클래스에 main() 메소드가 없기 때문에 오류가 난다. main()을 삽입하여 다음과 같이 작성하면 된다.

```
public class SampleProgram {
   public static void main(String[] args) {
      int i;
      int j;
      i = 20;
      j = 30;
      System.out.println(i+j);
   }
}
```

9. 다음과 같이 한 줄로 작성할 수 있다.

```
i = (j%2 == 0)?10:20;
```

3장 연습문제

이론 문제

1. ⑴ 0에서 10까지(혹은 2에서 10까지) 짝수만 더하는 프로그램이며, 실행 결과는 30
⑵

```
int i=0, sum=0;
while(true) {
   i = i + 2;
   sum += i;
   if(i == 10) break;
}
System.out.println(sum);
```

```
int i=0, sum=0;
do {
   i = i + 1;
   if(i%2 = 1) continue;
   sum += i;
} while(i<10);
System.out.println(sum);
```

3. ⑴ char [] c = new char [10];
⑵ int [] n = {0,1,2,3,4,5};
⑶ char [] day = {'일', '월', '화', '수', '목', '금', '토'};
⑷ double [][] d = new double[5][4];
⑸ int val [][] = {{1,2,3,4}, {5,6,7,8}, {9,10,11,12}};

5. 다음 2차원 배열 선언문에서 문법적으로 잘못된 것은 다음 보기이다.
④ int [3][2] n = { {1,2}, {3,4}, {4,5} }; // int [3][2] n에서 3,2를 사용하면 안 됨

7. ⑴ 빈칸을 채우면 다음과 같다.

```
double [] allocArray() {
   Scanner scanner = new Scanner(System.in);
   double [] n = new double [scanner.nextInt()]; // 입력된 정수 크기의 배열 생성
   return n; // 배열 리턴
}
```

⑵ double d [] = allocArray();

4장 연습문제

이론 문제

1. ④
   클래스의 멤버 변수들은 보호하기 위해 가능하면 private으로 선언하는 것이 바람직하다.

3. ③

5. 두 메소드 f()의 매개 변수 개수를 다르기 때문에, f()의 메소드 오버로딩은 성공한 경우이다. 리턴 타입이 같거나 다르건 오버로딩에 상관없다.

7. 가능하면 멤버 변수(필드)는 private으로 선언하고, 생성자나 메소드를 통해 접근하도록 하는 것이 바람직하다. 생성자를 이용하면 다음과 같이 할 수 있다.

```
class Person {
   private int age;
   public Person(int age) { this.age = age; }
}
public class Example {
   public static void main (String args[]) {
      Person a = new Person(17);
   }
}
```

또는 다음과 같이 메소드를 작성해도 된다.

```
class Person {
   private int age;
   public setAge(int age) { this.age = age; }
}
public class Example {
   public static void main (String args[]) {
      Person a = new Person();
      a.setAge(17);
   }
}
```

9.

```
public class Rectangle {
   int w, h;
   Rectangle(int w, int h) {
      this.w = w; this.h = h;
   }
    Rectangle(int w) {
      this.w = w; this.h = 2;
   }
    Rectangle() {
      this.w = 1; this.h = 2;
   }
}
```

11. ④

getB()는 non-static 메소드이므로, static 메소드인 g()에서 호출할 수 없다.

5장 연습문제

이론 문제

1. (1) objA의 멤버들은 int a, int b, void set()
   (2) objB의 멤버들은 int a, int b, void set(), int c, int d
   (3) objC의 멤버들은 int a, int b, void set(), int c, int d, int e, int f

3. ④
   오버라이딩된 메소드가 호출되면 동적 바인딩이 발생한다.

5.

```
class LCD {
   private int size;
   public LCD(int n) { size = n; }
}
class ColorLCD extends LCD {
   int colorSize;
   public B(int colorSize, int size) {
      super(size);
      this.colorSize = colorSize;
   }
}
```

7.

(1) ②
(2) new A()에 의해 생성된 객체는 B를 상속받지 않았기 때문에 (B)new A();와 같이 B 클래스의 객체로 다운 캐스팅될 수 없기 때문이다.

9. 빈칸에 들어가는 코드는 다음과 같다.
   (1) draw();
   (2) super.draw();

11. (1) ②, ④

(2)

```
class Circle extends Shape {
   private int radius;

   public Circle(int radius) { this.radius = radius; }
   double getArea() { return 3.14*radius*radius; }
   public void draw() { System.out.println("반지름="+radius); }
}
```

6장 연습문제

이론 문제

1. java.lang 패키지에 속한 클래스들은 import 문 없이 사용할 수 있다.

3. Circle 클래스를 drawable 패키지에 속하게 하고자 하기 위해, 다음과 같이 package 문을 첫 줄에 삽입한다.

```
package drawable;
public class Circle {
   int radius;
   public Circle(int radius) { this.radius = radius; }
}
```

그리고 Main 클래스를 app 패키지에 저장하고, Circle 클래스를 사용하기 위해 다음 코드와 같이 한다.

```
package app;
import drawable.Circle;
public class Main {
   public static void main(String[] args) {
      Circle c = new Circle(5);
   }
}
```

5. (1) Integer n = Integer.valueOf(20);
   Integer.valueOf(20); 부분은 정수 20을 박싱하는 코드이다.
   (2) double d = 1.2 + Double.valueOf(3.4);
   Double.valueOf(3.4);은 실수 3.4를 박싱하는 코드이며, 1.2와의 덧셈을 위해 Double.valueOf(3.4)의 객체는 자동으로 언박싱되어 3.4가 된다. 그리고 1.2와 3.4가 더해져서 4.6이 된다.
   (3) System.out.print(3 + Integer.valueOf(20));
   Integer.valueOf(20)는 정수 20을 박싱한 코드이며, 3과 더하기를 위해 다시 자동 언박싱되어 20으로 처리된다. 3과 20이 더해져서 23이 되어 System.out.print(23)에 의해 출력된다.
   (4) Boolean b = true;

true가 Boolean.valueOf(true)로 자동 박싱된다.

(5) float f = Float.valueOf("10.1");

Float.valueOf("10.1")으로 실수 10.1이 박싱되고, float 타입의 변수 f에 치환되기 위해 Float.valueOf("10.1")은 10.1로 자동 언박싱된다.

(6) String s = "abc";

String 클래스는 Wrapper 클래스가 아니므로 박싱 혹은 언박싱과 무관하다.

7. (1) a와 == 연산을 수행하였을 때 true가 되는 문자열을 b이다. "Hello"는 리터럴 테이블에 저장되기 때문에 a와 b 레퍼런스의 값은 같다.

(2) f와 equals() 연산을 수행하였을 때 true가 되는 문자열은 c, e이다. equals()는 객체의 내용을 비교하므로 c와 e는 f와 같은 문자열이다.

9. 코드 각 라인이 실행될 때 a, b, c가 변하는 것을 주석에 설명하였다.

```
String a = new String("  hello  "); // a = "  hello  "
String b = a; // a = "  hello  ", b = "  hello  "
String c = a.trim(); // c = "hello". a = "  hello  "
a = a.toUpperCase(); // a = "  HELLO  "
```

최종적으로 a, b, c는 다음과 같다.

```
a = "  HELLO  "
b = "  hello  "
c = "hello"
```

7장 연습문제

이론 문제

1. ④
   컬렉션은 배열과 달리 요소의 개수가 가변적이다.

3. ①
   v.size()는 현재 삽입되어 있는 요소의 개수를 리턴하므로 현재 삽입된 요소가 없으면 0을 리턴한다. new Vector<Integer>(3)에 주어진 3은 초기 벡터의 용량을 지정하는 것으로, 벡터의 현재 용량을 알고자 하면 v.capacity()를 호출하면 된다.

5. 제네릭 컬렉션을 사용할 때 제네릭 타입에 대입할 수 있는 타입으로 int, char 등의 기본 타입은 사용할 수 없다. 그러므로 다음과 같이 수정해야 한다.

```
Vector<Integer> v = new Vector<Integer>(100);
```

7.

```
ArrayList<Double> a = new ArrayList<Double>();
a.add(3.5); // 3.5가 new Double(3.5)로 자동 박싱된다.
double d = a.get(0); // a.get(0)가 리턴하는 Double 객체를 double 타입으로 만들기 위해
                        a.get(0)을 a.get(0).doubleValue()로 자동 언박싱된다.
```

9. 빈칸에 코드를 채우면 다음과 같다.

```
Vector<String> v = new Vector<String>();
v.add("Good"); // v에 "Good" 삽입
v.add("Bad"); // v에 "Bad" 삽입
System.out.println(v.size()); // v에 현재 삽입된 문자열 개수 출력
v.remove(1); // v의 인덱스 1에 있는 "Bad" 문자열 삭제
```

11. 4개의 문항에 대해 답하면 다음과 같다.

```
class MyGeneric<W> {
   private W x;
   public MyGeneric(W x) {
      this.x = x;
   }
   public W take() { return x; } // (2)
   public boolean compare(W x) { // (3)
      if(this.x.equals(x)) return true;
      else return false;
   }
}
```

(1) MyGeneric의 타입 매개 변수는 W이다.
(2) take() 메소드 코드 정답은 앞의 소스에 있다.
(3) compare() 메소드 코드 정답은 앞의 소스에 있다.
(4) String으로 구체화한 MyGeneric 객체를 생성하고 활용 예를 들면 다음과 같다.

```
MyGeneric<String> g = new MyGeneric<String>("Kitae");
System.out.println(g.take()); // "Kitae"를 출력한다.
System.out.println(g.compare("Kito")); // false를 출력한다.
```

8장 연습문제

이론 문제

1. ① Panel

3. 코드의 빈칸을 채우면 다음과 같다.

```
import java.awt.*;
import javax.swing.*;
public class MyFrame extends JFrame {
   public MyFrame() {
      Container c = getContentPane(); // 컨텐트팬에 대한 레퍼런스 얻기
      c.add(new JButton("hello")); // 컨텐트팬에 "hello" 버튼 달기
      setSize(200, 400);  // 프레임을 너비 200, 높이 400픽셀로 설정
      setVisible(true);
   }
   public static void main(String [] args) {
      MyFrame frame = new MyFrame(); // MyFrame 생성
   }
}
```

5. ②
컨테이너는 다른 컨테이너에 삽입될 수 있다.

7. (1)

```
Container c = getContentPane(); // 컨텐트팬 알아내기
c.setLayout(new BorderLayout(10, 20)); // 배치관리자 설정
```

(2)

```
Container c = getContentPane(); // 컨텐트팬 알아내기
c.setLayout(new FlowLayout(FlowLayout.CENTER, 10, 20)); // 배치관리자 설정
```

(3)

```
Container c = getContentPane(); // 컨텐트팬 알아내기
c.setLayout(new GridLayout(2, 5, 10, 20)); // 배치관리자 설정
```

9장 연습문제

이론 문제

1. ④

이벤트 리스너는 익명 클래스나 내부 클래스 혹은 외부 클래스로 작성할 수 있으며, 외부 클래스의 경우 별도의 자바 파일에 작성할 수 있다. 그러나 반드시 별도의 자바 파일에 작성할 필요는 없다.

3. 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");
   }
});
```

5. 틀린 부분을 수정하면 다음과 같다.

```
class MyActionListener implements ActionListener {
   public void actionPerformed(ActionEvent e) {
      System.out.println("Click");
   }
}
```

7. 유니코드 키가 아닌 것은 다음과 같다.

<Alt>, <Tab>, <Delete>, <Shift>, <Help>

9.

```
la.addMouseListener(new MyMouseListener()); // la에 마우스 리스너를 등록한다.
...
class MyMouseListener extends MouseAdapter { // 마우스 리스너를 선언한다.
   public void mouseReleased(MouseEvent e) { // 눌러진 마우스가 놓여지는 순간 처리
      JLabel label = (JLabel)e.getSource(); // 이벤트 소스를 알아낸다.
      label.setText("안녕");          // 문자열을 "안녕"으로 변경한다.
   }
}

* mouseReleased 대신 mouseClicked를 써도 된다.
```

11. ③ c.requestFocus();

10장 연습문제

이론 문제

1. ②

Container 클래스는 AWT 패키지에 속하는 클래스이다.

3. ③

JCheckBox 컴포넌트를 마우스로 선택하면 Item 이벤트가 발생한다.

5. 메뉴를 만들어 프레임에 붙이는 코드의 빈칸을 채우면 다음과 같다.

```
JMenuBar mb = new JMenuBar();
JMenu fileMenu = new JMenu("File"); // "File" 메뉴를 생성한다.
mb.add(fileMenu); // 메뉴바에 파일 메뉴를 붙인다.
fileMenu.add(new JMenu("New")); // "New" 메뉴아이템을 생성하여 붙인다.
fileMenu.addSeparator(); // 분리선을 삽입한다.
frame().setJMenuBar(mb); // 프레임에 메뉴바를 붙인다.
```

7. "sunny.jpg"를 가진 이미지 레이블 sunnyLabel을 만드는 코드는 다음과 같다.

```
ImageIcon icon = new ImageIcon("sunny.jpg"); // 이미지 파일 로딩, 이미지 객체 생성
JLabel sunnyLabel = new JLabel(icon); // 레이블 컴포넌트 sunnyLabel 생성
```

9. (1) JRadioButton, 3개

'아침', '점심', '저녁'을 나타내는 3개의 라디오 버튼을 사용하여 하나를 선택하도록 한다.

(2) JCheckBox, 3개

방법 1) '남/여'를 위해 1개, '내국인/외국인' 선택을 위해 1개, '성년/미성년' 선택을 위해 1개 총 3개의 체크박스를 둔다.

방법 2) 물론 JRadioButton을 6개 사용하여, 두 개는 '남/여' 선택, 두 개는 '내국인/외국인' 선택, 두 개는 '성년/미성년' 선택에 사용하게 할 수도 있지만, 총 6개의 라디오 버튼을 사용하고 각 두 개씩 버튼 그룹으로 묶어야 하므로 방법 1보다는 비효율적이다.

결론적으로 방법 1의 JCheckBox, 3개가 정답이다.

(3) JLabel, 4개
이미지는 출력하기에 적절한 컴포넌트는 JLabel이고 이미지가 총 4개이기 때문이다.
(4) JButton, 1개
'다음'으로 계속 진행을 지시하기 위해서는 JButton 컴포넌트 1개가 필요하며, Action 이벤트 리스너를 달면 된다.

11장 연습문제

이론 문제

1. ③ paintComponent(Graphics g)
   public void paintComponent(Graphics g)는 JComponent의 메소드이다.

3. ④ 애니메이션

5. ⑴ g.drawImage(img, 10, 20, null);
   ⑵ g.drawImage(img, 10, 10, getWidth()-20, getHeight()-20, null);

7. 컴포넌트를 이용하여 GUI를 구성하면 컴포넌트 모양의 한계를 벗어날 수 없어 정형화된 모양의 GUI 밖에 구성할 수 없다. 하지만 그래픽을 이용하면 컴포넌트로 표현할 수 없는 모양과 방식으로 GUI를 구성할 수 있다. 다만 개발자가 일일이 그래픽으로 화면을 구성해야 하는 어려움이 있다.

9. super.paintComponent()는 Jpanel의 paintComponent()를 호출하는 코드이다. Jpanel의 paintComponent()는 라인 7에서 배경색으로 지정된 노란색으로 패널의 배경을 칠해 이전에 그려진 내용을 모두 지운다. 만일 super.paintComponent()를 호출하지 않는다면, 이전에 그려진 잔상이 지워지지 않고 배경색도 칠해지지 않게 된다.

12장 연습문제

이론 문제

1. 영화 보면서 팝콘 먹기, 전화하면서 문서 작성하기

3. ⑴ `public void run();`
   ⑵ `public void start();`
   ⑶ `public void interrupt();`

5. JVM이 자바 응용프로그램을 실행하기 위해 생성하는 스레드는 main 스레드(메인 스레드)이다. 메인 스레드는 자바 응용프로그램의 main() 메소드에서부터 실행을 시작한다. main() 메소드의 실행을 끝내면 메인 스레드는 스스로 종료한다.

7. ②
   스마트폰에서 문자를 전송하는 스레드와 음악을 연주하는 스레드는 서로 공유하는 데이터가 없기 때문에 동기화 될 필요가 없다.

9. ② `synchronized`

13장 연습문제

이론 문제

1. ②
스트림은 다른 스트림과 연결될 수 있다.

3. ② FileReader
FileReader는 문자 스트림 클래스이다.

5. 총 5개의 문자를 읽어 출력하므로 다음과 같이 출력된다.

```
12345
```

7.

```
FileOutputStream fout;
FileInputStream fin;
try {
   fout = new FileOutputStream("c:\\Temp\\test2.txt");
   fin = new FileInputStream("c:\\Temp\\test.txt");
   byte [] buf = new byte [128]; // 버퍼 할당
   while(true) {
      int n = fin.read(buf); // 버퍼 크기만큼 읽는다.
      fout.write(buf, 0, n); // 읽은 바이트만큼 쓴다.
      if(n < buf.length) // 버퍼 크기보다 적게 읽었다면
         break; // 파일 끝에 도달했으므로 복사 완료
   }
   fin.close();
   fout.close();
} catch (IOException e) { }
```

9. (1) true
(2) "c:\Program Files\java\jre-10" 문자열
(3) "c:\Program Files\java\jre-10\HTML.html" 문자열
(4) "HTML.html" 문자열
(5)

```
File f = new File("c:\\Program Files\\java\\jre-10", "HTML.html");
```

14장 연습문제

이론 문제

1. ipconfig 명령을 입력하면 현재 PC의 네트워크 설정 상황을 보여준다. ipconfig 명령을 입력하고 출력된 정보 중에서 "IPv4 주소" 줄에서 IP 주소를 확인할 수 있다.

3. ③
   ServerSocket 클래스는 서버 쪽에서 클라이언트의 접속을 기다라는 목적으로만 사용한다.

5. ④
   소켓 객체 ss는 클라이언트로부터의 접속을 받는 목적으로만 사용된다.

7. ①