

웹파이선프로그래밍 기말고사 ('22년 2학기)

■ 시험 방법

- ① 답안 제출용 화일을 MS Visual Code를 사용하여 학번.py로 만듦
- ② 시험 문제에 대한 답안으로 개발하는 클래스들을 학번.py 화일 안의 내용으로 작성함
- ③ 시험 종료 전, 이캠퍼스의 기말 고사 메뉴를 통해서 학번.py 화일을 제출함

■ 시험 주의 사항

- ① 답안 화일은 시험 종료 시간 전에 반드시 제출되어야 함
 - A. 시험 종료 시간을 넘겨서 제출된 답안은 교수의 재량에 따라 0점 혹은 대규모 감점 처리 함
- ② 답안 화일 이름의 규칙 위반시 교수의 재량으로 감점 처리 함
 - A. 답안 화일을 이캠퍼스에서 여러 번 제출하는 경우, 자동으로 화일 이름 뒤에 숫자가 붙으며, 이로 인한 감점은 없음. 여러 번 제출한 경우 마지막 화일을 채점함
- ③ 다음의 경우 채점하지 않음
 - A. 답안 화일은 py 화일이여야 함 (ipynb 화일 제출시 채점하지 않음)
 - B. 답안 화일을 실행하였을 때, 무한 loop 혹은 input() 등으로 실행이 멈추면 채점하지 않음
 - C. 답안 화일을 제출한 그대로 실행하였을 때, 예러 발생으로 실행이 중단되면 채점하지 않음
 - 반드시 답안 제출 전, Visual Code에서 "Run and Debug"를 실행해서, 답안 프로그램이 중단 혹은 에러 없이, 제대로 시작부터 끝까지 동작하는지 확인함
 - D. 각 문항 별로 분리하여 채점하지 않음. "모든 문항의 기능이 포함된 프로그램을 하나로 보고", 각각의 문항이 제대로 동작하는 경우에 대해서 배점하는 방식으로 채점함
- ④ 클래스/메소드 이름에서 오타자 혹은 대소문자가 틀린 경우 부분 점수 반영 없음
- ⑤ 전역 변수 사용은 문제에서 요구하지 않는 한, 사용을 금지함
- ⑥ 문제별로 부분 점수 없음 (문항의 배점이 3점이면, 3점 혹은 0점임)
- ⑦ 유사도 검사 프로그램의 실행 결과 기준, 유사도가 높은 학생들은 재시험을 실시하며, 재시험시 컷닝 혹은 대리 작성으로 판단되면, 학교 교칙에 의거 전원 징계 처리 함
- ⑧ 문제에서 언급하지 않는 이상, 파이썬 기본 기능 외 3rd-party 라이브러리 사용을 금지함
 - A. Python 표준 라이브러리 이외의 라이브러리는 사용을 금지한다는 의미임
 - B. 본인의 Visual Code의 설정으로, 본인이 입력하지 않은 import가 포함되는 경우도, 평가용 컴퓨터에서 실행이 안되면 0점 처리함
- ⑨ 채점은 요구한 Return 값이 제대로 만들어지는 지에 대해서만 평가함

■ 시험 문제

[문제 1] 대부분의 문제에 대한 채점에서 활용되며, 오류 발생시 사실상 점수 획득이 어려움

- 클래스: MyClass
- 메소드: 생성자
- 입력 파라미터: 식별자 (문자열)
- 참조:
 - ✓ 채점과정에서, 입력 파라미터는 문자열만 전달할 것임
 - ✓ 객체는 입력 파라미터로 받은 식별자를 내부적으로 저장함

[문제 2] 다른 문제의 채점과정에서 사용됨

- 클래스: MyClass
- 메소드 이름: getId
- 입력 파라미터: 없음
- Return 값: 생성자를 사용해서 입력 받아 객체 내부에서 보관 중인 식별자 (문자열)

[문제 3]

- 클래스: MyClass
- 메소드 이름: setId
- 입력 파라미터: 식별자 (정수, 실수, 혹은 문자열)
- Return 값: 없음
- 참조:
 - ✓ 채점과정에서, 입력 파라미터는 정수, 실수 혹은 문자열만 전달할 것임
 - ✓ 객체는, 문제 1과 같은 용도로, 입력 파라미터로 받은 식별자를 내부적으로 저장함
 - ✓ 단, 입력 파라미터의 타입이 정수 혹은 실수인 경우는 식별자를 "XXXX"로 저장함
 - ✓ 채점과정에서, 문제2의 메소드를 사용할 예정임

[문제 4]

- 클래스: MyClass
- 메소드 이름: getNumberOfObject
- 입력 파라미터: 없음
- Return 값: MyClass 클래스를 사용해서 만든 객체(object)의 누적 개수

[문제 5] 다른 문제의 채점과정에서 사용됨

- 클래스: MyClass
- 메소드 이름: storeWordlistAsDictionary
- 입력 파라미터: 리스트 (문자열)
- Return 값: 딕셔너리 (key는 문자열, value는 정수)
- 참조:
 - ✓ 채점과정에서, 입력 파라미터는 문자열 리스트만 전달할 것임
 - ✓ 메소드는 작업 시작 전, 리스트를 작은 값부터 큰 값으로 나열되도록 정렬함
 - ✓ 메소드는 입력 파라미터로 받은 문자열, 그리고 문자열을 입력 받은 회수를 딕셔너리 타입을 사용해서 내부적으로 보관함
 - ✓ 메소드가 실행되면, 새롭게 전달받은 입력 파라미터의 리스트 문자열을 사용하여, 내부적으로 보관중인 딕셔너리를 추가적으로 업데이트 함
 - ✓ Return 값은 보관 중인 모든 "문자열"과 "문자열을 입력 받은 회수" 임

[문제 6] 다른 문제의 채점과정에서 사용됨

- 클래스: MyClass

- 메소드 이름: `getWordCount`
- 입력 파라미터: 문자열
- Return 값: 튜플 (문자열 하나와, 정수 하나로 구성) 혹은 `False`
- 참조:
 - ✓ 채점과정에서, 입력 파라미터는 문자열 만 전달할 것임
 - ✓ 메소드는 내부적으로 저장하고 있는 딕셔너리에 대해서, 입력 파라미터가 key인 경우가 존재하면, 해당 key와 key에 대한 value(입력 받은 회수)를 튜플로 return 함
 - ✓ Return 값의 예제는 ('name', 1)과 같은 형태임
 - ✓ 단, 입력 파라미터로 받은 문자열이 key로 존재하지 않으면, `False`를 return함

[문제 7] 다른 문제의 채점과정에서 사용됨

- 클래스: `MyClass`
- 메소드 이름: `getWordList`
- 입력 파라미터: 없음
- Return 값: 리스트 (문자열) 혹은 `False`
- 참조:
 - ✓ 메소드는 내부적으로 저장하고 있는 딕셔너리의 key 값으로만 이루어진 문자열 리스트를, 작은 값부터 큰 값의 순서로 정렬한 형태로 return 함
 - ✓ 만약 내부적으로 저장하고 있는 딕셔너리가 비어 있으면, `False`를 return함

[문제 8]

- 클래스: `MyDerivedClass`
- Base 클래스: `MyClass`
- 메소드 이름: 메소드 이름을 아는 지의 여부도 문제의 일부임
- Return 값: 문자열
- 참조:
 - ✓ 객체가 `str()` 혹은 `print()`처럼, 문자열로 변환되어야 하는 경우에 대한 메소드를 작성함
 - ✓ 메소드는 내부적으로 저장하고 있는 딕셔너리의 key 값으로만 만들어진 리스트를 return함
 - ✓ Return 값의 예제는 "<key1,key2>"이며, 이 경우 key1/key2는 key 값에 해당하는 문자열을 의미함
 - ✓ 내부적으로 저장하고 있는 딕셔너리가 비어 있다면, return 값은 "<>"의 문자열임
 - ✓ 예제와 동일한 형태여야 하며, 공백(space) 등을 포함하면 오답처리함

[문제 9]

- 클래스: `MyDerivedClass`
- Base 클래스: `MyClass`
- 메소드 이름: 메소드 이름을 아는 지의 여부도 문제의 일부임
- 입력 파라미터: `MyDerivedClass` 타입의 객체
- Return 값: `True` 혹은 `False`

- 참조:
 - ✓ 채점과정에서, 입력 파라미터는 MyDerivedClass 타입의 객체 만 전달할 것임
 - ✓ 메소드는 > 연산자 사용시 호출됨
 - ✓ 메소드는 객체가 저장하는 딕셔너리의 key 개수가 입력 파라미터로 받은 객체의 딕셔너리에 저장된 key 개수보다 크면 True, 그렇지 않으면 False를 return함

[문제 10]

- 클래스: MyDerivedClass
- Base 클래스: MyClass
- 메소드 이름: 메소드 이름을 아는 지의 여부도 문제의 일부임
- 입력 파라미터: MyDerivedClass 타입의 객체
- Return 값: MyDerivedClass 타입의 객체
- 참조:
 - ✓ 채점과정에서, 입력 파라미터는 MyDerivedClass 타입의 객체 만 전달할 것임
 - ✓ 메소드는 + 연산자 사용시 호출됨
 - ✓ 메소드는 내부적으로 저장하고 있는 딕셔너리 정보와 입력 파라미터로 전달받은 객체의 딕셔너리 정보를 통합한 결과를 return함
 - ✓ 통합과정에서, 이미 보관중인 key 값과 동일한 key 값을 입력 파라미터의 객체에서 발견하면, 이미 보관중인 key 값의 value에 입력 파라미터로 전달받은 객체의 value를 더하는 방식으로 처리함
 - ✓ 통합과정에서, 기존 객체들의 정보는 변경되지 않으며, 통합된 결과는 새로운 MyDerivedClass 타입의 객체에 저장하여 return 함

[문제 11]

- 클래스: MyDerivedClass
- Base 클래스: MyClass
- 메소드: 생성자
- 입력 파라미터: 식별자 (문자열) 혹은 없음
- 참조:
 - ✓ 채점과정에서, 입력 파라미터는 없거나 혹은 문자열 만 전달할 것임
 - ✓ 객체는, 문제 1과 같은 용도로, 입력 파라미터로 받은 식별자를 내부적으로 저장함
 - ✓ 생성자는 식별자를 저장하는 과정에서 Base 클래스의 생성자를 활용함
 - ✓ 만약, 입력 파라미터가 없는 경우는, 식별자를 “****” 로 저장함