연습문제 이론 홀수 정답

1장 연습문제

이론문제

1. 자바 소스 파일의 확장자는 .java이고 컴파일된 파일의 확장자는 .class이다.

3. C → C++ → Java → C#

5. ① 자바 가상 기계

7. ④. 자바에서는 하나의 클래스 파일(.class)에는 반드시 하나의 컴파일된 클래스만이 저장되어, 클래스가 들어 있는 클래스 파일을 찾기 쉽게 하였다.

9. ③. 자바에서는 변수나 함수는 반드시 클래스 내에 작성하여야 한다.

11. (1) A.java이다. 자바 소스 파일에 여러 개의 클래스가 작성되어 있을 때, public 속성을 가진 클래스의 이름으로 저장된다.
    (2) A.class, A$B.class, C.class, C$D.class의 4개의 클래스 파일이 생성되고, 한 클래스 파일에는 반드시 하나의 클래스만 컴파일되어 저장된다.

2장 연습문제

이론문제

1. 자바에서는 클래스를 선언할 때 class 키워드를 사용한다.

3. 잘못된 식별자와 이유는 다음과 같다.

```
int %j; // %는 특수문자로 사용할 수 없다.
double 1var; // 첫 번째 문자로 숫자를 사용할 수 없다.
```

5. (1) 67 + 12.8의 결과 값과 타입은 double 타입의 79.8
   (2) 10/3의 결과 값과 타입은 int 타입의 3
   (3) 10.0/3의 결과 값과 타입은 double 타입의 3.3333333333333333
   (4) 10==9의 결과 값과 타입은 boolean 타입의 false

7. 다음 각 항목의 잘못된 부분을 수정하면 다음과 같다.
   (1) while(1) { } → while(true) { }
   (2) int n = 3.5; → int n = (int)3.5; 혹은 double n = 3.5;
   (3) int b = (3<5)?true:false; → boolean b = (3<5)?true:false;
   (4) int score = 85;
   if(80 < score < 90) System.out.print(score);
   → 아래와 같이 수정
   int score = 85;
   if(80 < score && score < 90) System.out.print(score);

9. sum = (sum>100)?100:0;

11. text는 다음과 같고,

```
"\"를 출력하려면 \\ 다음에 \"를 붙여 \\\"과 같이 하면 됩니다."
```

화면에는 다음과 같이 출력한다.

```
System.out.println(text);
```

13. (1) grade가 'A'일 때, 100, 50, 30, 10이 모두 더해져서 190이 출력된다.
(2) grade가 'B'일 때, 50, 30, 10이 모두 더해져서 90이 출력된다.
(3) grade가 'C'일 때, 30, 10이 모두 더해져서 40이 출력된다.
(4) grade가 'F'일 때, 어떤 case 문으로도 분기되지 않아 0이 출력된다.

3장 연습문제

이론문제

1. 1에서 10 사이의 홀수를 출력하는 코드로 다음과 같이 출력된다.

```
1 3 5 7 9
```

3. continue

5. ④ int n[3] = new int [3];. int n[] = new int [3]으로 고쳐야 한다.

7. (1) char c [] = new char [10];
   (2) int [] n = {0,1,2,3,4,5};
   (3) char [] day = {'일', '월', '화', '수', '목', '금', '토' };

9. (1) 2, 3번 라인에 컴파일 오류가 발생한다. 첫 번째 라인에서 배열에 대한 레퍼런스 myArray만 선언한 상태로, myArray는 초기화되지 않는 상태이다. 그런 상태에서 myArrayp[0], myArrayp[1]을 사용하면 없는 배열을 참조하게 되므로 컴파일 오류가 발생한다.
   (2) 첫 번째 라인의 코드를 int myArray[] = new int [5]; 등과 같이 배열을 할당하면 컴파일 오류가 없어진다.

11. 배열 n은 5행으로 이루어진 2차원 배열이며, 각 행에는 순서대로 1개, 3개, 1개, 4개, 2개의 원소들이 들어 있다. 이 프로그램은 각 행에 있는 원소의 개수를 다음과 같이 출력한다.

```
1 3 1 4 2
```

13. ①. main()의 원형에서 abstract가 아니라 static이다.

4장 연습문제

이론문제

1. ④. 필드는 클래스의 중요한 속성을 나타내는 것으로 보호하기 위해 private으로 선언하는 것이 바람직하다.

3. ③

5. ①. void f(int a) { x = a; }와 int f(int b) { return x+b; }는 메소드 이름과 매개변수 개수 및 타입이 모두 같으므로 메소드 오버로딩이 실패한 사례이다. 리턴 타입이 다른 것은 오버로딩과 관계없다.

7. 컴파일 오류가 발생하는 부분은 다음 코드이며,

```
aPerson.age = 17;
```

오류 메시지는 다음과 같다.

```
The field Person.age is not visible
```

오류가 발생하는 이유는 age가 private으로 선언되어 있기 때문에, Person 클래스 바깥의 main() 메소드에서는 접근할 수 없기 때문이다.
오류를 수정하기 위해 age를 public으로 선언해서는 안 되며, Person 클래스에 이 필드를 접근할 수 있는 public 메소드를 만들어 둔다. 그러면 main()에서 이 메소드를 이용하여 age에 접근할 수 있다. 코드는 다음과 같다.

```
class Person {
    private int age;
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
}
public class Example {
```

```
    public static void main (String args[]) {
        Person aPerson = new Person();
        aPerson.setAge(17);
    }
}
```

이 글을 열심히 읽는 독자를 위해서 한 가지 덤으로 중요한 설명을 해보자. 앞의 정답처럼 Person 클래스를 만들어도, main()에서 setAge()를 통해서 age를 마음대로 바꿀 수 있는데 뭐 하러 age를 private으로 두고 굳이 복잡하게 setAge(), getAge()를 만드는가? 하는 의문을 가질 수 있다. 하지만, setAge()를 다음과 같이 만들어 보자.

```
public void setAge(int age) {
    if(age < 0)
        return;
    this.age = age;
}
```

setAge()를 이렇게 만들면 Person 클래스의 age 필드를 음수가 되지 않도록 유지할 수 있지만, setAge() 없이 age를 public으로 공개해버리면, main() 메소드에서 다음과 같이 age를 음수로 설정하는 잘못을 저질러도 막을 수 없게 된다.

```
public static void main (String args[]) {
    Person aPerson = new Person();
    aPerson.age = -20;
}
```

정리하면, 클래스의 주요 필드는 private으로 해두고, public 속성의 set/get 메소드를 별도로 만들어 이 메소드를 통해서만 필드를 접근하게 하여 필드의 무결성을 유지하는 것이 좋은 객체지향프로그래밍이다.

9.

자바에서는 객체를 임의로 소멸시킬 수 없으며, 이것은 개발자에게 매우 다행한 일이다. 참조하는 레퍼런스가 하나도 없는 객체를 가비지라고 판단하고, 이를 가용 메모리로 자동 수집하는 가비지 컬렉션을 진행시킨다. 응용프로그램에서 자바 플랫폼에 이 과정을 지시하고자 하면 System.gc() 코드를 호출하면 된다.

11. ④ `static int g() { return getB(); }`
`static` 메소드에서 `non-static` 멤버를 접근할 수 없다.

13. 코드에는 틀린 부분이 있다. `main()` 메소드 내에서 `int sum = f(2, 4);`의 호출은 잘못되었다. `main()`은 `static` 타입이므로 `f()`를 호출하려면 `f()`도 `static` 타입이어야 한다. `f()` 메소드를 다음과 같이 고쳐야 한다.

```
static public int f(int a, int b)  { return a + b; }
```

15. `new` 연산자를 이용하여 시스템으로부터 할당받아 사용하다 더 이상 사용하지 않는 객체나 배열 메모리를 가비지라 한다. 가비지가 많아지면 상대적으로 자바 가상 기계에서 응용프로그램에게 할당해줄 수 있는 가용 메모리의 양이 줄어들어 자바 응용프로그램의 실행에 영향을 줄 수 있으므로 자바 가상 기계는 가용 공간이 일정 크기 이하로 줄어들게 되면 자동으로 가비지를 회수하여 가용 메모리 공간을 늘린다. 이러한 가비지 컬렉션 때문에 개발자는 할당받은 메모리를 반환하는 코딩 부담을 덜게 된다.

5장 연습문제

이론문제

1. (1) 객체 objA의 멤버들은 총 2개로서 다음과 같다.

```
private int a;
public void set(int a) { this.a = a; }
```

(2) 객체 objB의 멤버들을 총 4개로서 다음과 같다.

```
private int a;
public void set(int a) { this.a = a; }
protected int b, c;
```

(3) 객체 objC의 멤버들은 총 6개로서 다음과 같다.

```
private int a;
public void set(int a) { this.a = a; }
protected int b, c;
public int d, e;
```

(4) a = 1; // ① 라인에서 오류가 발생한다. a는 클래스 A의 private 멤버이므로 상속받은 클래스 D에서 접근할 수 없기 때문이다. 하지만, protected 멤버는 마음대로 접근할 수 있다.

3. 독자마다 조금씩 다르게 할 수도 있겠지만, 현재와 미래에 확장성으로 고려하여 다음과 같이 작성하였다.

```
class Pen { // 모든 펜의 공통 속성
    private int amount; // 남은 량
    public int getAmount() { return amount; }
    public void setAmount(int amount) { this.amount = amount; }
}
class SharpPencil extends Pen {
    private int width; // 펜의 굵기
}
```

```
class BallPen extends Pen {
    private String color;
    public String getColor() { return color; }
    public void setColor(String color ) { this.color = color; }
}
class FountainPen extends BallPen {
    public void refill(int n) { setAmount(n); }
}
```

다음과 같이 작성할 수도 있다.

```
class Pen {
    private int amount;
    public int getAmount() { return amount; }
    public void setAmount(int amount) { this.amount = amount; }
}

class SharpPencil extends Pen {
    private int width; // 펜의 굵기
}

class ColorPen extends Pen {
    private String color;
    public String getColor() { return color; }
    public void setColor(String color ) { this.color = color; }
}

class BallPen extends ColorPen {

}

class FountainPen extends ColorPen {
    public void refill(int n) { setAmount(n); }
}
```

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

7. 다음과 같이 출력된다.

```
A
B:11
```

9. ①. 잘못됨. 추상 메소드 void f()를 abstract void f();로 수정하여야 한다.
②. 문제 없음. 추상 메소드가 없는 클래스로 추상 클래스로 선언할 수 있다.
③. 잘못됨. 추상 클래스 B를 상속받고 추상 메소드를 오버라이딩하지 않으면 클래스 C도 추상 클래스가 됨. 그러므로 클래스 C를 다음과 같이 수정하여야 한다.

```
abstract class C extends B {
}
```

④. 잘못됨. 추상 클래스 B의 추상 메소드는 리턴 타입이 int이지만, 상속받은 클래스 C에서는 void 타입의 메소드를 f()를 구현하였기 때문에 오버라이딩이 실패하였음. 다음과 같이 수정하여야 한다.

```
class C extends B {
    int f() { System.out.println("~"); return 0;}
}
```

11. (1) ② B b = new C(); ③ A a = new D();
(2)

```
true
false
```

(3)

```
true
true
```

13. (1) 추상 클래스의 객체는 생성할 수 없다. 그러므로 다음 두 경우가 오류이다.
② Shape s = new Shape(); ④ Circle c = new Circle(10);

⑵ Circle 클래스에 2군데를 수정해야 한다.

```
class Circle extends Shape { // abstract 삭제
    private int radius;
    public Circle(int radius) { this.radius = radius; }
    double getArea() { return 3.14*radius*radius; }

    // draw() 오버라이딩
    public void draw() { System.out.println("반지름=" +  radius); }
}
```

15. ②. 인터페이스는 클래스와 같이 멤버 변수(필드)의 선언이 가능하다.

6장 연습문제

이론문제

1. (1) import 문은 다른 패키지에 있는 클래스를 사용할 때 코드의 서두에 선언하는 것으로, 컴파일러에게 그 클래스의 경로명을 알려주는 문이다.

   (2) import java.util.Random;은 Random 클래스가 java.util 패키지에 있음을 컴파일러에게 알려주는 문이다. 자바 소스 프로그램에서 Random의 이름을 사용하면, 컴파일러가 Random 클래스의 경로명을 찾을 때, import 문을 참조하여 찾게 한다. import java.util.*;는 자바 프로그램 내에서 사용하는 클래스들의 경로명을 찾을 때, java.util 패키지에서도 확인할 것을 컴파일러에게 지시하는 문이다. 만일 자바 소스 프로그램에서 Random 클래스를 사용하면, 컴파일러는 Random 클래스가 어느 패키지에 있는지 찾기 위해 java.util 패키지에서 확인해본다. import java.util.*;는 import java.util.Random; import java.util.Vector; 등과 같이 여러 import 문을 줄여 사용할 때 유용하다.

   (3) import 문을 사용하지 않고도 자바 프로그램을 작성할 수 있다. 자바 프로그램을 작성할 때, 모든 클래스의 이름에 완전 경로명을 사용하면 된다. 예를 들면 java.util.Random r = new java.util.Random(); 과 같이 완전 경로명으로 작성하는 것이다. 하지만, 프로그램의 코드가 길어지고 복잡해져 가독성이 떨어지는 단점이 있어 import 문을 사용하는 것이 효과적이다.

   (4) java.lang 패키지에 속한 클래스들은 import 없이도 사용할 수 있다. java.lang 패키지에 속한 클래스들로는 Object, String, Math, System 등이 있다.

3. import 문을 사용하지 않도록 Example 클래스를 다시 작성하면 다음과 같다.

```
public class Example {
    public static void main(String[] args) {
        java.util.StringTokenizer st = new java.util.StringTokenizer("a=3,b=5,c=6", ",");
        while (st.hasMoreTokens())
            System.out.println(st.nextToken());
    }
}
```

5. (1)

```
package device;
public class TV {
    private int size;
    public TV(int size) { this.size = size; }
}
```

(2)

```
package app;
import device.TV;
public class Home {
    public Home() { TV myTv = new TV(65); }
}
```

package app;와 import device.TV;의 순서가 다르면 틀린 답이다. 반드시 package 선언이 먼저 와야 한다.

(3) TV 클래스를 컴파일한 TV.class 파일의 경로명은 device.TV.class이고, Home 클래스를 컴파일한 Home.class 파일의 경로명은 app.Home.class이다.

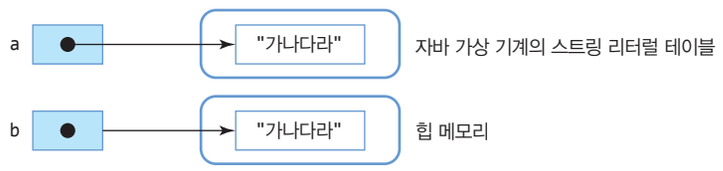
7. 다음 코드를 실행하면,

```
String a = "가나다라";
System.out.println(a == "가나다라");
String b = new String(a);
System.out.println(a == b);
```

다음과 같이 출력된다.

```
true
false
```

설명을 위해 다음 그림을 참고하라.



"가나다라" 문자열은 자바 가상 기계의 스트링 리터럴 테이블에 저장되고, String a의 레퍼런스 a는 리터럴 테이블을 가리키므로, "가나다라" 문자열의 레퍼런스와 레퍼런스 a의 값은 일치한다. 그러므로 a == "가나다라"의 비교 연산은 true이다. 하지만, new String(a)에 의해 생성되는 객체는 힙에 생성되므로 String b는 힙 영역을 가리킨다. 그러므로 a == b의 연산에서 a와 b의 레퍼런스 값은 서로 달라 연산 결과는 false가 된다.

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

11. (1) Math 클래스를 활용하여 100~255까지의 랜덤 정수 발생

```
for(int i=0; i<10; i++)
    System.out.println((int)(Math.random()*156 + 100));
```

(2) Random 클래스를 난수를 발생시키도록 코드 전체를 재작성하면 다음과 같다.

```
import java.util.Random;
public class Example {
    public static void main(String[] args) {
        Random r = new Random();
        for(int i=0; i<10; i++) {
            System.out.print(r.nextInt(156) + 100);
            System.out.print(" ");
        }
    }
}
```

7장 연습문제

이론문제

1. ③. int와 같은 기본 타입의 값은 Wrapper 클래스를 이용하여 객체로 만들어 저장하면 된다.

3. ②. 벡터 v의 초기 크기가 30이다. 30개 이상 저장할 수 있고 벡터는 스스로 저장 공간을 늘린다.

5. ① Stack<String> ss;
   다른 문항이 틀린 이유는 다음과 같다.
   ②에서는 E에 구체적인 타입을 지정하여야 사용할 수 있다.
   ③에서는 HashMap에서는 두 개의 타입 매개변수가 필요한데 String 타입 하나만 사용하였기 때문이다.
   ④에서 Set은 인터페이스이므로 new Set<Integer>()와 같이 객체를 생성할 수 없다.

7. 최종적으로 출력되는 벡터의 용량은 12이다. 벡터의 초기 용량은 3이며 루프가 4번째 돌 때 벡터의 공간이 부족하다. 이때 벡터는 용량을 2배 증가시켜 용량이 6이 된다. 다시 7번째 루프를 돌 때 용량이 12가 되며, 10번 루프를 돌았을 때의 용량은 여전이 12이다. 코드를 아래와 같이 고쳐서 실행해보면 더욱 분명해진다.

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

9.

```
Vector<Integer> v = new Vector<Integer>();
for(int i=0; i<10; i++) v.add(i); // 10개의 정수 저장

Iterator<Integer> it = v.iterator(); // Iterator 레퍼런스 얻기
while(it.hasNext()) { // it가 가리키는 객체가 있는 동안 루프
    int n = it.next(); // 하나씩 알아내기
    System.out.print(n + " "); // 출력
}
```

11. (1) JGeneric의 타입 매개변수는 1개이며 W이다.
(2), (3), (4), (5), (6)은 아래 코드와 같다.

```
class JGeneric<W> {
    private W x, y;
    public JGeneric(W x, W y) {
        this.x = x; this.y = y;
    }
    public W first() { return x; }  // (3) 여러 줄로 작성 가능
    public W second() { return y; }  // (4) 여러 줄로 작성 가능
    public boolean equal() { return x.equals(y); }  // (5) 여러 줄로 작성 가능
}

public class Example {
    public static void main(String[] args) {
        JGeneric<String> js = new JGeneric<String>("hello", "hellu"); // (2)
        // (6) 활용 예
        System.out.println(js.first());
        System.out.println(js.second());
        System.out.println(js.equal());

    }
}
```

```
hello
hellu
false
```

8장 연습문제

이론문제

1. ②. 자바에서 스트림은 다른 스트림과 연결하여 사용함으로써 다양한 데이터의 입출력을 가능하게 한다.

3. ④ FileInputStream

5. ① File

7. (1) true
   (2) "c:\\windows"
   (3) "c:\\windows\\system.ini"
   (4) "system.ini"
   (5) File file = new File("c:\\windows\\", "system.ini"); 또는
   File file = new File("c:\\windows", "system.ini");

9.

```
FileInputStream fin = null;
try {
    fin = new FileInputStream("c:\\temp\\test.txt");
    int c;
    while(true) {
        c = fin.read(); // 파일에서 한 바이트 읽기
        if (c == -1)  break; // 파일 끝까지 읽었음
        System.out.print((char)c);
    }
    fin.close(); // 파일 입력 스트림 닫기
} catch (FileNotFoundException e1) {
    System.out.println("파일을 찾을 수 없습니다.");
} catch (IOException e) {
    System.out.println("입출력 오류가 발생했습니다.");
}
```

9장 연습문제

이론문제

1. AWT 컴포넌트는 중량 컴포넌트이며, Swing 컴포넌트는 경량 컴포넌트이다. 중량 컴포넌트는 운영체제(다른 말로 native)가 가진 GUI 자원을 할당받고 이를 이용하여 화면에 출력되는 컴포넌트로서, 운영체제 자원을 소모하여 운영체제에 부담을 준다. 이에 반해 경량 컴포넌트란 운영체제의 자원을 활용하지 않고 그려지는 GUI 컴포넌트이다. 그러므로 AWT 컴포넌트는 운영체제에 따라 서로 다른 모양을 출력될 가능성이 높지만, 경량 컴포넌트는 운영체제와 관계없이 동일한 모양으로 출력된다. 또한 AWT 컴포넌트는 운영체제가 직접 화면에 그리기 때문에 그려지는 속도가 상대적으로 빠르다.

3. ④ Button

5. ④. 컴포넌트들은 컨테이너 없이도 출력된다.

7.

```
import javax.swing.*;  // 이곳에 필요한 import 문을 삽입하라.
public class MyFrame extends JFrame {
    public MyFrame() {
        setTitle("hello"); // 프레임 타이틀로 "hello" 문자열 출력
        setSize(200,300); // 프레임 크기를 200x300으로 설정
        setVisible(true); // 프레임 화면 출력
    }
    public static void main(String [] args) {
        MyFrame frame = new MyFrame();
    }
}
```

9. (1) c.setLayout(new BorderLayout(3, 4));
   (2) c.setLayout(new FlowLayout(FlowLayout.RIGHT, 5, 6));
   (3) c.setLayout(new GridLayout(5, 2, 7, 8));
   (4) c.setLayout(null);

10장 연습문제

이론문제

1. ③. 키 이벤트를 처리하는 도중 마우스 이벤트가 발생하면, 현재 이벤트를 모두 처리한 뒤 다음 이벤트를 처리한다. 이벤트는 발생 순서대로 처리된다.

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

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

(2) MouseListener를 implements하면 나머지 4개 mouseReleased(), mouseClicked(), mouseEntered(), mouseExited()를 모두 구현하여야 한다. 그러므로 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 메소드로 넘어온다. 그러므로 ActionEvent가 아니라 KeyEvent로 수정하여야 한다.

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

7. ① ItemListener

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

11. ⑴ a 키는 유니코드 키이므로 keyPressed(), keyTyped(), keyReleased() 순으로 호출된다.

    ⑵ <Esc> 키는 유니코드 키가 아니므로 keyPressed(), keyReleased() 순으로 호출되며, keyTyped()는 호출되지 않는다.

13. component.repaint();는 component의 위치나 크기, 색 등에 변화가 생겼으니 다시 그리도록 자바 플랫폼에 요청하는 코드이다. component.revalidate();는 component가 사실상 컨테이너인 경우로, 컨테이너의 자식 컴포넌트들을 다시 배치하도록 지시하는 코드이다. 컨테이너 내부의 자식 컴포넌트가 삭제되거나 새로 추가된 경우 등으로 컨테이너 내부에 변화가 일어난 경우 호출하면 된다.

11장 연습문제

이론문제

1. ③ 컴포넌트가 만들어진 시간

3.

```
ImageIcon icon = new ImageIcon("java.jpg"); // java.jpg 파일을 로딩한다.
JLabel label = new JLabel(); // 빈 JLabel 컴포넌트를 생성한다.
label.setIcon(icon); // 이미지를 레이블에 부착한다.
```

5. ④ 슬라이드바를 클릭한 경우

7.

```
b.setIcon(new ImageIcon("plain.jpg"));
b.setRolloverIcon(new ImageIcon("over.jpg"));
```

9.

```
slider.setValue(slider.getMinimum()+(slider.getMaximum()-slider.getMinimum())/2);
```

12장 연습문제

이론문제

1. ③. 그래픽 기반 프로그래밍과 컴포넌트 기반 프로그래밍은 함께 사용하면 보다 다양한 GUI를 만들 수 있다.

3. ③. Graphics로 선을 그릴 때 선의 두께를 조절할 수 없다.

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.

⑴ 이미지를 원본 크기로 (10, 20) 위치에 그리는 코드는 비교적 간단하다.

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

⑵ 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, getWidth()-10*2, getHeight()-10*2, this);
```

9. ①

13장 연습문제

이론문제

1. 영화 보면서 팝콘 먹기, 전화하면서 문서 작성하기

3. ① `run()`

5. (1) 자바 스레드의 우선순위 값의 범위는 `1~10` 사이
   (2) 우선순위가 높은 스레드를 무조건 먼저 실행시키는 우선순위 기반 정책. 우선순위가 같으면 돌아가면서 실행시키는 라운드로빈 정책
   (3) 스레드 `B`는 스레드 `A`가 종료하거나 스레드 `A`가 `sleep()`을 호출하여 잠을 자거나 스레드 `A`가 입출력을 호출하여 `block` 상태가 되거나 스레드 `A`가 `wait()`을 호출하여 누군가 `notify()`를 불러주기를 무한정 기다릴 때
   (4) 스레드 `A`, `B`, `C`가 실행 준비 상태일 때, 항상 스레드 `A`의 우선순위가 제일 높기 때문에 자바 가상 기계는 무조건 스레드 `A`를 선택하여 실행시킨다.
   (5) 스레드 `A`가 완전히 종료하면, 스레드 `B`와 `C`의 우선순위가 같으므로 짧은 시간 동안 나누어 번갈아 실행된다.
   (6) 스레드 `A`가 실행 도중 네트워크로부터 데이터 도착을 기다리게 되었을 때, 즉 `block` 상태이므로 스레드 `A`는 네트워크로부터 데이터가 도착할 때까지 실행시키지 않는다. 대신 스레드 `B`나 `C` 중 하나를 선택하여 실행 기회를 준다.

7. ② `synchronized`

9. ① 다른 스레드가 객체 `obj`의 `notify()`를 호출할 때

14장 연습문제

이론문제

1. ④ Separator

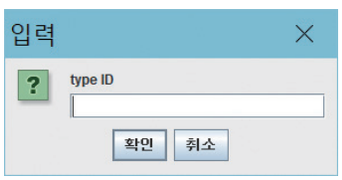
3. ④. 툴바의 핸들을 마우스 드래깅할 수 없게 만드는 메소드는 JToolBar의 setFloatable(false)이다.

5.

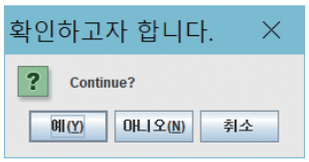
```
JButton b = new JButton("Hello");
b.setToolTipText("안녕하세요");
```

7. 모달 다이얼로그란 다이얼로그에 입력을 마치지 않고서는 다른 작업을 할 수 없도록 다이얼로그가 키 입력을 독점하는 것을 말한다. 일반적으로 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. ipconfig 명령을 입력하면 현재 PC의 네트워크 설정 상황을 보여준다. ipconfig 명령을 입력하고 출력된 정보 중에서 "IPv4 주소" 줄에서 IP 주소를 확인할 수 있다.

3. ④. 서버 소켓은 클라이언트로부터의 접속을 받기 위해 사용되는 소켓이며, 접속이 이루어지면 서버 소켓이 클라이언트와 통신을 할 수 있는 클라이언트 소켓을 생성해준다.

5. ③. 5050은 접속하고자 하는 서버의 포트 번호이다.

7. ③. 생성된 ss 소켓은 클라이언트로부터의 연결을 받기 위해서만 사용된다.

16장 연습문제

이론문제

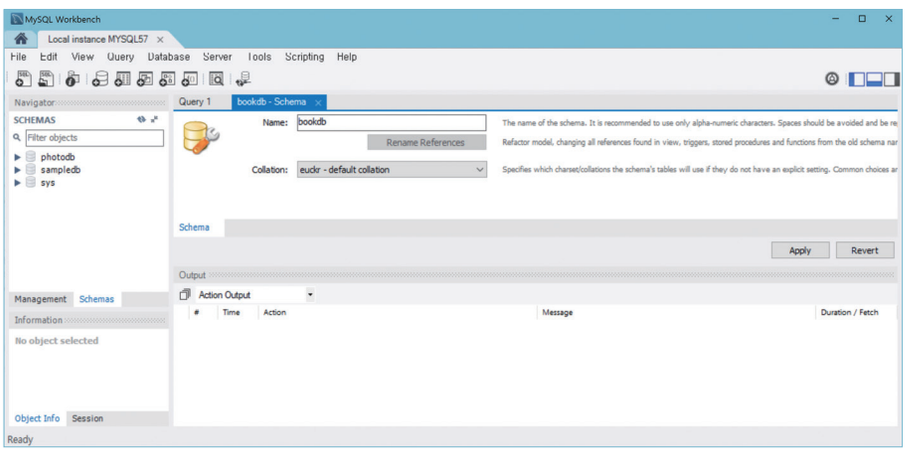
1. ④. JDBC는 관계형 데이터베이스에 대한 API를 제공한다.

3. SQL

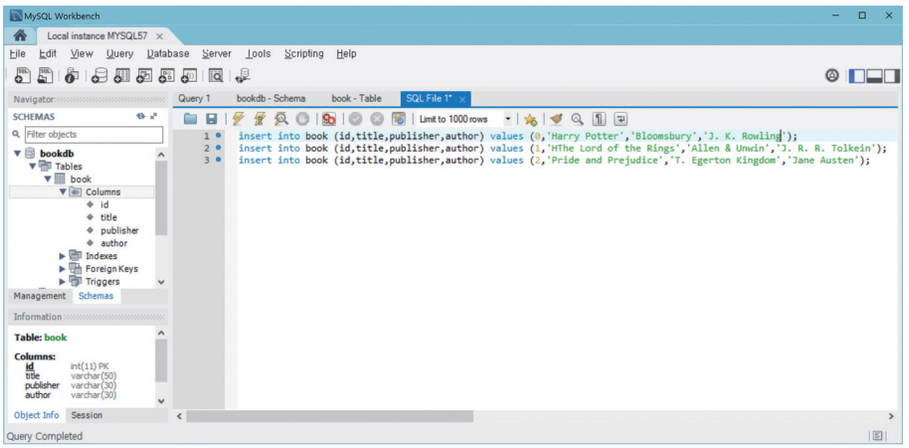
5. 열의 이름을 인자로 하거나 또는 열의 인덱스를 인자로 하여 각각 getInt("id") 또는 getInt(1)를 빈칸에 삽입할 수 있다.

실습문제

1.



3.


MySQL Workbench
Local instance MYSQL57
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
bookdb
Tables
book
Columns
id
title
publisher
author
Indexes
Foreign Keys
Triggers
Management Schemas
Information
Table: book
Columns:
id int(11) PK
title varchar(50)
publisher varchar(30)
author varchar(30)
Object Info Session
Query 1 bookdb - Schema book - Table SQL File 1*
Limit to 1000 rows
1 insert into book (id,title,publisher,author) values (0,'Harry Potter','Bloomsbury','J. K. Rowling');
2 insert into book (id,title,publisher,author) values (1,'HThe Lord of the Rings','Allen & Unwin','J. R. R. Tolkein');
3 insert into book (id,title,publisher,author) values (2,'Pride and Prejudice','T. Egerton Kingdom','Jane Austen');
Query Completed