



section1-class00-기능 구현 요구 사항

Stream API를 이용한 구구단 애플리케이션 제작

Stream API를 이용한 구구단 애플리케이션 제작

- ✓ 애플리케이션 설명
- ✓ 프로젝트 목표
- ✓ 기능 구현 요구 사항
 - 1 입력
 - 2 출력
 - 3 💡 제한 사항(필수)
 - 4 제한 사항(선택)
 - 5 입출력 예시

✓ 애플리케이션 설명

이번 섹션에서는 아주 간단한 구구단 애플리케이션을 제작해 보면서 여러분들 스스로 사고 하면서 애플리케이션을 제작할 수 있는 기본적인 사고 원칙에 익숙해지는 시간을 가져보도록 하겠습니다.

```

1 package com.itvillage.legacy;
2
3 > import ...
4
5
6 /**
7  * 🖱 명령형 방식의 구구단
8  */
9 public class GuGuDanApplicationLegacy {
10     public static void main(String[] args) {
11         Scanner scanner = new Scanner(System.in);
12
13         System.out.print("> 단을 입력해 주세요(2 - 9): ");
14
15         int dan = scanner.nextInt();
16
17         for (int i = 1; i < 9; i++) {
18             System.out.println(dan + " * " + i + " = " + dan * i);
19         }
20     }
21 }

```

[그림 1-1] 전통적인 명령형 방식의 구구단 출력 애플리케이션

[그림 1-1]은 특정 **단**을 입력으로 전달 받아서 해당 단을 출력하는 구구단 프로그램입니다. for문을 이용한 명령형 방식의 구구단 애플리케이션이고, 여러분들이 구현하는데 큰 어려움은 없을 거라고 생각합니다.

이번 섹션에서는 구구단을 출력하는 애플리케이션을 제작해보되 [그림 1-1]과 같이 for문을 이용하는 대신 **Java의 Stream API를 이용**해서 구구단 애플리케이션을 만들어보도록 하겠습니다.

구구단을 출력하는 로직 자체는 어렵지 않은 아주 간단한 애플리케이션이지만 꽤 괜찮아 보이는 구구단 애플리케이션을 제작하기 위한 연습을 해 보시기 바랍니다.

✅ 프로젝트 목표

- Java의 **람다 표현식과 함수형 인터페이스의 개념을 이해**할 수 있다.

- **Stream API의 특성과 기본 사용 방법을 이해**할 수 있다.
 - **코드 가독성의 중요성**을 이해할 수 있다.
 - 메서드 추출(Extract)을 통해 **기능별로 메서드의 로직을 분리**할 수 있다.
 - 클래스를 이용해 **역할 분리의 개념**을 이해할 수 있다.
-

✓ 기능 구현 요구 사항

1 입력

1. 특정 **단**을 입력한다.

2 출력

1. 사용자가 입력한 **단**에 해당하는 구구단을 출력할 수 있어야 한다.

3 💡 제한 사항(필수)

1. 하나의 **단**만 입력 가능하다.
2. **단**은 2단부터 9단까지 한 자리 숫자의 단만 사용할 수 있어야 한다.
3. 전통적인 for문이나 if문 대신에 ★ **Java의 Stream API**를 사용해서 구현해야 한다.

4 제한 사항(선택)

1. 입력 값이 올바르지 않을 경우 안내 메시지를 출력할 수 있어야 한다.
2. 사용자의 선택에 따라 애플리케이션 종료 여부를 결정할 수 있어야 한다.

5 입출력 예시

> 단을 입력해 주세요(2 - 9): 9

9 * 1 = 9

9 * 2 = 18

9 * 3 = 27

9 * 4 = 36

9 * 5 = 45

9 * 6 = 54

9 * 7 = 63

9 * 8 = 72

9 * 9 = 81