최신 버전 반영
Java 8~17 대응

이것이 자바다

개정판

교육 현장에서 가장 많이 쓰이는 JAVA 프로그래밍의 기본서

신용권, 임경균 지음

확인문제 + 정답

한빛미디어 Hanbit Media, Inc.

확인문제

Chapter 01

1. 자바 언어의 특징을 잘못 설명한 것은 무엇입니까?

❶ 안드로이드 애플리케이션뿐만 아니라 웹 사이트를 개발할 때 사용하는 언어이다.
❷ 한 번 작성으로 다양한 운영체제에서 실행할 수 있다.
❸ 객체 지향 프로그래밍 언어이다.
❹ 개발자가 코드로 메모리를 관리해야 한다.

2. Open JDK와 Oracle JDK를 잘못 설명한 것은 무엇입니까?

❶ 둘 다 학습용 및 개발용으로는 무료로 사용 가능하다.
❷ Oracle JDK는 개발 소스 공개 의무가 없지만, Open JDK는 있다.
❸ 둘 다 Java SE의 구현체이다.
❹ JDK 11 LTS 버전의 후속 LTS 버전은 JDK 17이다.

3. 환경 변수에 대해 잘못 설명한 것은 무엇입니까?

❶ 프로그램에서 사용할 수 있도록 운영체제가 관리한다.
❷ JAVA_HOME은 JDK가 설치된 디렉토리 경로를 가지고 있다.
❸ PATH는 명령 프롬프트 또는 터미널에서 명령어 파일을 찾을 때 이용된다.
❹ 환경 변수를 수정하면 기존 명령 프롬프트 또는 터미널에서 바로 적용된다.

4. 자바 가상 머신(JVM)에 대해 잘못 설명한 것은 무엇입니까?

❶ JVM은 java.exe 명령어에 의해 구동된다.
❷ JVM은 바이트코드를 기계어로 변환하고 실행시킨다.
❸ JVM은 운영체제에 독립적이다(운영체제별로 동일한 JVM이 사용된다).
❹ 바이트코드는 어떤 JVM에서도 실행 가능한 독립적 코드이다.

5. 자바 프로그램 개발 과정을 순서대로 적어보세요. ( ) ➔ ( ) ➔ ( ) ➔ ( )

❶ javac.exe로 바이트코드 파일(~.class)을 생성한다.

❷ java.exe로 JVM을 구동시킨다.

❸ 자바 소스 파일(~.java)을 작성한다.

❹ JVM은 main() 메소드를 찾아 메소드 블록을 실행시킨다.

6. 자바 소스 파일을 작성할 때 잘못된 것은 무엇입니까?

❶ 자바 소스 파일명과 클래스명은 대소문자가 동일해야 한다.

❷ 클래스 블록과 메소드 블록은 반드시 중괄호 {}로 감싸야 한다.

❸ 실행문 뒤에는 반드시 세미콜론(;)을 붙여야 한다.

❹ 주석은 문자열 안에도 작성할 수 있다.

7. 이클립스의 특징을 올바르게 설명한 것을 모두 선택하세요.

❶ 오픈 소스 통합 개발 환경(IDE)이다.

❷ 소스 파일을 저장하면 자동 컴파일되어 바이트코드 파일이 생성된다.

❸ 워크스페이스(Workspace)는 프로젝트들이 생성되는 기본 디렉토리를 말한다.

❹ Java 17을 지원하는 최소 버전은 Eclipse IDE 2021-12이다.

8. 다음과 같이 출력되도록 Example.java를 패키지 ch01.verify에서 작성해보세요.

```
개발자가 되기 위한 필수 개발 언어 Java
```

확인문제

Chapter 02

1. 변수에 대해 잘못 설명한 것은 무엇입니까?

❶ 변수는 하나의 값만 저장할 수 있다.

❷ 변수는 선언 시에 사용한 타입의 값만 저장할 수 있다.

❸ 변수는 변수가 선언된 중괄호 {} 안에서만 사용 가능하다.

❹ 변수는 초기값이 저장되지 않은 상태에서 읽을 수 있다.

2. 변수 이름으로 사용할 수 없는 것을 모두 선택하세요.

❶ modelName
❷ class
❸ 6hour
❹ $value
❺ _age
❻ #name
❼ int

3. 다음 표의 빈 칸에 자바의 기본 타입 8개를 적어보세요.

타입 \ 크기	1byte	2byte	4byte	8byte
정수타입	( )	( ) ( )	( )	( )
실수타입			( )	( )
논리타입	( )			

4. 다음 코드에서 타입, 변수 이름, 리터럴에 해당하는 것을 적어보세요.

```
int age;
age = 10;
double price = 3.14;
```

타입: ( ), ( )

변수 이름: ( ), ( )

리터럴: ( ), ( )

5. 다음 자동 타입 변환에서 컴파일 에러가 발생하는 것을 선택하세요.

```
byte byteValue = 10;
char charValue = 'A';
```

❶ int intValue = byteValue;

❷ int intValue = charValue;

❸ short shortValue = charValue;

❹ double doubleValue = byteValue;

6. 다음 강제 타입 변환에서 컴파일 에러가 발생하는 것을 선택하세요.

```
int intValue = 10;
char charValue = 'A';
double doubleValue = 5.7;
String strValue = "A";
```

❶ double var = (double) intValue;

❷ byte var = (byte) intValue;

❸ int var = (int) doubleValue;

❹ char var = (char) strValue;

7. 변수를 잘못 초기화한 것은 무엇입니까?

❶ int var1 = 10;

❷ long var2 = 10000000000L;

❸ char var3 = ‘ ’; //작은따옴표 두 개가 붙어 있음

❹ float var4 = 10;

❺ String var5 = “abc₩ndef”;

❻ String var6 = """
abc
def
""";

확인문제

8. 콘솔에 값을 입출력하는 방법에 대해 잘못 설명한 것을 선택하세요.

❶ System.out.print(변수)는 변수값을 출력시키고 행을 바꾸지 않는다.

❷ System.out.println(변수)는 변수값을 출력시키고 행을 바꾼다.

❸ System.out.printf("형식", 변수)는 주어진 형식대로 변수값을 바꾼다.

❹ Scanner의 nextLine() 메소드는 콘솔에 입력된 내용을 문자열로 읽는다.

9. 연산식의 타입 변환 중에서 컴파일 에러가 발생하는 것을 선택하세요.

```
byte byteValue = 10;
float floatValue = 2.5F;
double doubleValue = 2.5;
```

❶ byte result = byteValue + byteValue;

❷ int result = 5 + byteValue;

❸ float result = 5 + floatValue;

❹ double result = 5 + doubleValue;

10. 문자열을 기본 타입으로 변환하는 코드로 틀린 것을 고르세요.

```
String str = "5";
```

❶ byte var1 = Byte.parseByte(str);

❷ int var2 = Int.parseInt(str);

❸ float var3 = Float.parseFloat(str);

❹ double var4 = Double.parseDouble(str);

11. 다음 코드에서 컴파일 에러가 발생하는 라인을 모두 적어보세요.

```
1   int v1 = 1;
2   System.out.println("v1: " + v1);
3   if(true) {
4     int v2 = 2;
5     if(true) {
6       int v3 = 2;
7       System.out.println("v1: " + v1);
8       System.out.println("v2: " + v2);
9       System.out.println("v3: " + v3);
10    }
11    System.out.println("v1: " + v1);
12    System.out.println("v2: " + v2);
13    System.out.println("v3: " + v3);
14  }
15  System.out.println("v1: " + v1);
16  System.out.println("v2: " + v2);
```

확인문제

Chapter 03

1. 다음 코드를 실행했을 때 출력 결과를 작성해보세요.

```
int x = 10;
int y = 20;
int z = (++x) + (y--);
System.out.println(z);
```

2. 다음 코드를 실행했을 때 출력 결과를 작성해보세요.

```
int score = 85;
String result = (!(score>90))? "가":"나";
System.out.println(result);
```

3. 534자루의 연필을 30명의 학생들에게 똑같은 개수로 나누어 줄 때 1인당 몇 개를 가질 수 있고, 마지막에 몇 개가 남는지를 구하는 코드입니다. ( )에 들어갈 알맞은 코드를 차례대로 작성해보세요.

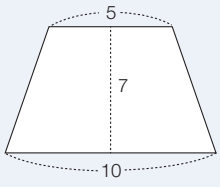
```
int pencils = 534;
int students = 30;

//학생 한 명이 가지는 연필 수
int pencilsPerStudent = (                    );
System.out.println(pencilsPerStudent);

//남은 연필 수
int pencilsLeft = (                    );
System.out.println(pencilsLeft);
```

4. 다음은 십의 자리 이하를 버리는 코드입니다. 변수 value의 값이 356이라면 300이 나올 수 있도록 ( )에 알맞은 코드를 작성하세요(산술 연산자만 사용).

```
int value = 356;
System.out.println(                );
```

5. 다음 코드는 사다리꼴의 넓이를 구하는 코드입니다. 정확히 소수 자릿수가 나올 수 있도록 ( )에 들어갈 수 있는 코드를 모두 선택하세요.

```
int lengthTop = 5;
int lengthBottom = 10;
int height = 7;
double area = (                    );
System.out.println(area);
```



❶ (lengthTop+lengthBottom) * height / 2.0

❷ (lengthTop+lengthBottom) * height * 1.0 / 2

❸ (double) (lengthTop+lengthBottom) * height / 2

❹ (double) ((lengthTop+lengthBottom) * height / 2)

6. 다음 코드는 비교 연산자와 논리 연산자의 복합 연산식입니다. 연산식의 출력 결과를 작성해보세요.

```
int x = 10;
int y = 5;

System.out.println( (x>7) && (y<=5) );
System.out.println( (x%3 == 2) || (y%2 != 1) );
```

7. 다음은 % 연산을 수행한 결과값에 10을 더하는 코드입니다. NaN 값을 검사해서 올바른 결과가 출력될 수 있도록 ( )에 들어갈 코드를 작성해보세요.

```
double x = 5.0;
double y = 0.0;
double z = 5 % y;
if (             ) {
  System.out.println("0.0으로 나눌 수 없습니다.");
} else {
  double result = z + 10;
  System.out.println("결과: " + result);
}
```

확인문제

Chapter 04

1. 조건문과 반복문에 대해 잘못 설명한 것은 무엇입니까?

❶ if 문은 조건식의 결과에 따라 실행 흐름을 달리할 수 있다.

❷ switch 문에서 사용할 수 있는 변수의 타입은 int, double이 될 수 있다.

❸ for 문은 카운터 변수로 지정한 횟수만큼 반복시킬 때 사용할 수 있다.

❹ break 문은 switch 문, for 문, while 문을 종료할 때 사용할 수 있다.

2. 왼쪽 switch 문을 Expression(표현식)으로 변경해서 오른쪽에 작성해보세요.

```
String grade = "B";

int score1 = 0;
switch (grade) {
case "A":
  score1 = 100;
  break;
  case "B":
  int result = 100 - 20;
  score1 = result;
  break;
default:
  score1 = 60;
}
```

3. for 문을 이용해서 1부터 100까지의 정수 중에서 3의 배수의 총합을 출력하는 코드를 작성해보세요.

4. while 문과 Math.random() 메소드를 이용해서 두 개의 주사위를 던졌을 때 나오는 눈을 (눈1, 눈2) 형태로 출력하고, 눈의 합이 5가 아니면 계속 주사위를 던지고, 눈의 합이 5이면 실행을 멈추는 코드를 작성해보세요. 눈의 합이 5가 되는 경우는 (1, 4), (4, 1), (2, 3), (3, 2)입니다.

5. 중첩 for 문을 이용하여 방정식 4x + 5y = 60의 모든 해를 구해서 (x, y) 형태로 출력하는 코드를 작성해보세요. 단, x와 y는 10 이하의 자연수입니다.

6. for 문을 이용해서 다음과 같은 실행 결과가 나오는 코드를 작성해보세요.

```
*
**
***
****
*****
```

7. while 문과 Scanner의 nextLine() 메소드를 이용해서 다음 실행 결과와 같이 키보드로부터 입력된 데이터로 예금, 출금, 조회, 종료 기능을 제공하는 코드를 작성해보세요.

```
----------------------------------
1.예금 | 2.출금 | 3.잔고 | 4.종료
----------------------------------
선택> 1
예금액>10000

----------------------------------
1.예금 | 2.출금 | 3.잔고 | 4.종료
----------------------------------
선택> 2
출금액>2000

----------------------------------
1.예금 | 2.출금 | 3.잔고 | 4.종료
----------------------------------
선택> 3
잔고>8000

----------------------------------
1.예금 | 2.출금 | 3.잔고 | 4.종료
----------------------------------
선택> 4

프로그램 종료
```

확인문제

Chapter 05

1. 참조 타입에 대한 설명으로 틀린 것은 무엇입니까?

❶ 참조 타입에는 배열, 열거, 클래스, 인터페이스가 있다.

❷ 참조 타입 변수의 메모리 생성 위치는 스택이다.

❸ 참조 타입에서 ==, != 연산자는 객체 번지를 비교한다.

❹ 참조 타입은 null 값으로 초기화할 수 없다.

2. 자바에서 메모리 사용에 대한 설명으로 틀린 것은 무엇입니까?

❶ 로컬 변수는 스택 영역에 생성되며 실행 블록이 끝나면 소멸된다.

❷ 메소드 코드나 상수, 열거 상수는 정적(메소드) 영역에 생성된다.

❸ 참조되지 않는 객체는 프로그램에서 직접 소멸 코드를 작성하는 것이 좋다.

❹ 배열 및 객체는 힙 영역에 생성된다.

3. String 타입에 대한 설명으로 틀린 것은 무엇입니까?

❶ String은 클래스이므로 참조 타입이다.

❷ String 타입의 문자열 비교는 ==를 사용해야 한다.

❸ 동일한 문자열 리터럴을 저장하는 변수는 동일한 String 객체를 참조한다.

❹ new String ("문자열")은 문자열이 동일하더라도 다른 String 객체를 생성한다.

4. 배열을 생성하는 방법으로 틀린 것은 무엇입니까?

❶ int[ ] array = { 1, 2, 3 };

❷ int[ ] array; array = { 1, 2, 3 };

❸ int[ ] array = new int[3];

❹ int[ ][ ] array = new int[3][2];

5. 배열의 기본 초기값에 대한 설명으로 틀린 것은 무엇입니까?

❶ 정수 타입 배열 항목의 기본 초기값은 0이다.

❷ 실수 타입 배열 항목의 기본 초기값은 0.0f 또는 0.0이다.

❸ boolean 타입 배열 항목의 기본 초기값은 true이다.

❹ 참조 타입 배열 항목의 기본 초기값은 null이다.

6. 다음은 배열 길이를 출력하는 코드입니다. 실행 결과를 작성해보세요.

```
int[][] array = {
    {95, 86},
    {83, 92, 96},
    {78, 83, 93, 87, 88}
};

System.out.println(array.length);
System.out.println(array[2].length);
```

7. 주어진 배열 항목에서 최대값을 출력하는 코드를 작성해보세요(for 문 이용).

```
int[] array = { 1, 5, 3, 8, 2 };
```

8. 주어진 배열 항목의 전체 합과 평균을 구해 출력하는 코드를 작성해보세요(중첩 for 문 이용).

```
int[][] array = {
    {95, 86},
    {83, 92, 96},
    {78, 83, 93, 87, 88}
};
```

9. 학생들의 점수를 분석하는 프로그램을 만들려고 합니다. 키보드로부터 학생 수와 각 학생들의 점수를 입력받고 while 문과 Scanner의 nextLine() 메소드를 이용해서 최고 점수 및 평균 점수를 출력하는 코드를 작성해보세요.

```
-----------------------------------------------------
1.학생수 | 2.점수입력 | 3.점수리스트 | 4.분석 | 5.종료
-----------------------------------------------------
선택> 1
학생수> 3
```

확인문제

```
-----------------------------------------------------
1.학생수 | 2.점수입력 | 3.점수리스트 | 4.분석 | 5.종료
-----------------------------------------------------
선택> 2
scores[0]> 85
scores[1]> 95
scores[2]> 93
-----------------------------------------------------
1.학생수 | 2.점수입력 | 3.점수리스트 | 4.분석 | 5.종료
-----------------------------------------------------
선택> 3
scores[0]: 85
scores[1]: 95
scores[2]: 93
-----------------------------------------------------
1.학생수 | 2.점수입력 | 3.점수리스트 | 4.분석 | 5.종료
-----------------------------------------------------
선택> 4
최고 점수: 95
평균 점수: 91.0
-----------------------------------------------------
1.학생수 | 2.점수입력 | 3.점수리스트 | 4.분석 | 5.종료
-----------------------------------------------------
선택> 5
프로그램 종료
```

Chapter 06

1. 객체와 클래스에 대한 설명으로 틀린 것은 무엇입니까?

❶ 클래스는 객체를 생성하기 위한 설계도(청사진)와 같은 것이다.
❷ new 연산자로 클래스의 생성자를 호출함으로써 객체가 생성된다.
❸ 하나의 클래스로 하나의 객체만 생성할 수 있다.
❹ 객체는 클래스의 인스턴스이다.

2. 클래스의 구성 멤버가 아닌 것은 무엇입니까?

❶ 필드(field)
❷ 생성자(constructor)
❸ 메소드(method)
❹ 로컬 변수(local variable)

3. 필드, 생성자, 메소드에 대한 설명으로 틀린 것은 무엇입니까?

❶ 필드는 객체의 데이터를 저장한다.
❷ 생성자는 객체의 초기화를 담당한다.
❸ 메소드는 객체의 동작 부분으로, 실행 코드를 가지고 있는 블록이다.
❹ 클래스는 반드시 필드와 메소드를 가져야 한다.

4. 필드에 대한 설명으로 틀린 것은 무엇입니까?

❶ 필드는 메소드에서 사용할 수 있다.
❷ 인스턴스 필드 초기화는 생성자에서 할 수 있다.
❸ 필드는 반드시 생성자 선언 전에 선언되어야 한다.
❹ 필드는 초기값을 주지 않더라도 기본값으로 자동 초기화된다.

5. 생성자에 대한 설명으로 틀린 것은 무엇입니까?

❶ 객체를 생성하려면 생성자 호출이 반드시 필요한 것은 아니다.
❷ 생성자는 다른 생성자를 호출하기 위해 this()를 사용할 수 있다.
❸ 생성자가 선언되지 않으면 컴파일러가 기본 생성자를 추가한다.
❹ 외부에서 객체를 생성할 수 없도록 생성자에 private 접근 제한자를 붙일 수 있다.

확인문제

6. 메소드에 대한 설명으로 틀린 것은 무엇입니까?

1. 리턴값이 없는 메소드는 리턴 타입을 void로 해야 한다.
2. 리턴 타입이 있는 메소드는 리턴값을 지정하기 위해 반드시 return 문이 있어야 한다.
3. 매개값의 수를 모를 경우 "…"를 이용해서 매개변수를 선언할 수 있다.
4. 메소드의 이름은 중복해서 선언할 수 없다.

7. 메소드 오버로딩에 대한 설명으로 틀린 것은 무엇입니까?

1. 동일한 이름의 메소드를 여러 개 선언하는 것을 말한다.
2. 반드시 리턴 타입이 달라야 한다.
3. 매개변수의 타입, 수, 순서를 다르게 선언해야 한다.
4. 매개값의 타입 및 수에 따라 호출될 메소드가 선택된다.

8. 인스턴스 멤버와 정적 멤버에 대한 설명으로 틀린 것은 무엇입니까?

1. 정적 멤버는 static으로 선언된 필드와 메소드를 말한다.
2. 인스턴스 필드는 생성자 및 정적 블록에서 초기화될 수 있다.
3. 정적 필드와 정적 메소드는 객체 생성 없이 클래스를 통해 접근할 수 있다.
4. 인스턴스 필드와 메소드는 객체를 생성하고 사용해야 한다.

9. final 필드와 상수(static final)에 대한 설명으로 틀린 것은 무엇입니까?

1. final 필드와 상수는 초기값이 저장되면 값을 변경할 수 없다.
2. final 필드와 상수는 생성자에서 초기화될 수 있다.
3. 상수의 이름은 대문자로 작성하는 것이 관례이다.
4. 상수는 객체 생성 없이 클래스를 통해 사용할 수 있다.

10. 패키지에 대한 설명으로 틀린 것은 무엇입니까?

1. 패키지는 클래스들을 그룹화시키는 기능을 한다.
2. 클래스가 패키지에 소속되려면 패키지 선언을 반드시 해야 한다.
3. import 문은 다른 패키지의 클래스를 사용할 때 필요하다.
4. com.mycom 패키지에 소속된 클래스는 com.yourcom에 옮겨 놓아도 동작한다.

11. 접근 제한에 대한 설명으로 틀린 것은 무엇입니까?

❶ 접근 제한자는 클래스, 필드, 생성자, 메소드의 사용을 제한한다.

❷ public 접근 제한은 아무런 제한 없이 해당 요소를 사용할 수 있게 한다.

❸ default 접근 제한은 해당 클래스 내부에서만 사용을 허가한다.

❹ 외부에서 접근하지 못하도록 하려면 private 접근 제한을 해야 한다.

12. 다음 클래스에서 해당 멤버가 필드, 생성자, 메소드 중 어떤 것인지 ( ) 안에 적어보세요.

```
public class Member {
    private String name; ──────────→ (          )
    public Member(String name) { … } ──────────→ (          )
    public void setName(String name) { … } ──────────→ (          )
}
```

13. 현실 세계의 회원을 Member 클래스로 모델링하려고 합니다. 회원의 데이터로는 이름, 아이디, 패스워드, 나이가 있습니다. 이 데이터들을 가지는 Member 클래스를 선언해보세요.

데이터 이름	필드 이름	타입
이름	name	문자열
아이디	id	문자열
패스워드	password	문자열
나이	age	정수

14. 13번 문제에서 작성한 Member 클래스에 생성자를 추가하려고 합니다. 다음과 같이 name 필드와 id 필드를 외부에서 받은 값으로 초기화하도록 생성자를 선언해보세요.

```
Member user1 = new Member("홍길동", "hong");
```

확인문제

15. login() 메소드를 호출할 때에는 매개값으로 id와 password를 제공하고, logout() 메소드는 id만 매개값으로 제공하려고 합니다. 다음 조건과 예제 코드를 보고 MemberService 클래스에서 login(), logout() 메소드를 선언해보세요.

❶ login() 메소드는 매개값 id가 "hong", 매개값 password가 "12345" 일 경우에만 true로 리턴

❷ logout() 메소드는 id + "님이 로그아웃 되었습니다"가 출력되도록 할 것

리턴 타입	메소드 이름	매개변수(타입)
boolean	login	id(String), password(String)
void	logout	id(String)

```
MemberService memberService = new MemberService();
boolean result = memberService.login("hong", "12345");
if(result) {
  System.out.println("로그인 되었습니다.");
  memberService.logout("hong");
} else {
  System.out.println("id 또는 password가 올바르지 않습니다.");
}
```

16. println() 메소드는 매개값을 콘솔에 출력합니다. println() 메소드의 매개값으로는 int, boolean, double, String 타입 값을 줄 수 있습니다. 다음 조건과 예제 코드를 보고 Printer 클래스에서 println() 메소드를 오버로딩하여 선언해보세요.

리턴 타입	메소드 이름	매개변수(타입)
void	logout	value (int, boolean, double, String)

```
Printer printer = new Printer();
printer.println(10);
printer.println(true);
printer.println(5.7);
printer.println("홍길동");
```

17. 16번 문제에서는 Printer 객체를 생성하고 println() 메소드를 호출했습니다. 이번에는 Printer 객체를 생성하지 않고도 다음과 같이 호출할 수 있도록 Printer 클래스를 수정해보세요.

```
Printer.println(10);
Printer.println(true);
Printer.println(5.7);
Printer.println("홍길동");
```

18. 다음 예제 코드가 실행되면 "같은 ShopService 객체입니다."라는 메시지가 출력되도록, 싱글톤 패턴을 사용해서 ShopService 클래스를 작성해보세요.

```
ShopService obj1 = ShopService.getInstance();
ShopService obj2 = ShopService.getInstance();

if(obj1 == obj2) {
  System.out.println("같은 ShopService 객체입니다.");
} else {
  System.out.println("다른 ShopService 객체입니다.");
}
```

19. 은행 계좌 객체인 Account 객체는 잔고(balance) 필드를 가지고 있습니다. balance 필드는 음수값이 될 수 없고, 최대 백만 원까지만 저장할 수 있습니다. 외부에서 balance 필드를 마음대로 변경하지 못하도록 하고, 0 <= balance <= 1,000,000 범위의 값만 가질 수 있도록 Account 클래스를 작성해보세요.

❶ Setter와 Getter를 이용

❷ 0과 1,000,000은 MIN_BALANCE와 MAX_BALANCE 상수를 선언해서 이용

❸ Setter의 매개값이 음수이거나 백만 원을 초과하면 현재 balance 값을 유지

```
Account account = new Account();

account.setBalance(10000);
System.out.println("현재 잔고: " + account.getBalance());    //현재 잔고: 10000

account.setBalance(-100);
System.out.println("현재 잔고: " + account.getBalance());    //현재 잔고: 10000
```

확인문제

```
account.setBalance(2000000);
System.out.println("현재 잔고: " + account.getBalance());    //현재 잔고: 10000

account.setBalance(300000);
System.out.println("현재 잔고: " + account.getBalance());    //현재 잔고: 300000
```

20. 다음은 키보드로부터 계좌 정보를 입력받아 계좌를 관리하는 프로그램입니다. 계좌는 Account 객체로 생성되고 BankApplication에서 길이 100인 Account[ ] 배열로 관리됩니다. 실행 결과를 보고, Account와 BankApplication 클래스를 작성해보세요(키보드로 입력받을 때는 Scanner의 nextLine() 메소드를 사용).

[계좌생성 실행 결과] ❶

```
-----------------------------------------------------------
1.계좌생성 | 2.계좌목록 | 3.예금 | 4.출금 | 5.종료
-----------------------------------------------------------
선택> 1
-------------
계좌생성
-------------
계좌번호: 111-111
계좌주: 홍길동
초기입금액: 10000
결과: 계좌가 생성되었습니다.
-----------------------------------------------------------
1.계좌생성 | 2.계좌목록 | 3.예금 | 4.출금 | 5.종료
-----------------------------------------------------------
선택> 1
-------------
계좌생성
-------------
계좌번호: 111-222
계좌주: 강자바
초기입금액: 20000
결과: 계좌가 생성되었습니다.
```

[계좌목록 실행 결과] ❷

```
-----------------------------------------------------------
1.계좌생성 | 2.계좌목록 | 3.예금 | 4.출금 | 5.종료
-----------------------------------------------------------
선택> 2
-------------
계좌목록
-------------
111-111    홍길동    10000
111-222    강자바    20000
```

[계좌목록 실행 결과] ❸

```
-----------------------------------------------------------
1.계좌생성 | 2.계좌목록 | 3.예금 | 4.출금 | 5.종료
-----------------------------------------------------------
선택> 3
-------------
예금
-------------
계좌번호: 111-111
예금액: 5000
```

[계좌목록 실행 결과] ❹

```
-----------------------------------------------------------
1.계좌생성 | 2.계좌목록 | 3.예금 | 4.출금 | 5.종료
-----------------------------------------------------------
선택> 4
-------------
출금
-------------
계좌번호: 111-222
출금액: 3000
결과: 출금이 성공되었습니다.
```

[계좌목록 실행 결과] ❺

```
-----------------------------------------------------------
1.계좌생성 | 2.계좌목록 | 3.예금 | 4.출금 | 5.종료
-----------------------------------------------------------
선택> 2
-------------
계좌목록
-------------
111-111    홍길동    15000
111-222    강자바    17000
-----------------------------------------------------------
1.계좌생성 | 2.계좌목록 | 3.예금 | 4.출금 | 5.종료
-----------------------------------------------------------
선택> 5
프로그램 종료
```

Chapter 07

1. 자바의 상속에 대한 설명 중 틀린 것은 무엇입니까?

❶ 자바는 다중 상속을 허용한다.
❷ 부모의 메소드를 자식 클래스에서 재정의(오버라이딩)할 수 있다.
❸ 부모의 private 접근 제한을 갖는 필드와 메소드는 상속의 대상이 아니다.
❹ final 클래스는 상속할 수 없고, final 메소드는 오버라이딩할 수 없다.

2. 클래스 타입 변환에 대한 설명 중 틀린 것은 무엇입니까?

❶ 자식 객체는 부모 타입으로 자동 타입 변환된다.
❷ 부모 객체는 어떤 자식 타입으로도 강제 타입 변환된다.
❸ 자동 타입 변환을 이용해서 필드와 매개변수의 다형성을 구현한다.
❹ 강제 타입 변환 전에 instanceof 연산자로 변환 가능한지 검사하는 것이 좋다.

3. final 키워드에 대한 설명으로 틀린 것은 무엇입니까?

❶ final 클래스는 부모 클래스로 사용할 수 있다.
❷ final 필드는 초기화된 후에는 변경할 수 없다.
❸ final 메소드는 재정의(오버라이딩)할 수 없다.
❹ static final 필드는 상수를 말한다.

4. 오버라이딩(Overriding)에 대한 설명으로 틀린 것은 무엇입니까?

❶ 부모 메소드의 시그너처(리턴 타입, 메소드명, 매개변수)와 동일해야 한다.
❷ 부모 메소드보다 좁은 접근 제한자를 붙일 수 없다.(예: public (부모) ➜ private (자식)).
❸ @Override 어노테이션을 사용하면 재정의가 확실한지 컴파일러가 검증한다.
❹ protected 접근 제한을 갖는 메소드는 다른 패키지의 자식 클래스에서 재정의할 수 없다.

5. 추상 클래스에 대한 설명으로 틀린 것은 무엇입니까?

❶ 직접 객체를 생성할 수 없고, 상속만 할 수 있다.
❷ 추상 메소드를 반드시 가져야 한다.
❸ 추상 메소드는 자식 클래스에서 재정의(오버라이딩)할 수 있다.
❹ 추상 메소드를 재정의하지 않으면 자식 클래스도 추상 클래스가 되어야 한다.

확인문제

6. Parent 클래스를 상속해서 Child 클래스를 다음과 같이 작성했는데, Child 생성자에서 컴파일 에러가 발생했습니다. 그 이유와 해결 방법을 설명해보세요.

```
public class Parent {
  public String name;

  public Parent(String name) {
    this.name = name;
  }
}
```

```
public class Child extends Parent {
  public int studentNo;

  public Child(String name, int studentNo) {
    this.name = name;
    this.studentNo = studentNo;
  }
}
```

7. Parent 클래스를 상속받아 Child 클래스를 다음과 같이 작성했습니다. ChildExample 클래스를 실행했을 때 호출되는 각 클래스의 생성자의 순서를 생각하면서 출력 결과를 작성해보세요.

```
public class Parent {
  public String nation;

  public Parent() {
    this("대한민국");
    System.out.println("Parent() call");
  }

  public Parent(String nation) {
    this.nation = nation;
    System.out.println("Parent(String nation) call");
  }
}
```

```
public class Child extends Parent {
  public String name;

  public Child() {
    this("홍길동");
    System.out.println("Child() call");
  }

  public Child(String name) {
    this.name = name;
    System.out.println("Child(String name) call");
  }
}
```

```
public class ChildExample {
  public static void main(String[] args) {
    Child child = new Child();
  }
}
```

8. Tire 클래스를 상속받아 SnowTire 클래스를 다음과 같이 작성했습니다. SnowTireExample 클래스를 실행했을 때 출력 결과를 작성해보세요.

```
public class Tire {
  public void run() {
    System.out.println("일반 타이어가 굴러갑니다.");
  }
}
```

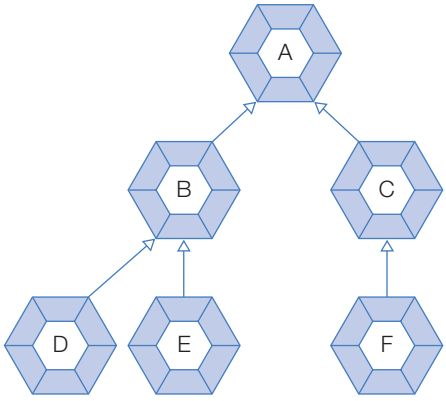
```
public class SnowTire extends Tire {
  @Override
  public void run() {
    System.out.println("스노우 타이어가 굴러갑니다.");
  }
}
```

확인문제

```
public class SnowTireExample {
  public static void main(String[] args) {
    SnowTire snowTire = new SnowTire();
    Tire tire = snowTire;

    snowTire.run();
    tire.run();
  }
}
```

9. A, B, C, D, E, F 클래스가 다음과 같이 상속 관계에 있을 때 다음 빈칸에 들어올 수 없는 코드를 선택하세요.

```
//변수 대입
B b = [          ];

-----------------------------------

//메소드 선언
void method(B b) { … }

//메소드 호출
method([          ])
```



❶ new B()
❷ (B) new A()
❸ new D()
❹ new E()

10. 다음과 같이 작성한 Computer 클래스에서 컴파일 에러가 발생했습니다. 그 이유를 설명해보세요.

```
public abstract class Machine {
  public void powerOn() { }
  public void powerOff() { }
  public abstract void work();
}
```

```
public class Computer extends Machine {
}
```

11. MainActivity의 onCreate()를 실행할 때 Activity의 onCreate()도 실행시키고 싶습니다. 밑줄에 들어갈 코드를 작성해보세요.

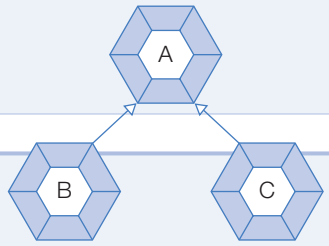
```
public class Activity {
  public void onCreate() {
    System.out.println("기본적인 실행 내용");
  }
}
```

```
public class MainActivity extends Activity {
    @Override
    public void onCreate() {
         __________.onCreate();
        System.out.println("추가적인 실행 내용");
    }
}
```

12. 다음과 같은 Example 클래스에서 action() 메소드를 호출할 때 매개값이 C 객체일 경우에만 method2()가 호출되도록 밑줄에 들어갈 코드를 작성해보세요.

```
public class A {
  public void method1() {
    System.out.println("A-method1()");
  }
}
```

```
public class B extends A {
  public void method1() {
    System.out.println("B-method1()");
  }
}
```



확인문제

```
public class C extends A {
  public void method1() {
    System.out.println("C-method1()");
  }
  public void method2() {
    System.out.println("C-method2()");
  }
}
```

```
public class Example {
  public static void action(A a) {
    a.method1();
    if( ____________________________ ) {
          c.method2();
    }
  }

  public static void main(String[] args) {
    action(new A());
    action(new B());
    action(new C());
  }
}
```

Chapter 08

1. 인터페이스에 대한 설명으로 틀린 것은 무엇입니까?

❶ 인터페이스로 객체(인스턴스)를 생성할 수 있다.
❷ 인터페이스는 다형성의 주된 기술로 사용된다.
❸ 인터페이스를 구현한 객체는 인터페이스로 동일하게 사용할 수 있다.
❹ 인터페이스를 사용함으로써 객체 교체가 쉬워진다.

2. 인터페이스의 구성 멤버에 대한 설명으로 틀린 것은 무엇입니까?

❶ 인터페이스는 인스턴스 필드가 없고 상수를 멤버로 가진다.
❷ 추상 메소드는 구현 클래스가 재정의해야 하는 멤버이다.
❸ 디폴트 메소드는 구현 클래스에서 재정의할 수 없다.
❹ 정적 멤버는 구현 객체가 없어도 사용할 수 있는 멤버이다.

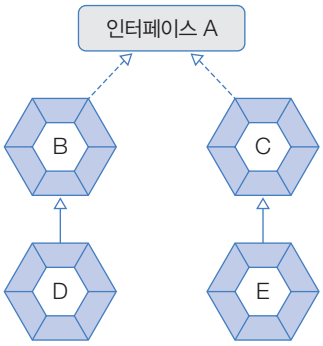
3. 인터페이스 다형성에 대한 설명으로 틀린 것은 무엇입니까?

❶ 필드가 인터페이스 타입일 경우 다양한 구현 객체를 대입할 수 있다.
❷ 매개변수가 인터페이스 타입일 경우 다양한 구현 객체를 대입할 수 있다.
❸ 배열이 인터페이스 타입일 경우 다양한 구현 객체를 저장할 수 있다.
❹ 구현 객체를 인터페이스 타입으로 변환하려면 강제 타입 변환을 해야 한다.

4. 인터페이스 A를 B와 C가 구현하고 B를 상속해서 D 클래스를, C를 상속해서 E 클래스를 만들었습니다. 다음 빈칸에 들어올 수 있는 것을 모두 선택하세요.

```
//메소드 선언
void method(A a) { … }

//메소드 호출
method(                    );
```

❶ new B()
❷ new C()
❸ new D()
❹ new E()



확인문제

5. TV 클래스를 실행했을 때 "TV를 켰습니다."라고 출력되도록 밑줄과 박스에 들어갈 코드를 작성해보세요.

```
public interface Remocon {
  public void powerOn();
}
```

```
public class TV ____________________ {
  ┌──────────────────────────────────────┐
  │                                      │
  └──────────────────────────────────────┘
  public static void main(String[] args) {
    Remocon r = new TV();
    r.powerOn();
  }
}
```

6. Soundable 인터페이스는 다음과 같은 sound() 추상 메소드를 가지고 있습니다. SoundableExample 클래스의 printSound() 메소드는 매개변수 타입으로 Soundable 인터페이스를 가집니다. printSound()를 호출할 때 Cat과 Dog 객체를 주고 실행하면 각각 "야옹"과 "멍멍"이 출력되도록 Cat과 Dog 클래스를 작성해보세요.

```
public interface Soundable {
  public String sound();
}
```

```
public class SoundableExample {
  public static void printSound(Soundable soundable) {
    System.out.println(soundable.sound());
  }

  public static void main(String[] args) {
    printSound(new Cat());
    printSound(new Dog());
  }
}
```

7. DaoExample 클래스의 main() 메소드에서 dbWork() 메소드를 호출할 때 OracleDao와 MySqlDao 객체를 매개값으로 주고 호출했습니다. dbWork() 메소드는 두 객체를 모두 매개값으로 받기 위해 DataAccessObject 타입의 매개변수를 가지고 있습니다. 실행 결과를 보고 DataAccessObject 인터페이스와 OracleDao, MySqlDao 구현 클래스를 각각 작성해보세요.

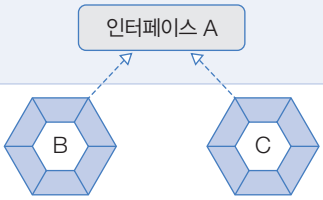
```
public class DaoExample {
  public static void dbWork(DataAccessObject dao) {
    dao.select();
    dao.insert();
    dao.update();
    dao.delete();
  }

  public static void main(String[] args) {
    dbWork(new OracleDao());
    dbWork(new MySqlDao());
  }
}
```

실행 결과

```
Oracle DB에서 검색
Oracle DB에 삽입
Oracle DB를 수정
Oracle DB에서 삭제
MySql DB에서 검색
MySql DB에 삽입
MySql DB를 수정
MySql DB에서 삭제
```

8. 다음과 같이 인터페이스와 클래스가 선언되어 있습니다. action() 메소드를 호출할 때 매개값이 C 객체일 경우에만 method2()가 호출되도록 밑줄에 들어갈 코드를 작성해보세요.

```
public interface A {
  public void method1();
}
```



확인문제

```
public class B implements A {
  @Override
  public void method1() {
    System.out.println("B - method1()");
  }
}
```

```
public class C implements A {
  @Override
  public void method1() {
    System.out.println("C - method1()");
  }

  public void method2() {
    System.out.println("C - method2()");
  }
}
```

```
public class Example {
  public static void action(A a) {
    a.method1();
    if ( ____________________________ ) {
      c.method2();
    }
  }

  public static void main(String[] args) {
    action(new B());
    action(new C());
  }
}
```

Chapter 09

1. 중첩 멤버 클래스에 대한 설명으로 틀린 것은 무엇입니까?

❶ 인스턴스 멤버 클래스는 바깥 클래스의 객체가 있어야 사용될 수 있다.

❷ 정적 멤버 클래스는 바깥 클래스의 객체가 없어도 사용될 수 있다.

❸ 인스턴스 멤버 클래스 내부에는 바깥 클래스의 모든 필드와 메소드를 사용할 수 있다.

❹ 정적 멤버 클래스 내부에는 바깥 클래스의 인스턴스 필드를 사용할 수 있다.

2. 로컬 클래스에 대한 설명으로 틀린 것은 무엇입니까?

❶ 로컬 클래스는 메소드 내부에 선언된 클래스를 말한다.

❷ 로컬 클래스는 바깥 클래스의 필드와 메소드를 사용할 수 있다.

❸ 로컬 클래스는 static 키워드를 이용해서 정적 클래스로 만들 수 있다.

❹ final 특성을 가진 매개변수나 로컬 변수만 로컬 클래스 내부에서 사용할 수 있다.

3. 익명 객체에 대한 설명으로 틀린 것은 무엇입니까?

❶ 익명 객체는 클래스를 상속하거나 인터페이스를 구현해야만 생성될 수 있다.

❷ 익명 객체는 필드, 매개변수, 로컬 변수의 초기값으로 주로 사용된다.

❸ 익명 객체에는 생성자를 선언할 수 있다.

❹ 익명 객체는 주로 재정의된 메소드를 멤버로 가진다.

4. 다음과 같이 Car 클래스 내부에 Tire와 Engine 클래스가 멤버로 선언되어 있습니다. CarExample 클래스에서 Tire와 Engine 객체를 생성하는 코드를 작성해보세요.

```
public class Car {
  class Tire {}
  static class Engine {}
}
```

```
public class CarExample {
  public static void main(String[] args) {
    Car myCar = new Car();
    Car.Tire tire = ____________________;
```

확인문제

```
    Car.Engine engine = _____________________________;
  }
}
```

5. Action 인터페이스는 다음과 같이 work() 추상 메소드를 가지고 있습니다. ActionExample 클래스의 main() 메소드에서 Action의 익명 구현 객체를 만들어 실행 결과와 동일하게 나오도록 박스 안에 들어갈 코드를 작성해보세요.

```
public interface Action {
  public void work();
}
```

```
public class ActionExample {
  public static void main(String[] args) {
    Action action =
    action.work();
  }
}
```

실행 결과

```
복사를 합니다.
```

6. AnonymousExample 클래스의 실행 결과를 보고, Vehicle 인터페이스의 익명 구현 객체를 필드와 로컬 변수의 초기값 그리고 메소드의 매개값으로 대입해보세요.

```
public interface Vehicle {
  public void run();
}
```

```
public class Anonymous {
  Vehicle field = 
  void method1() {
    Vehicle localVar = 
    localVar.run();
  }

  void method2(Vehicle v) {
    v.run();
  }
}
```

```
public class AnonymousExample {
  public static void main(String[] args) {
    Anonymous anony = new Anonymous();
    anony.field.run();
    anony.method1();
    anony.method2(
    );
  }
}
```

실행 결과

```
자전거가 달립니다.
승용차가 달립니다.
트럭이 달립니다.
```

확인문제

7. 다음 Chatting 클래스는 컴파일 에러가 발생합니다. 원인을 설명해보세요.

```
public class Chatting {
  class Chat {
    void start() {}
    void sendMessage(String message) {}
  }

  void startChat(String chatId) {
    String nickName = null;
    nickName = chatId;

    Chat chat = new Chat() {
      @Override
      public void start() {
        while(true) {
          String inputData = "안녕하세요";
          String message = "[" + nickName + "] " + inputData;
          sendMessage(message);
        }
      }
    };

    chat.start();
  }
}
```

Chapter 10

1. 자바 라이브러리에 대한 설명으로 틀린 것은 무엇입니까?

❶ 일반적으로 JAR(~.jar) 파일 형태로 존재한다.
❷ JAR 파일 안에는 클래스 및 인터페이스의 소스 파일이 있다.
❸ 라이브러리에 포함된 모든 패키지는 프로그램에서 접근이 가능하다.
❹ 이클립스 프로젝트에서 사용할 경우 Build Path에 JAR 파일을 추가한다.

2. 모듈에 대한 설명으로 틀린 것은 무엇입니까?

❶ 모듈은 패키지 관리 기능까지 포함된 라이브러리라고 볼 수 있다.
❷ 모듈에 포함된 일부 패키지는 은닉해서 접근할 수 없도록 할 수 있다.
❸ 모듈은 모듈 기술자가 반드시 존재할 필요는 없다.
❹ 모듈도 라이브러리이므로 JAR 파일 형태로 배포될 수 있다.

3. 모듈 기술자(module-info.java)에 기술되는 내용으로 틀린 것은 무엇입니까?

❶ exports는 외부에서 접근할 수 있는 패키지를 기술한다.
❷ requires는 의존 모듈을 기술한다.
❸ requires를 기술할 때에는 exports를 기술할 수 없다.
❹ transitive는 전이 의존 모듈을 기술한다.

4. 집합 모듈에 대한 설명으로 틀린 것은 무엇입니까?

❶ 한 번의 의존 설정으로 여러 모듈을 사용할 수 있도록 해준다.
❷ 집합 모듈 기술자에는 requires transitive로 의존 모듈을 기술한다.
❸ 집합 모듈 기술자에는 requires transitive로 다른 집합 모듈을 기술할 수 있다.
❹ 집합 모듈을 의존 설정할 경우에는 다른 모듈을 의존 설정할 수 없다.

5. 자바 표준 모듈에 대한 설명으로 틀린 것은 무엇입니까?

❶ java.base 모듈은 기본 모듈이므로 requires하지 않아도 사용할 수 있다.
❷ java.base 모듈에 속한 패키지는 import 없이도 사용할 수 있다.
❸ java.se 모듈은 JDK의 전체 모듈을 사용할 수 있도록 구성된 집합 모듈이다.
❹ 자바 표준 모듈은 작은 자바 실행 환경을 만들기 위해 설계되었다.

확인문제

Chapter 11

1. 예외에 대한 설명 중 틀린 것은 무엇입니까?

❶ 예외는 사용자의 잘못된 조작, 개발자의 잘못된 코딩으로 인한 프로그램 오류를 말한다.

❷ RuntimeException의 하위 예외는 컴파일러가 예외 처리 코드를 체크하지 않는다.

❸ 예외는 try-catch 블록을 사용해서 처리된다.

❹ 자바 표준 예외만 프로그램에서 처리할 수 있다.

2. try-catch-finally 블록에 대한 설명 중 틀린 것은 무엇입니까?

❶ try {} 블록에는 예외가 발생할 수 있는 코드를 작성한다.

❷ catch {} 블록은 try {} 블록에서 발생한 예외를 처리하는 블록이다.

❸ try {} 블록에서 return 문을 사용하면 finally {} 블록은 실행되지 않는다.

❹ catch {} 블록은 예외의 종류별로 여러 개를 작성할 수 있다.

3. throws에 대한 설명으로 틀린 것은 무엇입니까?

❶ 생성자나 메소드의 선언 끝 부분에 사용되어 내부에서 발생된 예외를 떠넘긴다.

❷ throws 뒤에는 떠넘겨야 할 예외를 쉼표(,)로 구분해서 기술한다.

❸ 모든 예외를 떠넘기기 위해 간단하게 throws Exception으로 작성할 수 있다.

❹ 새로운 예외를 발생시키기 위해 사용된다.

4. throw에 대한 설명으로 틀린 것은 무엇입니까?

❶ 예외를 최초로 발생시키는 코드이다.

❷ 예외를 호출한 곳으로 떠넘기기 위해 메소드 선언부에 작성된다.

❸ throw로 발생된 예외는 일반적으로 생성자나 메소드 선언부에 throws로 떠넘겨진다.

❹ throw 키워드 뒤에는 예외 객체 생성 코드가 온다.

5. 메소드가 다음과 같이 선언되어 있습니다. 잘못된 예외 처리를 선택하세요.

```
public void method1() throws NumberFormatException, ClassNotFoundException { … }
```

❶ try { method1(); } catch (Exception e) {}

❷ void method2() throws Exception { method1(); }

❸ try { method1(); }
catch (Exception e) {}
catch (ClassNotFoundException e) {}

❹ try { method1(); }
catch (ClassNotFoundException e) {}
catch (NumberFormatException e) {}

6. 다음 코드가 실행되었을 때 출력 결과를 작성해보세요.

```
String[] strArray = { "10", "2a" };
int value = 0;
for(int i=0; i<=2; i++) {
  try {
    value = Integer.parseInt(strArray[i]);
  } catch(ArrayIndexOutOfBoundsException e) {
    System.out.println("인덱스를 초과했음");
  } catch(NumberFormatException e) {
    System.out.println("숫자로 변환할 수 없음");
  } finally {
    System.out.println(value);
  }
}
```

확인문제

7. login() 메소드에서 존재하지 않는 ID를 입력하면 NotExistIDException을 발생시키고, 잘못된 패스워드를 입력하면 WrongPasswordException을 발생시키려고 합니다. 다음 LoginExample의 실행 결과를 보고 빈칸을 채워보세요.

```
public class NotExistIDException extends Exception {
  public NotExistIDException() {}
  public NotExistIDException(String message) {
    
  }
}
```

```
public class WrongPasswordException extends Exception {
  public WrongPasswordException() {}
  public WrongPasswordException(String message) {
    
  }
}
```

```
public class LoginExample {
  public static void main(String[] args) {
    try {
      login("white", "12345");
    } catch(Exception e) {
      System.out.println(e.getMessage());
    }

    try {
      login("blue", "54321");
    } catch(Exception e) {
      System.out.println(e.getMessage());
    }
  }

  public static void login(String id, String password)  {
    //id가 blue가 아니라면 NotExistIDException을 발생시킴
    if(!id.equals("blue")) {
      
    }
```

```
        //password가 12345가 아니라면 WrongPasswordException을 발생시킴
        if(!password.equals("12345")) {
            
        }
    }
}
```

실행 결과

```
아이디가 존재하지 않습니다.
```

실행 결과

```
패스워드가 틀립니다.
```

8. FileWriter는 파일을 열고 데이터를 저장하는 클래스입니다. 예외 발생 여부와 상관 없이 마지막에는 close() 메소드를 실행해서 파일을 닫아주려고 합니다. 왼쪽 코드는 try-catch-finally를 이용해서 작성한 코드로, 리소스 자동 닫기를 이용하도록 수정하고 싶습니다. 수정한 코드를 오른쪽에 작성해보세요.

```
import java.io.IOException;

public class FileWriter implements AutoCloseable {
  public FileWriter(String filePath) throws IOException {
    System.out.println(filePath + " 파일을 엽니다.");
  }

  public void write(String data) throws IOException {
    System.out.println(data + "를 파일에 저장합니다.");
  }

  @Override
  public void close() throws IOException {
    System.out.println("파일을 닫습니다.");
  }
}
```

확인문제

```
import java.io.IOException;

public class FileWriterExample {
  public static void main(String[]
      args) {
    FileWriter fw = null;
    try {
      fw = new FileWriter("file.txt");
      fw.write("Java");
    } catch (IOException e) {
      e.printStackTrace();
    } finally {
      try { fw.close(); } catch
        (IOException e) {}
    }
  }
}
```

```
import java.io.IOException;

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

Chapter 12

1. API 도큐먼트에 대한 설명으로 틀린 것은 무엇입니까?

❶ 자바 표준 라이브러리를 프로그램에서 어떻게 사용할 수 있는지를 설명하고 있다.

❷ 클래스의 상속 관계 및 자식 클래스들이 무엇이 있는지 알 수 있다.

❸ 생성자 선언부, 필드의 타입, 메소드의 선언부를 확인할 수 있다.

❹ public, protected, default, private 접근 제한을 가지는 멤버들을 확인할 수 있다.

2. java.base 모듈에 대한 설명으로 틀린 것은 무엇입니까?

❶ 모든 표준 모듈이 의존하는 기본 모듈이다.

❷ 모듈 기술자에 requires를 하지 않아도 사용할 수 있는 모듈이다.

❸ java.base의 패키지에는 java.lang, java.util, java.io, java.net, java.sql 등이 있다.

❹ java.lang 패키지를 제외한 다른 패키지는 import 문을 필요로 한다.

3. Object 클래스에 대한 설명 중 틀린 것은 무엇입니까?

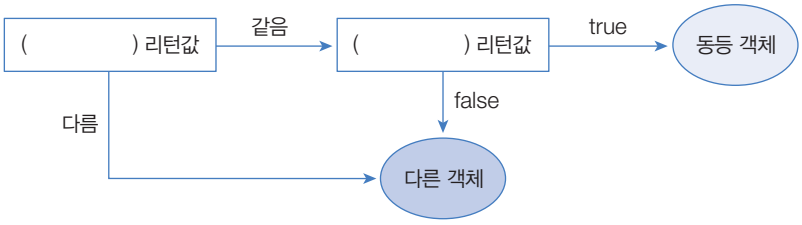
❶ 모든 자바 클래스의 최상위 부모 클래스이다.

❷ Object의 equals() 메소드는 == 연산자와 동일하게 번지를 비교한다.

❸ Object의 hashCode() 메소드는 동등 비교 시 활용된다.

❹ Object의 toString() 메소드는 객체의 필드값을 문자열로 리턴한다.

4. 객체의 동등 비교를 위해 Object의 equals()와 hashCode() 메소드를 오버라이딩했다고 가정할 경우, 메소드 호출 순서를 생각하고 다음 ( ) 안을 채워보세요.



확인문제

5. Object의 equals()와 hashCode() 메소드를 오버라이딩해서 Student의 학번(studentNum)이 같으면 동등 객체가 될 수 있도록 Student 클래스를 작성해보세요(hashCode() 메소드의 리턴값은 studentNum 필드값으로 설정).

```
public class Student {
  private String studentNum;

  public Student(String studentNum) {
    this.studentNum = studentNum;
  }

  public String getStudentNum() {
    return studentNum;
  }

  //여기에 코드를 작성하세요.
}
```

```
import java.util.HashSet;

public class StudentExample {
  public static void main(String[] args) {
    //Student를 저장하는 HashSet 생성
    HashSet<Student> hashSet = new HashSet<Student>();

    //Student 저장
    hashSet.add(new Student("1"));
    hashSet.add(new Student("1"));   //같은 학번이므로 중복 저장이 안됨
    hashSet.add(new Student("2"));

    //저장된 Student 수 출력
    System.out.println("저장된 Student 수: " + hashSet.size());
  }
}
```

6. Member 클래스에서 Object의 toString() 메소드를 오버라이딩해서 MemberExample 클래스의 실행 결과처럼 나오도록 작성해보세요.

```
public class Member {
  private String id;
  private String name;

  public Member(String id, String name) {
    this.id = id;
    this.name = name;
  }

  //여기에 코드를 작성하세요.
}
```

```
public class MemberExample {
  public static void main(String[] args) {
    Member member = new Member("blue", "이파란");
    System.out.println(member);
  }
}
```

실행 결과

```
blue: 이파란
```

7. System 클래스에 대한 설명 중 틀린 것은 무엇입니까?

❶ System 클래스는 정적 필드와 정적 메소드만 제공한다.

❷ System.out은 콘솔에 출력할 때, System.in은 키보드에서 입력받을 때 사용한다.

❸ milisTime()과 nanoTime() 메소드는 현재 시간에 대한 long값을 리턴한다.

❹ exit() 메소드는 프로세스(JVM)를 종료시킨다.

확인문제

8. 다음 전체 코드를 실행하는 데 걸린 시간을 구하는 코드를 작성해보세요(단위 나노초).

```
int[] scores = new int[1000];
for(int i=0; i<scores.length; i++) {
  scores[i] = i;
}

int sum = 0;
for(int score : scores) {
  sum += score;
}

double avg = sum / scores.length;
System.out.println(avg);
```

9. 다음 바이트 배열은 UTF-8 문자셋으로 인코딩한 데이터로, 다시 문자열로 디코딩해서 변수 data에 저장하려고 합니다. 밑줄 친 곳에 들어갈 코드를 작성해보세요.

```
public class DecodingExample {
  public static void main(String[] args) throws Exception {
    byte[] bytes = { -20, -107, -120, -21, -123, -107 };
    String str = _____________________________;
    System.out.println("str: " + str);
  }
}
```

10. 다음 코드는 1부터 100까지의 숫자를 통 문자열로 만들기 위해 += 연산자를 이용해 100번 반복하고 있습니다. 이것은 곧 100개 이상의 String 객체를 생성하는 결과를 만들기 때문에 좋은 코드라고 볼 수 없습니다. StringBuilder를 사용해서 좀 더 효율적인 코드로 개선해보세요.

```
public class StringBuilderExample {
  public static void main(String[] args) {
    String str = "";
    for(int i=1; i<=100; i++) {
      str += i;
    }
```

```
    System.out.println(str);
  }
}
```

11. 다음 문자열에서 쉼표(,)로 구분되어 있는 문자열을 StringTokenizer를 이용해서 분리시키고 출력해보세요.

```
아이디,이름,패스워드
```

12. 숫자 100과 300으로 각각 박싱된 Integer 객체를 == 연산자로 비교한 결과 100을 박싱한 Integer 객체는 true가 나오지만, 300을 박싱한 Integer 객체는 false가 나왔습니다. 그 이유를 설명하고, 값만 비교할 수 있도록 코드를 수정해보세요.

```
public class IntegerCompareExample {
  public static void main(String[] args) {
    Integer obj1 = 100;
    Integer obj2 = 100;
    Integer obj3 = 300;
    Integer obj4 = 300;

    System.out.println( obj1 == obj2 );
    System.out.println( obj3 == obj4 );
  }
}
```

13. Math 클래스가 제공하는 메소드의 리턴값이 잘못된 것은 무엇입니까?

❶ Math.ceil(5.3) ➜ 6.0

❷ Math.floor(5.3) ➜ 5.0

❸ Math.max(5.3, 2.5) ➜ 5.3

❹ Math.round(5.7) ➜ 6.0

확인문제

14. 난수를 얻는 방법을 잘못 설명한 것은 무엇입니까?

❶ Math.random() 메소드는 0.0 <= … < 1.0 사이의 실수 난수를 리턴한다.

❷ Random의 nextDouble() 메소드는 0.0 <= … < 1.0 사이의 실수 난수를 리턴한다.

❸ Random의 nextInt() 메소드는 int 타입의 허용 범위에서 난수를 리턴한다.

❹ Random의 nextInt(int n) 메소드는 0<= … <=n 사이의 정수 난수를 리턴한다.

15. 올해 12월 31일까지 몇 일이 남았는지를 구하는 코드를 작성해보세요.

16. SimpleDateFormat 클래스를 이용해서 오늘 날짜를 다음과 같이 출력하도록 코드를 작성해보세요.

```
xxxx년 xx월 xx일 x요일 xx시 xx분
```

17. 정규 표현식을 이용해 첫 번째는 알파벳으로 시작하고 두 번째부터 숫자와 알파벳으로 구성된 8~12자 사이의 ID 값인지 검사하고 싶습니다. 알파벳은 대소문자를 모두 허용한다고 할 때, 다음 밑줄에 들어갈 코드를 작성해보세요.

```
import java.util.regex.Pattern;

public class PatternMatcherExample {
  public static void main(String[] args) {
    String id = "5Angel1004";
    String regExp = _____________________________;
    boolean isMatch = ___________________________________;
    if(isMatch) {
      System.out.println("ID로 사용할 수 있습니다.");
    } else {
      System.out.println("ID로 사용할 수 없습니다.");
    }
  }
}
```

18. Class 객체에 대한 설명 중 틀린 것은 무엇입니까?

❶ Class.forName() 메소드 또는 객체의 getClass() 메소드로 얻을 수 있다.

❷ 패키지와 클래스 이름을 알 수 있다.

❸ 클래스의 생성자, 필드, 메소드에 대한 정보를 알아낼 수 있다.

❹ getResource() 메소드는 프로젝트 경로를 기준으로 리소스의 URL을 리턴한다.

19. 어노테이션(Annotation)에 대한 설명 중 틀린 것은 무엇입니까?

❶ 컴파일하거나 실행할 때 어떻게 처리해야 할 것인지를 알려주는 역할을 한다.

❷ 클래스, 필드, 생성자, 메소드를 선언하기 전에 @어노테이션을 붙일 수 있다.

❸ @어노테이션("*")일 경우 value 속성값이 *가 된다.

❹ @어노테이션("*", prop=3)일 경우 value 속성값은 *, prop 속성값은 3이 된다.

확인문제

Chapter 13

1. 제네릭에 대한 설명으로 틀린 것은 무엇입니까?

❶ 컴파일 시 강한 타입 체크를 할 수 있다.

❷ 타입 변환(casting)을 제거한다.

❸ 제네릭 타입은 타입 파라미터를 가지는 제네릭 클래스와 인터페이스를 말한다.

❹ 제네릭 메소드는 리턴 타입으로 타입 파라미터를 가질 수 없다.

2. ContainerExample 클래스의 main() 메소드는 Container 제네릭 타입을 사용하고 있습니다. main() 메소드에서 사용하는 방법을 참고해서 Container 제네릭 타입을 선언해보세요.

```
public class ContainerExample {
  public static void main(String[] args) {
    Container<String> container1 = new Container<String>();
    container1.set("홍길동");
    String str = container1.get();

    Container<Integer> container2 = new Container<Integer>();
    container2.set(6);
    int value = container2.get();
  }
}
```

3. ContainerExample 클래스의 main() 메소드는 Container 제네릭 타입을 사용하고 있습니다. main() 메소드에서 사용하는 방법을 참고해서 Container 제네릭 타입을 선언해보세요.

```
public class ContainerExample {
  public static void main(String[] args) {
    Container<String, String> container1 = new Container<String, String>();
    container1.set("홍길동", "도적");
    String name1 = container1.getKey();
    String job = container1.getValue();

    Container<String, Integer> container2 = new Container<String, Integer>();
    container2.set("홍길동", 35);
    String name2 = container2.getKey();
```

```
    int age = container2.getValue();
  }
}
```

4. 다음 Util 클래스의 정적 getValue() 메소드는 첫 번째 매개값으로 Pair 타입과 하위 타입만 받고, 두 번째 매개값으로 키값을 받습니다. 리턴값은 키값이 일치할 경우 Pair에 저장된 값을 리턴하고, 일치하지 않으면 null을 리턴하도록 Util 클래스와 getValue() 제네릭 메소드를 작성해보세요.

```
public class UtilExample {
  public static void main(String[] args) {
    Pair<String, Integer> pair = new Pair<>("홍길동", 35 );
    Integer age = Util.getValue(pair, "홍길동");
    System.out.println(age);

    ChildPair<String, Integer> childPair = new ChildPair<>("홍삼원", 20 );
    Integer childAge = Util.getValue(childPair, "홍삼순");
    System.out.println(childAge);

    /*OtherPair<String, Integer> otherPair = new OtherPair<>("홍삼원", 20);
    //OtherPair는 Pair를 상속하지 않으므로 컴파일 에러가 발생
    int otherAge = Util.getValue(otherPair, "홍삼원");
    System.out.println(otherAge);*/
  }
}
```

일치

불일치

```
public class Pair<K, V> {
  private K key;
  private V value;

  public Pair(K key, V value) {
    this.key = key;
    this.value = value;
  }

  public K getKey()   { return key; }
  public V getValue() { return value; }
}
```

확인문제

```
public class ChildPair<K, V> extends Pair<K,V> {
  public ChildPair(K k, V v) {
    super(k, v);
  }
}
```

```
public class OtherPair<K, V> {
  private K key;
  private V value;

  public OtherPair(K key, V value) {
    this.key = key;
    this.value = value;
  }

  public K getKey()   { return key; }
  public V getValue() { return value; }
}
```

Chapter 14

1. 스레드에 대한 설명 중 틀린 것은 무엇입니까?

❶ 자바 애플리케이션은 메인(main) 스레드가 main() 메소드를 실행시킨다.

❷ 작업 스레드 클래스는 Thread 클래스를 상속해서 만들 수 있다.

❸ Runnable 객체는 스레드가 실행해야 할 코드를 가지고 있는 객체라고 볼 수 있다.

❹ 스레드 실행을 시작하려면 run() 메소드를 호출해야 한다.

2. 동영상과 음악을 재생하기 위해 두 가지 스레드를 실행하려고 합니다. 밑줄 친 부분에 적당한 코드를 작성해보세요.

```
public class ThreadExample {
  public static void main(String[] args) {
    Thread thread1 = new MovieThread();
    thread1.start();

    Thread thread2 = new Thread(_____________________________);
    thread2.start();
  }
}
```

```
public class MovieThread _____________________________ {
  @Override
  public void run() {
    for(int i=0;i<3;i++) {
      System.out.println("동영상을 재생합니다.");
      try {
        Thread.sleep(1000);
      } catch (InterruptedException e) {
      }
    }
  }
}
```

확인문제

```
public class MusicRunnable _____ {
  @Override
  public void run() {
    for(int i=0;i<3;i++) {
      System.out.println("음악을 재생합니다.");
      try {
        Thread.sleep(1000);
      } catch (InterruptedException e) {
      }
    }
  }
}
```

3. 동기화 메소드와 동기화 블록에 대한 설명 중 틀린 것은 무엇입니까?

❶ 동기화 메소드와 동기화 블록은 싱글(단일) 스레드 환경에서는 필요 없다.

❷ 스레드가 동기화 메소드를 실행할 때 다른 스레드는 일반 메소드를 호출할 수 없다.

❸ 스레드가 동기화 메소드를 실행할 때 다른 스레드는 동기화 메소드를 호출할 수 없다.

❹ 스레드가 동기화 블록을 실행할 때 다른 스레드는 동기화 메소드를 호출할 수 없다.

4. 스레드 일시 정지 상태에 대한 설명 중 틀린 것은 무엇입니까?

❶ sleep() 메소드는 주어진 시간 동안 스레드가 일시 정지 상태가 된다.

❷ 스레드가 동기화 메소드를 실행할 때 다른 스레드가 동기화 메소드를 호출하게 되면 일시 정지 상태가 된다.

❸ 동기화 메소드 내에서 wait() 메소드를 호출하면 현재 스레드가 일시 정지 상태가 된다.

❹ yield() 메소드를 호출하면 현재 스레드가 일시 정지 상태가 된다.

5. interrupt() 메소드를 호출한 효과에 대한 설명 중 틀린 것은 무엇입니까?

❶ 일시 정지 상태에서 InterruptedException을 발생시킨다.

❷ 스레드를 즉시 종료한다.

❸ 스레드가 일시 정지 상태가 될 때까지 InterruptedException이 발생하지 않는다.

❹ InterruptedException이 발생하지 않았다면 isInterrupted() 메소드는 true를 리턴한다.

6. 메인 스레드에서 3초 후 MovieThread의 interrupt() 메소드를 호출해서 MovieThread를 안전하게 종료하고 싶습니다. 비어있는 부분에 적당한 코드를 작성해보세요.

```
public class ThreadExample {
  public static void main(String[] args) {
    Thread thread = new MovieThread();
    thread.start();

    try { Thread.sleep(3000); } catch (InterruptedException e) {}

    thread.interrupt();
  }
}
```

```
public class MovieThread extends Thread {
  @Override
  public void run() {
    while(true) {
      System.out.println("동영상을 재생합니다.");
      [                                    ]
    }
  }
}
```

7. wait()와 notify() 메소드에 대한 설명 중 틀린 것은 무엇입니까?

❶ 스레드가 wait()를 호출하면 일시 정지 상태가 된다.

❷ notify()를 호출하면 wait()로 일시 정지 상태에 있던 스레드가 실행 대기 상태가 된다.

❸ wait()와 notify()는 동기화 메소드 또는 블록에서 호출할 필요가 없다.

❹ wait()와 notify()는 두 스레드가 균등하게 번갈아 가면서 실행할 때 사용할 수 있다.

확인문제

8. 3초 뒤에 메인 스레드가 종료하면 MovieThread도 같이 종료되게 만들고 싶습니다. 밑줄 친 부분에 적당한 코드를 넣어보세요.

```
public class ThreadExample {
  public static void main(String[] args) {
    Thread thread = new MovieThread();
    ____________________________
    thread.start();

    try { Thread.sleep(3000); } catch (InterruptedException e) {}
  }
}
```

```
public class MovieThread extends Thread {
  @Override
  public void run() {
    while(true) {
      System.out.println("동영상을 재생합니다.");
      try { Thread.sleep(1000); } catch (InterruptedException e) {}
    }
  }
}
```

9. while 문으로 반복적인 작업을 하는 스레드를 종료시키는 방법에 대한 설명 중 최선의 방법이 아닌 것은 무엇입니까?

1. 조건식에 boolean 타입의 stop 플래그를 이용해서 while 문을 빠져나가게 한다.
2. 스레드가 반복적으로 일시 정지 상태가 된다면 InterruptedException을 발생시켜 예외 처리 코드에서 break 문으로 while 문을 빠져나가게 한다.
3. 스레드가 일시 정지 상태로 가지 않는다면 isInterrupted()나 interrupted() 메소드의 리턴값을 조사해서 true일 경우 break 문으로 while 문을 빠져나가게 한다.
4. stop() 메소드를 호출한다.

10. 스레드풀에 대한 설명 중 틀린 것은 무엇입니까?

❶ 갑작스러운 작업의 증가로 스레드의 폭증을 막기 위해 사용된다.

❷ ExecutorService 객체가 스레드풀이며 newFixedThreadPool() 메소드로 얻을 수 있다.

❸ 작업은 Runnable 또는 Callable 인터페이스를 구현해서 정의한다.

❹ execute() 메소드로 작업 처리 요청을 하면 작업이 완료될 때까지 대기(블로킹)된다.

확인문제

Chapter 15

1. 자바의 컬렉션 프레임워크에 대한 설명으로 틀린 것은 무엇입니까?

❶ List 컬렉션은 인덱스로 객체를 관리하며 중복 저장을 허용한다.

❷ Set 컬렉션은 순서를 유지하지 않으며 중복 저장을 허용하지 않는다.

❸ Map 컬렉션은 키와 값으로 구성된 Map.Entry를 저장한다.

❹ Stack은 FIFO(선입선출) 자료구조를 구현한 클래스이다.

2. List 컬렉션에 대한 설명 중 틀린 것은 무엇입니까?

❶ 대표적인 구현 클래스로는 ArrayList, Vector, LinkedList가 있다.

❷ 멀티 스레드 환경에서는 ArrayList보다는 Vector가 스레드에 안전하다.

❸ ArrayList에서 객체를 삭제하면 삭제된 위치는 비어 있게 된다.

❹ 중간 위치에 객체를 빈번히 삽입하거나 제거할 경우 LinkedList를 사용하는 것이 좋다.

3. Set 컬렉션에 대한 설명 중 틀린 것은 무엇입니까?

❶ 대표적인 구현 클래스로는 HashSet, LinkedHashSet, TreeSet이 있다.

❷ Set 컬렉션에서 객체를 하나씩 꺼내오고 싶다면 Iterator를 이용한다.

❸ HashSet은 hashCode()와 equals() 메소드를 이용해서 중복된 객체를 판별한다.

❹ Set 컬렉션에는 null을 저장할 수 없다.

4. Map 컬렉션에 대한 설명 중 틀린 것은 무엇입니까?.

❶ 대표적인 구현 클래스로는 HashMap, Hashtable, TreeMap, Properties가 있다.

❷ HashMap과 Hashtable은 hashCode()와 equals() 메소드를 이용해서 중복 키를 판별한다.

❸ 멀티 스레드 환경에서는 Hashtable보다는 HashMap이 스레드에 안전하다.

❹ Properties는 키와 값이 모두 String 타입이다.

5. 단일(싱글) 스레드 환경에서 Board 객체를 저장 순서에 맞게 읽고 싶습니다. 가장 적합한 컬렉션을 생성하도록 밑줄 친 부분에 코드를 작성해보세요.

_____ 변수 = new _____

(타입) (생성자 호출)

6. 단일(싱글) 스레드 환경에서 학번(String)를 키로, 점수(Integer)를 값으로 저장하는 가장 적합한 컬렉션을 생성하도록 밑줄 친 부분에 코드를 작성해보세요.

_____ 변수 = new _____

(타입) (생성자 호출)

7. BoardDao 객체의 getBoardList() 메소드를 호출하면 List<Board> 타입의 컬렉션을 리턴합니다. ListExample 클래스의 실행 결과를 보고, BoardDao 클래스와 getBoardList() 메소드를 작성해보세요.

```
public class Board {
  private String title;
  private String content;

  public Board(String title, String content) {
    this.title = title;
    this.content = content;
  }

  public String getTitle() { return title; }
  public String getContent() { return content; }
}
```

```
import java.util.List;

public class ListExample {
  public static void main(String[] args) {
    BoardDao dao = new BoardDao();
    List<Board> list = dao.getBoardList();
    for(Board board : list) {
      System.out.println(board.getTitle() + "-" + board.getContent());
    }
  }
}
```

실행 결과

```
제목1-내용1
```

제목2-내용2
제목3-내용3

8. HashSet에 Student 객체를 저장하려고 합니다. 학번이 같으면 동일한 Student라고 가정하고 중복 저장이 되지 않도록 하고 싶습니다. Student 객체의 해시코드는 학번이라고 가정하고 Student 클래스를 작성해보세요.

```
public class Student {
  public int studentNum;
  public String name;

  public Student (int studentNum, String name) {
    this.studentNum = studentNum;
    this.name = name;
  }

  //여기에 코드를 작성하세요.
}
```

```
import java.util.HashSet;
import java.util.Set;

public class HashSetExample {
  public static void main(String[] args) {
    Set<Student> set = new HashSet<Student>();

    set.add(new Student(1, "홍길동"));
    set.add(new Student(2, "신용권"));
    set.add(new Student(1, "조민우"));

    System.out.println("저장된 객체 수: " + set.size());
    for(Student s : set) {
      System.out.println(s.studentNum + ":" + s.name);
    }
  }
}
```

실행 결과

```
저장된 객체 수: 2
1:홍길동
2:신용권
```

9. HashMap에 아이디(String)와 점수(Integer)가 저장되어 있습니다. 실행 결과와 같이 평균 점수, 최고 점수, 최고 점수를 받은 아이디를 출력하도록 코드를 작성해보세요.

```
import java.util.HashMap;
import java.util.Map;
import java.util.Set;

public class MapExample {
  public static void main(String[] args) {
    Map<String,Integer> map = new HashMap<String,Integer>();
    map.put("blue", 96);
    map.put("hong", 86);
    map.put("white", 92);

    String name = null;   //최고 점수를 받은 아이디를 저장하는 변수
    int maxScore = 0;     //최고 점수를 저장하는 변수
    int totalScore = 0;   //점수 합계를 저장하는 변수

    //여기에 코드를 작성하세요.
  }
}
```

실행 결과

```
평균 점수: 91
최고 점수: 96
최고 점수를 받은 아이디: blue
```

확인문제

10. TreeSet에 Student 객체를 저장할 때, Student의 score 필드값을 기준으로 자동 정렬되도록 구현하고 싶습니다. TreeSet의 last() 메소드를 호출했을 때 가장 높은 score의 Student 객체가 리턴되도록 Student 클래스에서 밑줄 친 곳과 비어있는 곳에 코드를 작성해보세요.

```
public class Student _____ {
  public String id;
  public int score;

  public Student (String id, int score) {
    this.id = id;
    this.score = score;
  }

}
```

```
import java.util.TreeSet;

public class TreeSetExample {
  public static void main(String[] args) {
    TreeSet<Student> treeSet = new TreeSet<Student>();
    treeSet.add(new Student("blue", 96));
    treeSet.add(new Student("hong", 86));
    treeSet.add(new Student("white", 92));

    Student student = treeSet.last();
    System.out.println("최고 점수: " + student.score);
    System.out.println("최고 점수를 받은 아이디: " + student.id);
  }
}
```

실행 결과

```
최고 점수: 96
최고 점수를 받은 아이디: blue
```

11. LIFO와 FIFO 컬렉션에 대한 설명으로 틀린 것은 무엇입니까?

❶ Stack은 LIFO이고 Queue는 FIFO 구조를 가지고 있다.

❷ Stack에 넣는 행위는 push이고, 빼는 행위는 pop이다.

❸ Queue에 넣는 행위는 offer이고, 빼는 행위는 poll이다.

❹ Stack과 Queue는 자바에서 클래스 타입이다.

12. 동기화된 컬렉션에 대한 설명으로 틀린 것은 무엇입니까?

❶ 멀티 스레드 환경에서 안전하게 사용할 수 있는 컬렉션이다.

❷ 동기화된 컬렉션의 메소드는 synchronized 처리가 되어 있다.

❸ Vector와 HashMap은 동기화된 컬렉션이다.

❹ Collections의 synchronizedMap() 메소드는 동기화된 Map을 리턴한다.

13. 수정할 수 없는 List 컬렉션에 대한 설명으로 틀린 것은 무엇입니까?

❶ 요소를 추가, 삭제할 수 없는 List 컬렉션을 말한다.

❷ List의 of() 메소드는 수정할 수 없는 컬렉션을 생성한다.

❸ List의 copyOf() 메소드는 기존 컬렉션을 복사한 수정할 수 없는 컬렉션을 생성한다.

❹ Array.asList() 메소드는 배열로부터 수정할 수 있는 List 컬렉션을 생성한다.

확인문제

Chapter 16

1. 람다식에 대한 설명으로 틀린 것은 무엇입니까?

❶ 람다식은 함수형 인터페이스의 익명 구현 객체를 생성한다.

❷ 매개변수가 없을 경우 ( ) ➔ { … } 형태로 작성한다.

❸ (x,y) ➔ { return x+y; }는 (x,y) ➔ x+y로 바꿀 수 있다.

❹ @FunctionalInterface가 기술된 인터페이스만 람다식으로 표현이 가능하다.

2. 메소드 참조와 생성자 참조에 대한 설명으로 틀린 것은 무엇입니까?

❶ 메소드 참조는 함수적 인터페이스의 익명 구현 객체를 생성한다.

❷ 인스턴스 메소드는 "참조변수::메소드"로 기술한다.

❸ 정적 메소드는 "클래스::메소드"로 기술한다.

❹ 생성자 참조인 "클래스::new"는 매개변수가 없는 디폴트 생성자만 호출한다.

3. 다음 중 잘못 작성된 람다식은 무엇입니까?

❶ a ➔ a+3

❷ a,b ➔ a*b

❸ x ➔ System.out.println(x/5)

❹ (x,y) ➔ Math.max(x, y)

4. 다음 코드의 실행 결과를 보고 빈 곳에 들어갈 람다식을 작성해보세요.

```
public class Example {
  public static void main(String[] args) {
    Thread thread = new Thread(
      
    );
    thread.start();
  }
}
```

실행 결과

```
작업 스레드가 실행됩니다.
작업 스레드가 실행됩니다.
```

작업 스레드가 실행됩니다.

5. 다음 코드의 실행 결과를 보고 밑줄 친 곳에 들어갈 람다식을 작성해보세요.

```
public class Button {
  //정적 멤버 인터페이스(함수형 인터페이스)
  @FunctionalInterface
  public static interface ClickListener {
    void onClick();
  }

  private ClickListener clickListener;

  public void setClickListener(ClickListener clickListener) {
    this.clickListener = clickListener;
  }

  public void click() {
    this.clickListener.onClick();
  }
}
```

```
public class Example {
  public static void main(String[] args) {
    Button btnOk = new Button();
    btnOk.setClickListener( ____________________________________________ );
    btnOk.click();

    Button btnCancel = new Button();
    btnCancel.setClickListener( ____________________________________________ );
    btnCancel.click();
  }
}
```

실행 결과

```
Ok 버튼을 클릭했습니다.
Cancel 버튼을 클릭했습니다.
```

확인문제

6. 다음 코드를 보고, Function 함수형 인터페이스를 작성해보세요.

```
public class Example {
  public static double calc(Function fun) {
    double x = 10;
    double y = 4;
    return fun.apply(x, y);
  }

  public static void main(String[] args) {
    double result = calc( (x, y) -> (x / y) );
    System.out.println("result: " + result);
  }
}
```

실행 결과

```
result: 2.5
```

7. 다음은 배열 항목 중에 최대값 또는 최소값을 찾는 코드입니다. maxOrMin() 메소드를 호출할 때 빈 곳에 람다식을 작성해보세요.

```
@FunctionalInterface
public interface Operator {
  public int apply(int x, int y);
}
```

```
public class Example {
  private static int[] scores = { 10, 50, 3 };

  public static int maxOrMin(Operator operator) {
    int result = scores[0];
    for(int score : scores) {
      result = operator.apply(result, score);
    }
    return result;
  }
```

```
  public static void main(String[] args) {
    //최대값 얻기
    int max = maxOrMin(
      
    );
    System.out.println("최대값: " + max);

    //최소값 얻기
    int min = maxOrMin(
      

    );
    System.out.println("최소값: " + min);
  }
}
```

실행 결과

```
최대값: 50
최소값: 3
```

8. 다음은 학생의 영어 평균 점수와 수학 평균 점수를 계산하는 코드입니다. 빈 곳에 avg() 메소드를 작성해보세요.

```
@FunctionalInterface
public interface Function<T> {
  public double apply(T t);
}
```

```
public class Student {
  private String name;
  private int englishScore;
  private int mathScore;

  public Student(String name, int englishScore, int mathScore) {
    this.name = name;
    this.englishScore = englishScore;
```

확인문제

```
    this.mathScore = mathScore;
  }

  public String getName() { return name; }
  public int getEnglishScore() { return englishScore; }
  public int getMathScore() { return mathScore; }
}
```

```
public class Example {
  private static Student[] students = {
    new Student("홍길동", 90, 96),
    new Student("신용권", 95, 93)
  };

  //avg() 메소드 작성
  
  public static void main(String[] args) {
    double englishAvg = avg( s -> s.getEnglishScore() );
    System.out.println("영어 평균 점수: " + englishAvg);

    double mathAvg = avg( s -> s.getMathScore() );
    System.out.println("수학 평균 점수: " + mathAvg);
  }
}
```

9. 8번 문제에서 Example 클래스의 main() 메소드를 실행할 때, avg() 메소드의 매개값으로 람다식을 사용하지 않고 메소드 참조로 변경해보세요.

```
double englishAvg = avg( s -> s.getEnglishScore() );
➜ double englishAvg = avg( ____________________ );

double mathAvg = avg( s -> s.getMathScore() );
➜ double mathAvg = avg( ____________________ );
```

Chapter 17

1. 스트림에 대한 설명으로 틀린 것은 무엇입니까?

❶ 스트림은 내부 반복자를 사용하기 때문에 코드가 간결해진다.
❷ 스트림은 요소를 분리해서 병렬 처리시킬 수 있다.
❸ 스트림은 람다식을 사용해서 요소 처리 내용을 기술한다.
❹ 스트림은 요소를 모두 처리하고 나서 처음부터 요소를 다시 반복시킬 수 있다.

2. 스트림을 얻을 수 있는 소스가 아닌 것은 무엇입니까?

❶ 컬렉션(List)
❷ int, long, double 범위
❸ 디렉토리
❹ 배열

3. 스트림 파이프라인에 대한 설명으로 틀린 것은 무엇입니까?

❶ 스트림을 연결해서 중간 처리와 최종 처리를 할 수 있다.
❷ 중간 처리 단계에서는 필터링, 매핑, 정렬, 그룹핑을 한다.
❸ 최종 처리 단계에서는 합계, 평균, 카운팅, 최대값, 최소값 등을 얻을 수 있다.
❹ 최종 처리가 없더라도 중간 처리를 할 수 있다.

4. 스트림 병렬 처리에 대한 설명으로 틀린 것은 무엇입니까?

❶ 전체 요소를 분할해서 처리한다.
❷ 내부적으로 포크조인 프레임워크를 이용한다.
❸ 병렬 처리는 순차적 처리보다 항상 빠른 처리를 한다.
❹ 내부적으로 스레드풀을 이용해서 스레드를 관리한다.

5. List에 저장되어 있는 String 요소에서 대소문자와 상관없이 'java'라는 단어가 포함된 문자열만 필터링해서 출력하려고 합니다. 빈칸에 알맞은 코드를 작성해보세요.

```
import java.util.Arrays;
import java.util.List;

public class Example {
  public static void main(String[] args) {
    List<String> list = Arrays.asList(
```

```
      "This is a java book",
      "Lambda Expressions",
      "Java8 supports lambda expressions"
    );
    list.stream()
      [          ]
  }
}
```

실행 결과

```
This is a java book
Java8 supports lambda expressions
```

6. List에 저장되어 있는 Member의 평균 나이를 출력하려고 합니다. 빈칸에 알맞은 코드를 작성해 보세요.

```
public class Member {
  private String name;
  private int age;

  public Member(String name, int age) {
    this.name = name;
    this.age = age;
  }

  public String getName() { return name; }
  public int getAge() { return age; }
}
```

```
import java.util.Arrays;
import java.util.List;

public class Example {
  public static void main(String[] args) {
    List<Member> list = Arrays.asList(
      new Member("홍길동", 30),
      new Member("신용권", 40),
```

```
      new Member("감자바", 26)
    );

    double avg = list.stream()
        
    System.out.println("평균 나이: " + avg);
  }
}
```

실행 결과

```
평균 나이: 32.0
```

7. List에 저장되어 있는 Member 중에서 직업이 '개발자'인 사람만 별도의 List에 수집하려고 합니다. 빈칸에 알맞은 코드를 작성해보세요.

```
public class Member {
  private String name;
  private String job;

  public Member(String name, String job) {
    this.name = name;
    this.job = job;
  }

  public String getName() { return name; }
  public String getJob() { return job; }
}
```

```
import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;

public class Example {
  public static void main(String[] args) {
    List<Member> list = Arrays.asList(
      new Member("홍길동", "개발자"),
```

확인문제

```
        new Member("김나리", "디자이너"),
        new Member("신용권", "개발자")
      );

      List<Member> developers = list.stream()
        [                                    ]

      developers
        .stream()
        .forEach(m -> System.out.println(m.getName()));
    }
}
```

실행 결과

```
홍길동
신용권
```

8. List에 저장되어 있는 Member를 직업별로 그룹핑해서 Map<String,List<Member>> 객체로 생성하려고 합니다. 키는 Member의 직업이고, 값은 해당 직업을 갖는 Member들을 저장하고 있는 List입니다. 실행 결과를 보고 빈칸에 알맞은 코드를 작성해보세요.

```
public class Member {
  private String name;
  private String job;

  public Member(String name, String job) {
    this.name = name;
    this.job = job;
  }

  public String getName() { return name; }
  public String getJob() { return job; }
  @Override
  public String toString() {
    return "{name:" + name + ", job:" + job + "}";
  }
}
```

```
import java.util.Arrays;
import java.util.List;
import java.util.Map;
import java.util.stream.Collectors;

public class Example {
  public static void main(String[] args) {
    List<Member> list = Arrays.asList(
      new Member("홍길동", "개발자"),
      new Member("김나리", "디자이너"),
      new Member("신용권", "개발자")
    );

    Map<String, List<Member>> groupingMap = list.stream()
      [                                                    ]

    System.out.println("[개발자]");
      [                                                    ]
      [                                                    ]
    System.out.println();

    System.out.println("[디자이너]");
      [                                                    ]
  }
}
```

실행 결과

```
[개발자]
{name:홍길동, job:개발자}
{name:신용권, job:개발자}

[디자이너]
{name:김나리, job:디자이너}
```

확인문제

Chapter 18

1. 입출력 스트림에 대한 설명 중 틀린 것은 무엇입니까?

❶ 하나의 스트림으로 입력과 출력이 동시에 가능하다.

❷ 프로그램을 기준으로 데이터가 들어오면 입력 스트림이다.

❸ 프로그램을 기준으로 데이터가 나가면 출력 스트림이다.

❹ 콘솔에 출력하거나 파일에 저장하려면 출력 스트림을 사용해야 한다.

2. InputStream과 Reader에 대한 설명으로 틀린 것은 무엇입니까?

❶ 이미지 데이터는 InputStream 또는 Reader로 모두 읽을 수 있다.

❷ Reader의 read() 메소드는 1문자를 읽는다.

❸ InputStream의 read() 메소드는 1바이트를 읽는다.

❹ InputStreamReader를 이용하면 InputStream을 Reader로 변환시킬 수 있다.

3. InputStream의 read(byte[] b) 메소드에 대한 설명으로 틀린 것은 무엇입니까?

❶ 메소드의 리턴값은 읽은 바이트 수이다.

❷ 매개값 b에는 읽은 데이터가 저장된다.

❸ 읽을 수 있는 바이트 수는 제한이 없다.

❹ 매개값 b에는 이전에 읽은 바이트가 남아 있을 수 있다.

4. 출력 스트림에서 데이터를 출력 후 flush() 메소드를 호출하는 이유는 무엇입니까?

❶ 출력 스트림의 버퍼에 있는 데이터를 모두 출력시키고 버퍼를 비운다.

❷ 출력 스트림을 메모리에서 제거한다.

❸ 출력 스트림의 버퍼에 있는 데이터를 모두 삭제한다.

❹ 출력 스트림을 닫는 역할을 한다.

5. 보조 스트림에 대한 설명으로 틀린 것은 무엇입니까?

❶ InputStreamReader는 InputStream을 Reader로 변환시키는 보조 스트림이다.

❷ BufferedInputStream은 데이터 읽기 성능을 향상시키는 보조 스트림이다.

❸ DataInputStream은 객체를 입출력하는 보조 스트림이다.

❹ PrintStream은 print(), println() 메소드를 제공하는 보조 스트림이다.

6. ObjectInputStream, ObjectOutputStream에 대한 설명으로 틀린 것은 무엇입니까?

❶ 객체를 직렬화해서 출력하고 역직렬화해서 복원시킨다.

❷ Serializable 인터페이스를 구현한 객체만 입출력할 수 있다.

❸ 클래스의 serialVersionUID는 입출력할 때 달라도 상관없다.

❹ transient 필드는 출력에서 제외된다.

7. 소스 파일을 읽고 실행 결과와 같이 행의 라인 번호를 추가시켜 출력하도록 밑줄과 빈 곳에 코드를 작성해보세요.

```
import java.io.BufferedReader;
import java.io.FileReader;

public class Example {
  public static void main(String[] args) throws Exception {
    String filePath = "C:/ThisIsJavaSecondEdition/workspace/thisisjava/src/ch02/
                      sec01/VariableUseExample.java";

    FileReader fr = _____;
    BufferedReader br = _____;

    int rowNumber = 0;
    String rowData;
    while(true) {
      
    }

    br.close(); fr.close();
  }
}
```

실행 결과

```
1: package ch02.sec01;
2:
3: public class VariableUseExample {
4:   public static void main(String[] args) {
```

확인문제

```
 5:     int hour = 3;
 ...
10:     System.out.println("총" + totalMinute + "분");
11:   }
12: }
```

8. PrintStream에 대한 설명으로 틀린 것은 무엇입니까?

❶ out 필드는 콘솔로 출력하는 PrintStream 타입이다.

❷ print(), println(), printf() 메소드를 제공한다.

❸ println() 메소드는 매개값의 타입에 따라 오버로딩되어 있다.

❹ PrintStream은 문자 기반 출력 스트림에 연결된다.

9. File과 Files 클래스에 대한 설명으로 틀린 것은 무엇입니까?

❶ File 객체는 파일이 실제로 존재하지 않아도 생성할 수 있다.

❷ File 객체는 파일 정보만 제공하고, 디렉토리 정보는 제공하지 않는다.

❸ Files 클래스는 정적 메소드로 구성되어 있기 때문에 객체를 만들 필요가 없다.

❹ File 객체는 파일의 크기를 제공하는 length() 메소드를 제공한다.

10. 실행하면 다음과 같이 원본 파일 경로와 복사 파일 경로를 입력받고 원본 파일을 복사하는 프로그램을 만들어보세요. (바이트 기반 스트림과 성능 향상 보조 스트림을 반드시 사용)

[조건]

❶ 원본 파일이 존재하지 않을 경우, "원본 파일이 존재하지 않습니다."를 출력할 것

❷ 복사 파일 경로에서 디렉토리가 존재하지 않으면 경로상의 모든 디렉토리를 자동 생성할 것

❸ 복사가 성공되었을 때 "복사가 성공되었습니다."를 출력할 것

실행 결과

```
원본 파일 경로: C:/Temp/dir1/photo1.jpg
복사 파일 경로: C:/Temp/dir2/photo2.jpg
원본 파일이 존재하지 않습니다.
```

실행 결과

```
원본 파일 경로: C:/Temp/dir1/photo1.jpg
복사 파일 경로: C:/Temp/dir2/photo2.jpg
복사가 성공되었습니다.
```

Chapter 19

1. 서버와 클라이언트에 대한 설명으로 틀린 것은 무엇입니까?

❶ 서비스를 제공하는 쪽이 서버이고, 서비스를 요청하는 쪽이 클라이언트이다.

❷ 클라이언트가 서버에 연결하기 위해서는 IP 주소만 있으면 된다.

❸ 포트(Port)는 여러 서버 중에 특정 서버와 연결하기 위해 필요한 정보다.

❹ 서버와 클라이언트는 양쪽 모두 포트가 배정되어야 한다.

2. TCP와 UDP에 대한 설명으로 틀린 것을 모두 선택하세요.

❶ TCP는 데이터 입출력에 앞서 연결 요청과 수락 과정이 필요하다.

❷ TCP는 여러 회선으로 데이터를 전달하므로, 데이터의 전달 순서가 달라질 수 있다.

❸ UDP는 연결 수락 과정이 없기 때문에 TCP보다 상대적으로 빠르다.

❹ UDP는 고정된 회선으로 데이터를 전달하기 때문에 전달 신뢰도가 높다.

3. 다음은 TCP 클라이언트가 서버로 연결 요청을 하고 서버는 연결을 수락하는 코드이다. 빈칸에 들어갈 코드를 작성하세요. (단, 클라이언트와 서버는 같은 컴퓨터에서 실행하고 있습니다.)

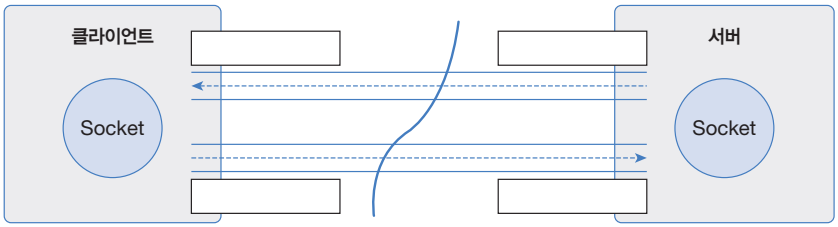
[클라이언트]

```
Socket socket = 
```

[서버]

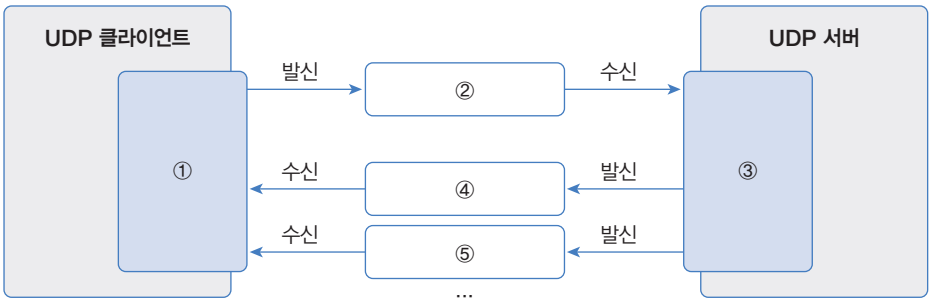
```
ServerSocket serverSocket = new ServerSocket(5001);
Socket socket = ;
```

4. TCP를 사용하는 클라이언트를 서버에 연결해 Socket으로 데이터 입출력을 하려고 합니다. 빈칸에 Socket을 통해 얻는 입출력 스트림의 타입을 적어보세요.



확인문제

5. UDP를 사용하는 클라이언트를 서버에서 사용되는 클래스 이름 및 데이터와 수신 · 발신할 때, 사용되는 클래스 이름을 ①~⑤ 빈칸에 작성해보세요.



6. 서버 측 DatagramSocket에 대한 설명으로 틀린 것은 무엇입니까?

❶ 서버에서는 고정된 Port 번호를 제공하고 생성해야 한다.

❷ receive() 메소드는 데이터를 수신할 때까지 블로킹된다.

❸ 클라이언트의 IP 주소와 Port 번호는 수신된 DatagramPacket에서 얻을 수 있다.

❹ 클라이언트로 DatagramPacket을 발신할 때 write() 메소드를 사용한다.

7. 상품 관리 프로그램을 TCP를 이용해서 개발하려고 합니다. 다음 내용에 맞게 서버와 클라이언트를 직접 개발해보세요.

[실행 내용]

1. 클라이언트를 실행하면 콘솔에서 다음과 같이 [상품 목록]과 메뉴가 출력된다.

```
[상품 목록]
-------------------------------------------------------------------
no    name                          price          stock
-------------------------------------------------------------------
-------------------------------------------------------------------
메뉴: 1.Create ¦ 2.Update ¦ 3.Delete ¦ 4.Exit
선택:
```

2. 선택에서 1을 입력하고 Enter 키를 누르면 다음과 같이 상품 생성을 위한 정보를 입력할 수 있다.

```
메뉴: 1.Create ¦ 2.Update ¦ 3.Delete ¦ 4.Exit
선택: 1
[상품 생성]
상품 이름: Television
상품 가격: 3000000
상품 재고: 10
```

3. 상품 이름, 상품 가격, 상품 재고까지 모두 입력하고 Enter 키를 누르면 다시 [상품 목록]으로 되돌아간다.

```
[상품 목록]
---------------------------------------------------------------------
no    name                          price          stock
---------------------------------------------------------------------
1     Television                      3000000       10

---------------------------------------------------------------------
메뉴: 1.Create ¦ 2.Update ¦ 3.Delete ¦ 4.Exit
선택:
```

4. 선택에서 2를 입력하고 Enter 키를 누르면 다음과 같이 상품 수정을 위한 정보를 입력할 수 있다.

```
메뉴: 1.Create ¦ 2.Update ¦ 3.Delete ¦ 4.Exit
선택: 2

[상품 수정]
상품 번호: 1
이름 변경: SmartTV
가격 변경: 3500000
재고 변경: 20
```

확인문제

5. 상품 번호, 이름 변경, 가격 변경, 재고 변경까지 모두 입력하고 Enter 키를 누르면 다시 [상품 목록]으로 되돌아간다.

```
[상품 목록]
-------------------------------------------------------------------
no    name                          price          stock
-------------------------------------------------------------------
1     SmartTV                       3500000        20

-------------------------------------------------------------------
메뉴: 1.Create ¦ 2.Update ¦ 3.Delete ¦ 4.Exit
선택:
```

6. 선택에서 3을 입력하고 Enter 키를 누르면 다음과 같이 상품 삭제를 위한 정보를 입력할 수 있다.

```
메뉴: 1.Create ¦ 2.Update ¦ 3.Delete ¦ 4.Exit
선택: 3

[상품 삭제]
상품 번호: 1
```

7. 삭제할 상품 번호를 입력하고 Enter 키를 누르면 다시 [상품 목록]으로 되돌아간다. 목록에서는 삭제된 상품이 보이지 않아야 한다.

```
[상품 목록]
-------------------------------------------------------------------
no    name                          price          stock
-------------------------------------------------------------------
-------------------------------------------------------------------
메뉴: 1.Create ¦ 2.Update ¦ 3.Delete ¦ 4.Exit
선택:
```

8. 마지막으로 선택에서 4를 입력하고 Enter 키를 누르면 클라이언트 프로그램이 종료된다.

[요약]

클라이언트 요청부터 서버 응답까지의 전 과정을 요약하면 다음과 같다.

❶ 선택에서 메뉴 번호를 입력한다.

❷ 필요한 정보를 키보드로 추가 입력받는다.

❸ 메뉴 번호와 입력된 데이터를 JSON 형식으로 만들고, 서버로 처리 요청을 한다.

❹ 서버는 요청 JSON을 해석하고 처리한다(동시 요청 처리를 위한 스레드풀 적용).

❺ 서버는 처리 결과를 JSON 형식으로 만들고 클라이언트로 응답을 보낸다.

❻ 클라이언트는 응답 JSON을 해석하고 status가 success일 경우 다시 목록과 메뉴를 보여준다.

[실행 조건]

1. 클라이언트가 서버로 보내는 요청 JSON은 다음과 같은 구조로 작성한다. menu는 입력된 메뉴 번호이며, data는 서버에서 요청을 처리하는 데 필요한 데이터이다.

```
{
  menu: 메뉴 번호,
  data: { … }
}
```

2. 서버가 클라이언트로 보내는 응답 JSON은 다음과 같은 구조로 작성한다. status는 처리 상태이므로 "success" 또는 "fail"로 작성하고, data는 클라이언트로 전달하고자 하는 데이터를 넣는다. 상품 목록을 보낼 경우 data는 배열 형태가 된다.

```
{
  result: "success" 또는 "fail",
  data: { … } 또는 [ … ]
}
```

확인문제

3. 상품 정보를 담을 때 사용하는 Product 클래스는 다음과 같이 설계한다. 그리고 서버는 상품을 List〈Product〉 컬렉션으로 메모리에 저장해서 관리한다.

```
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class Product {
  private int no;
  private String name;
  private int price;
  private int stock;
}
```

[작성할 파일]

이 문제를 위해 작성해야 할 소스 파일은 다음 3가지이다.

❶ 서버: ProductServer.java (ProductServer 클래스와 SocketClient 중첩 클래스 선언)

❷ 클라이언트: ProductClient.java (ProductClient 클래스 선언)

❸ 공통: Product.java (Product 클래스 선언)

Chapter 20

1. JDBC에 대한 설명으로 틀린 것은 무엇입니까?

❶ java.sql에서 제공하는 표준 라이브러리를 말한다.
❷ DBMS의 종류와 상관없이 사용할 수 있는 클래스와 인터페이스로 구성되어 있다.
❸ JDBC 인터페이스들을 구현한 것이 JDBC Driver이다.
❹ JDBC Driver는 DBMS의 종류와 상관없이 동일한 것을 사용할 수 있다.

2. JDBC가 DB와 연결할 때 필요한 정보가 아닌 것은 무엇입니까?

❶ DBMS가 설치된 컴퓨터의 IP 주소와 Port 번호가 필요하다.
❷ DBMS에 생성된 DB의 이름과 사용자 및 비밀번호가 필요하다.
❸ DB에 생성된 테이블 이름을 알아야 한다.
❹ DBMS별로 제공되는 JDBC Driver 클래스 이름을 알아야 한다.

3. JDBC로 SQL 실행 결과를 얻기 위한 코드 작성 순서는? ( ) ➔ ( ) ➔ ( ) ➔ ( )

❶ DriverManager부터 Connection을 얻는다.
❷ Class.forName() 메소드를 이용해서 JDBC Driver 클래스를 로딩한다.
❸ ResultSet에서 SQL 실행 결과를 얻는다.
❹ PreparedStatement를 얻고 SQL 문을 실행한다.

4. PreparedStatement에 대한 설명으로 틀린 것은 무엇입니까?

❶ 매개변수화된 SQL 문을 사용할 수 있다.
❷ INSERT, UPDATE, DELETE 문은 executeUpdate() 메소드로 실행한다.
❸ SELECT 문은 executeQuery() 메소드로 실행한다.
❹ 매개변수화된 SQL 문의 ? 순번은 0번부터 시작한다.

5. ResultSet에 대한 설명으로 틀린 것은 무엇입니까?

❶ ResultSet은 executeQuery의 리턴값이다.
❷ next() 메소드로 afterLast로 이동할 때 true를 리턴한다.
❸ ResultSet은 한 번에 하나의 행만 읽을 수 있다.
❹ ResultSet은 다음 행으로 커서를 이동할 때 next() 메소드를 사용한다.

확인문제

6. 프로시저와 함수를 실행하는 방법으로 틀린 것은 무엇입니까?

❶ CallableStatement를 이용한다.

❷ 프로시저 호출 문자열로 "{call 프로시저명(?, ?, …)}"을 사용한다.

❸ 함수 호출 문자열로 "{? = 함수명(?, ?, …)}"을 사용한다.

❹ 리턴값인 ?을 지정할 때에는 registerOutParameter() 메소드를 이용한다.

7. 트랜잭션에 대한 설명으로 틀린 것은 무엇입니까?

❶ 기능 처리의 최소 단위를 말한다.

❷ 커밋(commit)은 내부 작업을 모두 성공 처리한다.

❸ 롤백(rollback)은 내부 작업 중에서 성공한 작업까지 되돌린다.

❹ 트랜잭션을 코드로 제어하려면 setAutoCommit(false) 메소드를 먼저 호출해야 한다.

8. 20장 12절에서 구현한 게시판에서 다음 내용과 같이 새 사용자를 가입하는 기능을 추가해보세요.

[실행 내용]

1. 메인 메뉴에 '4.Join' 메뉴를 다음과 같이 추가한다.

```
메인 메뉴: 1.Create | 2.Read | 3.Clear | 4.Join | 5.Exit
메뉴 선택:
```

2. '4.Join' 메뉴를 선택했을 때 새 사용자 정보를 입력받도록 하고, 보조 메뉴를 다음과 같이 출력한다.

```
[새 사용자 입력]
아이디: java
이름: 김자바
비밀번호: 12345
나이: 26
이메일: java@mycompany.com
------------------------------------------------------------------------
보조 메뉴: 1.Ok | 2.Cancel
메뉴 선택: 1
```

3. User 클래스를 다음과 같이 선언하고, 입력된 새 사용자 정보를 User 객체에 저장한다.

```
import lombok.Data;

@Data
public class User {
  private String userId;
  private String userName;
  private String userPassword;
  private int userAge;
  private String userEmail;
}
```

4. 보조 메뉴에서 '1.Ok'를 선택하면 새 사용자는 데이터베이스 users 테이블에 저장된다. 새 사용자를 users 테이블에 저장하는 방법은 20.6절을 참고한다. 성공적으로 가입되면 다시 목록으로 되돌아온다.

5. 보조 메뉴에서 '2.Cancel'을 선택하면 새 사용자 정보를 무시하고 다시 목록으로 되돌아온다.

9. 8번 문제에서 만든 결과물에 다음과 같이 로그인 기능을 추가해보세요.

[실행 내용]

1. 메인 메뉴에 '5.Login' 메뉴를 다음과 같이 추가한다.

```
메인 메뉴: 1.Create | 2.Read | 3.Clear | 4.Join | 5.Login | 6.Exit
메뉴 선택:
```

2. '5.Login' 메뉴를 선택했을 때 로그인에 필요한 사용자 정보를 입력받도록 하고, 보조 메뉴를 다음과 같이 출력한다.

```
[로그인]
아이디: java
비밀번호: 12345
------------------------------------------------------------------------
보조 메뉴: 1.Ok | 2.Cancel
메뉴 선택: 1
```

확인문제

3. 보조 메뉴에서 '1.Ok'를 선택하면 입력된 아이디로 데이터베이스 users 테이블에서 비밀번호를 가져온다. 사용자 정보를 가져오는 방법은 20장 9절의 사용자 정보 읽기를 참고한다.

4. 입력된 비밀번호와 DB에서 가져온 비밀번호가 일치한다면 목록으로 돌아오는데, 로그인한 사용자 아이디를 [게시물 목록] 옆에 출력한다. 그리고 메인 메뉴에 다음과 같이 '4.Logout'이 나오도록 재구성한다.

```
[게시물 목록] 사용자: java
------------------------------------------------------------------------
no    writer      date            title
------------------------------------------------------------------------
3     winter      2022-01-27      봄의 정원
2     winter      2022-01-27      크리스마스
1     winter      2022-01-27      눈 오는 날

------------------------------------------------------------------------
메인 메뉴: 1.Create | 2.Read | 3.Clear | 4.Logout | 5.Exit
메뉴 선택:
```

5. 입력된 아이디 또는 비밀번호가 맞지 않을 경우에는 "아이디가 존재하지 않습니다." 또는 "비밀번호가 일치하지 않습니다."를 출력하고 [새 사용자 정보]를 다시 입력하도록 해준다.

```
아이디가 존재하지 않습니다.

[새 사용자 정보]
아이디: java
비밀번호: 12345
------------------------------------------------------------------------
보조 메뉴: 1.Ok | 2.Cancel
메뉴 선택: 1
```

6. 보조 메뉴에서 '2.Cancel'을 선택하면 로그인 정보를 무시하고 다시 목록으로 돌아온다.

7. 로그인한 상태에서 메인 메뉴의 '4.Logout'을 선택하면 [게시물 목록] 옆에 사용자 정보가 사라지도록 로그아웃 기능을 구현한다.

10. 9번 문제에서 만든 결과물에 다음과 같은 조건에 맞는 프로그램으로 수정해보세요.

[실행 내용]

1. 로그인한 상태에서 메인 메뉴의 '1.Create'를 선택했을 때 나오는 [새 게시물 입력] 중에서 작성자는 제거하는 대신 로그인한 사용자 아이디를 작성자로 사용한다.

2. 로그인하지 않은 상태에서 메인 메뉴의 '2.Read'를 선택하고, 게시물을 읽었다면 다음과 같은 보조 메뉴를 출력하지 않고 바로 목록으로 되돌아간다. 로그인한 상태에서도 로그인한 사용자 아이디와 작성자가 동일할 경우에만 보조 메뉴가 출력되도록 한다.

```
보조 메뉴: 1.Update ¦ 2.Delete ¦ 3.List
메뉴 선택:
```

3. 보조 메뉴의 '1.Update'를 선택했을 때 나오는 [수정 내용 입력] 중에서 작성자는 제거해서 수정할 수 없도록 한다.

정답

Chapter 01

1. ❹
2. ❷
3. ❹
4. ❸
5. ❸ ➔ ❶ ➔ ❷ ➔ ❹
6. ❹
7. ❶, ❷, ❸, ❹
8.

```
public class Example {
  public static void main(String[] args) {
    System.out.println("개발자가 되기 위한 필수 개발 언어 Java");
  }
}
```

Chapter 02

1. ❹
2. ❷, ❸, ❻, ❼
3.

byte	short char	int	long
		float	double
boolean			

4. int, double

   age, price

   10, 3.14
5. ❸
6. ❹
7. ❸

8. ❸

9. ❶

10. ❷

11. 13, 16

Chapter 03

1. 31

2. 가

3. `pencils / students`
   `pencils % students`

4. `value / 100 * 100`

5. ❶, ❷, ❸

6. `true, false`

7. `Double.isNaN(z)`

Chapter 04

1. ❷

2.

```
public class Example {
  public static void main(String[] args) {
    String grade = "B";

    int score = switch (grade) {
      case "A" -> 100;
      case "B" -> {
        int result = 100 - 20;
        yield result;
      }
      default -> 60;
    };

    System.out.println(score);
  }
}
```

정답

3.

```
public class Example {
  public static void main(String[] args) {
    int sum = 0;
    for(int i=1; i<10; i++) {
      if(i%3 == 0) {
        sum += i;
      }
    }
    System.out.println("3의 배수의 합: " + sum);
  }
}
```

4.

```
public class Example {
  public static void main(String[] args) {
    while(true) {
      int num1 = (int)(Math.random()*6) + 1;
      int num2 = (int)(Math.random()*6) + 1;
      System.out.println("(" + num1 + ", " + num2 + ")");
      if( (num1+num2) == 5) {
        break;
      }
    }
  }
}
```

5.

```
public class Example {
  public static void main(String[] args) {
    for(int x=1; x<=10; x++) {
      for(int y=1; y<=10; y++) {
        if( (4*x + 5*y) == 60) {
          System.out.println("(" + x + ", " + y + ")");
        }
      }
    }
  }
}
```

6.

```
public class Example {
  public static void main(String[] args) {
    for(int i=1; i<=5; i++) {
      for(int j=1; j<=i; j++) {
        System.out.print("*");
        if(j==i) {
          System.out.println();
        }
      }
    }
  }
}
```

7.

```
import java.util.Scanner;

public class Example {
  public static void main(String[] args) {
    boolean run = true;

    int balance = 0;

    Scanner scanner = new Scanner(System.in);

    while(run) {
      System.out.println("-----------------------------------");
      System.out.println("1.예금 ¦ 2.출금 ¦ 3.잔고 ¦ 4.종료");
      System.out.println("-----------------------------------");
      System.out.print("선택> ");

      int menuNum = scanner.nextInt();

      switch(menuNum) {
        case 1:
          System.out.print("예금액> ");
          balance += scanner.nextInt();
          break;
        case 2:
          System.out.print("출금액> ");
          balance -= scanner.nextInt();
```

정답

```
            break;
          case 3:
            System.out.print("잔고> ");
            System.out.println(balance);
            break;
          case 4:
            run = false;
            break;
        }

        System.out.println();
      }

      System.out.println("프로그램 종료");
    }
  }
```

Chapter 05

1. ❹
2. ❸
3. ❷
4. ❷
5. ❸
6. 3
   5
7.

```
public class Example {
  public static void main(String[] args) {
      int max = 0;
      int[] array = { 1, 5, 3, 8, 2 };

    for(int i=0; i<array.length; i++) {
      if(max<array[i]) {
        max = array[i];
      }
```

```
    }

      System.out.println("max: " + max);
  }
}
```

8.

```
public class Example {
  public static void main(String[] args) {
    int[][] array = {
        {95, 86},
        {83, 92, 96},
        {78, 83, 93, 87, 88}
    };

    int sum = 0;
    double avg = 0.0;

    int count = 0;
    for(int i=0; i<array.length; i++) {
      for(int j=0; j<array[i].length; j++) {
        sum += array[i][j];
        count++;
      }
    }
    avg = (double) sum / count;

    System.out.println("sum: " + sum);
    System.out.println("avg: " + avg);
  }
}
```

9.

```
import java.util.Scanner;

public class Example {
  public static void main(String[] args) {
    boolean run = true;

    int studentNum = 0;
    int[] scores = null;
```

정답

```
    Scanner scanner = new Scanner(System.in);

    while(run) {
      System.out.println("-----------------------------------------------------------");
      System.out.println("1.학생수 ¦ 2.점수입력 ¦ 3.점수리스트 ¦ 4.분석 ¦ 5.종료");
      System.out.println("-----------------------------------------------------------");
      System.out.print("선택> ");

      int selectNo = scanner.nextLine();

      if(selectNo == 1) {
        System.out.print("학생수> ");
        studentNum = scanner.nextLine();
        scores = new int[studentNum];
      } else if(selectNo == 2) {
        for(int i=0; i<scores.length; i++) {
          System.out.print("scores[" + i + "]> ");
          scores[i] = scanner.nextLine();
        }
      } else if(selectNo == 3) {
        for(int i=0; i<scores.length; i++) {
          System.out.println("scores[" + i + "]: " + scores[i]);
        }
      } else if(selectNo == 4) {
        int max = 0;
        int sum = 0;
        double avg = 0;
        for(int i=0; i<scores.length; i++) {
          max = (max<scores[i])? scores[i] : max;
          sum += scores[i];
        }
        avg = (double) sum / studentNum;
        System.out.println("최고 점수: " + max);
        System.out.println("평균 점수: " + avg);
      } else if(selectNo == 5) {
        run = false;
      }
    }

    System.out.println("프로그램 종료");
  }
}
```

Chapter 06

1. ❸
2. ❹
3. ❹
4. ❸
5. ❶
6. ❹
7. ❷
8. ❷
9. ❷
10. ❹
11. ❸
12. 필드, 생성자, 메소드
13.

```
public class Member {
  String name;
  String id;
  String password;
  int age;
}
```

14.

```
public class Member {
  String name;
  String id;
  String password;
  int age;

  Member(String name, String id) {
    this.name = name;
    this.id = id;
  }
}
```

정답

15.

```
public class MemberService {
  boolean login(String id, String password) {
    if(id.equals("hong") && password.equals("12345")) {
      return true;
    } else {
      return false;
    }
  }

  void logout(String id) {
    System.out.println(id + "님이 로그아웃 되었습니다.");
  }
}
```

16.

```
public class Printer {
  public void println(int value) {
    System.out.println(value);
  }

  public void println(boolean value) {
    System.out.println(value);
  }

  public void println(double value) {
    System.out.println(value);
  }

  public void println(String value) {
    System.out.println(value);
  }
}
```

17.

```
public class Printer {
  public static void println(int value) {
    System.out.println(value);
  }
```

정답

```
    public static void println(boolean value) {
      System.out.println(value);
    }

    public static void println(double value) {
      System.out.println(value);
    }

    public static void println(String value) {
      System.out.println(value);
    }
  }
```

18.

```
public class ShopService {
  private static ShopService singleton = new ShopService();

  private ShopService() {}

  static ShopService getInstance() {
    return singleton;
  }
}
```

19.

```
public class Account {
  public static final int MIN_BALANCE = 0;
  public static final int MAX_BALANCE = 1000000;
  private int balance;

  public int getBalance() {
    return balance;
  }

  public void setBalance(int balance) {
    if(balance<Account.MIN_BALANCE || balance>Account.MAX_BALANCE) {
      return;
    }
    this.balance = balance;
  }
}
```

정답

20.

[Account.java]

```
public class Account {
  private String ano;
  private String owner;
  private int balance;

  public Account(String ano, String owner, int balance) {
    this.ano = ano;
    this.owner = owner;
    this.balance = balance;
  }

  public String getAno() { return ano; }
  public void setAno(String ano) { this.ano = ano; }
  public String getOwner() { return owner; }
  public void setOwner(String owner) { this.owner = owner; }
  public int getBalance() { return balance; }
  public void setBalance(int balance) { this.balance = balance; }
}
```

[BankApplication.java]

```
package ch06.exam20;
import java.util.Scanner;

public class BankApplication {
  private static Account[] accountArray = new Account[100];
  private static Scanner scanner = new Scanner(System.in);

  public static void main(String[] args) {
    boolean run = true;
    while(run) {
      System.out.println("-----------------------------------------------------");
      System.out.println("1.계좌생성 | 2.계좌목록 | 3.예금 | 4.출금 | 5.종료");
      System.out.println("-----------------------------------------------------");
      System.out.print("선택> ");

      int selectNo = Integer.parseInt(scanner.nextLine());
      if(selectNo == 1) {
        createAccount();
```

정답

```
      } else if(selectNo == 2) {
        accountList();
      } else if(selectNo == 3) {
        deposit();
      } else if(selectNo == 4) {
        withdraw();
      } else if(selectNo == 5) {
        run = false;
      }
    }
    System.out.println("프로그램 종료");
  }

  //계좌생성하기
  private static void createAccount() {
    System.out.println("----------");
    System.out.println("계좌생성");
    System.out.println("----------");

    System.out.print("계좌번호: ");
    String ano = scanner.nextLine();

    System.out.print("계좌주: ");
    String owner = scanner.nextLine();

    System.out.print("초기입금액: ");
    int balance = Integer.parseInt(scanner.nextLine());

    Account newAccount = new Account(ano, owner, balance);
    for(int i=0; i<accountArray.length; i++) {
      if(accountArray[i] == null) {
        accountArray[i] = newAccount;
        System.out.println("결과: 계좌가 생성되었습니다.");
        break;
      }
    }
  }

  //계좌목록보기
  private static void accountList() {
    System.out.println("----------");
    System.out.println("계좌목록");
```

정답

```
    System.out.println("--------------");
    for(int i=0; i<accountArray.length; i++) {
      Account account = accountArray[i];
      if(account != null) {
        System.out.print(account.getAno());
        System.out.print("     ");
        System.out.print(account.getOwner());
        System.out.print("     ");
        System.out.print(account.getBalance());
        System.out.println();
      }
    }
  }

  //예금하기
  private static void deposit() {
    System.out.println("--------------");
    System.out.println("예금");
    System.out.println("--------------");
    System.out.print("계좌번호: ");
    String ano = scanner.nextLine();
    System.out.print("예금액: ");
    int money = Integer.parseInt(scanner.nextLine());
    Account account = findAccount(ano);
    if(account == null) {
      System.out.println("결과: 계좌가 없습니다.");
      return;
    }
    account.setBalance(account.getBalance() + money);
    System.out.println("결과: 예금이 성공되었습니다.");
  }

  //출금하기
  private static void withdraw() {
    System.out.println("--------------");
    System.out.println("출금");
    System.out.println("--------------");
    System.out.print("계좌번호: ");
    String ano = scanner.nextLine();
    System.out.print("출금액: ");
    int money = Integer.parseInt(scanner.nextLine());
```

```
    Account account = findAccount(ano);
    if(account == null) {
      System.out.println("결과: 계좌가 없습니다.");
      return;
    }
    account.setBalance(account.getBalance() - money);
    System.out.println("결과: 출금이 성공되었습니다.");
  }

  //Account 배열에서 ano와 동일한 Account 객체 찾기
  private static Account findAccount(String ano) {
    Account account = null;
    for(int i=0; i<accountArray.length; i++) {
      if(accountArray[i] != null) {
        String dbAno = accountArray[i].getAno();
        if(dbAno.equals(ano)) {
          account = accountArray[i];
          break;
        }
      }
    }
    return account;
  }
}
```

Chapter 07

1. ❶
2. ❷
3. ❶
4. ❹
5. ❷
6. 부모 생성자를 올바르게 호출하지 않았다.

   수정 ➜ Child.java에서 this.name=name; 라인을 지우고 super(name); 코드를 넣는다.
7.
```
Parent(String nation) call
Parent() call
Child(String name) call
Child() call
```

정답

8. 스노우 타이어가 굴러갑니다.
   스노우 타이어가 굴러갑니다.

9. ❷

10. work() 추상 메소드를 재정의하지 않았기 때문이다. 만약 재정의하지 않는다면 public abstract class Computer extends Machine {}과 같이 추상 클래스로 선언해야 한다.

11. super

12. a instanceof C c

Chapter 08

1. ❶

2. ❸

3. ❹

4. ❶, ❷, ❸, ❹

5. implements Remocon,

```
@Override
public void powerOn() {
  System.out.println("TV를 켰습니다.");
}
```

6.

```
public class Cat implements Soundable {
  @Override
  public String sound() {
    return "야옹";
  }
}
```

```
public class Dog implements Soundable {
  @Override
  public String sound() {
    return "멍멍";
  }
}
```

정답

7.

```
public interface DataAccessObject {
  public void select();
  public void insert();
  public void update();
  public void delete();
}
```

```
public class OracleDao implements DataAccessObject  {
  @Override
  public void select() {
    System.out.println("Oracle DB에서 검색");
  }

  @Override
  public void insert() {
    System.out.println("Oracle DB에 삽입");
  }

  @Override
  public void update() {
    System.out.println("Oracle DB를 수정");
  }

  @Override
  public void delete() {
    System.out.println("Oracle DB에서 삭제");
  }
}
```

```
public class MySqlDao implements DataAccessObject  {
  @Override
  public void select() {
    System.out.println("MySql DB에서 검색");
  }

  @Override
  public void insert() {
    System.out.println("MySql DB에 삽입");
  }

  @Override
  public void update() {
```

정답

```
    System.out.println("MySql DB를 수정");
  }

  @Override
  public void delete() {
    System.out.println("MySql DB에서 삭제");
  }
}
```

8. a instanceof C c

Chapter 09

1. ❹
2. ❸
3. ❸
4. myCar.new Tire( )
   new Car.Engine( )
5.

```
public class ActionExample {
  public static void main(String[] args) {
    Action action = new Action() {
      @Override
      public void work() {
        System.out.println("복사를 합니다.");
      }
    };
    action.work();
  }
}
```

6.

```
public class Anonymous {
  Vehicle field = new Vehicle() {
    @Override
    public void run() {
      System.out.println("자전거가 달립니다.");
    }
```

정답

```
  };

  void method1() {
    Vehicle localVar = new Vehicle() {
      @Override
      public void run() {
        System.out.println("승용차가 달립니다.");
      }
    };
    localVar.run();
  }

  void method2(Vehicle v) {
    v.run();
  }
}
```

```
public class AnonymousExample {
  public static void main(String[] args) {
    Anonymous anony = new Anonymous();
    //익명 객체 필드 사용
    anony.field.run();
    //익명 객체 로컬변수 사용
    anony.method1();
    //익명 객체 매개값 사용
    anony.method2(
      new Vehicle() {
        @Override
        public void run() {
          System.out.println("트럭이 달립니다.");
        }
      }
    );
  }
}
```

7. nickName은 final 특성을 갖기 때문에 startChat() 메소드에서 nickName = chatId와 같이 값을 변경할 수 없다. 따라서 String nickName = null;과 nickName = chatId;를 제거하고 대신 String nickName = chatId;를 넣어야 한다.

정답

Chapter 10

1. ❷
2. ❸
3. ❸
4. ❹
5. ❷

Chapter 11

1. ❹
2. ❸
3. ❹
4. ❷
5. ❸
6. 10
   숫자로 변환할 수 없음
   10
   인덱스를 초과했음
   10
7.

```
public class NotExistIDException extends Exception {
  public NotExistIDException() {}
  public NotExistIDException(String message) {
    super(message);
  }
}
```

```
public class WrongPasswordException extends Exception {
  public WrongPasswordException() {}
  public WrongPasswordException(String message) {
    super(message);
  }
}
```

```
public class LoginExample {
  public static void main(String[] args) {
    try {
      login("white", "12345");
    } catch(Exception e) {
      System.out.println(e.getMessage());
    }

    try {
      login("blue", "54321");
    } catch(Exception e) {
      System.out.println(e.getMessage());
    }
  }

  public static void login(String id, String password)
throws NotExistIDException, WrongPasswordException {
    //id가 blue가 아니라면 NotExistIDException을 발생시킴
    if(!id.equals("blue")) {
      throw new NotExistIDException("아이디가 존재하지 않습니다.");
    }

    //password가 12345가 아니라면 WrongPasswordException을 발생시킴
    if(!password.equals("12345")) {
      throw new WrongPasswordException("패스워드가 틀립니다.");
    }
  }
}
```

8.

```
try( FileWriter fw = new FileWriter("file.txt") ) {
  fw.write("Java");
} catch (IOException e) {
  e.printStackTrace();
}
```

Chapter 12

1. ❹
2. ❸

정답

3. ❹

4. hashCode, equals

5.

```
@Override
public boolean equals(Object obj) {
  if(obj instanceof Student) {
    Student student = (Student) obj;
    if(studentNum.equals(student.getStudentNum())) {
      return true;
    }
  }
  return false;
}

@Override
public int hashCode() {
  return studentNum.hashCode();
}
```

6.
```
@Override
public String toString() {
  return id + ": " + name;
}
```

7. ❸

8.

```
public class Example {
  public static void main(String[] args) {
    long start = System.nanoTime();

    int[] scores = new int[1000];
    for(int i=0; i<scores.length; i++) {
      scores[i] = i;
    }

    int sum = 0;
    for(int score : scores) {
      sum += score;
    }
```

```
    double avg = sum / scores.length;
    System.out.println(avg);

    long end = System.nanoTime();

    System.out.println((end-start) + " ns");
  }
}
```

9. new String(bytes, "UTF-8")

10.

```
public class StringBuilderExample {
  public static void main(String[] args) {
    String str = "";
    StringBuilder sb = new StringBuilder();
    for(int i=1; i<=100; i++) {
      sb.append(i);
    }
    str = sb.toString();
    System.out.println(str);
  }
}
```

11.

```
import java.util.StringTokenizer;

public class StringTokenizerExample {
  public static void main(String[] args) {
    String str = "아이디,이름,패스워드";

    StringTokenizer st = new StringTokenizer(str, ",");
    while(st.hasMoreTokens()) {
      String token = st.nextToken();
      System.out.println(token);
    }
  }
}
```

정답

12. 값의 범위가 −128~127이면 ==은 값을 비교하고 그 이외에는 번지를 비교하기 때문이다.

```
public class IntegerCompareExample {
  public static void main(String[] args) {
    Integer obj1 = 100;
    Integer obj2 = 100;
    Integer obj3 = 300;
    Integer obj4 = 300;

    System.out.println( obj1.equals(obj2) );
    System.out.println( obj3.equals(obj4) );
  }
}
```

13. ❹

14. ❹

15.

```
import java.time.LocalDateTime;
import java.time.temporal.ChronoUnit;

public class Example {
  public static void main(String[] args) {
    LocalDateTime startDateTime = LocalDateTime.now();

    LocalDateTime endDateTime = LocalDateTime.of(
        startDateTime.getYear(),  12, 31, 0, 0, 0);

    long remainDay = startDateTime.until(endDateTime, ChronoUnit.DAYS);
    System.out.println("남은 일자: " + remainDay);
  }
}
```

16.

```
import java.text.SimpleDateFormat;
import java.util.Date;

public class Example {
  public static void main(String[] args) {
    Date now = new Date();
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy년 MM월 dd일 E요일 HH시 mm분");
    System.out.println( sdf.format(now) );
  }
}
```

17. "[a-zA-Z][a-zA-Z0-9]{7,11}"
    Pattern.matches(regExp, id)

18. ❹

19. ❹

Chapter 13

1. ❹

2.

```
public class Container<T> {
  private T t;

  public T get() {
    return t;
  }

  public void set(T t) {
    this.t = t;
  }
}
```

3.

```
public class Container<K, V> {
  private K key;
  private V value;

  public K getKey() {
    return this.key;
  }

  public V getValue() {
    return this.value;
  }

  public void set(K key, V value) {
    this.key = key;
    this.value = value;
  }
}
```

정답

4.

```
public class Util {
  //how1
  public static <K, V> V getValue(Pair<K, V> p, K k) {
    if(p.getKey() == k) {
      return p.getValue();
    } else {
      return null;
    }
  }

  //how2
  /*public static <P extends Pair<K, V>, K, V> V getValue(P p, K k) {
    if(p.getKey() == k) {
      return p.getValue();
    } else {
      return null;
    }
  }*/
}
```

Chapter 14

1. ❹
2. new MusicRunnable()
   extends Thread
   implements Runnable
3. ❷
4. ❹
5. ❷
6. if(this.isInterrupted()) {
     break;
   }
7. ❸
8. thread.setDaemon(true);
9. ❹
10. ❹

Chapter 15

1. ❹
2. ❸
3. ❹
4. ❸
5. List〈Board〉 또는 ArrayList〈Board〉

   new ArrayList〈Board〉(); 또는 new ArrayList〈〉();
6. Map〈String, Integer〉 또는 HashMap〈String, Integer〉

   new HashMap〈String, Integer〉(); 또는 new HashMap〈〉();
7.

```
import java.util.ArrayList;
import java.util.List;

public class BoardDao {
  public List<Board> getBoardList() {
    List<Board> list = new ArrayList<Board>();
    list.add(new Board("제목1", "내용1"));
    list.add(new Board("제목2", "내용2"));
    list.add(new Board("제목3", "내용3"));
    return list;
  }
}
```

8.

```
public class Student {
  public int studentNum;
  public String name;

  public Student (int studentNum, String name) {
    this.studentNum = studentNum;
    this.name = name;
  }

  @Override
  public int hashCode() {
    return studentNum;
```

정답

```
  }

  @Override
  public boolean equals(Object obj) {
    if(!(obj instanceof Student)) return false;
    Student student = (Student) obj;
    if(studentNum != student.studentNum) return false;
    return true;
  }
}
```

9.

```
import java.util.HashMap;
import java.util.Map;
import java.util.Set;

public class MapExample {
  public static void main(String[] args) {
    Map<String,Integer> map = new HashMap<String,Integer>();
    map.put("blue", 96);
    map.put("hong", 86);
    map.put("white", 92);

    String name = null;
    int maxScore = 0;
    int totalScore = 0;

    Set<Map.Entry<String,Integer>> entrySet = map.entrySet();
    for(Map.Entry<String,Integer> entry : entrySet) {
      if(entry.getValue()>maxScore) {
        name = entry.getKey();
        maxScore = entry.getValue();
      }
      totalScore += entry.getValue();
    }

    int avgScore = totalScore / map.size();
    System.out.println("평균 점수: " + avgScore);

    System.out.println("최고 점수: " + maxScore);
    System.out.println("최고 점수를 받은 아이디: " + name);
  }
}
```

10. implements Comparable<Student>

```
@Override
public int compareTo(Student o) {
  if(score < o.score) return -1;
  else if(score == o.score) return 0;
  else return 1;
}
```

11. ❹

12. ❸

13. ❹

Chapter 16

1. ❹

2. ❹

3. ❷

4.
```
() -> {
  for(int i=0; i<3; i++) {
    System.out.println("작업 스레드가 실행됩니다.");
  }
}
```

5.
```
(() -> {System.out.println("Ok 버튼을 클릭했습니다.");})
(() -> {System.out.println("Cancel 버튼을 클릭했습니다.");})
```

6.
```
@FunctionalInterface
public interface Function {
  public double apply(double x, double y);
}
```

7.
```
i(a, b) -> {
  if(a>=b) {
    return a;
  } else {
    return b;
  }
}

(a, b) -> (a<=b)?a:b
```

정답

8.

```
public static double avg(Function<Student> function) {
  int sum = 0;
  for(Student student : students) {
    sum += function.apply(student);
  }
  double avg = (double) sum / students.length;
  return avg;
}
```

9. Student::getEnglishScore
   Student::getMathScore

Chapter 17

1. ❹
2. ❷
3. ❹
4. ❸
5. .filter(a -> a.toLowerCase().contains("java"))
   .forEach(a -> System.out.println(a));
6. .mapToInt(Member::getAge)
   .average()
   .getAsDouble();
7. .filter(m -> m.getJob().equals("개발자"))
   .collect(Collectors.toList());
8. .collect(Collectors.groupingBy(m -> m.getJob()));

   groupingMap.get("개발자").stream()
     .forEach(m -> System.out.println(m));

   groupingMap.get("디자이너").stream()
     .forEach(m -> System.out.println(m));

Chapter 18

1. ❶
2. ❶
3. ❸
4. ❶
5. ❸
6. ❸
7.
```
new FileReader(filePath);
new BufferedReader(fr);

rowData=br.readLine();
if(rowData == null) {
  break;
}
System.out.println(++rowNumber + ": " + rowData);
```
8. ❹
9. ❷
10.

```
import java.io.BufferedInputStream;
import java.io.BufferedOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.util.Scanner;

public class Example {
  public static void main(String[] args) {
    try {
      Scanner scanner = new Scanner(System.in);

      //경로 입력 받기
      System.out.print("원본 파일 경로: ");
      String originalFilePath = scanner.nextLine();

      System.out.print("복사 파일 경로: ");
      String targetFilePath = scanner.nextLine();
```

정답

```
        //원본 파일 존재 여부 확인
        File originalFile = new File(originalFilePath);
        if(!originalFile.exists()) {
          System.out.println("원본 파일이 존재하지 않습니다.");
          System.exit(0);
        }

        //복사 파일 경로상에 없는 모든 디렉토리 생성
        File targetFile = new File(targetFilePath);
        File parentFile = targetFile.getParentFile();
        if(!parentFile.exists()) {
          parentFile.mkdirs();
        }

        //입출력 스트림 얻기
        BufferedInputStream bis = new BufferedInputStream(
            new FileInputStream(originalFilePath));
        BufferedOutputStream bos = new BufferedOutputStream(
            new FileOutputStream(targetFilePath));

        //파일 데이터를 읽고 출력하기
        byte[] data = new byte[1024];
        int num = -1;
        while(true) {
          num = bis.read(data);
          if(num == -1) break;
          bos.write(data, 0, num);
        }
        System.out.println("복사가 성공되었습니다.");

        //입출력 스트림 닫기
        bis.close();
        bos.close();
      } catch(Exception e) {
        e.printStackTrace();
      }
    }
}
```

Chapter 19

1. ❷

2. ❷, ❹

3. `new Socket("localhost", 5001);`
   `serverSocket.accept()`

4. InputStream / OutputStream
   OutputStream / InputStream

5. ① Datagram Socket ② DatagramPacket ③ Datagram Socket ④ DatagramPacket
   ⑤ DatagramPacket

6. ❹

7.

[Product.java]

```
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok. NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class Product {
      private int no;
      private String name;
      private int price;
      private int stock;
}
```

[ProductServer.java]

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Iterator;
import java.util.List;
import java.util.Vector;
```

정답

```
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;

import org.json.JSONArray;
import org.json.JSONObject;

public class ProductServer {
	//필드
	private ServerSocket serverSocket;
	private ExecutorService threadPool;
	private List<Product> products;
	private int sequence;

	//메소드: 서버 시작
	public void start() throws IOException {
	  serverSocket = new ServerSocket(50001);
	  threadPool = Executors.newFixedThreadPool(100);
	  products = new Vector<Product>();

	  System.out.println( "[서버] 시작됨");

	  while(true) {
	    //연결 수락
	    Socket socket = serverSocket.accept();
	    //요청 처리용 SocketClient 생성
	    SocketClient sc = new SocketClient(socket);
	  }
	}

	//메소드: 서버 종료
	public void stop() {
	  try {
	    serverSocket.close();
	    threadPool.shutdownNow();
	    System.out.println( "[서버] 종료됨 ");
	  } catch (IOException e1) {}
	}

	//중첩 클래스: 요청 처리
	public class SocketClient {
	  //필드
	  private Socket socket;
```

정답

```
private DataInputStream dis;
private DataOutputStream dos;

//생성자
public SocketClient(Socket socket) {
  try {
    this.socket = socket;
    this.dis = new DataInputStream(socket.getInputStream());
    this.dos = new DataOutputStream(socket.getOutputStream());
    receive();
  } catch(IOException e) {
    close();
  }
}

//메소드: 요청 받기
public void receive() {
  threadPool.execute(() -> {
    try {
      while(true) {
       String receiveJson = dis.readUTF();

       JSONObject request = new JSONObject(receiveJson);
       int menu = request.getInt("menu");

       switch(menu) {
                case 0 -> list(request);
                case 1 -> create(request);
                case 2 -> update(request);
                case 3 -> delete(request);
       }
      }
    } catch(IOException e) {
      close();
    }
  });
}

public void list(JSONObject request) throws IOException {
  //응답 보내기
  JSONArray data = new JSONArray();
  for(Product p : products) {
```

정답

```
        JSONObject product = new JSONObject();
        product.put("no", p.getNo());
        product.put("name", p.getName());
        product.put("price", p.getPrice());
        product.put("stock", p.getStock());
        data.put(product);
      }

      JSONObject response = new JSONObject();
      response.put("status", "success");
      response.put("data", data);
      dos.writeUTF(response.toString());
      dos.flush();
    }

    public void create(JSONObject request) throws IOException {
      //요청 처리하기
      JSONObject data = request.getJSONObject("data");
      Product product = new Product();
      product.setNo(++sequence);
      product.setName(data.getString("name"));
      product.setPrice(data.getInt("price"));
      product.setStock(data.getInt("stock"));
      products.add(product);

      //응답 보내기
      JSONObject response = new JSONObject();
      response.put("status", "success");
      response.put("data", new JSONObject());
      dos.writeUTF(response.toString());
      dos.flush();
    }

    public void update(JSONObject request) throws IOException {
      //요청 처리하기
      JSONObject data = request.getJSONObject("data");
      int no = data.getInt("no");
      for(int i=0; i<products.size(); i++) {
        Product product = products.get(i);
        if(product.getNo() == no) {
          product.setName(data.getString("name"));
          product.setPrice(data.getInt("price"));
```

정답

```
            product.setStock(data.getInt("stock"));
          }
        }

        //응답 보내기
        JSONObject response = new JSONObject();
        response.put("status", "success");
        response.put("data", new JSONObject());
        dos.writeUTF(response.toString());
        dos.flush();
      }

      public void delete(JSONObject request) throws IOException {
        //요청 처리하기
        JSONObject data = request.getJSONObject("data");
        int no = data.getInt("no");
        Iterator<Product> iterator = products.iterator();
        while(iterator.hasNext()) {
          Product product = iterator.next();
          if(product.getNo() == no) {
            iterator.remove();
          }
        }

        //응답 보내기
        JSONObject response = new JSONObject();
        response.put("status", "success");
        response.put("data", new JSONObject());
        dos.writeUTF(response.toString());
        dos.flush();
      }

      //메소드: 연결 종료
      public void close() {
        try {
          socket.close();
        } catch(Exception e) {}
      }
    }


    //메소드: 메인
```

정답

```
        public static void main(String[] args) {
          ProductServer productServer = new ProductServer();
          try {
            productServer.start();
          } catch(IOException e) {
            System.out.println(e.getMessage());
            productServer.stop();
          }
        }
}
```

[ProductClient.java]

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.Socket;
import java.util.Scanner;

import org.json.JSONArray;
import org.json.JSONObject;

public class ProductClient {
        //필드
        private Socket socket;
        private DataInputStream dis;
        private DataOutputStream dos;
        private Scanner scanner;

        //메소드: 서버 연결
        public  void start() throws IOException {
          //서버 연결하기
          socket = new Socket("localhost", 50001);
          dis = new DataInputStream(socket.getInputStream());
          dos = new DataOutputStream(socket.getOutputStream());
          System.out.println("[클라이언트] 서버에 연결됨");

          scanner = new Scanner(System.in);

          //상품 목록 보여주기
          list();
        }
```

정답

```
//메소드: 클라이언트 종료
public void stop() {
  try {
    socket.close();
    scanner.close();
  } catch(Exception e) {}
  System.out.println("[클라이언트] 종료됨");
}

//메소드: 메뉴
public void menu() throws IOException {
  System.out.println();
  System.out.println("------------------------------------------------------------");
  System.out.println("메뉴: 1.Create | 2.Update | 3.Delete | 4.Exit");
  System.out.print("선택: ");
  String menuNo = scanner.nextLine();
  System.out.println();

  switch(menuNo) {
    case "1" -> create();
    case "2" -> update();
    case "3" -> delete();
    case "4" -> exit();
  }
}

//메소드: 상품 목록
public void list() throws IOException{
  //타이틀 및 컬럼명 출력
  System.out.println();
  System.out.println("[상품 목록]");
  System.out.println("------------------------------------------------------------");
  System.out.printf("%-6s%-30s%-15s%-10s\n", "no", "name", "price", "stock");
  System.out.println("------------------------------------------------------------");

  //상품 목록 요청하기
  JSONObject request = new JSONObject();
  request.put("menu", 0);
  request.put("data", new JSONObject());
  dos.writeUTF(request.toString());
  dos.flush();
```

정답

```
    //응답 받기
    JSONObject response = new JSONObject(dis.readUTF());
    if(response.getString("status").equals("success")) {
      //상품 목록 출력
      JSONArray data = response.getJSONArray("data");
      for(int i=0; i<data.length(); i++) {
        JSONObject product = data.getJSONObject(i);
        System.out.printf(
           "%-6d%-30s%-15d%-10d\n",
           product.getInt("no"),
           product.getString("name"),
           product.getInt("price"),
           product.getInt("stock")
        );
      }
    }

    //메뉴 출력
    menu();
  }

  //메소드: 상품 생성
  public void create() throws IOException {
    //상품 정보 입력
    System.out.println("[상품 생성]");
    Product product = new Product();
    System.out.print("상품 이름: ");
    product.setName(scanner.nextLine());
    System.out.print("상품 가격: ");
    product.setPrice(Integer.parseInt(scanner.nextLine()));
    System.out.print("상품 재고: ");
    product.setStock(Integer.parseInt(scanner.nextLine()));

    //상품 생성 요청하기
    JSONObject data = new JSONObject();
    data.put("name", product.getName());
    data.put("price", product.getPrice());
    data.put("stock", product.getStock());

    JSONObject request = new JSONObject();
    request.put("menu", 1);
```

정답

```
    request.put("data", data);

    dos.writeUTF(request.toString());
    dos.flush();

    //응답 받기
    JSONObject response = new JSONObject(dis.readUTF());
    if(response.getString("status").equals("success")) {
      list();
    }
  }

  //메소드: 상품 수정
  public void update() throws IOException {
    //상품 수정 내용 입력
    System.out.println("[상품 수정]");
    Product product = new Product();
    System.out.print("상품 번호: ");
    product.setNo(Integer.parseInt(scanner.nextLine()));
    System.out.print("이름 변경: ");
    product.setName(scanner.nextLine());
    System.out.print("가격 변경: ");
    product.setPrice(Integer.parseInt(scanner.nextLine()));
    System.out.print("재고 변경: ");
    product.setStock(Integer.parseInt(scanner.nextLine()));

    //상품 수정 요청하기
    JSONObject data = new JSONObject();
    data.put("no", product.getNo());
    data.put("name", product.getName());
    data.put("price", product.getPrice());
    data.put("stock", product.getStock());

    JSONObject request = new JSONObject();
    request.put("menu", 2);
    request.put("data", data);

    dos.writeUTF(request.toString());
    dos.flush();

    //응답 받기
    JSONObject response = new JSONObject(dis.readUTF());
```

정답

```
    if(response.getString("status").equals("success")) {
      list();
    }
  }

  //메소드: 상품 삭제
  public void delete() throws IOException {
    //상품 삭제 내용 입력
    System.out.println("[상품 삭제]");
    System.out.print("상품 번호: ");
    int no = Integer.parseInt(scanner.nextLine());

    //상품 수정 요청하기
    JSONObject data = new JSONObject();
    data.put("no", no);

    JSONObject request = new JSONObject();
    request.put("menu", 3);
    request.put("data", data);

    dos.writeUTF(request.toString());
    dos.flush();

    //응답 받기
    JSONObject response = new JSONObject(dis.readUTF());
    if(response.getString("status").equals("success")) {
      list();
    }
  }

  //메소드: 종료
  public void exit() {
    stop();
  }

  //메소드: 메인
  public static void main(String[] args) {
    ProductClient productClient = new ProductClient();
    try {
      productClient.start();
    } catch(IOException e) {
      System.out.println(e.getMessage());
```

```
            productClient.stop();
          }
        }
    }
```

Chapter 20

1. ❹
2. ❸
3. ❷ ➔ ❶ ➔ ❹ ➔ ❸
4. ❹
5. ❷
6. ❸
7. ❸
8.

[Board.java]

```
import java.util.Date;
import lombok.Data;

@Data
public class Board {
  private int bno;
  private String btitle;
  private String bcontent;
  private String bwriter;
  private Date bdate;
}
```

[User.java]

```
import lombok.Data;

@Data
public class User {
  private String userId;
  private String userName;
```

정답

```
  private String userPassword;
  private int userAge;
  private String userEmail;
}
```

[BoardExample.java]

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

public class BoardExample {
  //Field
  private Scanner scanner = new Scanner(System.in);
  private Connection conn;

  //Constructor
  public BoardExample() {
    try {
      //JDBC Driver 등록
      Class.forName("oracle.jdbc.OracleDriver");

      //연결하기
      conn = DriverManager.getConnection(
        "jdbc:oracle:thin:@localhost:1521/orcl",
        "java",
        "oracle"
      );
    } catch(Exception e) {
      e.printStackTrace();
      exit();
    }
  }

  //Method
  public void list() {
    //타이틀 및 컬럼명 출력
    System.out.println();
    System.out.println("[게시물 목록]");
```

정답

```
    System.out.println("------------------------------------------------------------------");
    System.out.printf("%-6s%-12s%-16s%-40s\n", "no", "writer", "date", "title");
    System.out.println("------------------------------------------------------------------");

    //boads 테이블에서 게시물 정보를 가져와서 출력하기
    try {
      String sql = "" +
        "SELECT bno, btitle, bcontent, bwriter, bdate " +
        "FROM boards " +
        "ORDER BY bno DESC";
      PreparedStatement pstmt = conn.prepareStatement(sql);
      ResultSet rs = pstmt.executeQuery();
      while(rs.next()) {
        Board board = new Board();
        board.setBno(rs.getInt("bno"));
        board.setBtitle(rs.getString("btitle"));
        board.setBcontent(rs.getString("bcontent"));
        board.setBwriter(rs.getString("bwriter"));
        board.setBdate(rs.getDate("bdate"));
        System.out.printf("%-6s%-12s%-16s%-40s \n",
            board.getBno(),
            board.getBwriter(),
            board.getBdate(),
            board.getBtitle());
      }
      rs.close();
      pstmt.close();
    } catch(SQLException e) {
      e.printStackTrace();
      exit();
    }

    //메인 메뉴 출력
    mainMenu();
  }

  public void mainMenu() {
    System.out.println();
    System.out.println("------------------------------------------------------------------");
    //>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
    System.out.println("메인 메뉴: 1.Create ¦ 2.Read ¦ 3.Clear ¦ 4.Join ¦ 5.Exit");
    System.out.print("메뉴 선택: ");
```

정답

```
    String menuNo = scanner.nextLine();
    System.out.println();

    switch(menuNo) {
      case "1" -> create();
      case "2" -> read();
      case "3" -> clear();
      case "4" -> join();
      case "5" -> exit();
    }
    //<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
  }

  public void create() {
    //입력 받기
    Board board = new Board();
    System.out.println("[새 게시물 입력]");
    System.out.print("제목: ");
    board.setBtitle(scanner.nextLine());
    System.out.print("내용: ");
    board.setBcontent(scanner.nextLine());
    System.out.print("작성자: ");
    board.setBwriter(scanner.nextLine());

    //보조 메뉴 출력
    System.out.println("------------------------------------------------------------------------");
    System.out.println("보조 메뉴: 1.Ok | 2.Cancel");
    System.out.print("메뉴 선택: ");
    String menuNo = scanner.nextLine();
    if(menuNo.equals("1")) {
      //boards 테이블에 게시물 정보 저장
      try {
        String sql = "" +
          "INSERT INTO boards (bno, btitle, bcontent, bwriter, bdate) " +
          "VALUES (SEQ_BNO.NEXTVAL, ?, ?, ?, SYSDATE)";
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, board.getBtitle());
        pstmt.setString(2, board.getBcontent());
        pstmt.setString(3, board.getBwriter());
        pstmt.executeUpdate();
        pstmt.close();
      } catch (Exception e) {
```

정답

```
        e.printStackTrace();
        exit();
      }
    }

    //게시물 목록 출력
    list();
  }

  public void read() {
    //입력 받기
    System.out.println("[게시물 읽기]");
    System.out.print("bno: ");
    int bno = Integer.parseInt(scanner.nextLine());

    //boards 테이블에서 해당 게시물을 가져와 출력
    try {
      String sql = "" +
        "SELECT bno, btitle, bcontent, bwriter, bdate " +
        "FROM boards " +
        "WHERE bno=?";
      PreparedStatement pstmt = conn.prepareStatement(sql);
      pstmt.setInt(1, bno);
      ResultSet rs = pstmt.executeQuery();
      if(rs.next()) {
        Board board = new Board();
        board.setBno(rs.getInt("bno"));
        board.setBtitle(rs.getString("btitle"));
        board.setBcontent(rs.getString("bcontent"));
        board.setBwriter(rs.getString("bwriter"));
        board.setBdate(rs.getDate("bdate"));
        System.out.println("############");
        System.out.println("번호: " + board.getBno());
        System.out.println("제목: " + board.getBtitle());
        System.out.println("내용: " + board.getBcontent());
        System.out.println("작성자: " + board.getBwriter());
        System.out.println("날짜: " + board.getBdate());
        //보조 메뉴 출력
        System.out.println("------------------------------------------------------------");
        System.out.println("보조 메뉴: 1.Update | 2.Delete | 3.List");
        System.out.print("메뉴 선택: ");
        String menuNo = scanner.nextLine();
```

정답

```
        System.out.println();

        if(menuNo.equals("1")) {
          update(board);
        } else if(menuNo.equals("2")) {
          delete(board);
        }
      }
      rs.close();
      pstmt.close();
    } catch (Exception e) {
      e.printStackTrace();
      exit();
    }

    //게시물 목록 출력
    list();
  }

  public void update(Board board) {
    //수정 내용 입력 받기
    System.out.println("[수정 내용 입력]");
    System.out.print("제목: ");
    board.setBtitle(scanner.nextLine());
    System.out.print("내용: ");
    board.setBcontent(scanner.nextLine());
    System.out.print("작성자: ");
    board.setBwriter(scanner.nextLine());

    //보조 메뉴 출력
    System.out.println("------------------------------------------------------------------");
    System.out.println("보조 메뉴: 1.Ok | 2.Cancel");
    System.out.print("메뉴 선택: ");
    String menuNo = scanner.nextLine();
    if(menuNo.equals("1")) {
      //boards 테이블에서 게시물 정보 수정
      try {
        String sql = "" +
          "UPDATE boards SET btitle=?, bcontent=?, bwriter=? " +
          "WHERE bno=?";
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, board.getBtitle());
```

정답

```
      pstmt.setString(2, board.getBcontent());
      pstmt.setString(3, board.getBwriter());
      pstmt.setInt(4, board.getBno());
      pstmt.executeUpdate();
      pstmt.close();
    } catch (Exception e) {
      e.printStackTrace();
      exit();
    }
  }

  //게시물 목록 출력
  list();
}

public void delete(Board board) {
  //boards 테이블에 게시물 정보 삭제
  try {
    String sql = "DELETE FROM boards WHERE bno=?";
    PreparedStatement pstmt = conn.prepareStatement(sql);
    pstmt.setInt(1, board.getBno());
    pstmt.executeUpdate();
    pstmt.close();
  } catch (Exception e) {
    e.printStackTrace();
    exit();
  }

  //게시물 목록 출력
  list();
}

public void clear() {
  System.out.println("[게시물 전체 삭제]");
  System.out.println("------------------------------------------------------------");
  System.out.println("보조 메뉴: 1.Ok | 2.Cancel");
  System.out.print("메뉴 선택: ");
  String menuNo = scanner.nextLine();
  if(menuNo.equals("1")) {
    //boards 테이블에 게시물 정보 전체 삭제
    try {
      String sql = "TRUNCATE TABLE boards";
```

정답

```
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.executeUpdate();
        pstmt.close();
      } catch (Exception e) {
        e.printStackTrace();
        exit();
      }
    }

    //게시물 목록 출력
    list();
  }

  //>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
  public void join() {
    //입력 받기
    User user = new User();
    System.out.println("[새 사용자 입력]");
    System.out.print("아이디: ");
    user.setUserId(scanner.nextLine());
    System.out.print("이름: ");
    user.setUserName(scanner.nextLine());
    System.out.print("비밀번호: ");
    user.setUserPassword(scanner.nextLine());
    System.out.print("나이: ");
    user.setUserAge(Integer.parseInt(scanner.nextLine()));
    System.out.print("이메일: ");
    user.setUserEmail(scanner.nextLine());

    //보조 메뉴 출력
    System.out.println("------------------------------------------------------------------");
    System.out.println("보조 메뉴: 1.Ok | 2.Cancel");
    System.out.print("메뉴 선택: ");
    String menuNo = scanner.nextLine();
    if(menuNo.equals("1")) {
      //boards 테이블에 게시물 정보 저장
      try {
        String sql = "" +
            "INSERT INTO users (userid, username, userpassword, userage, useremail) " +
            "VALUES (?, ?, ?, ?, ?)";
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, user.getUserId());
```

정답

```
          pstmt.setString(2, user.getUserName());
          pstmt.setString(3, user.getUserPassword());
          pstmt.setInt(4, user.getUserAge());
          pstmt.setString(5, user.getUserEmail());
          pstmt.executeUpdate();
          pstmt.close();
        } catch (Exception e) {
          e.printStackTrace();
          exit();
        }
      }

      //게시물 목록 출력
      list();
    }
    //<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

    public void exit() {
      if(conn != null) {
        try {
          conn.close();
        } catch (SQLException e) {
        }
      }
      System.out.println("** 게시판 종료 **");
      System.exit(0);
    }

    public static void main(String[] args) {
      BoardExample boardExample = new BoardExample();
      boardExample.list();
    }
  }
```

9.

[BoardExample.java]

```
  import java.sql.Connection;
  import java.sql.DriverManager;
  import java.sql.PreparedStatement;
  import java.sql.ResultSet;
```

정답

```
import java.sql.SQLException;
import java.util.Scanner;

public class BoardExample {
  //Field
  private Scanner scanner = new Scanner(System.in);
  private Connection conn;
  //>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
  private String loginId;
  //<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

  //Constructor
  public BoardExample() {
    try {
      //JDBC Driver 등록
      Class.forName("oracle.jdbc.OracleDriver");

      //연결하기
      conn = DriverManager.getConnection(
        "jdbc:oracle:thin:@localhost:1521/orcl",
        "java",
        "oracle"
      );
    } catch(Exception e) {
      e.printStackTrace();
      exit();
    }
  }

  //Method
  public void list() {
    //타이틀 및 컬럼명 출력
    System.out.println();
    //>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
    System.out.println("[게시물 목록] " + ((loginId != null)? ("사용자: " + loginId) : ""));
    //<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
    System.out.println("-------------------------------------------------------------------");
    System.out.printf("%-6s%-12s%-16s%-40s\n", "no", "writer", "date", "title");
    System.out.println("-------------------------------------------------------------------");

    //boards 테이블에서 게시물 정보를 가져와서 출력하기
    try {
```

정답

```
      String sql = "" +
        "SELECT bno, btitle, bcontent, bwriter, bdate " +
        "FROM boards " +
        "ORDER BY bno DESC";
      PreparedStatement pstmt = conn.prepareStatement(sql);
      ResultSet rs = pstmt.executeQuery();
      while(rs.next()) {
        Board board = new Board();
        board.setBno(rs.getInt("bno"));
        board.setBtitle(rs.getString("btitle"));
        board.setBcontent(rs.getString("bcontent"));
        board.setBwriter(rs.getString("bwriter"));
        board.setBdate(rs.getDate("bdate"));
        System.out.printf("%-6s%-12s%-16s%-40s \n",
            board.getBno(),
            board.getBwriter(),
            board.getBdate(),
            board.getBtitle());
      }
      rs.close();
      pstmt.close();
    } catch(SQLException e) {
      e.printStackTrace();
      exit();
    }

    //메인 메뉴 출력
    mainMenu();
  }

  public void mainMenu() {
    System.out.println();
    System.out.println("-----------------------------------------------------------------");
    //>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
    if(loginId == null) {
      System.out.println("메인 메뉴: 1.Create ¦ 2.Read ¦ 3.Clear ¦ 4.Join ¦ 5.Login ¦
          6.Exit");
      System.out.print("메뉴 선택: ");
      String menuNo = scanner.nextLine();
      System.out.println();

      switch(menuNo) {
```

정답

```
        case "1" -> create();
        case "2" -> read();
        case "3" -> clear();
        case "4" -> join();
        case "5" -> login();
        case "6" -> exit();
      }
    } else {
      System.out.println("메인 메뉴: 1.Create | 2.Read | 3.Clear | 4.Logout | 5.Exit");
      System.out.print("메뉴 선택: ");
      String menuNo = scanner.nextLine();
      System.out.println();

      switch(menuNo) {
        case "1" -> create();
        case "2" -> read();
        case "3" -> clear();
        case "4" -> logout();
        case "5" -> exit();
      }
    }
    //<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
  }

  public void create() {
    //입력 받기
    Board board = new Board();
    System.out.println("[새 게시물 입력]");
    System.out.print("제목: ");
    board.setBtitle(scanner.nextLine());
    System.out.print("내용: ");
    board.setBcontent(scanner.nextLine());
    System.out.print("작성자: ");
    board.setBwriter(scanner.nextLine());

    //보조 메뉴 출력
    System.out.println("-----------------------------------------------------------------------");
    System.out.println("보조 메뉴: 1.Ok | 2.Cancel");
    System.out.print("메뉴 선택: ");
    String menuNo = scanner.nextLine();
    if(menuNo.equals("1")) {
      //boards 테이블에 게시물 정보 저장
```

정답

```
      try {
        String sql = "" +
          "INSERT INTO boards (bno, btitle, bcontent, bwriter, bdate) " +
          "VALUES (SEQ_BNO.NEXTVAL, ?, ?, ?, SYSDATE)";
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, board.getBtitle());
        pstmt.setString(2, board.getBcontent());
        pstmt.setString(3, board.getBwriter());
        pstmt.executeUpdate();
        pstmt.close();
      } catch (Exception e) {
        e.printStackTrace();
        exit();
      }
    }

    //게시물 목록 출력
    list();
  }

  public void read() {
    //입력 받기
    System.out.println("[게시물 읽기]");
    System.out.print("bno: ");
    int bno = Integer.parseInt(scanner.nextLine());

    //boards 테이블에서 해당 게시물을 가져와 출력
    try {
      String sql = "" +
        "SELECT bno, btitle, bcontent, bwriter, bdate " +
        "FROM boards " +
        "WHERE bno=?";
      PreparedStatement pstmt = conn.prepareStatement(sql);
      pstmt.setInt(1, bno);
      ResultSet rs = pstmt.executeQuery();
      if(rs.next()) {
        Board board = new Board();
        board.setBno(rs.getInt("bno"));
        board.setBtitle(rs.getString("btitle"));
        board.setBcontent(rs.getString("bcontent"));
        board.setBwriter(rs.getString("bwriter"));
        board.setBdate(rs.getDate("bdate"));
```

정답

```
        System.out.println("############");
        System.out.println("번호: " + board.getBno());
        System.out.println("제목: " + board.getBtitle());
        System.out.println("내용: " + board.getBcontent());
        System.out.println("작성자: " + board.getBwriter());
        System.out.println("날짜: " + board.getBdate());
        //보조 메뉴 출력
        System.out.println("------------------------------------------------------------");
        System.out.println("보조 메뉴: 1.Update | 2.Delete | 3.List");
        System.out.print("메뉴 선택: ");
        String menuNo = scanner.nextLine();
        System.out.println();

        if(menuNo.equals("1")) {
          update(board);
        } else if(menuNo.equals("2")) {
          delete(board);
        }
      }
      rs.close();
      pstmt.close();
    } catch (Exception e) {
      e.printStackTrace();
      exit();
    }

    //게시물 목록 출력
    list();
  }

  public void update(Board board) {
    //수정 내용 입력 받기
    System.out.println("[수정 내용 입력]");
    System.out.print("제목: ");
    board.setBtitle(scanner.nextLine());
    System.out.print("내용: ");
    board.setBcontent(scanner.nextLine());
    System.out.print("작성자: ");
    board.setBwriter(scanner.nextLine());

    //보조 메뉴 출력
    System.out.println("------------------------------------------------------------");
```

정답

```
    System.out.println("보조 메뉴: 1.Ok | 2.Cancel");
    System.out.print("메뉴 선택: ");
    String menuNo = scanner.nextLine();
    if(menuNo.equals("1")) {
      //boards 테이블에서 게시물 정보 수정
      try {
        String sql = "" +
          "UPDATE boards SET btitle=?, bcontent=?, bwriter=? " +
          "WHERE bno=?";
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, board.getBtitle());
        pstmt.setString(2, board.getBcontent());
        pstmt.setString(3, board.getBwriter());
        pstmt.setInt(4, board.getBno());
        pstmt.executeUpdate();
        pstmt.close();
      } catch (Exception e) {
        e.printStackTrace();
        exit();
      }
    }

    //게시물 목록 출력
    list();
  }

  public void delete(Board board) {
    //boards 테이블에 게시물 정보 삭제
    try {
      String sql = "DELETE FROM boards WHERE bno=?";
      PreparedStatement pstmt = conn.prepareStatement(sql);
      pstmt.setInt(1, board.getBno());
      pstmt.executeUpdate();
      pstmt.close();
    } catch (Exception e) {
      e.printStackTrace();
      exit();
    }

    //게시물 목록 출력
    list();
  }
```

정답

```
public void clear() {
  System.out.println("[게시물 전체 삭제]");
  System.out.println("------------------------------------------------------------------");
  System.out.println("보조 메뉴: 1.Ok | 2.Cancel");
  System.out.print("메뉴 선택: ");
  String menuNo = scanner.nextLine();
  if(menuNo.equals("1")) {
    //boards 테이블에 게시물 정보 전체 삭제
    try {
      String sql = "TRUNCATE TABLE boards";
      PreparedStatement pstmt = conn.prepareStatement(sql);
      pstmt.executeUpdate();
      pstmt.close();
    } catch (Exception e) {
      e.printStackTrace();
      exit();
    }
  }

  //게시물 목록 출력
  list();
}

public void join() {
  //입력 받기
  User user = new User();
  System.out.println("[새 사용자 입력]");
  System.out.print("아이디: ");
  user.setUserId(scanner.nextLine());
  System.out.print("이름: ");
  user.setUserName(scanner.nextLine());
  System.out.print("비밀번호: ");
  user.setUserPassword(scanner.nextLine());
  System.out.print("나이: ");
  user.setUserAge(Integer.parseInt(scanner.nextLine()));
  System.out.print("이메일: ");
  user.setUserEmail(scanner.nextLine());

  //보조 메뉴 출력
  System.out.println("------------------------------------------------------------------");
  System.out.println("보조 메뉴: 1.Ok | 2.Cancel");
```

정답

```
    System.out.print("메뉴 선택: ");
    String menuNo = scanner.nextLine();
    if(menuNo.equals("1")) {
      //boards 테이블에 게시물 정보 저장
      try {
        String sql = "" +
            "INSERT INTO users (userid, username, userpassword, userage, useremail) " +
            "VALUES (?, ?, ?, ?, ?)";
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, user.getUserId());
        pstmt.setString(2, user.getUserName());
        pstmt.setString(3, user.getUserPassword());
        pstmt.setInt(4, user.getUserAge());
        pstmt.setString(5, user.getUserEmail());
        pstmt.executeUpdate();
        pstmt.close();
      } catch (Exception e) {
        e.printStackTrace();
        exit();
      }
    }

    //게시물 목록 출력
    list();
  }

  //>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
  public void login() {
    //입력 받기
    User user = new User();
    System.out.println("[로그인]");
    System.out.print("아이디: ");
    user.setUserId(scanner.nextLine());
    System.out.print("비밀번호: ");
    user.setUserPassword(scanner.nextLine());

    //보조 메뉴 출력
    System.out.println("-----------------------------------------------------------------");
    System.out.println("보조 메뉴: 1.Ok | 2.Cancel");
    System.out.print("메뉴 선택: ");
    String menuNo = scanner.nextLine();
    if(menuNo.equals("1")) {
```

정답

```
    //boards 테이블에 게시물 정보 저장
    try {
      String sql = "SELECT userpassword FROM users WHERE userid=?";
      PreparedStatement pstmt = conn.prepareStatement(sql);
      pstmt.setString(1, user.getUserId());
      ResultSet rs = pstmt.executeQuery();
      if(rs.next()) {
        String dbPassword = rs.getString("userpassword");
        if(dbPassword.equals(user.getUserPassword())) {
          loginId = user.getUserId();
        } else {
          System.out.println("비밀번호가 일치하지 않습니다.");
        }
      } else {
        System.out.println("아이디가 존재하지 않습니다.");
      }
      rs.close();
      pstmt.close();
    } catch (Exception e) {
      e.printStackTrace();
      exit();
    }
  }

  //게시물 목록 출력
  list();
}

public void logout() {
  //로그인 아이디 없애기
  loginId = null;

  //게시물 목록 출력
  list();
}
//<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

public void exit() {
  if(conn != null) {
    try {
      conn.close();
    } catch (SQLException e) {
```

```
        }
      }
      System.out.println("** 게시판 종료 **");
      System.exit(0);
    }

    public static void main(String[] args) {
      BoardExample boardExample = new BoardExample();
      boardExample.list();
    }
  }
```

10.

[BoardExample.java]

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

public class BoardExample {
  //Field
  private Scanner scanner = new Scanner(System.in);
  private Connection conn;
  private String loginId;

  //Constructor
  public BoardExample() {
    try {
      //JDBC Driver 등록
      Class.forName("oracle.jdbc.OracleDriver");

      //연결하기
      conn = DriverManager.getConnection(
        "jdbc:oracle:thin:@localhost:1521/orcl",
        "java",
        "oracle"
      );
    } catch(Exception e) {
```

정답

```
    e.printStackTrace();
    exit();
  }
}

//Method
public void list() {
  //타이틀 및 컬럼명 출력
  System.out.println();
  System.out.println("[게시물 목록] " + ((loginId != null)? ("사용자: " + loginId) : ""));
  System.out.println("------------------------------------------------------------------");
  System.out.printf("%-6s%-12s%-16s%-40s\n", "no", "writer", "date", "title");
  System.out.println("------------------------------------------------------------------");

  //boards 테이블에서 게시물 정보를 가져와 출력하기
  try {
    String sql = "" +
      "SELECT bno, btitle, bcontent, bwriter, bdate " +
      "FROM boards " +
      "ORDER BY bno DESC";
    PreparedStatement pstmt = conn.prepareStatement(sql);
    ResultSet rs = pstmt.executeQuery();
    while(rs.next()) {
      Board board = new Board();
      board.setBno(rs.getInt("bno"));
      board.setBtitle(rs.getString("btitle"));
      board.setBcontent(rs.getString("bcontent"));
      board.setBwriter(rs.getString("bwriter"));
      board.setBdate(rs.getDate("bdate"));
      System.out.printf("%-6s%-12s%-16s%-40s \n",
          board.getBno(),
          board.getBwriter(),
          board.getBdate(),
          board.getBtitle());
    }
    rs.close();
    pstmt.close();
  } catch(SQLException e) {
    e.printStackTrace();
    exit();
  }
```

정답

```
    //메인 메뉴 출력
    mainMenu();
  }

  public void mainMenu() {
    System.out.println();
    System.out.println("-----------------------------------------------------------------");
    if(loginId == null) {
      System.out.println("메인 메뉴: 1.Create | 2.Read | 3.Clear | 4.Join | 5.Login |
          6.Exit");
      System.out.print("메뉴 선택: ");
      String menuNo = scanner.nextLine();
      System.out.println();

      switch(menuNo) {
        case "1" -> create();
        case "2" -> read();
        case "3" -> clear();
        case "4" -> join();
        case "5" -> login();
        case "6" -> exit();
      }
    } else {
      System.out.println("메인 메뉴: 1.Create | 2.Read | 3.Clear | 4.Logout | 5.Exit");
      System.out.print("메뉴 선택: ");
      String menuNo = scanner.nextLine();
      System.out.println();

      switch(menuNo) {
        case "1" -> create();
        case "2" -> read();
        case "3" -> clear();
        case "4" -> logout();
        case "5" -> exit();
      }
    }
  }

  public void create() {
    //입력 받기
    Board board = new Board();
    System.out.println("[새 게시물 입력]");
```

정답

```
    System.out.print("제목: ");
    board.setBtitle(scanner.nextLine());
    System.out.print("내용: ");
    board.setBcontent(scanner.nextLine());
    //>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
    if(loginId == null) {
      System.out.print("작성자: ");
      board.setBwriter(scanner.nextLine());
    } else {
      board.setBwriter(loginId);
    }
    //<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

    //보조 메뉴 출력
    System.out.println("------------------------------------------------------------------");
    System.out.println("보조 메뉴: 1.Ok ¦ 2.Cancel");
    System.out.print("메뉴 선택: ");
    String menuNo = scanner.nextLine();
    if(menuNo.equals("1")) {
      //boards 테이블에 게시물 정보 저장
      try {
        String sql = "" +
          "INSERT INTO boards (bno, btitle, bcontent, bwriter, bdate) " +
          "VALUES (SEQ_BNO.NEXTVAL, ?, ?, ?, SYSDATE)";
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, board.getBtitle());
        pstmt.setString(2, board.getBcontent());
        pstmt.setString(3, board.getBwriter());
        pstmt.executeUpdate();
        pstmt.close();
      } catch (Exception e) {
        e.printStackTrace();
        exit();
      }
    }

    //게시물 목록 출력
    list();
  }

  public void read() {
    //입력 받기
```

정답

```
System.out.println("[게시물 읽기]");
System.out.print("bno: ");
int bno = Integer.parseInt(scanner.nextLine());

//boards 테이블에서 해당 게시물을 가져와 출력
try {
  String sql = "" +
    "SELECT bno, btitle, bcontent, bwriter, bdate " +
    "FROM boards " +
    "WHERE bno=?";
  PreparedStatement pstmt = conn.prepareStatement(sql);
  pstmt.setInt(1, bno);
  ResultSet rs = pstmt.executeQuery();
  if(rs.next()) {
    Board board = new Board();
    board.setBno(rs.getInt("bno"));
    board.setBtitle(rs.getString("btitle"));
    board.setBcontent(rs.getString("bcontent"));
    board.setBwriter(rs.getString("bwriter"));
    board.setBdate(rs.getDate("bdate"));
    System.out.println("############");
    System.out.println("번호: " + board.getBno());
    System.out.println("제목: " + board.getBtitle());
    System.out.println("내용: " + board.getBcontent());
    System.out.println("작성자: " + board.getBwriter());
    System.out.println("날짜: " + board.getBdate());
    //>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
    //보조 메뉴 출력
    if(loginId != null && loginId.equals(board.getBwriter())) {
      System.out.println("------------------------------------------------------------");
      System.out.println("보조 메뉴: 1.Update ¦ 2.Delete ¦ 3.List");
      System.out.print("메뉴 선택: ");
      String menuNo = scanner.nextLine();
      System.out.println();

      if(menuNo.equals("1")) {
        update(board);
      } else if(menuNo.equals("2")) {
        delete(board);
      }
    }
    //<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
```

정답

```
      }
      rs.close();
      pstmt.close();
    } catch (Exception e) {
      e.printStackTrace();
      exit();
    }

    //게시물 목록 출력
    list();
  }

  //>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
  public void update(Board board) {
    //수정 내용 입력 받기
    System.out.println("[수정 내용 입력]");
    System.out.print("제목: ");
    board.setBtitle(scanner.nextLine());
    System.out.print("내용: ");
    board.setBcontent(scanner.nextLine());

    //보조 메뉴 출력
    System.out.println("------------------------------------------------------------------");
    System.out.println("보조 메뉴: 1.Ok | 2.Cancel");
    System.out.print("메뉴 선택: ");
    String menuNo = scanner.nextLine();
    if(menuNo.equals("1")) {
      //boards 테이블에서 게시물 정보 수정
      try {
        String sql = "" +
          "UPDATE boards SET btitle=?, bcontent=? " +
          "WHERE bno=?";
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, board.getBtitle());
        pstmt.setString(2, board.getBcontent());
        pstmt.setInt(3, board.getBno());
        pstmt.executeUpdate();
        pstmt.close();
      } catch (Exception e) {
        e.printStackTrace();
        exit();
      }
```

```
  }

  //게시물 목록 출력
  list();
}
//<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

public void delete(Board board) {
  //boards 테이블에 게시물 정보 삭제
  try {
    String sql = "DELETE FROM boards WHERE bno=?";
    PreparedStatement pstmt = conn.prepareStatement(sql);
    pstmt.setInt(1, board.getBno());
    pstmt.executeUpdate();
    pstmt.close();
  } catch (Exception e) {
    e.printStackTrace();
    exit();
  }

  //게시물 목록 출력
  list();
}

public void clear() {
  System.out.println("[게시물 전체 삭제]");
  System.out.println("-------------------------------------------------------------------");
  System.out.println("보조 메뉴: 1.Ok | 2.Cancel");
  System.out.print("메뉴 선택: ");
  String menuNo = scanner.nextLine();
  if(menuNo.equals("1")) {
    //boards 테이블에 게시물 정보 전체 삭제
    try {
      String sql = "TRUNCATE TABLE boards";
      PreparedStatement pstmt = conn.prepareStatement(sql);
      pstmt.executeUpdate();
      pstmt.close();
    } catch (Exception e) {
      e.printStackTrace();
      exit();
    }
  }
```

정답

```
    //게시물 목록 출력
    list();
  }

  public void join() {
    //입력 받기
    User user = new User();
    System.out.println("[새 사용자 입력]");
    System.out.print("아이디: ");
    user.setUserId(scanner.nextLine());
    System.out.print("이름: ");
    user.setUserName(scanner.nextLine());
    System.out.print("비밀번호: ");
    user.setUserPassword(scanner.nextLine());
    System.out.print("나이: ");
    user.setUserAge(Integer.parseInt(scanner.nextLine()));
    System.out.print("이메일: ");
    user.setUserEmail(scanner.nextLine());

    //보조 메뉴 출력
    System.out.println("-----------------------------------------------------------------");
    System.out.println("보조 메뉴: 1.Ok | 2.Cancel");
    System.out.print("메뉴 선택: ");
    String menuNo = scanner.nextLine();
    if(menuNo.equals("1")) {
      //boards 테이블에 게시물 정보 저장
      try {
        String sql = "" +
            "INSERT INTO users (userid, username, userpassword, userage, useremail) " +
            "VALUES (?, ?, ?, ?, ?)";
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, user.getUserId());
        pstmt.setString(2, user.getUserName());
        pstmt.setString(3, user.getUserPassword());
        pstmt.setInt(4, user.getUserAge());
        pstmt.setString(5, user.getUserEmail());
        pstmt.executeUpdate();
        pstmt.close();
      } catch (Exception e) {
        e.printStackTrace();
        exit();
```

정답

```
      }
    }

    //게시물 목록 출력
    list();
  }

  public void login() {
    //입력 받기
    User user = new User();
    System.out.println("[로그인]");
    System.out.print("아이디: ");
    user.setUserId(scanner.nextLine());
    System.out.print("비밀번호: ");
    user.setUserPassword(scanner.nextLine());

    //보조 메뉴 출력
    System.out.println("------------------------------------------------------------------");
    System.out.println("보조 메뉴: 1.Ok | 2.Cancel");
    System.out.print("메뉴 선택: ");
    String menuNo = scanner.nextLine();
    if(menuNo.equals("1")) {
      //boards 테이블에 게시물 정보 저장
      try {
        String sql = "SELECT userpassword FROM users WHERE userid=?";
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, user.getUserId());
        ResultSet rs = pstmt.executeQuery();
        if(rs.next()) {
          String dbPassword = rs.getString("userpassword");
          if(dbPassword.equals(user.getUserPassword())) {
            loginId = user.getUserId();
          } else {
            System.out.println("비밀번호가 일치하지 않습니다.");
          }
        } else {
          System.out.println("아이디가 존재하지 않습니다.");
        }
        rs.close();
        pstmt.close();
      } catch (Exception e) {
        e.printStackTrace();
```

정답

```
          exit();
        }
      }

      //게시물 목록 출력
      list();
    }

    public void logout() {
      //로그인 아이디 없애기
      loginId = null;

      //게시물 목록 출력
      list();
    }

    public void exit() {
      if(conn != null) {
        try {
          conn.close();
        } catch (SQLException e) {
        }
      }
      System.out.println("** 게시판 종료 **");
      System.exit(0);
    }

    public static void main(String[] args) {
      BoardExample boardExample = new BoardExample();
      boardExample.list();
    }
}
```