

Chapter 01

1. ④
2. ②
3. ④
4. ③
5. ③ → ① → ② → ④
6. ④
7. ①, ②, ③, ④
- 8.

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

Chapter 02

1. ④
2. ②, ③, ⑥, ⑦
- 3.

byte	short char	int	long
		float	double
boolean			

4. int, double
age, price
10, 3, 14
5. ③
6. ④
7. ③

- 8. ③
- 9. ①
- 10. ②
- 11. 13, 16

Chapter 03

- 1. 31
- 2. 가
- 3. pencils / students
pencils % students
- 4. value / 100 * 100
- 5. ①, ②, ③
- 6. true, false
- 7. Double.isNaN(z)

Chapter 04

- 1. ②
- 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);  
    }  
}
```

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 = Integer.parseInt(scanner.nextLine());

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

```

```

        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. ④
2. ③
3. ②
4. ②
5. ③
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;
    }
}

```

```

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 = Integer.parseInt(scanner.nextLine());

    if(selectNo == 1) {
        System.out.print("학생수> ");
        studentNum = Integer.parseInt(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] = Integer.parseInt(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. ③

2. ④

3. ④

4. ③

5. ①

6. ④

7. ②

8. ②

9. ②

10. ④

11. ③

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;  
    }  
}
```


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);
    }
}
```

```
    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;
    }
}
```

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();
            }
        }
    }
}
```

```

        } 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");
}

```

```

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. ①
2. ②
3. ①
4. ④
5. ②
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

8. 스노우 타이어가 굴러갑니다.
스노우 타이어가 굴러갑니다.
9. ②
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. ①
2. ③
3. ④
4. ①, ②, ③, ④
5. implements Remocon,

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

6.

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

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

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() {
```



```

        System.out.println("MySQL DB를 수정");
    }

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

```

8. a instanceof C c

Chapter 09

1. ④

2. ③

3. ③

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.");
        }
    }
}

```

```

};

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.

Chapter 10

1. ②
2. ③
3. ③
4. ④
5. ②

Chapter 11

1. ④
2. ③
3. ④
4. ②
5. ③
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. ④

2. ③

3. ④

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. ③

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);
        }
    }
}

```

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. ④

14. ④

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. ④

19. ④

Chapter 13

1. ④

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;
    }
}
```


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

Chapter 15

1. ④

2. ③

3. ④

4. ③

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;
    }
}
```

```

    }

    @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. 4

12. 3

13. 4

Chapter 16

1. 4

2. 4

3. 2

```
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

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. ④

2. ②

3. ④

4. ③

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. ❷

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();
```

```

//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. ②
2. ②, ④
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. ④
- 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;
```



```
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;
```

```
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) {
```

```

        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"));

```

```

        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

```

```

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();
    }
}

```

```

//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();
}

```

```

//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);

```

```
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());
```



```
        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. ④
2. ③
3. ② → ① → ④ → ③
4. ④
5. ②
6. ③
7. ③
- 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;
}

```

```
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]");  
    }  
}
```



```
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: 1.0k | 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
    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();

```

```

        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());

```

```

        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: 1.0k | 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";

```



```

        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;
```



```

    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"));

```

```

        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("-----");

```

```

System.out.println("보조 메뉴: 1.0k | 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();
}

```



```

public void clear() {
    System.out.println("[게시물 전체 삭제]");
    System.out.println("-----");
    System.out.println("보조 메뉴: 1.0k | 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: 1.0k | 2.Cancel");
}

```



```
//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) {

```

```

        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();
    }
}

```

```
//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]");
}
```



```

    }

    //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: 1.0k | 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");
        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: 1.0k | 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();
        }
    }
}

```

```
        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();
}
}
```