

## 웹파이선프로그래밍 2020년 동계계절학기 기말고사 (이성원 교수)

### ■ 일반 주의사항

- ① **미리 배포한 시험 가이드라인에서, 동영상 촬영 종료 시점을 수업 종료 시점으로 변경함**
  - 시험 답안을 수업 종료 시간 전에 미리 제출한 사람도, 수업 종료 시간(낮12시50분)까지 스마트폰을 통한 시험 장면을 촬영하고, 수업 종료 시간에 촬영을 종료함
  - 미리 답안을 제출한 사람은 답안이 맞는지 재검 하도록 하고, 인터넷 접속 등 행위는 금지함
- ② 기업체의 일반적인 비대면 실기 시험에 준하는 원칙으로 진행함
  - 시험 시간 중 시험 문제에 대한 질의응답은 없음
  - 스마트폰은 비행모드로 변경하여, 시험 중 전화/문자 등이 오지 않도록 함
  - 문제에서 요구하는 사항 만 평가하므로, 문제에서 요구한 부분에만 충실하게 답변을 작성함
  - 문제 외에 대한 부분은, 평가에 일체 반영하지 않으며, 본인 판단에 따라 실시함
  - **시험 중, 본인이 아무리 생각해도 위급 상황이라고, 생각되면 010-9530-3622로 전화함**
- ③ 모든 변수와 코드는 Class 혹은 Function/Method 안에 있어야 함
  - 전역 변수(global variable)와 전역 지역(Class/Function/Method 밖)의 코드는 없도록 함
  - 필요한 module을 포함하기 위한 import 구문은 예외로 인정함
- ④ Class의 member에서, self는 별도로 언급하지 않으니, 알아서 필요시 추가함
- ⑤ 제출한 상태에서 바로 실행하였을 때, 에러가 나서 동작을 안하는 답안은 채점 안함
- ⑥ 채점은 채점 소프트웨어가 수행하므로, 대소문자/철자가 정확하게 문제와 동일해야 함
- ⑦ ecampus 시스템의 문제 발생시, 이로 인한 불이익은 없을 테니, 걱정하지 말 것. 제출 시간이 넘으면 바로, 담당교수 이메일( drsungwon@khu.ac.kr )로 제출함

### ■ 답안 제출 관련 주의사항

- ① 답변서를 제대로 된 "학번.py" 파일로 제출함 (예: 2020123456.py)
  - Jupyter Notebook의 ipynb 파일을, 확장자만 py로 바꾼 경우, 원칙적으로 0점 처리함
- ② **시험 종료 시간인 오후2시50분 이전에 ecampus로 답안의 업로드를 마칩**
  - 시험 종료 시간 안에 ecampus로 제출을 하지 못하면, 즉시 이메일로 답안을 제출함
  - 지연 제출로 인한, 이메일 답안 제출은 원칙적으로 0점 처리 대상임
  - 교수 재량으로 지연 시간과 사유 등을 고려하여 큰 감점 후 채점 가능함 (1:1 면담 실시)

### ■ 영상 제출 관련 주의사항

- ① 시험 촬영 영상을 정해진 용량으로 제출함
  - ecampus는 1GB 용량 제한이 있기에, 전체 시험 촬영 영상의 용량은 1GB 이하여야 함
  - 스마트폰 기능상, 복수 파일로 영상이 나뉘는 경우는, 파일 이름을 영상1, 영상2 등으로 순서에 맞춰서 변경한 후, 하나의 압축 파일로 만들어서 ecampus로 업로드 함
  - 복수의 영상 녹화 파일로 제출하는 경우에도, 전체 파일의 용량은 총1GB 이하여야 함
- ② **영상 제출 마감 시한인 시험 당일 오후11시59분 이전에 ecampus로 영상의 업로드를 마칩**
  - 제출 마감 시한 안에 ecampus로 제출을 하지 못하면, 즉시 이메일로 파일을 제출함
  - 이메일에는 담당 교수가 영상을 다운받을 수 있는 방법을 알려주는 정보를 포함해야 함
  - 정해진 시간과 용량으로 ecampus에 업로드 못하면, 원칙적으로 0점 처리 대상임
  - 교수 재량으로 사유 등을 고려하여 큰 감점 후 채점 가능함 (1:1 면담 실시)

■ 문제 1: 다음의 Class를 만듦 (50점/총100점. 문제별 5점. 8문항 모두 맞으면 10점 추가 부여)

- ① Class 이름: MySelfClass
  - A. Default constructor를 지원해야 함 (사용자가 별도로 입력 파라메타를 주지 않는 생성자)
- ② Member method (or function) 1: getStudentID( )
  - A. 입력 파라메타: 없음
  - B. Return 값: 본인 학번(10자리)의 문자열 하나 (10자리 보다 작으면, 끝에 '0'으로 채움)
- ③ Member method 2: getBirthday( )
  - A. 입력 파라메타: 없음
  - B. Return 값: 본인이 태어난 년도(4자리) + 월(2자리) + 일(2자리)의 문자열 하나  
(ㄱ) 예를 들어, 2001년 2월 3일은 "20010203" 임  
(ㄴ) 학교의 인포21(혹은 Portal) 시스템에 등록된 생년월일 이어야 함
- ④ Member method 3: getStudentIDByInteger( )
  - A. 입력 파라메타: 없음
  - B. Return 값: 본인의 학번(10자리)을 정수로 바꾼 숫자 하나
- ⑤ Member method 4: getBirthdayByIntegerSum( )
  - A. 입력 파라메타: 없음
  - B. Return 값: 정수 하나  
(ㄱ) getBirthday()의 return 값에서, 각 자리 글자를 정수로 환산 후, 모든 정수를 더한 값인 정수 하나  
(ㄴ) 예를 들어, getBirthday()의 return 값이 "20010203"이면, return 값은 정수 8임
- ⑥ Member method 5: getMaxInteger( )
  - A. 입력 파라메타: 없음
  - B. Return 값: 정수 하나  
(ㄱ) getStudentID( )와 getBirthday( )의 return 값으로 받은, 두개의 문자열에 속한 글자들을 각각 정수로 변환 했을 때, 가장 큰 수인 글자를 정수로 바꾼 숫자 하나  
(ㄴ) 예를 들어, getBirthday()의 return 값이 "20010203"이고, getStudentID()의 return 값이 "9900000029"이면, 이 method의 return 값은 문자 '9'를 정수로 바꾼 9임
- ⑦ Member method 6: getAverageInteger( )
  - A. 입력 파라메타: 없음
  - B. Return 값: 실수 하나  
(ㄱ) getStudentID( )와 getBirthday( )의 return 값으로 받은, 두개의 문자열에 속한 글자들을 각각 실수로 변환한 후, 모든 실수를 더한 값의 평균 값인 실수 하나  
(ㄴ) 예를 들어, getBirthday()의 return 값이 "20010203"이고, getStudentID()의 return 값이 "9900000029"이면, return 값은 실수 2.0555555... 임
- ⑧ Member method 7: getBirthdayByMonth( )
  - A. 입력 파라메타: 없음
  - B. Return 값: 본인이 태어난 월에 해당하는 문자열 하나  
(ㄱ) getBirthday( )의 return 값 중 '월'에 해당하는 값에 대해서, 다음 중 하나여야 함:  
JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC  
(ㄴ) 예를 들어, 2001년 2월 3일은 "FEB" 문자열 임

■ 문제 2: 다음의 Class를 만듬 (50점/총100점. 3문항 모두 맞으면 10점 추가 부여)

(A) Class 이름: MyRecordClass

- ① Default constructor 를 지원해야 함 (사용자가 별도로 입력 파라메타를 주지 않는 생성자)

(B) Member method (or function) 1: makeRecord()

{10 점/총 100 점}

- ① 입력 파라메타: 담당 학생 학번, 종료 기한 (Deadline), 수행 항목 번호, 우선 순위
- 담당 학생 학번: 문자열 (문제 1 과 동일하게 10 자리)
  - 종료 기한: 문자열, 년월일을 나타내는 8 글자 (예: 20200718), 편의상 모든 월은 31 일임
  - 수행 항목 번호: 문자열, 알파벳과 숫자로 이루어진 4 글자 (예: A001)
  - 우선 순위: 정수, 1/2/3 중 하나의 값
- ② Return 값: 다음 중 하나임
- List: 입력 값(입력 파라메타)로 받은 정보에 에러가 없는 경우, 입력 값을 순서대로 저장하여 만들어진, ["9900000029", "20210718", "A001", 1] 형태의 리스트
  - -1: 정수, 입력 값이 정해진 데이터 타입(문자열, 정수 등)이 아닌 경우
  - -2: 정수, 입력 값이 정해진 형태(10 자리, 8 자리, 4 자리 및 1~3 의 정수)가 아닌 경우
- ③ 추가 동작
- makeRecord()의 입력 값으로 받은 정보에 에러가 없는 경우, 객체는 모두 입력 받은 Record 정보를 저장하고 있어야 함
  - 입력 값들의 데이터 타입이 잘못 되었으면, 형태에 대한 조건 검사 없이 -1 을 return 함

(C) Member method 2: checkDelay()

{10 점/총 100 점}

- ① 입력 파라메타: 수행 항목 번호, 오늘 날짜
- 수행 항목 번호: makeRecord()의 수행 항목 번호와 동일 타입
  - 오늘 날짜: makeRecord()의 종료 기한과 동일 타입
- ② Return 값: 다음 중 하나임
- True: Boolean, 수행 항목 번호에 해당하는 Record 의 종료 기한이, 오늘 이전인 경우 (과거)
  - False: Boolean, 수행 항목 번호에 해당하는 Record 의 종료 기한이, 오늘 혹은 내일 이후인 경우 (현재 & 미래)
  - -1: 정수, 수행 항목 번호에 해당하는 Record 가 없는 경우

(D) Member method 3: deferDeadline()

{20 점/총 100 점}

- ① 입력 파라메타: 수행 항목 번호, 종료 기한 연장 날짜
- 수행 항목 번호: makeRecord()의 수행 항목 번호와 동일 타입
  - 종료 기한 연장 날짜: 정수, Deadline 을 연장할 날짜(day)의 갯수
- ② Return 값: 다음 중 하나임
- List: 수행 항목 번호에 대응하는 Record 의 종료 기한을 연장한 List. 예를 들어, ["9900000029", "20210718", "A001", 1]의 항목에 7 일을 연장하면, ["9900000029", "20210725", "A001", 1]임
  - -1: 정수, 수행 항목 번호에 해당하는 Record 가 없는 경우
  - -2: 정수, 입력 파라메타 중 연장할 날짜가 14 일 보다 큰 경우
- ③ 추가 동작
- 한 달은 총 31 일 이므로, 이에 맞춰서 년/월/일이 변경되어야 함

==== 수고 하셨습니다. 제출 전 주의사항을 잘 지켰는지, 다시 한번 확인 후 제출합니다 ====