



# 숫자 야구 - 미니과제

## 미니과제 진행 방식

### 진행 방식

- 미션은 과제 진행 요구 사항, 기능 요구 사항, 프로그래밍 요구 사항 세 가지로 구성되어 있다.
- 세 개의 요구 사항을 만족하기 위해 노력한다. 특히 기능을 구현하기 전에 기능 목록을 만들고, 기능 단위로 커밋 하는 방식으로 진행한다.
- 기능 요구 사항에 기재되지 않은 내용은 스스로 판단하여 구현한다.

### 미션 제출 방법

- 미션 구현을 완료한 후 GitHub을 통해 제출해야 한다.
  - GitHub을 활용한 제출 방법은 미니과제 진행 가이드 문서를 참고해 제출한다.
  - 풀 리퀘스트 제목은 '[OO대 트랙\_본명] 미션 제출합니다.'로 작성한다.
- GitHub에 미션을 제출한 후 [구글 폼](#)에 PR 링크를 포함하여 최종 제출한다.

### 과제 진행 소감

과제를 수행하면서 느낀 점, 배운 점, 많은 시간을 투자한 부분 등 자유롭게 작성한다.

#### 예시

이번 미션은 생각만큼 쉽지 않았습니다.  
특히 기능을 분리해서 기능 단위로 커밋하는 것이 쉽지 않다는 것을 깨달았습니다.  
기능 단위로 분리하고 커밋하기 위해 다양한 방법으로 연습하였습니다.  
하지만 일단 기능을 분리해서 구현하고 나니 좀 더 명확하게 구현할 수 있다는 것을 느낄 수 있었습니다.  
미션을 수행하면서 더욱 성장한 걸 느꼈어요. :)

### 과제 제출 전 체크 리스트

- 터미널에서 `java -version` 을 실행하여 Java 버전이 17인지 확인한다. Eclipse 또는 IntelliJ IDEA와 같은 IDE에서 Java 17로 실행되는지 확인한다.
- 터미널에서 Mac 또는 Linux 사용자의 경우 `./gradlew clean test` 명령을 실행하고, Windows 사용자의 경우 `gradlew.bat clean test` 또는 `./gradlew.bat clean test` 명령을 실행할 때 모든 테스트가 아래와 같이 통과하는지 확인한다.

```
BUILD SUCCESSFUL in 0s
```

## 숫자 야구

### 과제 진행 요구 사항

- 미션은 [숫자 야구](#) 저장소를 포크하고 클론하는 것으로 시작한다.
- 기능을 구현하기 전 `README.md` 에 구현할 기능 목록을 정리해 추가한다.
- Git의 커밋 단위는 앞 단계에서 `README.md` 에 정리한 기능 목록 단위로 추가한다.
  - [AngularJS Git Commit Message Conventions](#)을 참고해 커밋 메시지를 작성한다.

- 자세한 과제 진행 방법은 미니과제 진행 가이드 문서를 참고한다.

## 기능 요구 사항

기본적으로 1부터 9까지 서로 다른 수로 이루어진 3자리의 수를 맞추는 게임이다.

- 같은 수가 같은 자리에 있으면 스트라이크, 다른 자리에 있으면 볼, 같은 수가 전혀 없으면 낫싱이란 힌트를 얻고, 그 힌트를 이용해서 먼저 상대방(컴퓨터)의 수를 맞추면 승리한다.

e.g.  
상대방(컴퓨터)의 수가 425일 때,  
- 123을 제시한 경우: 1스트라이크  
- 456을 제시한 경우: 1볼 1스트라이크  
- 789를 제시한 경우: 낫싱

- 위 숫자 야구 게임에서 상대방의 역할을 컴퓨터가 한다. 컴퓨터는 1에서 9까지 서로 다른 임의의 수 3개를 선택한다. 게임 플레이어는 컴퓨터가 생각하고 있는 3개의 숫자를 입력하고, 컴퓨터는 입력한 숫자에 대한 결과를 출력한다.
- 이 같은 과정을 반복해 컴퓨터가 선택한 3개의 숫자를 모두 맞히면 게임이 종료된다.
- 게임을 종료한 후 게임을 다시 시작하거나 완전히 종료할 수 있다.
- 사용자가 잘못된 값을 입력할 경우 `IllegalArgumentException` 을 발생시킨 후 애플리케이션은 종료되어야 한다.

## 실행 결과

```
숫자를 입력해 주세요 : 123
1볼 1스트라이크
숫자를 입력해 주세요 : 145
1볼
숫자를 입력해 주세요 : 671
2볼
숫자를 입력해 주세요 : 216
1스트라이크
숫자를 입력해 주세요 : 713
3스트라이크
3개의 숫자를 모두 맞히셨습니다! 게임 종료
게임을 새로 시작하려면 1, 종료하려면 2를 입력하세요.
1
숫자를 입력해 주세요 : 123
1볼
...
```

## 프로그래밍 요구 사항 1

- JDK 17 버전에서 실행 가능해야 한다.
- 프로그램 실행의 시작점은 `Application` 의 `main()` 이다.
- `build.gradle` 파일은 변경할 수 없으며, 제공된 라이브러리 이외의 외부 라이브러리는 사용하지 않는다.
- 프로그램 종료 시 `System.exit()` 를 호출하지 않는다.
- 프로그래밍 요구 사항에서 달리 명시하지 않는 한 파일, 패키지 등의 이름을 바꾸거나 이동하지 않는다.

## 프로그래밍 요구 사항 2

- 자바 코드 컨벤션을 지키면서 프로그래밍한다.
  - 기본적으로 [Google Java Style Guide](#)을 원칙으로 한다.
  - 단, 들여쓰기는 '2 spaces'가 아닌 '4 spaces'로 한다.
- indent(인덴트, 들여쓰기) depth를 3이 넘지 않도록 구현한다. 2까지만 허용한다.
  - 예를 들어 while문 안에 if문이 있으면 들여쓰기는 2이다.
  - 힌트: indent(인덴트, 들여쓰기) depth를 줄이는 좋은 방법은 함수(또는 메서드)를 분리하면 된다.
- 3행 연산자를 쓰지 않는다.
- 함수(또는 메서드)가 한 가지 일만 하도록 최대한 작게 만들어라.
- JUnit 5와 AssertJ를 이용하여 정리한 기능 목록이 정상적으로 작동하는지 테스트 코드로 확인한다.

- 테스트 도구 사용법이 익숙하지 않다면 아래 문서를 참고하여 학습한 후 테스트를 구현한다.
  - [JUnit 5 User Guide](#)
  - [AssertJ User Guide](#)
  - [AssertJ Exception Assertions](#)
  - [Guide to JUnit 5 Parameterized Tests](#)