

## Programming Assignment 1

- Template: Assignment01
- 업로드: 각자의 프로젝트 폴더 (기존의 Self-Test 업로드 방식과 동일)
- Assignment01\_pt1, Assignment01\_pt2 두 개의 폴더 제출
- Commit Message: Assignment01 (git commit -m "Assignment01")
- 제출 기한: 5/4(화) 23시 59분까지

교수가 과제를 생성하고 학생들이 과제의 답을 작성하고 다시 교수가 과제의 점수를 산출하는 일련의 과정이 이루어지고 있다. 이번 과제에서 우리는 이러한 과정을 클래스로 작성하고 전체적인 구조를 작성할 것이다. 클래스는 총 7개로, Professor, Student, Assignment, Answer, GradeList, Git, GradeNode 클래스가 존재하고 Part 1과 Part 2로 나누어서 작성을 할 것이다.

### PART 1

- 과제에 대한 정보를 저장하는 **Assignment** 클래스에는 **instance variable**로 **subject**(과목명), **question**(과제 내용)이 있다. 이에 관련된 모든 메소드 및 생성자를 작성할 것

- 모든 변수를 'None'으로 초기화하는 기본 생성자 작성
- 모든 변수를 인자로 받은 값으로 초기화하는 생성자 작성
- subject, question의 값을 가져오는 accessor 메소드 작성
- subject, question의 값을 설정하는 mutator 메소드 작성
- 인자로 받은 객체와 일치하는지 확인하는 equals() 메소드 작성
- subject, question의 값을 출력하는 toString() 메소드 오버라이딩  
(출력 형태는 최종 출력 형태 참고)

- 과제의 답에 대한 정보를 저장하는 **Answer** 클래스에는 **instance variable**로 **solution**(과제에 대한 답)이 있다. 이에 관련된 모든 메소드 및 생성자를 작성할 것

- 모든 변수를 'None'으로 초기화하는 기본 생성자 작성
- 모든 변수를 인자로 받은 값으로 설정하는 생성자 작성
- solution에 대한 accessor, mutator method 작성
- 인자로 받은 객체와 일치하는지 확인하는 equals() 메소드 작성

- solution의 값을 출력하는 toString() 메소드 오버라이딩  
(출력 형태는 최종 출력 형태 참고)

- 과제를 생성하고 점수를 산출하는 **Professor** 클래스에는 **instance variable**로 **SCORE**(과제 점수)가 있다. 이에 관련된 모든 메소드를 작성할 것

- subject, question의 값을 인자로 받아서 Assignment 클래스의 객체를 리턴하는 makeAssignment() 메소드 작성
- 0과 10사이의 난수를 통해 받아온 랜덤 SCORE 변수 값을 String 타입으로 리턴하는 grading() 메소드 작성 (인자로 Assignment 객체와 Answer 객체를 받지만 사용은 하지 않음)

- 과제의 답을 작성하는 **Student** 클래스에는 **instance variable**로 **studentName**(학생 이름), **studentId**(학번)가 있다. 이에 관련된 모든 메소드 및 생성자를 작성할 것

- 모든 변수를 인자로 받은 값으로 초기화하는 생성자 작성
- studentName, studentId의 값을 가져오는 accessor 메소드 작성
- studentName, studentId의 값을 초기화하는 mutator 메소드 작성
- 인자로 받은 Assignment 객체와 solution의 값 중 solution의 값만 사용하여 생성한 Answer 객체를 리턴하는 makeAnswer() 메소드 작성
- studentName, studentId의 값을 출력하는 toString() 메소드 오버라이딩  
(출력 형태는 최종 출력 형태 참고)

- 과제에 대한 모든 정보를 저장하는 **GradeNode** 클래스에는 **assignment**(과제), **answer**(과제에 대한 답), **score**(과제 점수)가 있다. 이에 관련된 모든 메소드 및 생성자를 작성할 것

- 모든 변수를 null로 초기화하는 기본 생성자 작성
- nextNode를 제외한 모든 변수를 인자로 받은 값으로 초기화하는 생성자 작성  
(단, 인자로 객체를 받으면 각 객체의 모든 변수를 전달해야 함)
- score에 대한 accessor, mutator method 작성
- 인자로 받은 객체와 일치하는지 확인하는 equals() 메소드 작성
- nextNode를 제외한 모든 변수의 값을 출력하는 toString() 메소드 오버라이딩  
(변수가 class 타입일 경우 해당 class의 toString() 메소드 호출)

- 과제 생성, 과제의 답 생성, 점수 산출 및 모든 정보 저장이 이뤄질 **Git** 클래스에는 **assignmentNum**(과제 수), **studentNum**(학생 수), **repository**(**GradeNode** 객체를 담은 2차원 배열) 및 **main()** 메소드가 있다. 이에 관련된 모든 코드를 작성할 것

- repository 2차원 배열의 원소에 **GradeNode** 객체를 생성하는 생성자 작성
- Assignment 객체, Answer 객체, score 값을 인자로 받고 받은 모든 인자로 생성한 **GradeNode** 객체를 리턴하는 **pushGradeNode()** 메소드 작성

◆ main() 메소드는 다음과 같이 동작함

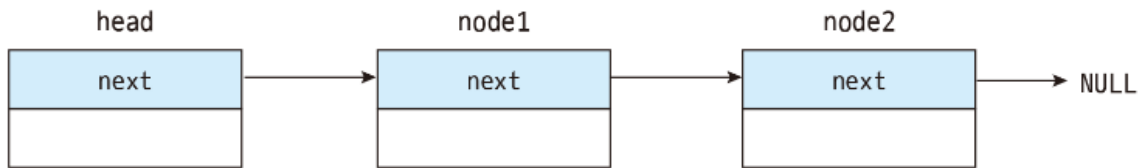
- ✓ **makeAssignment()** 메소드 호출로 생성한 Assignment 객체를 assignment 배열에 삽입
- ✓ **makeAnswer()** 메소드 호출로 생성한 Answer 객체를 answer 배열에 삽입
- ✓ **grading** 메소드 호출로 저장한 solution 문자열을 score 배열에 삽입
- ✓ **pushGradeNode()** 메소드 호출로 생성한 **GradeNode** 객체를 repository 2차원 배열에 삽입
- ✓ repository 2차원 배열 출력
- ✓ **showRepository()** 메소드를 통한 repository 출력은 다음과 같다 (Score는 다를 수 있음)

```
Solution:          1          2          3          4
Score:             C          F          C          A0

Solution:          Speed          m/s          I have no idea          Running
Score:             C0             F             C0             D

Solution:          Computer          Apple          Iphone          Nothing
Score:             B0             AA             B0             B0
```

## PART 2



연결 리스트(linked list)란 각 노드가 데이터와 포인터를 가지고 한 줄로 연결되어 있는 방식으로 데이터를 저장하는 자료 구조이다. 이름에서 말하듯이 데이터를 담고 있는 노드들이 연결되어 있는데, 노드의 포인터가 다음이나 이전의 노드와의 연결을 담당하게 된다. C언어의 연결리스트에서 포인터를 java에서는 참조 변수로 구현한다. 따라서 Part 2에서는 참조 변수를 사용하여 연결 리스트를 구현할 것이다.

- **GradeNode** 클래스에 **nextNode**(연결리스트로 연결된 다음 노드)가 추가됨

- 과제에 대한 모든 정보를 저장하는 **GradeNode** 클래스의 객체로 연결리스트를 생성할

**GradeList** 클래스에는 **head**(연결리스트의 첫 노드)가 있다. 이에 관련된 모든 메소드 및 생성자를 작성할 것

- head 노드를 null로 초기화하는 기본 생성자 작성
- 인자로 받은 gradeList 연결리스트를 복사하는 Copy Constructor 생성자 작성
- 인자로 받은 변수 및 객체들로 생성한 노드를 연결리스트에 추가하는 메소드 작성
- GradeList 연결리스트 출력 메소드 작성

◆ main() 메소드는 다음과 같이 동작함

(Part1 표시가 있는 부분은 Part1의 main() 코드 복사해서 사용)

- ✓ 학생 수만큼 원소를 가지는 GradeList 객체 배열 생성
- ✓ addNode() 메소드 호출로 생성한 GradeNode 객체를 gradeList 클래스에 연결리스트로 삽입
- ✓ 생성된 연결리스트를 첫 노드부터 마지막 노드까지 모든 정보를 출력
- ✓ 기존에 생성된 gradeList 전부 복사 후 출력

- ✓ main() 메소드에서 gradeList와 gradeList를 복사한 copyList의 출력 결과는 다음과 같다  
(왼쪽이 gradeList, 오른쪽이 copyList)

```
Student Name: Kevin, Student Id: 2021104233
=====
Subject: Mathematics, Question: 1 + 1 = ?
Solution: 1
Grade: C
Subject: Physics, Question: What is Velocity?
Solution: Speed
Grade: C0
Subject: Philosophy, Question: What is this?
Solution: Computer
Grade: B0

Student Name: Judy, Student Id: 2002548215
=====
Subject: Mathematics, Question: 1 + 1 = ?
Solution: 2
Grade: F
Subject: Physics, Question: What is Velocity?
Solution: m/s
Grade: F
Subject: Philosophy, Question: What is this?
Solution: Apple
Grade: AA

Student Name: Charlie, Student Id: 1997487865
=====
Subject: Mathematics, Question: 1 + 1 = ?
Solution: 3
Grade: C
Subject: Physics, Question: What is Velocity?
Solution: I have no idea
Grade: C0
Subject: Philosophy, Question: What is this?
Solution: Iphone
Grade: B0

Student Name: Puth, Student Id: 15048655
=====
Subject: Mathematics, Question: 1 + 1 = ?
Solution: 4
Grade: A0
Subject: Physics, Question: What is Velocity?
Solution: Running
Grade: D
Subject: Philosophy, Question: What is this?
Solution: Nothing
Grade: B0
```

```
=====GradeList copied=====
Subject: Mathematics, Question: 1 + 1 = ?
Solution: 1
Grade: C
Subject: Physics, Question: What is Velocity?
Solution: Speed
Grade: C0
Subject: Philosophy, Question: What is this?
Solution: Computer
Grade: B0

Subject: Mathematics, Question: 1 + 1 = ?
Solution: 2
Grade: F
Subject: Physics, Question: What is Velocity?
Solution: m/s
Grade: F
Subject: Philosophy, Question: What is this?
Solution: Apple
Grade: AA

Subject: Mathematics, Question: 1 + 1 = ?
Solution: 3
Grade: C
Subject: Physics, Question: What is Velocity?
Solution: I have no idea
Grade: C0
Subject: Philosophy, Question: What is this?
Solution: Iphone
Grade: B0

Subject: Mathematics, Question: 1 + 1 = ?
Solution: 4
Grade: A0
Subject: Physics, Question: What is Velocity?
Solution: Running
Grade: D
Subject: Philosophy, Question: What is this?
Solution: Nothing
Grade: B0
```

- ✓ main() 메소드에서 gradeList의 head 노드들과 복사된 copyList의 head 노드들이 일치하는지 확인하는 출력 결과는 다음과 같다

```
Are the head nodes properly copied?
true
```