객체지향설계및프로그래밍 실습과제#2

- 아마도 우리 실습 마지막 기본 복습 과제 –

# 0. 과제를 시작하기 전에

이번 과제부터 몇 가지 기본 규칙들이 추가됩니다:

* 각 부분과제에는 60, 30, 10점과 같이 점수가 매겨져 있으며,  
  과제점수는 여러분이 수행한 모든 부분과제 득점의 총합입니다.
  + 마지막 과제는 ‘나 이 과제 할게요!’라고 선언한 학생들만 할 수 있습니다.
* 부분과제들 중 하나 이상을 성공한 경우 집에 갈 수 있습니다.
* (중요)여러분은 반드시 1, 2, 3번과 같은 순서대로 과제를 수행해야 합니다.
* 과제 검사(채점)는 언제나 여러분이 수행한 가장 어려운 부분과제에 대해서만 수행됩니다.

이러한 기본 규칙들에 대한 자세한 내용은 <http://fb.com/OODP.TA> 게시물을 참고해 주세요.

# 내 Eclipse를 ‘내 Eclipse’로 만들기 (60점)

별도의 실습보조자료 문서를 참고하면서 Eclipse가 제공하는 다양한 커스터마이즈 설정들을 직접 다루어 보고, Ctrl + Shift + F를 누를 때마다 ‘내 코드’를 만들어 주는 기특한 Eclipse를 만들어 봅시다.

## 과제 수행 방법

* 여러분이 만족할 수 있을 만큼 Eclipse의 각 설정들을 건드려 보세요.
* 아직 모르는 Java 개념에 대해서는, 가급적이면 일단 원래 설정을 유지해 두세요.
* 뭔가 바뀌긴 하는데 무슨 의미인지 궁금해서 터질 것 같다면, 조교에게 설명을 요청해 주세요.
* 설정을 다 마친 다음 적당한 테스트를 통해 내 마음에 잘 드는지 한 번 확인해 보세요.
* 집에 있는 Eclipse에서도 이 설정을 쓰고 싶다면 문서에 설명된 export 기능을 사용하세요.

## 과제 수행 조건

* 이 부분과제의 성공 여부는 여러분 스스로 판단하며, 조교는 여러분의 그런 선택을 존중합니다. 따라서, 이 부분과제는 조교에게 검사받지 않으며, 만약 자신이 이것만 하고 집에 가고 싶은 경우 느낀 점을 제출하고 바로 집에 가도 좋습니다.

# 배열에 임의의 수를 채우고 몇 가지 연산을 수행하는 프로그램 (30점)

이번 과제는 여러분의 기본 구문 작성 능력을 한 번 더 확인하면서, 이후 더 재미있는 프로그램을 만들기 위해 필요한 몇 가지 기능들의 사용법을 살펴 봅니다.

## 필요한 것

* Java에서 배열을 사용하는 방법
* Java에서 pseudo-random 값을 사용하는 방법

## 과제 수행 방법

* 중간 휴식시간 이후, 조교와 함께 프로그램의 기본적인 골격을 함께 만들어 볼 예정입니다.
  + 이번 내용은 따라오기 쉽지 않으므로 별도 문서에 세세한 진행 순서를 기록해 두었습니다.
* 조교와의 즐거운 시간이 끝나면, 여러분이 원하는 몇 가지 연산을 더 추가해 본 다음, 실행했을 때 자신이 의도한 대로 동작하는지 한 번 확인해 보도록 합시다.

## 과제 수행 조건

* 최소 두 개 이상의 배열 연산 메서드를 추가해 보아야 합니다.
* 잘 동작하는 것이 확인되면 조교에게 검사를 요청해 주세요.

## 주의사항

* 여러분이 원하는 연산들 중 일부는 아직 만들기 어려울 수 있습니다. 막히는 경우 조교의 도움을 요청해 주세요.

# ‘모든 경우의 수’에 대해 혼자 숫자 야구를 수행하고, 자신의 평균 성능(맞추는데 걸린 이닝 수)을 출력하는 프로그램 (10점)

우리 실습의 첫 번째 도전과제는, 뭔가 강렬한 실습을 원하는 여러분 개개인이 현재 어떤 코드 작성 습관을 보유하고 있는지 알아보기 위해 출제되었습니다(그래서 이름도 두 줄이나 됩니다). 적혀 있는 대로, 123 ~ 987까지의 모든 경우의 수에 대해 혼자 숫자 야구를 수행(답 추측🡪공 던지기🡪S:B:O판정 과정을 혼자 반복)하고 평균 이닝 수를 집계하여 마지막에 출력하는 프로그램을 만들어 보도록 합시다.

## 필요한 것

* static 필드 사용 방법
  + ‘static 메서드’ 만들 때처럼 메서드 밖 & Program class 안에 변수 선언을 수행하고 static을 붙여 주면 됩니다. 필드 초기화는 일단 main() 최상단에서 수행해 줍시다.
* 다차원 배열 사용 방법
  + 소소한 문법 차이만 빼면, Java의 다차원 배열은 C의 것과 유사하게 쓸 수 있습니다.

## 과제 수행 방법

* 아래 메서드들을 구현해서 활용해 보도록 합시다:
  + boolean IsValid(int[] arg)
    - 주어진 3개짜리 배열이 1~9 사이의 겹치지 않는 수로 이루어져 있는지 판단합니다.
  + int[] NextBase(int[] previousBase)
    - 다음 게임에서 맞춰야 할 base 배열(예: 이전에 123이었다면 124)을 반환합니다. Base 배열을 static 필드로 만들어 둔 경우 단순히 해당 필드 값을 바꿔 주어도 되겠군요.
  + void ResetData()
    - Base를 맞추기 위해서는 몇 가지 추가 데이터가 필요합니다. 이 메서드를 구현해서 매 게임마다 데이터 필드들을 싹 초기화해서 혹시 생길지 모르는 오류를 방지해 봅시다.
  + int[] Pitch(추가 데이터들)
    - Base를 맞추기 위한 새로운 수열을 만들어 반환합니다. 추가 데이터는 argument로 넘겨 주거나 아니면 그냥 static 필드로 놓고 쓰면 되겠군요.
  + int[] Catch(int[] pitch, 추가 데이터들)
    - Pitch()의 호출 결과를 입력받아 base와 비교하여 판정 결과(S:B:O)를 3개짜리 배열로 반환해 주는 메서드입니다.
* 위의 메서드들은 예시일 뿐이며, 사실 성능 측면에서 그리 좋은 편은 아닙니다. 여러분이 생각하는 더 적절한 방법이 있는 경우 얼마든지 해당 방법을 사용해도 좋습니다.

## 과제 수행 조건

* 여러분이 작성한 프로그램은 다음 조건을 모두 만족해야 합니다:
  + ‘집에 빨리 가고 싶어서’ 대강 만들지 않고,  
    여러분 본연의 능력을 충분히 발휘한 결과물이어야 합니다.
  + 당연하겠지만, base를 안 보고 정공법으로 맞추어야 합니다.  
    비슷한 이유로, 매 게임은 독립적으로 수행되어야 합니다.  
    (‘이전에 123이 나왔으니 이제 123은 안 나온다’같은 사실을 이용하면 안 됩니다)
  + 게임이 다 끝나면 평균 이닝 수를 소수 3자리 이상 집계하여 출력해야 합니다.
* 다른 과제들과 달리, 위에 나열된 각 항목에 대한 만족 여부 판단은 조교가 수행합니다.  
  겁먹지는 말고, 자신이 과제를 얼추 끝냈다 싶으면 조교를 불러 토론을 진행해 봅시다.

## 주의사항

* 과제 설명서의 첫 페이지에서 언급했듯, 이 부분과제를 수행하기 전에 이전 과제들을 모두 마무리해 두는 것이 좋습니다.
* 이 과제는 여러분의 습관을 묻는 것이지 여러분의 실력을 묻는 것이 아닙니다. 그러니 자신이 생각하고 있는 아이디어를 Java 코드로 옮기기 어려운 경우 망설이지 말고 조교에게 문의해 주세요.