

1. 변수와 데이터 타입

- 문제: 두 정수를 입력받아 합을 출력하는 프로그램을 작성하세요.
힌트: Scanner를 사용합니다.

2. 조건문

- 문제: 입력받은 숫자가 짝수인지 홀수인지 판별하는 프로그램을 작성하세요.

3. 반복문

- 문제: 1부터 100까지의 숫자 중 3의 배수의 합을 구하는 프로그램을 작성하세요.

4. 배열

- 문제: 정수형 배열에서 가장 큰 값을 찾는 프로그램을 작성하세요.

5. 문자열 처리

- 문제: 사용자가 입력한 문자열이 회문(앞뒤가 똑같은 문자열)인지 판별하는 프로그램을 작성하세요.
예시: "madam" → 회문

6. 메서드

- 문제: 두 정수를 매개변수로 받아 더한 값을 반환하는 메서드를 작성하고 사용 하세요.

7. 클래스와 객체

- 문제: 학생의 이름과 점수를 저장하는 Student 클래스를 만들고, 학생 객체를 생성하여 값을 출력하세요.

8. 생성자

- 문제: Rectangle 클래스를 작성하고, 생성자를 통해 가로와 세로 값을 초기화한 뒤 넓이를 계산하는 프로그램을 작성하세요.

9. 상속

- 문제: Animal 클래스를 작성하고, 이를 상속받는 Dog 클래스를 작성하여 Dog 객체의 메서드를 호출하세요.

10. 인터페이스

- 문제: Vehicle 인터페이스를 작성하고, 이를 구현하는 Car와 Bus 클래스를 작성하세요. 각각의 클래스에서 고유의 메시지를 출력하는 메서드를 구현하세요.

11. 예외 처리

- 문제: 두 숫자를 입력받아 나누는 프로그램을 작성하되, **0으로 나눌 경우** 적절한 메시지를 출력하도록 예외 처리를 추가하세요.

12. 컬렉션 - ArrayList

- 문제: ArrayList에 정수를 추가하고, 모든 값을 출력하는 프로그램을 작성하세요.

13. 컬렉션 - HashMap

- 문제: 학생의 이름과 점수를 저장하는 HashMap을 생성하고, 특정 학생의 점수를 검색하는 프로그램을 작성하세요.

14. 파일 입출력

- 문제: 텍스트 파일을 읽어와 내용을 출력하는 프로그램을 작성하세요.
힌트: BufferedReader , FileReader

15. 정렬

- 문제: 사용자 정의 객체(예: Book)를 ArrayList에 저장하고, 제목 기준으로 정렬하는 프로그램을 작성하세요.
힌트: Comparator를 사용하세요.

16. 큐(Queue) 사용하기 - 이부분은 답지를 먼저 보세요

문제: Queue를 사용하여, 정수 1부터 5까지 값을 큐에 넣고 하나씩 꺼내서 출력하는 프로그램을 작성하세요.

17. 스레드

- 문제: 두 개의 스레드를 생성하고, 각각 1부터 10까지 숫자를 출력하도록 프로그램을 작성하세요.

18. 랴다식

- 문제: 숫자 리스트에서 짝수만 출력하는 랴다식을 작성하세요.
힌트: List.stream().filter()를 사용하세요.

19. 스트림

- 문제: 정수 리스트에서 각 값을 제곱한 리스트를 반환하는 프로그램을 작성하세요.
예시: [1, 2, 3] → [1, 4, 9]

20. JDBC

- 문제: 데이터베이스에 연결하여 users 테이블에서 모든 데이터를 출력하는 프로그램을 작성하세요.
힌트: DriverManager와 ResultSet을 사용하세요.

데이터베이스연결정보

```
Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/testdb", "scott", "tiger");
```

1. 변수와 데이터 타입

문제: 두 정수를 입력받아 합을 출력하는 프로그램.

```
import java.util.Scanner;

public class Test1{
    public static void main (String[] args) {
        Scanner scanner =new Scanner(System.in);
        System.out.print("첫 번째 숫자를 입력하세요: ");
        int num1 = scanner.nextInt();
        System.out.print("두 번째 숫자를 입력하세요: ");
        int num2 = scanner.nextInt();
        System.out.println("두 숫자의 합: " + (num1 + num2));
    }
}
```

2. 조건문

문제: 입력받은 숫자가 짝수인지 홀수인지 판별.

```
import java.util.Scanner;

public class Test2{
    public static void main (String[] args) {
        Scanner scanner =new Scanner(System.in);
        System.out.print("숫자를 입력하세요: ");
        int num = scanner.nextInt();
        if (num % 2 == 0) {
            System.out.println(num + "은(는) 짝수입니다.");
        } else {
            System.out.println(num + "은(는) 홀수입니다.");
        }
    }
}
```

3. 반복문

문제: 1부터 100까지 3의 배수의 합.

```
public class Test3{
    public static void main (String[] args) {
        int sum =0;
        for (int i =1; i <= 100; i++) {
            if (i % 3 == 0) {
                sum += i;
            }
        }
        System.out.println("1부터 100까지 3의 배수의 합: " + sum);
    }
}
```

4. 배열

문제: 배열에서 가장 큰 값 찾기.

```
public class Test4{
    public static void main (String[] args) {
        int[] numbers = {10, 20, 5, 8, 30};
        int max = numbers[0];
        for (int num : numbers) {
            if (num > max) {
                max = num;
            }
        }
        System.out.println("가장 큰 값: " + max);
    }
}
```

5. 문자열 처리

문제: 회문인지 판별. (level ,refer)

```
import java.util.Scanner;
public class Test5{
    public static void main (String[] args) {
        Scanner scanner =new Scanner(System.in);
        System.out.print("문자열을 입력하세요: ");
        String str = scanner.nextLine();
        String reversed =new StringBuilder(str).reverse().toString();
        if (str.equals(reversed)) {
            System.out.println(str + "은(는) 회문입니다.");
        } else {
            System.out.println(str + "은(는) 회문이 아닙니다.");
        }
    }
}
```

//회문인지 판별하는 매서드를 직접작성

import java.util.Scanner;

```
public class Test5_1{
    public static void main(String[] args) {
        // Scanner 객체 생성
        Scanner scanner = new Scanner(System.in);

        // 사용자로부터 문자열 입력받기
        System.out.print("문자열을 입력하세요: ");
        String str = scanner.nextLine();

        // 회문 판별
        if (isPalindrome(str)) {
            System.out.println(str + "은(는) 회문입니다.");
        } else {
            System.out.println(str + "은(는) 회문이 아닙니다.");
        }

        // Scanner 닫기
        scanner.close();
    }
}
```

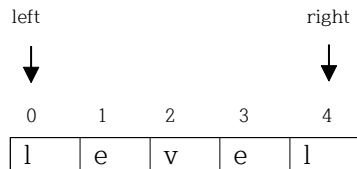
// 회문 판별하는 메서드

```
public static boolean isPalindrome(String str) {

    // 문자열을 앞에서부터와 뒤에서부터 비교
    int left = 0;
    int right = str.length() - 1;

    // 양 끝에서부터 비교하며 중앙으로 이동
    while (left < right) {
        if (str.charAt(left) != str.charAt(right)) {
            return false; // 하나라도 다르면 회문 아님
        }
        left++;
        right--;
    }

    return true; // 모든 문자 비교 후 동일하면 회문
}
}
```



6. 메서드

문제: 두 정수의 합을 반환하는 메서드.

```
public class Test6 {
    public static void main (String[] args) {
        int result = add(10, 20);
        System.out.println("두 숫자의 합: " + result);
    }

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

7. 클래스와 객체

문제: Student 클래스 정의 및 객체 생성.

```
class Student {
    String name;
    int score;

    // 생성자
    Student(String name, int score) {
        this.name = name;
        this.score = score;
    }

    // toString 메서드 오버라이드
    @Override
    public String toString() {
        return "이름: " + name + ", 점수: " + score;
    }
}

public class Test7{
    public static void main(String[] args) {
        // Student 객체 생성
        Student student = new Student("홍길동", 95);

        // student 객체를 출력
        System.out.println(student);
    }
}
```

8. 생성자

문제: Rectangle 클래스와 넓이 계산.

```
class Rectangle {
    int width, height;

    Rectangle(int width, int height) {
        this.width = width;
        this.height = height;
    }

    int getArea () {
        return width * height;
    }
}
```

```
public class Test8{
    public static void main (String[] args) {
        Rectangle rect =new Rectangle(5, 10);
        System.out.println("사각형의 넓이: " + rect.getArea());
    }
}
```

9. 상속

문제: Animal 클래스와 상속.

```
class Animal {
    void 짖다() {
        System.out.println("동물이 짖는다");
    }
}

class Dog extends Animal {
    @Override
    void 짖다() {
        System.out.println("멍멍!");
    }
}

public class Test9{
    public static void main (String[] args) {
        Animal animal=new Dog();
        animal.짖다();
    }
}
```

10. 인터페이스

문제: Vehicle 인터페이스와 구현.

```
interface Vehicle {
    void move ();
}

class Car implements Vehicle {
    public void move () {
        System.out.println("자동차가 달립니다.");
    }
}

class Bus implements Vehicle {
    public void move () {
        System.out.println("버스가 달립니다.");
    }
}

public class Test10{
    public static void main (String[] args) {
        Vehicle car =new Car();
        Vehicle bus =new Bus();
        car.move();
        bus.move();
    }
}
```

11. 예외 처리

문제: 0으로 나누기 예외 처리.

```
import java.util.Scanner;

public class Test11{
    public static void main (String[] args) {
        Scanner scanner =new Scanner(System.in);
        System.out.print("숫자를 입력하세요: ");
        int a = scanner.nextInt();
        System.out.print("나눌 숫자를 입력하세요: ");
        int b = scanner.nextInt();

        try {
            System.out.println("결과: " + (a / b));
        } catch (ArithmeticException e) {
            System.out.println("0으로 나눌 수 없습니다.");
        }
    }
}
```

12. 컬렉션 - ArrayList

문제: ArrayList에 정수를 추가하고, 모든 값을 출력하는 프로그램.

```
import java.util.ArrayList;

public class Test12{
    public static void main (String[] args) {
        ArrayList<Integer> numbers = new ArrayList<>();
        numbers.add(10);
        numbers.add(20);
        numbers.add(30);

        for (int num : numbers) {
            System.out.println(num);
        }
    }
}
```

13. 컬렉션 - HashMap

문제: 학생의 이름과 점수를 저장하는 HashMap을 생성하고, 특정 학생의 점수를 검색하는 프로그램.

```
import java.util.HashMap;

public class Test13{
    public static void main (String[] args) {
        HashMap<String, Integer> students = new HashMap<>();
        students.put("김학생", 95);
        students.put("이학생", 88);
        students.put("최학생", 90);

        String name ="김학생";
        System.out.println(name + "의 점수: " + students.get(name));
    }
}
```

14. 파일 입출력

문제: 텍스트 파일을 읽어와 내용을 출력하는 프로그램.

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

public class Test14{
    public static void main (String[] args) {
        try (BufferedReader br =new BufferedReader(new FileReader("test.txt"))) {
            String line;
            while ((line = br.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
            System.out.println("파일을 읽는 중 오류 발생: " + e.getMessage());
        }
    }
}
```

15. 정렬

문제: 사용자 정의 객체(예: Book)를 ArrayList에 저장하고, 제목 기준으로 정렬하는 프로그램.

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;

class Book {
    String title;
    int price;

    Book(String title, int price) {
        this.title = title;
        this.price = price;
    }

    @Override
    public String toString () {
        return "책 제목: " + title + ", 가격: " + price;
    }
}

public class Test15{
    public static void main (String[] args) {
        ArrayList<Book> books = new ArrayList<>();
        books.add(new Book("자바의 정석", 30000));
        books.add(new Book("스프링 입문", 25000));
        books.add(new Book("알고리즘", 20000));

        Collections.sort(books, Comparator.comparing(b -> b.title));

        for (Book book : books) {
            System.out.println(book);
        }
    }
}
```

// Comparator 인터페이스 직접구현한 방법으로 해결 (1. 이름있는 클래스 ,2.익명 3.람다식)
// 익명과 람다는 직접 작성해 보세요

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;

class Book {
    String title;
    int price;

    Book(String title, int price) {
        this.title = title;
        this.price = price;
    }

    @Override
    public String toString() {
        return "책 제목: " + title + ", 가격: " + price;
    }
}

class BookTitleComparator implements Comparator<Book> {
    @Override
    public int compare(Book b1, Book b2) {
        return b1.title.compareTo(b2.title); // 제목을 기준으로 정렬
    }
}

public class Main {
    public static void main(String[] args) {
        ArrayList<Book> books = new ArrayList<>();
        books.add(new Book("자바의 정석", 30000));
        books.add(new Book("스프링 입문", 25000));
        books.add(new Book("알고리즘", 20000));

        // Comparator 구현체를 사용하여 정렬
        Collections.sort(books, new BookTitleComparator());

        for (Book book : books) {
            System.out.println(book);
        }
    }
}
```

16. 큐(Queue) 사용하기

문제: Queue를 사용하여, 정수 1부터 5까지 값을 큐에 넣고 하나씩 꺼내서 출력하는 프로그램을 작성하세요.

```
import java.util.LinkedList;
import java.util.Queue;

public class Test16{
    public static void main(String[] args) {
        Queue<Integer> queue = new LinkedList<>(); //큐를 생성하는 방법

        // 큐에 값 추가
        for (int i = 1; i <= 5; i++) {
            queue.add(i);
        }

        // 큐에서 값을 하나씩 꺼내서 출력
        while (!queue.isEmpty()) {
            System.out.println(queue.poll());    // poll메서드로 꺼낸다. 가장먼저 넣은 것이 꺼내짐
        }
    }
}
```

!! 우선순위 큐 사용해 보기

```
import java.util.PriorityQueue;

class Student {
    String name;
    int score;

    // 생성자
    public Student(String name, int score) {
        this.name = name;
        this.score = score;
    }

    // 학생 정보 출력용
    @Override
    public String toString() {
        return "학생 이름: " + name + ", 점수: " + score;
    }
}

public class Test16_1{
    public static void main(String[] args) {
        // PriorityQueue에 학생 객체를 점수 순으로 정렬하기 위해 람다식 사용 ( Comparator를 구현해야함)
        PriorityQueue<Student> priorityQueue = new PriorityQueue<>((student1, student2) -> student1.score - student2.score);

        // 학생 객체 추가
        priorityQueue.add(new Student("이학생", 90));
        priorityQueue.add(new Student("김학생", 85));
        priorityQueue.add(new Student("최학생", 95));
        priorityQueue.add(new Student("전학생", 80));

        // 큐에서 학생 객체를 점수가 낮은 순서대로 꺼내기 (poll로 꺼내기)
        while (!priorityQueue.isEmpty()) {
            System.out.println(priorityQueue.poll()); // 가장 점수가 낮은 학생부터 꺼낸다.
        }
    }
}
```

17. 스레드

문제: 두 개의 스레드를 생성하고, 각각 1부터 10까지 숫자를 출력.

```
class 숫자출력하기 implements Runnable {
    public void run() {
        for (int i = 1; i <= 10; i++) {
            System.out.println(i);
        }
    }
}

public class Test17{
    public static void main(String[] args) {
        PrintNumbers printNumbers = new PrintNumbers();

        // Runnable을 사용하여 스레드 생성
        Thread thread1 = new Thread(숫자출력하기 );
        Thread thread2 = new Thread(숫자출력하기 );

        thread1.start();
        thread2.start();
    }
}
```

18. 랴다식

문제: 숫자 리스트에서 짝수만 출력하는 랴다식을 filter메서드에서 사용하시오

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

public class Test18 {
    public static void main (String[] args) {
        List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5, 6);
        numbers.stream()
            .filter(n -> n % 2 == 0)
            .forEach( item -> System.out.println( item) );
    }
}
```

19. 스트림

문제: 정수 리스트에서 각 값을 제공한 리스트를 반환.

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

public class Test19{
    public static void main (String[] args) {
        List<Integer> 숫자목록= Arrays.asList(1, 2, 3);
        List<Integer> 제공한목록 = 숫자목록.stream()
                                          .map(n -> n * n)
                                          .collect(Collectors.toList());

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

20. JDBC

문제: 데이터베이스에 연결하여 users 테이블에서 모든 데이터를 출력하는 프로그램.

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

public class Test20{
    public static void main (String[] args) {
        try {
            Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/testdb", "scott", "tiger");
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM users");

            while (rs.next()) {
                System.out.println("ID: " + rs.getInt("id") + ", Name: " + rs.getString("name"));
            }

            rs.close();
            stmt.close();
            conn.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```