**1장 연습문제 해답**

# 이론문제

1. 자바는 객체지향 개념이 실제 프로그램 개발에 정착되는데 많은 기여를 하였으며 오늘날 웹 기반 시스템 발전에도 큰 영향을 미쳤다.

2. 자바는 썬에서 개발한 객체지향 언어이다. 가정용 단말기에 적용하려는 목적으로 1992년에 만든 ‘오크(OAK)’ 언어에서 비롯되었다. 오크는 상품화되지 못했지만, 인터넷이 발전하면서 인터넷 친화적인 개발 언어로 방향을 전환하여 1995년 ‘자 바’라는 이름으로 새롭게 소개되었다.

3. 서블릿(Servlet), JSP(Java Server Pages)

4. 이클립스(Eclipse) IDE

5. 자바 가상머신, 가상머신, Java Virtual Machine(JVM)

6.

JDK : Java Development Kit

JRE : Java Runtime Environment

API : Application Programming Interface

7. 잘못된 설명 : 2,3,4,5

**2장 연습문제 해답**

# 이론문제

1. 바이트코드 혹은 클래스 파일 , 자바 가상머신

2. 자바의 대표적인 특징인 가상머신 때문으로 이를 통해 특정 운영체제나 하드웨어에 종속되지 않고 동일한 소스를 수정없이 실행 할 수 있다.

3. 프로젝트 생성.

# 실습문제

1.

2.

- HelloExam.java

- (b)

- HelloExam.class

3.

|  |
| --- |
| public class HelloExam {  public static void main(String[] args) {  System.out.println(“Hello!!”);  System.out.println(“Hello!!”);  System.out.println(“Hello!!”);  }  } |

**3장 연습문제 해답**

# 이론문제

1. 변수, 자료형, 변수

2.

(1) int x = 10;

(2) double y = 1.5; (세미콜론 누락)

(3) float f1 = 21.34f;

(4) int a = 1, b = 2, c = 3;

(5) boolean b = true; (소문자, 대문자X)

3.

(1) 변수 이름의 길이는 32/64 비트 운영체제 여부에 상관없다.

(2) 변수이름에 연산자 기호를 사용할 수 없다.

(3) 변수명의 첫글자는 숫자로 시작할 수 없다.

(4) 맞게 설명함.

(5) 변수명에 공백을 사용할 수 없으며 필요시 “\_”, Camel Case 를 활용한다.

(6) 맞게 설명함.

4. byte, short, int, long, char

5. byte c = a + b; -> int c = a + b; 가 되어야 함. 정수형 사칙연산의 결과는 int형이 기본임.

6. 123123123123 은 long 형 범위에 포함되는 숫자이나 그 자체로는 정수형의 기본인 int 형으로 인식이 되어 int 범위를 초과하기 때문에 오류가 발생된다. 123123123123L 혹은 “l”(소문자 L) 을 숫자 뒤에 붙여 주어야 한다.

7. float , double

8.

(1) float f1 = 28.8f;

(2) int max = 200;

(3) speed += 100;

(4) char ch = ‘A’;

(5) boolean sel = false;

9.

|  |
| --- |
| class TestClass {  int num1; -> 인스턴스 변수  static num2; -> 클래스 변수  void int sum(int num1 -> 파라미터) {  int result -> 지역변수 = this.num1 + num1;  return result;  }  } |

10.

(1) 15

(2) 10

(3) 18

(4) true

(5) 13

(6) 1

11. 10000.0 , num2 는 연산 결과를 double형으로 처리 했기 때문에 소숫점이 표시됨.

12. 연산결과값이 int 범위를 초과해 오버플로우가 발생하므로 부호부가 변경되어 음수 값이 출력됨.

13.

int a = 10 -> 0000 1010

int b = 15 -> 0000 1111

int c = a ^ b -> 0000 0101

14. 수정된 코드

|  |
| --- |
| for(int i = 0; i < 10; i++)  int k = i + 10;  System.out.println(k);  기본적으로 for 문 사용에 대한 잘못된 부분을 찾는 것으로 위와 같이 for 문의 잘못된 부분만 수정하면됨. 다만 for 문 안에서 k 를 매번 초기화 하는 것은 바람직 하지 않기 때문에 for 블럭 앞에서 미리 선언해 주는 것이 좋음. |

15.

(1) sel

(2) case

(3) break

(4) default

16.

(1) int[] num = new int[10];

(2) 이상 없음

(3) System.out.println(num[0,1]); // 배열 사용에서의 잘못된 부분을 찾는것이므로 여기서는 num[][] 이라는 2차원 배열이 선언되어 있다고 가정해야 한다.

(4) String[] adata = {“aaa”,”bbb”,”900”,”1000”};

(5) 이상 없음

**4장 연습문제 해답**

# 이론문제

1. 속성 , 행위(동작)

2. 아래 내용을 참고해서 작성

(1)속성이 같은 객체들을 대표할 수 있는 대상을 클래스(Class)라고 한다. 예를 들어 소나무는 클래스가 되고, 주 변에 보이는 각 소나무들은 모두 소나무 클래스의 인스턴스(객체)가 되는 것이다.

(2) 클래스는 객체를 정의하는 틀이며, 필드(속성)와 메서드(행위)로 구성한다. 클래스는 추상화를 이용하여 슈퍼 클래스(상위 클래스, 부모 클래스)와 서브 클래스(하위 클래스, 자식 클래스)로 구분한다.

(3) 인스턴스는 클래스에서 생성한 객체로, 고유한 상태가 있다. 동일한 클래스에서 생성된 객체라 할지라도 필드 값과메서드의내용은다를수있다.

3. 인스턴스(객체)에 비유. 만들어진 붕어빵은 굽기정도, 따뜻함, 팥양, 바삭함, 만든시간 등의 속성에서 차이가 발생.

4. 추상적, 구체적, 상속, 슈퍼클래스, 서브클래스

5. 캡슐화

6. 다형성

7. 메서드 오버라이딩

8. class, extends, implements

9. 현재 클래스를 상속받는 클래스를 생성하지 못하도록 한다.

10. 현재 클래스 내부

11.

- 클래스 변수는 static 으로 선언된 변수로 인스턴스 생성 없이 클래스이름.변수명 으로 접근 가능하며 모든 클래스 인스턴스에서 같은 값을 공유 한다.

- 인스턴스 변수는 인스턴스 생성시 초기화 되는 변수로 인스턴스이름(참조변수).변수명으로 접근 가능 하다.

12. Product p = new Product();

13. 생성자의 이름은 클래스명과 같아야 하며 void를 포함해 리턴값을 명시하지 않는다.

14.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18 | final class ClassA { -> final x  public void prtData() {  ....,  }  }  public class MyClass extend ClassA { -> extends  Int num1; -> int or Integer  public void MyClass() { -> void X  num1 = 10;  }  public void prtNum1() {  System.out.println(num1);  }  public static void main(String args) { -> String[]  MyClass mc = MyClass(); -> new MyClass()  prtNum1(); -> mc.prtNum1();  }  } |

* 출제 의도는 클래스 상속 구문에 대한 이해를 포함하고 있기 때문에 여기서는 위와 같이 답해야 함.
* 단순히 코드만 정상적으로 동작하게 하는 경우 06라인 extend ClassA 를 빼도 됨.

15. 인쇄오류 -> 11번과 동일문제임.

16.

(1) 40

(2) 40

(3) 30

(4) 20

17. 객체지향 개념을 실제 프로그램 개발에 쉽게 적용하고 유연한 설계를 지원하기 위해서이며 현재 구체화 되지 않은 내용에 대해 서브 클래스에서 구체화 할 수 있는 구조를 제공한다. 이를 통해 규격에 따른 프로그램 개발이 가능 함.

18. 틀린문항

(2) 추상 클래스는 일반 클래스와 같이 멤버와 메서드를 포함할 수 있음.

(3) 추상 클래스는 인스턴스 생성이 불가능 하고 반드시 상속을 통해 서브 클래스 정의후 메서드 오버라이딩으로 추상메서드를 구현해야 함.

오탈자

* 총 6개 항목이 있는데 3개씩 구분해서 각각 추상클래스, 인터페이스로 구분하고 항목마다 번호 필요.
* 4번항목에서 public, static, final -> public static final 로 수정
* 다음 인쇄때 반영 요망.

19. implements

20. 메서드 오버로딩, 메서드 오버라이딩

**5장 연습문제 해답**

# 이론문제

1.

|  |
| --- |
| public class MyExam {  String msg = “MyExam Message !!”;  public static void main(String[] args) {  MyExam app = new MyExam();  System.out.println(app.msg);  }  } |

2.

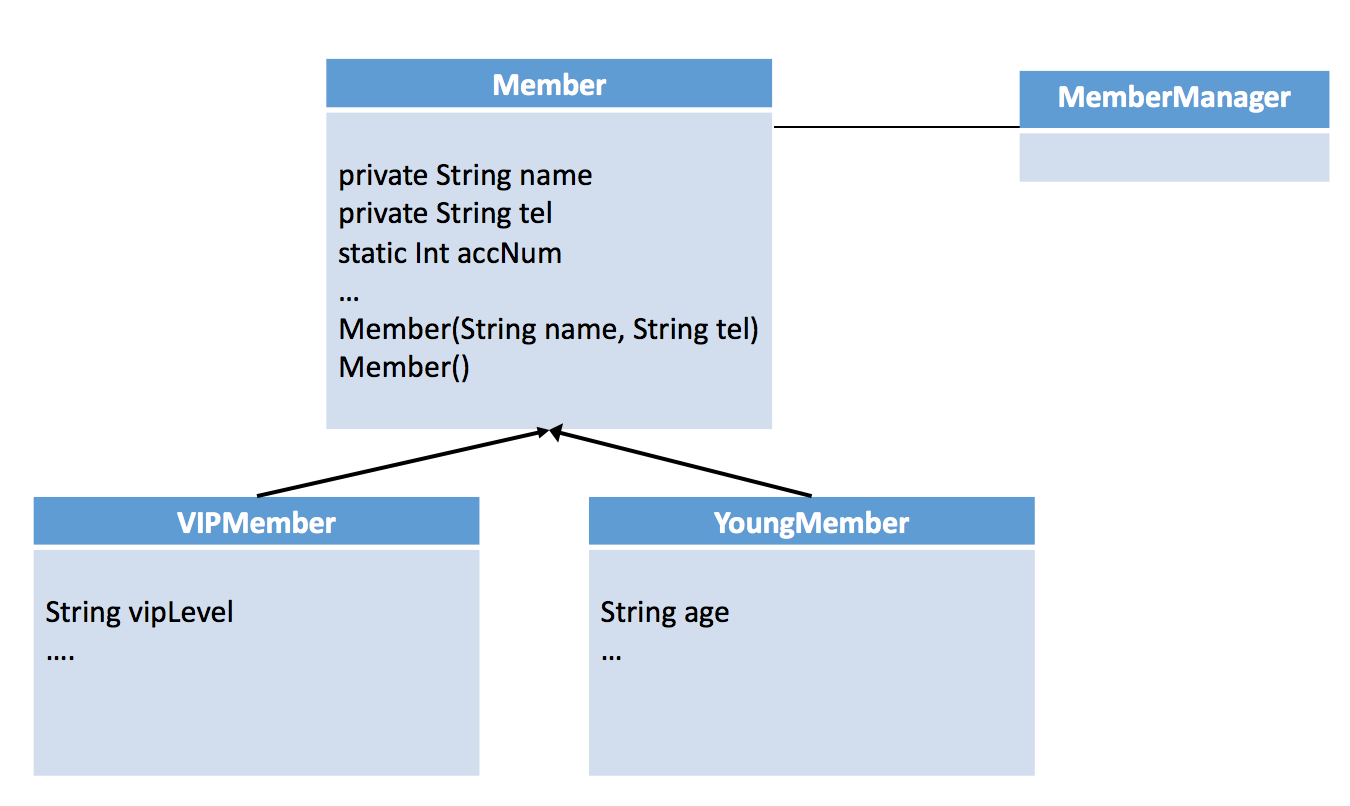
|  |
| --- |
| public class MyExam {  String msg;  public MyExam(String m) {  msg = m;  }  public MyExam(String m1, String m2) {  msg = m1+m2;  }  public String print() {  return msg;  }  public static void main(String[] args) {  MyExam me1 = new MyExam(“MyExam Message !!”);  MyExam me2 = new MyExam(“MyExam”, “Message !!”);  System.out.println(me1.print() + “ / ” +me2.print());  }  } |

3. 대략 다음과 같은 부분에 중점을 두고 클래스 다이어그램이 나오면 됨.

- Member 클래스 생성자 지정

- Member 클래스를 상속하는 VIP, Young 클래스 정의

- 회원번호 공유를 위해 static 으로 accNum 선언



**6장 연습문제 해답**

# 이론문제

1. Application Programming Interface, 라이브러리

2.

String str = new String(“Hello”);

3. 자바에서 문자열 “+” 연산시 매번 새로운 문자열 객체를 생성 하므로 성능상에 문제가 발생할 수 있다. 따라서 반복되는 문자열 결합에는 StringBuffer 클래스를 사용하는 것이 좋다.

4. 랩퍼클래스, Byte

5. 일관된 객체지향 프로그램 구조를 유지하려는 이유와 각 원시자료형에 대한 부가 기능(예를들어 변환등) 필요시 보다 효과적인 라이브러리 지원을 위해.

6. 랩퍼, 박싱, 언박싱, 오토 박싱/언박싱

7.

out 은 System 클래스에 static 으로 선언된 클래스 변수로 PrintStream 타입으로 콘솔과 연결된 출력 스트림이다. 그리고 println()은 PrintStream 클래스에서 제공하는 메서드 중 하나이기 때문에 System.out.println() 과 같이 사용해야 한다.

8. Calendar cal = Calendar.getInstance();

9.

(1){0,date}.{0,time}에 {1}과 {2}가

(2){new Date(),”홍길동”,”아무개”}

(3)(4) MessageFormat

**7장 연습문제 해답**

# 이론문제

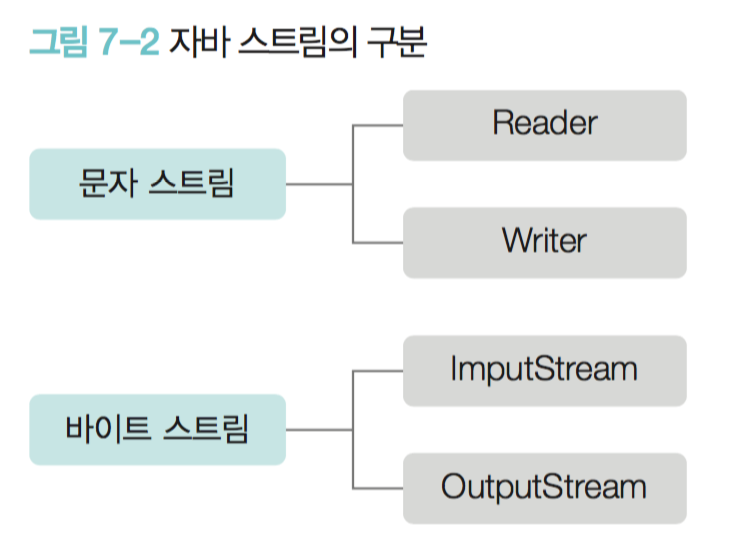
1. 교재 262페이지 내용을 참고해서 작성

2.

(2) 스트림은 단방향으로 입력과 출력이 모두 필요한 경우 입력과 출력 별도로 스트림 생성이 필요.

(3) 스트림 사용이 끝날때 까지 기다려야 하는 경우도 있지만 그렇지 않은 경우 스레드나 NIO 의 넌블로킹 스트림을 사용해 문제 해결이 가능하다.

3.



4.

(1)문자

(2)바이트

(3) Reader (4) Writer

(5) InputStream (6)OutputStream

5. 입력에 대한 제한이 없기 때문에 문자열이나 1자리 이상의 숫자를 입력할 수 있으나 스트림에서는 바이트스트림인 InputStream 을 사용하는 관계로 1byte 만 처리 되며 입력값을 int 형으로 처리하기 때문에 첫자리 문자의 아스키 코드값이 출력되게 된다.

6. 루트(root), 경로(path)

7. 프로토콜

8. 교재 283 ~ 284, 그림 7-7 참고

**8장 연습문제 해답**

# 이론문제

1. Graphical User Interface, 컴포넌트, AWT, Swing

2. BorderLayout



3. CardLayout

4. 이벤트, 키보드, 마우스

5. 교재 322페이지 참조

6.

(1) ActionListener

(2) addActionListener(this);

(3) actionPerformed(ActionEvent e)

7.

|  |
| --- |
| public class EventExam extends JFrame {  ….  button1.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  ….  }  });  } |

**9장 연습문제 해답**

# 이론문제

1. 다음에 대해 교재 333 ~ 338 내용 참조

(1) 연결리스트

(2) 스택

(3) 트리

(4) 배열

2. 스택에 데이터를 넣는 것을 push라 하고, 꺼내는 작업을 pop이라고 한다.

3. (1), (3) -> (3)은 다양하게 구현된 자료구조 API 를 배우는데 필요한 노력을 줄일 수 있다. 가 정확한 표현임. 여기서는 조금 애매하게 설명되어 있음.(추후 문장 수정 필요)

4. Map, List, Set

5. ArrayList

6.

HashMap<String> hm = new HashMap<String>();

**10장 연습문제 해답**

# 이론문제

1.

(1) 데이터베이스 관리 시스템은 전문가에 의한 관리가 필요하다.

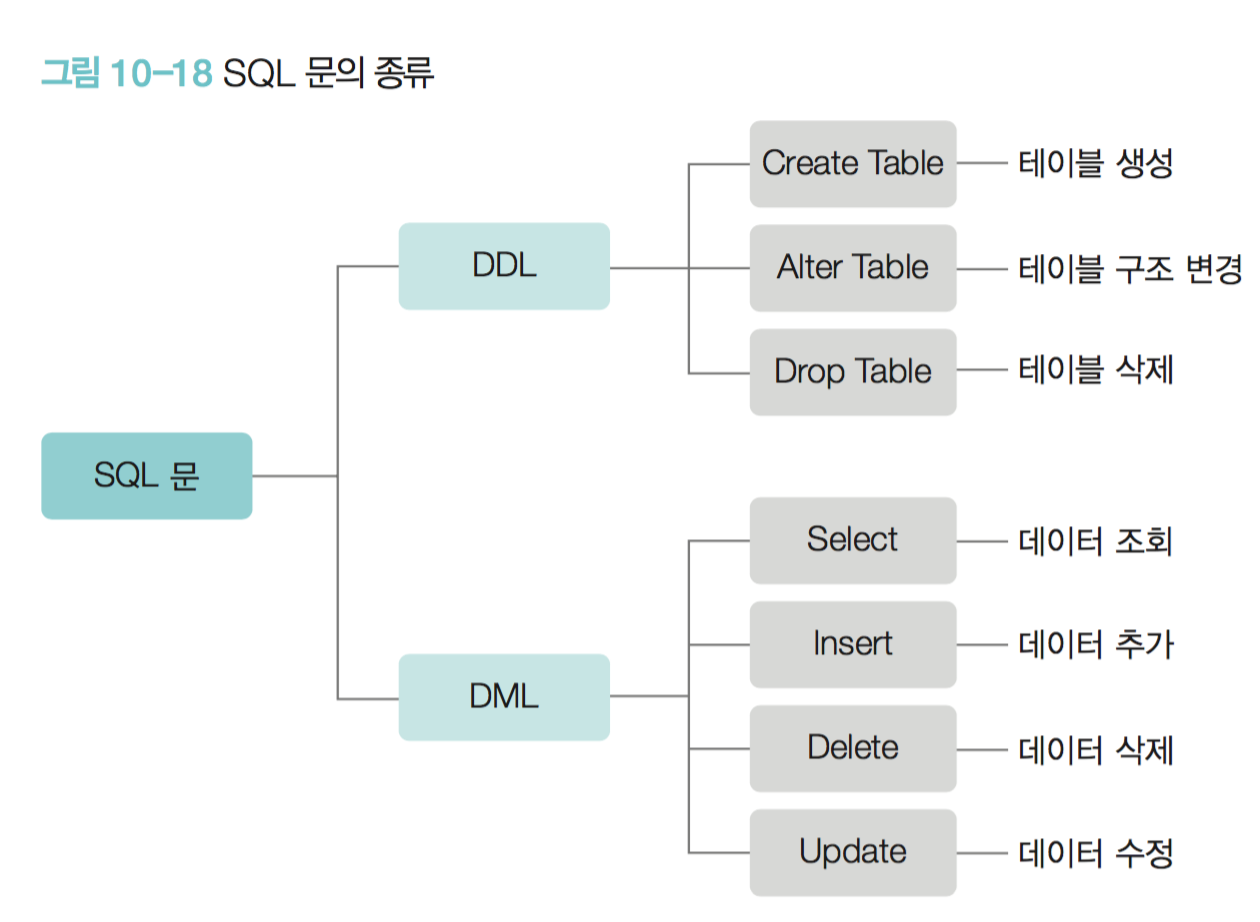
(2) 현재 가장 널리 사용되는 데이터베이스는 관계형 데이터베이스 이다.

(4) 데이터 무결성 유지는 데이터베이스 사용으로 얻을 수 있는 잇점중 하나이다.

(5) 데이터 베이스는 테이블을 기반의 셀 형태로 자료를 관리 한다.

2. 컬럼, 로우

3.



4.

CREATE TABLE PRODUCT (

pid int NOT NULL Primary Key,

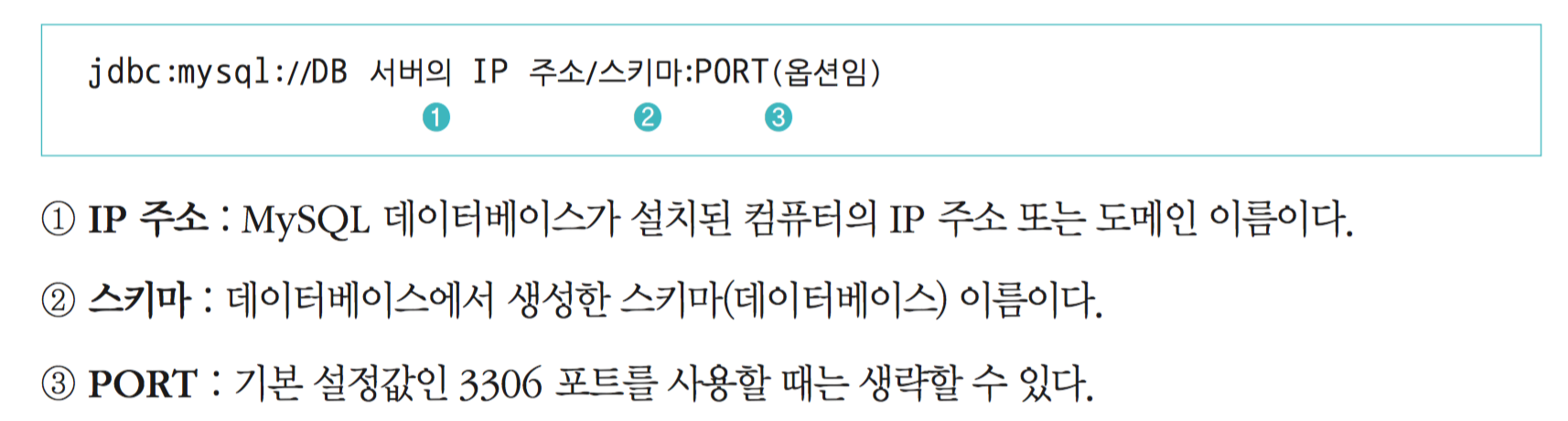
pname varchar(20),

price int,

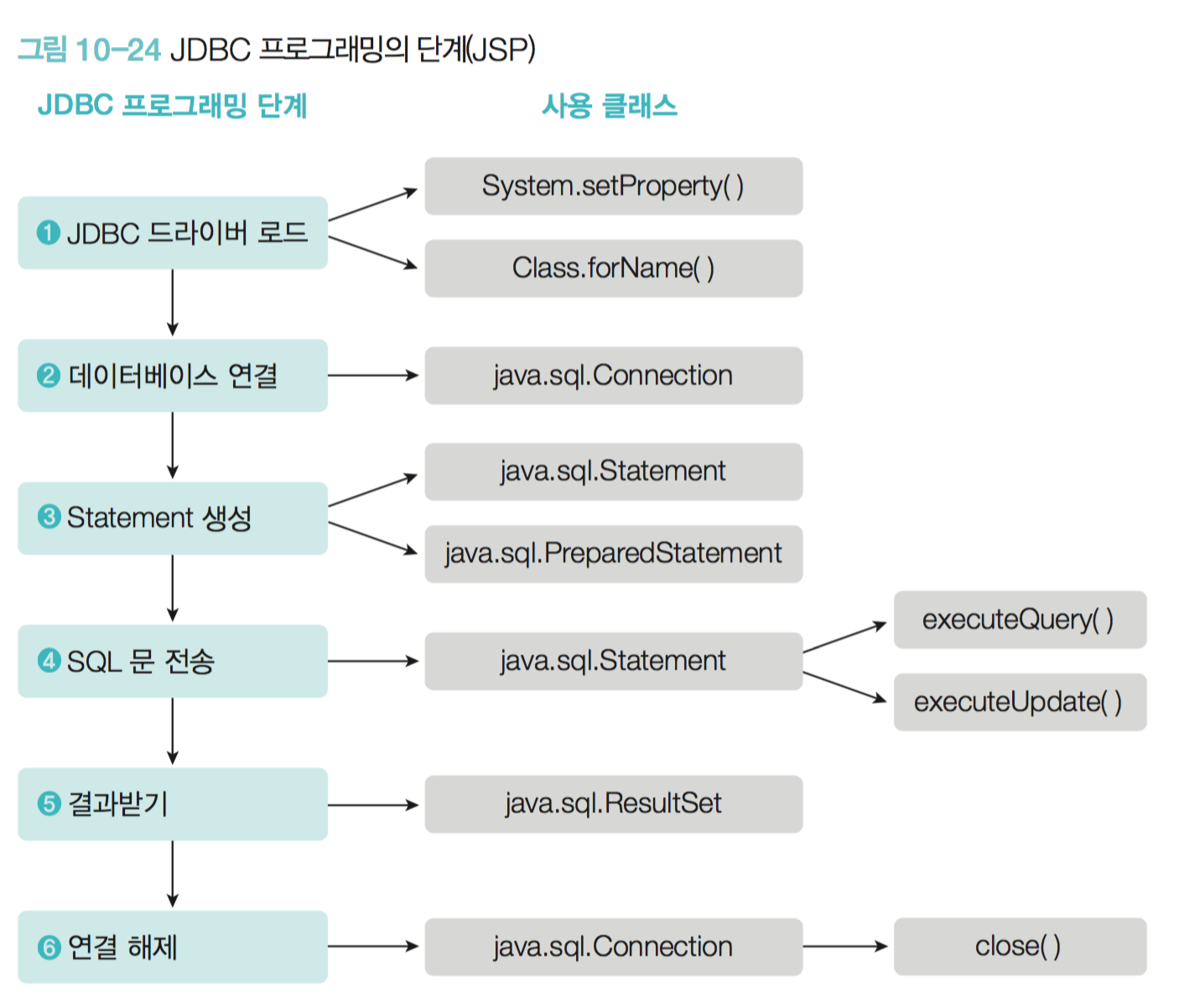
mdate date

);

5.



6.



7. PreparedStatement

**11장 연습문제 해답**

# 이론문제

1. 예외처리

2.

•컴파일 오류 : 컴파일러가 코드를 컴파일하지 못하는 문제로, 보통 프로그램 소스 코드의 문법이 잘못되었을 때 나 타난다. 예전에는(지금도 일부는 그렇지만) 코드를 작성하고 나서 컴파일을 따로 했는데, 이때 코드에 문법적 오류가 있으면 컴파일 오류가 발생한다. 최근에는 많은 프로그래밍 도구가 실시간 오류 점검 기능을 제공하며, 코드를 저장 하면 자동으로 컴파일할 때가 많아 따로 컴파일 오류를 처리하는 경우는 드물다. 물론 컴파일 시간이 오래 걸릴 정 도로 복잡한 프로그램이라면, 실시간 컴파일이나 오류 점검 기능이 오히려 개발 효율을 떨어뜨릴 수 있다.

• 런타임 오류 : 프로그램을 실행하는 과정에서 발생하는 오류이다. 문법적인 문제는 없지만, 프로그램의 논리 흐름(로 직)상의 실수로 잘못된 파라미터를 전달하거나 변수 또는 객체를 초기화하지 않는 문제 등이 이 오류에 해당한다. 경 우에 따라서는 프로그램이 잘 실행되다가도 특정 상황에서만 오류가 발생할 수도 있어 디버깅의 주요 대상이다.

3.

• 파일입출력을할때파일또는디렉터리이름을잘못기재했거나없을때

• 네트워크 프로그램에서 접속할 서버의 주소가 틀렸거나 서버가 다운되었을 때

• 키보드에서 값을 입력받아야 하는데, 키보드를 연결하지 않았을 때

• 배열의크기를초과해서배열원소에접근할때

• 초기화하지않은객체를사용할때

• 데이터베이스 프로그램에서 데이터베이스의 주소, 아이디, 비밀번호, 스키마, SQL 쿼리 등이 잘못되었을 때

4.

(1) try (2) catch

5. 메서드 선언부에 throws 구문 추가

void fileRead() throws FileNotFoundException {

…

}

6.

(1) 프로세스 (2) 스레드

7.

(1) 스레드를 생성 하는 것이 프로세스를 생성하는 것 보다 시스템 자원을 절약할 수 있다.

(4) 프로세스 밖에서 스레드를 제어할 수 없다.

(5) 동기화, 교착상태 같은 문제들을 고려해야 한다.

8. 교재 442, 예제 11-6 참조

9. synchronized

10. 함수형 프로그램언어, 람다식

11.

(String msg) -> {System.out.println(msg);}

12. 함수형 인터페이스