

# 프리코스 진행방식

# 진행 방식

- 미션은 기능 요구사항, 프로그래밍 요구사항, 과제 진행 요구사항 세 가지로 구성되어 있다.
- 세 개의 요구사항을 만족하기 위해 노력한다. 특히 기능을 구현하기 전에 **기능 목록을 만들고, 기능 단위로 commit하는 방식으로** 진행한다.

# 미션 제출 방법

- 미션 구현을 완료한 후 GitHub을 통해 제출해야 한다.
- GitHub을 활용한 제출 방법은 [프리코스 과제 제출](#) 문서 참고해 제출한다.
- GitHub에 미션을 제출한 후 [edu.nextstep@gmail.com](mailto:edu.nextstep@gmail.com) 로 메일을 발송한다.

# email 템플릿

- 제목 양식
- 본문 양식
  - 교육 과정 신청시 이메일 주소
  - Pull Request URL
  - 미션 진행 중 느낀점, 배운점

제목 : [\$이름] 프리코스 미션 제출합니다.

내용 :

다음 두 개의 정보를 반드시 포함해 메일을 보낸다.

\* 교육 과정 신청시 email 주소:

\* Pull Request URL:

미션을 진행하면서 느끼고, 배운점, 많은 시간을 투자한 부분 등도 포함하면 더 좋을 것 같아요.

# email 예시

새 메일

받는사람 edu.nextstep@gmail.com

제목 [박재성] 숫자야구게임 미션 제출합니다.

안녕하세요.

이번 미션 생각보다 쉽지 않네요.

특히 기능을 분리하고 기능 단위로 commit하는 것이 쉽지 않다는 것을 느꼈어요.

기능 단위로 분리하고 commit하기 위해 이런 이런 과정으로 연습했어요.

하지만 기능을 분리하고 구현했더니 더 명확하게 구현할 수 있다는 것을 느낄 수 있는 기회가 된 것 같아요.

\* **Email 주소:** edu.nextstep@gmail.com

\* **Pull Request URL:** <https://github.com/next-step/java-baseball-precourse/pull/1>

# 숫자 야구 게임

# 기능 요구사항

- 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 포볼 또는 낫싱이란 힌트를 얻고, 그 힌트를 이용해서 먼저 상대방(컴퓨터)의 수를 맞추면 승리한다.
  - [예] 상대방(컴퓨터)의 수가 425일 때, 123을 제시한 경우 : 1스트라이크, 456을 제시한 경우 : 1스트라이크 1볼, 789를 제시한 경우 : 낫싱
- 위 숫자 야구게임에서 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 [ERROR]로 시작하는 에러 메시지를 출력하고 게임을 계속 진행할 수 있어야 한다.

# 프로그램 실행 결과

숫자를 입력해주세요 : 123

1스트라이크 1볼

숫자를 입력해주세요 : 145

1볼

숫자를 입력해주세요 : 671

2볼

숫자를 입력해주세요 : 216

1스트라이크

숫자를 입력해주세요 : 713

3스트라이크

3개의 숫자를 모두 맞히셨습니다! 게임 끝

게임을 새로 시작하려면 1, 종료하려면 2를 입력하세요.

1

숫자를 입력해주세요 : 123

1볼

...



# 프로그래밍 요구사항1 - 제약사항

- 숫자 야구게임을 실행하는 시작점은 src/main/java 폴더의 baseball.Application의 main()이다.
- 숫자 야구게임은 JDK 8 버전에서 실행가능해야 한다. JDK 8에서 정상 동작하지 않을 경우 0점 처리한다.
- JDK에서 기본 제공하는 Random, Scanner API 대신 nextstep.utils 패키지에서 제공하는 Randoms, Console API를 활용해 구현해야 한다.
  - Random 값 추출은 nextstep.utils.Randoms의 pickNumberInRange()를 활용한다.
  - 사용자가 입력하는 값은 nextstep.utils.Console의 readLine()을 활용한다.
- 프로그램 구현을 완료했을 때 src/test/java 폴더의 baseball.ApplicationTest에 있는 2개의 Test Case가 성공해야 한다.
  - ApplicationTest에서 제공하는 2개의 Test Case는 숫자 야구게임 구현을 위한 최소한의 Test Case이다.
  - 필수 요구사항은 아니지만 제공하는 소스 코드를 참고해 숫자 야구게임을 위한 모든 Test Case를 추가해 보는 것도 테스트에 대한 좋은 연습이 될 수 있다.

# 프로그래밍 요구사항2 - 제약사항

- 자바 코드 컨벤션을 지키면서 프로그래밍한다.
  - <https://naver.github.io/hackday-conventions-java/>
- indent(인덴트, 들여쓰기) depth를 2가 넘지 않도록 구현한다. 1까지만 허용한다.
  - 예를 들어 while문 안에 if문이 있으면 들여쓰기는 2이다.
  - 힌트: indent(인덴트, 들여쓰기) depth를 줄이는 좋은 방법은 함수(또는 메소드)를 분리하면 된다.
- 자바 8에 추가된 stream api를 사용하지 않고 구현해야 한다. 단, 람다는 사용 가능하다.
- else 예약어를 쓰지 않는다.
  - 힌트: if 조건절에서 값을 return하는 방식으로 구현하면 else를 사용하지 않아도 된다.
  - else를 쓰지 말라고 하니 switch/case로 구현하는 경우가 있는데 switch/case도 허용하지 않는다.
- 함수(또는 메소드)의 길이가 10라인을 넘어가지 않도록 구현한다.
  - 함수(또는 메소드)가 한 가지 일만 잘 하도록 구현한다.

# 프로그래밍 요구사항3 - 단위 테스트

- 도메인 로직에 단위 테스트를 구현해야 한다. 단, UI(System.out, System.in, Scanner) 로직은 제외
  - 핵심 로직을 구현하는 코드와 UI를 담당하는 로직을 분리해 구현한다.
  - 힌트는 MVC 패턴 기반으로 구현한 후 View, Controller를 제외한 Model에 대한 단위 테스트를 추가하는 것에 집중한다.
- JUnit5와 AssertJ 사용법에 익숙하지 않은 개발자는 첨부한 "학습테스트를 통해 JUnit 학습하기.pdf" 문서를 참고해 사용법을 학습한 후 JUnit5 기반 단위 테스트를 구현한다.

# 과제 진행 요구사항

- 미션은 <https://github.com/next-step/java-baseball-precourse> 저장소를 fork/clone해 시작한다.
- 기능을 구현하기 전에 java-baseball-precourse/README.md 파일에 구현할 기능 목록을 정리해 추가한다.
- git의 commit 단위는 앞 단계에서 README.md 파일에 정리한 기능 목록 단위 또는 의미있는 단위로 Commit한다.
  - [AngularJS Commit Message Conventions](#) 참고해 commit log를 남기려고 노력해 본다.
- 과제 진행 및 제출 방법은 [프리코스 과제 제출](#) 문서를 참고한다.

# 미션 마감 및 기준

- 2021년 10월 6일(수) 23시 59분까지 GitHub을 통한 미션 제출과 메일까지 완료해야 한다.
  - 2021년 10월 7일(목) 00시 이후 추가 push도 허용하지 않는다.
- 2021년 10월 7일(목) 00시 이후 제출한 경우 미션을 제출하지 않은 것으로 한다.