

웹파이선프로그래밍 00분반 2022년 1학기 기말고사

[주의 사항]

- 답안 화일은 시험 종료 시간 전에 반드시 제출되어야 함
 - 시험 종료 시간을 넘겨서 제출된 답안은 교수의 재량에 따라 0점 혹은 대규모 감점 처리 함
- 답안 화일의 이름은 학번.py 이며, 규칙 위반시 교수의 재량으로 감점 처리 함
 - 답안 화일을 이캠퍼스에서 여러 번 제출하는 경우, 자동으로 화일 이름 뒤에 숫자가 붙으며, 이로 인한 감점은 없음. 여러 번 제출한 경우 마지막 화일을 채점함
- **다음의 경우 채점하지 않음**
 - 답안 화일은 py 화일이여야 함 (ipynb 화일 제출시 채점하지 않음)
 - 답안 화일을 실행하였을 때, 무한 loop 혹은 input() 등으로 실행이 멈추면 채점하지 않음
 - 답안 화일을 제출한 그대로 실행하였을 때, 에러 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
 - 특히, 각 문항 별로 분리하여 채점하지 않음. "모든 문항의 기능이 포함된 프로그램을 하나로 보고", 각각의 문항이 제대로 동작하는 경우에 대해서 배점하는 방식으로 채점함
- 함수 이름에서 오탈자 혹은 대소문자가 틀린 경우 부분 점수 반영 없음
- 전역 변수 사용은 문제에서 요구하지 않는 한, 사용을 금지함
- 문제별로 부분 점수 없음 (문항의 배점이 3점이면, 3점 혹은 0점임)
- 유사도 검사 프로그램의 실행 결과 기준, 유사도가 높은 학생들은 오프라인 (대면) 재시험을 실시 하며, 재시험시 컨닝 혹은 대리 작성으로 판단되면, 학교 교칙에 의거 전원 징계 처리 함
- 문제에서 언급하지 않는 이상, 파이썬 기본 기능 외 3rd-party 라이브러리 사용을 금지함
- 채점은 요구한 Return 값이 제대로 만들어지는 지에 대해서만 평가함
- **다음의 사항에 특히 주의함 (기말고사 신규사항)**
 - 문제의 메소드들은 Database 클래스의 멤버 메소드 들임
 - 채점시에는 모든 문제에서 새롭게 Database 객체를 만들어서 사용할 것임
 - 채점시에는 모든 문제에서 본인이 작성한 registNewCustomer()를 사용하여 항목을 추가 함. 따라서 해당 메소드가 틀리면 연쇄적으로 오점 처리됨

[문제1] 6점

- 메소드 이름: registNewCustomer
- 입력 파라미터
 - 첫번째 파라미터: 문자열, 고객 식별자
 - 두번째 파라미터: 문자열, 고객 이름
- 내부 동작
 - Database 객체는 registNewCustomer를 통해서 입력 받은 고객 정보를 객체 내부에서 dictionary 형태로 저장하고 관리함
- Return 값
 - 만약 입력 파라미터로 받은 고객 식별자가 객체 내부에 저장/관리하는 dictionary에 없다면, 입력 파라미터로부터 만들어진, { 고객 식별자 (문자열): 고객 이름 (문자열) }의 dictionary
 - 만약 입력 파라미터로 받은 고객 식별자가 이미 객체 내부에서 저장/관리하는 dictionary에

있다면, 정수 -1

[문제2] 6점

- 메소드 이름: getCustomerNumber
- 입력 파라미터
 - 없음
- Return 값
 - 객체 내부적으로 저장/관리하는 dictionary의 길이(length), 정수

[문제3] 6점

- 메소드 이름: getCustomerNameByID
- 입력 파라미터
 - 문자열, 고객 식별자
- Return 값
 - 만약 입력 파라미터로 받은 고객 식별자가 객체 내부에서 저장/관리하는 dictionary에 있다면, 입력 파라미터의 고객 식별자에 해당하는 { 고객 식별자 (문자열): 고객 이름 (문자열) }의 dictionary
 - 만약 입력 파라미터로 받은 고객 식별자가 객체 내부에 저장/관리하는 dictionary에 없다면, 정수 -1

[문제4] 6점

- 메소드 이름: getCustomerIDByName
- 입력 파라미터
 - 문자열, 고객 이름
- Return 값
 - 만약 입력 파라미터로 받은 고객 이름이 객체 내부에 저장/관리하는 dictionary에 있다면, 입력 파라미터의 고객 이름에 해당하는 { 고객 식별자 (문자열): 고객 이름 (문자열) }의 dictionary들을 모두 포함하는 dictionary. 즉, 고객 이름이 복수로 존재하면, 고객 이름을 갖는 모든 dictionary 들을 return 함. 이때 순서는 객체에 정보를 추가한 순서임
 - 만약 입력 파라미터로 받은 고객 이름이 객체 내부에 저장하여 관리하는 dictionary에 없다면, 정수 -1

[문제5] 6점

- 메소드 이름: getAllCustomer
- 입력 파라미터
 - 없음
- Return 값
 - 객체 내부적으로 저장/관리하는 dictionary. 이때 순서는 객체에 정보를 추가한 순서임

[문제6] 6점

- 메소드 이름: removeCustomerByID
- 입력 파라미터

- 문자열, 고객 식별자
- Return 값
 - 만약 입력 파라미터로 받은 고객 식별자가 객체 내부에 저장/관리하는 dictionary에 있다면, 입력 파라미터로 받은 고객 식별자의 정보를 객체 내부적으로 저장/관리하는 dictionary에서 삭제한 후, 입력 파라미터 항목이 삭제된 후의 dictionary를 return 함
 - 만약 입력 파라미터로 받은 고객 식별자가 객체 내부에 저장하여 관리하는 dictionary에 없다면, 정수 -1

[문제7] 6점

- 메소드 이름: removeCustomerByName
- 입력 파라미터
 - 문자열, 고객 이름
- Return 값
 - 만약 입력 파라미터로 받은 고객 이름이 객체 내부에 저장/관리하는 dictionary에 있다면, 입력 파라미터로 받은 고객 이름이 포함된 모든 정보를 객체 내부적으로 저장/관리하는 dictionary에서 삭제한 후, 입력 파라미터 항목이 삭제된 후의 dictionary를 return 함
 - 만약 입력 파라미터로 받은 고객 이름이 객체 내부에 저장하여 관리하는 dictionary에 없다면, 정수 -1

[문제8] 6점

- 메소드 이름: getAllCustomerNameSorted
- 입력 파라미터
 - 없음
- Return 값
 - 객체 내부적으로 저장/관리하는 dictionary의 고객 이름들을 오름차순으로 정렬한 리스트. 중복된 이름은 한번만 리스트에 포함되도록 함

[문제9] 6점

- 메소드 이름: getAllCustomerIDSorted
- 입력 파라미터
 - 없음
- Return 값
 - 객체 내부적으로 저장/관리하는 dictionary의 고객 식별자들을 오름차순으로 정렬한 리스트

[문제10] 6점

- 메소드 이름: getDuplicatedCustomerNames
- 입력 파라미터
 - 없음
- Return 값
 - 객체 내부적으로 저장/관리하는 dictionary의 고객 이름이 중복된 경우, 중복된 이름을 갖는 모든 { 고객 식별자 (문자열): 고객 이름 (문자열) } 정보를 포함하는 dictionary
 - 해당 정보들은 객체에 추가한 순서대로 배치함