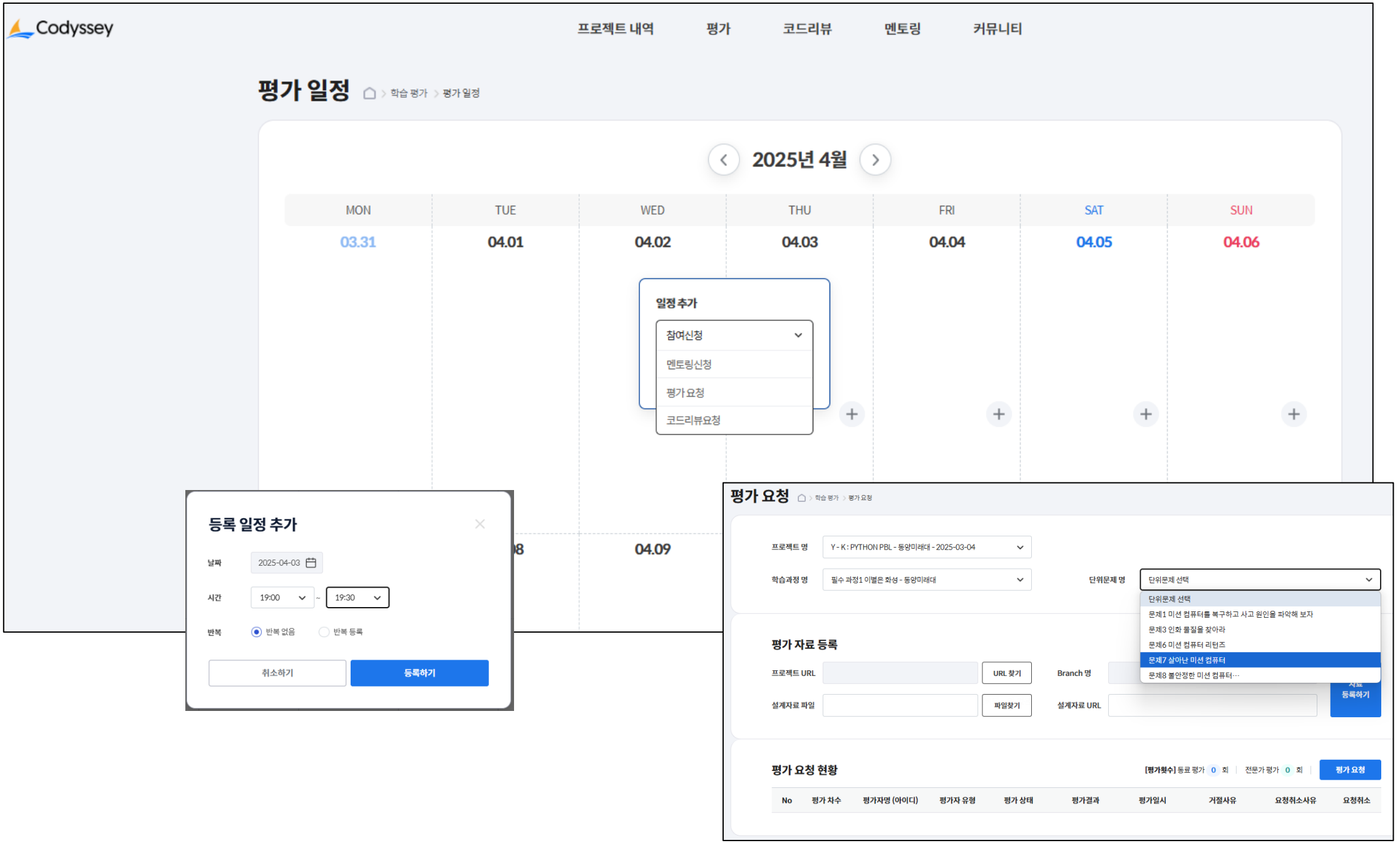
Codyssey
프로젝트 내역
평가
코드리뷰
멘토링
커뮤니티
평가 일정
학습 평가 > 평가 일정
2025년 4월
MON
TUE
WED
THU
FRI
SAT
SUN
03.31
04.01
04.02
04.03
04.04
04.05
04.06
04.09
일정 추가
참여신청
멘토링신청
평가 요청
코드리뷰요청
등록 일정 추가
날짜
2025-04-03
시간
19:00
19:30
반복
반복 없음
반복 등록
취소하기
등록하기
평가 요청
학습 평가 > 평가 요청
프로젝트 명
Y - K : PYTHON PBL - 동양미래대 - 2025-03-04
학습과정 명
필수 과정1 이별은 화성 - 동양미래대
단위문제 명
단위문제 선택
단위문제 선택
문제1 미션 컴퓨터를 복구하고 사고 원인을 파악해 보자
문제3 인화 물질을 찾아라
문제6 미션 컴퓨터 리턴즈
문제7 살아난 미션 컴퓨터
문제8 불안정한 미션 컴퓨터…
평가 자료 등록
프로젝트 URL
URL 찾기
Branch 명
설계자료 파일
파일찾기
설계자료 URL
자료 등록하기
평가 요청 현황
[평가횟수] 등록 평가 0 회 | 전문가 평가 0 회
평가 요청
No
평가 차수
평가자명 (아이디)
평가자 유형
평가 상태
평가결과
평가일시
거절사유
요청취소사유
요청취소

Codyssey 문제6 미션컴퓨터 리턴즈

동양미래대학교 컴퓨터공학부

기본 평가 운영 원칙

학생 / 시간	학번 % 2 == 0	학번 % 2 == 1
19:00-19:30	평가자-참여신청	피평가자-평가요청
19:30-20:00	피평가자-평가요청	평가자-참여신청
20:00-20:30	평가-피평가 자율	
20:30-	우수 수행 과제 공개 발표	

- 평가시간 30분 '가득' 채우면, 실력도 쑥쑥 !!!
- 반드시 실행 결과 확인

평가 시 유의 사항 (Check list)

» **프로그램 유지 보수 편이성이 고려되었는가 ?**

- DummySensor 클래스 메소드 구성이 적절한가 ?
- 환경변수 범위 값을 변경할 경우, 변경이 용이한가 ?

» **요구 기능을 충족하는가 ?**

- 환경 변수 범위 내 random 값이 잘 설정되는가 ?
- 반복 실행 시 random 값이 반복되어 설정되지는 않는가?
- (보너스 과제) log header가 저장되고, 반복 실행 시 header 중복 저장은 안되었나?

» **프로그램 안정성 ?**

- 클래스/인스턴스 변수가 불필요하게 노출되지는 않았나 ?
- 객체지향 철학이 잘 반영되어 설계되었나?

» **기타 논의 사항**

- 외부 라이브러리 사용 불가가 '표준 라이브러리' 까지를 의미하나?
- datetime/time 사용 불가시 구현 전략은 ?

Codyssey

문제6 미션컴퓨터 리턴즈

동양미래대학교 컴퓨터공학부

수행과제

- 더미 센서에 해당하는 클래스를 생성한다. 클래스의 이름은 DummySensor로 정의한다.
- DummySensor의 멤버로 env_values라는 사전 객체를 추가한다. 사전 객체에는 다음과 같은 항목들이 추가 되어 있어야 한다.
  - 화성 기지 내부 온도 (**mars_base_internal_temperature**)
  - 화성 기지 외부 온도 (**mars_base_external_temperature**)
  - 화성 기지 내부 습도 (**mars_base_internal_humidity**)
  - 회성 기지 외부 광량 (**mars_base_external_illuminance**)
  - 화성 기지 내부 이산화탄소 농도 (**mars_base_internal_co2**)
  - 화성 기지 내부 산소 농도 (**mars_base_internal_oxygen**)
- DummySensor는 테스트를 위한 객체이므로 데이터를 램덤으로 생성한다.
- DummySensor 클래스에 set_env() 메소드를 추가한다. set_env() 메소드는 random으로 주어진 범위 안의 값을 생성해서 env_values 항목에 채워주는 역할을 한다. 각 항목의 값의 범위는 다음과 같다.
  - 화성 기지 내부 온도 (18~30도)
  - 화성 기지 외부 온도 (0~21도)
  - 화성 기지 내부 습도 (50~60%)
  - 화성 기지 외부 광량 (500~715 W/m2)
  - 화성 기지 내부 이산화탄소 농도 (0.02~0.1%)
  - 화성 기지 내부 산소 농도 (4%~7%)
- DummySensor 클래스는 get_env() 메소드를 추가하는데 get_env() 메소드는 env_values를 return 한다.
- DummySensor 클래스를 ds라는 이름으로 인스턴스(Instance)로 만든다.
- 인스턴스화 한 DummySensor 클래스에서 set_env()와 get_env()를 차례로 호출해서 값을 확인한다.
- 전체 코드를 mars_mission_computer.py 파일로 저장한다.

스토리

화성 기지에 돔을 새로 만들어 연결하고 기지를 보강하고 나니 드디어 우주복을 벗을 수 있었다.

우주복을 벗고나니 한결 마음의 여유가 생긴다. 하지만 여전히 가장 큰 문제가 남아 있다. 미션 컴퓨터가 여전히 지금 상태를 제대로 작동을 못하고 있다는 점이다. 한송희 박사는 미션 컴퓨터의 메뉴를 구성하고 앞으로 생존에 필요한 기능들을 하나씩 추가하면서 화성을 탈출 할 수 있는 실마리를 만달어야 가야 한다고 생각했다.

그러기 위해서는 먼저 화성 기지의 남은 센서들을 사용해서 환경 값을 읽어 들이고 출력하는 기능을 추가 해야했다. 그리고 실제 센서를 만들기 전에 더미 센서(dummy sensor) 부터 만들어서 테스트를 시작해야 했다.

제약사항

- Python에서 기본 제공되는 명령어만 사용해야 하며 별도의 라이브러리나 패키지를 사용해서는 안된다.
- 단 random을 다루는 라이브러리는 사용 가능하다.
- Python의 coding style guide를 확인하고 가이드를 준수해서 코딩한다.
- 경고 메시지 없이 모든 코드는 실행 되어야 한다.

보너스 과제

- 출력하는 내용을 날짜와시간, 화성 기지 내부 온도, 화성 기지 외부 온도, 화성 기지 내부 습도 ,화성 기지 외부 광량, 화성 기지 내부 이산화탄소 농도, 화성 기지 내부 산소 농도 와 같이 파일에 log를 남기는 부분을 get_env()에 추가 한다.

Codyssey 문제7 살아난 미션컴퓨터

동양미래대학교 컴퓨터공학부

기본 평가 운영 원칙

학생 / 시간	학번 % 2 == 0	학번 % 2 == 1
19:00-19:30	평가자-참여신청	피평가자-평가요청
19:30-20:00	피평가자-평가요청	평가자-참여신청
20:00-20:30	평가-피평가 자율	
20:30-	우수 수행 과제 공개 발표	

- 평가시간 30분 '가득' 채우면, 실력도 쑥쑥 !!!
- 반드시 실행 결과 확인

평가 시 유의 사항 (Check list)

» **요구 기능을 충족하는가 ?**

- MissionComputer 클래스를 RunComputer로 인스턴스화 했나?
- Dummysensor 클래스를 ds로 인스턴스화 했나?
- RunComputer의 get_sensor_data() 메소드 호출시 DummySensor 전달은 어떤 방법으로 수행 했나?
- json 형식으로 화면 출력을 별도 모듈 사용 없이 직접 구현 했나요?
- (보너스 과제) 특정 키 입력 작업과 환경 변수 반복 출력 작업의 동시 처리를 위해 모색한 방법은(thread 등)

» **객체지향 ?**

- Public / Private 변수, 메서드
- Class 변수 / 인스턴스 변수
- @staticmethod / @classmethod

Codyssey

문제7 살아난 미션컴퓨터

동양미래대학교 컴퓨터공학부

수행과제

- 미션 컴퓨터에 해당하는 클래스를 생성한다. 클래스의 이름은 MissionComputer로 정의한다.
- 미션 컴퓨터에는 화성 기지의 환경에 대한 값을 저장할 수 있는 사전(Dict) 객체가 env_values라는 속성으로 포함되어야 한다.
- env_values라는 속성 안에는 다음과 같은 내용들이 구현 되어야 한다.
  - 화성 기지 내부 온도 (**mars_base_internal_temperature**)
  - 화성 기지 외부 온도 (**mars_base_external_temperature**)
  - 화성 기지 내부 습도 (**mars_base_internal_humidity**)
  - 회성 기지 외부 광량 (**mars_base_external_illuminance**)
  - 화성 기지 내부 이산화탄소 농도 (**mars_base_internal_co2**)
  - 화성 기지 내부 산소 농도 (**mars_base_internal_oxygen**)
- 문제 3에서 제작한 DummySensor 클래스를 ds라는 이름으로 인스턴스화 시킨다.
- MissionComputer에 get_sensor_data() 메소드를 추가한다.
- get_sensor_data() 메소드에 다음과 같은 세 가지 기능을 추가한다.
  - 센서의 값을 가져와서 env_values에 담는다.
  - env_values의 값을 출력한다. 이때 환경 정보의 값은 json 형태로 화면에 출력한다.
  - 위의 두 가지 동작을 5초에 한번씩 반복한다.
- MissionComputer 클래스를 RunComputer 라는 이름으로 인스턴스화 한다.
- RunComputer 인스턴스의 get_sensor_data() 메소드를 호출해서 지속적으로 환경에 대한 값을 출력할 수 있도록 한다.
- 전체 코드를 mars_mission_computer.py 파일로 저장한다.

스토리

더미 센서를 만들어 놓고 나니 이제는 미션 컴퓨터에서 직접 센서 데이터의 결과를 출력하고 내용을 확인할 수 있게 구성하는게 필요했다. 화성에서 사람이 살아가는데 필수적인 기지 내외부의 온도 그리고 광량, 이산화탄소의 농도와 산소 농도등을 확인 할 수 있게 구성해야 했다.

지구에서라면 수치 하나 하나가 실험실의 결과일 수도 있지만 여기서는 모두 생존과 관련되어 있어서 심각한 표정으로 한송희 박사는 코드를 들여다 보기 시작했다.

제약사항

- Python에서 기본 제공되는 명령어만 사용해야 하며 별도의 라이브러리나 패키지를 사용해서는 안된다.
- 단 시간을 다루는 라이브러리는 사용 가능하다.
- Python의 coding style guide를 확인하고 가이드를 준수해서 코딩한다.
- 경고 메시지 없이 모든 코드는 실행 되어야 한다.

보너스 과제

- 특정 키를 입력할 경우 반복적으로 출력되던 화성 기지의 환경에 대한 출력을 멈추고 'Sytem stoped....' 를 출력 할 수 있어야 한다.
- 5분에 한번씩 각 환경값에 대한 5분 평균 값을 별도로 출력한다.

동양미래대학교
컴퓨터공학부

기본 평가 운영 원칙

학생 / 시간	학번 % 2 == 0	학번 % 2 == 1
19:00-19:30	평가자-참여신청	피평가자-평가요청
19:30-20:00	피평가자-평가요청	평가자-참여신청
20:00-20:30	평가-피평가 자율	
20:30-	우수 수행 과제 공개 발표	

- 평가시간 30분 '가득' 채우면, 실력도 쑥쑥 !!!
- 반드시 실행 결과 확인

평가 시 유의 사항 (Check list)

» 미션 컴퓨터의 시스템 정보를 제공하는 SystemInfo Class를 선언할까?

- 각 항목 데이터 수집하는 메소드 생성? "예) get_os()"
  - » 시스템 정보 : OS, OS version, CPU type, CPU core 수, 메모리 크기
  - » 시스템 부하 : CPU 실시간 사용량, 메모리 실시간 사용량
- 해당 메소드 선언 : @staticmethod, @classmethod

» 보너스 과제

- setting.txt에 항목별 출력 여부(true, false) 설정 방법(json 등)
- 출력 항목만 출력 방법 : getattr() 활용 ?

Codyssey

문제8 불안정한 미션 컴퓨터....

동양미래대학교
컴퓨터공학부

수행과제

- 파이썬 코드를 사용해서 다음과 같은 미션 컴퓨터의 정보를 알아보는 메소드를 get_mission_computer_info() 라는 이름으로 만들고 문제 7에서 완성한 MissionComputer 클래스에 추가한다.

  - 필요한 미션 컴퓨터의 시스템 정보
    - 운영체계
    - 운영체계 버전
    - CPU의 타입
    - CPU의 코어 수
    - 메모리의 크기
- get_mission_computer_info()에 가져온 시스템 정보를 JSON 형식으로 출력하는 코드를 포함한다.
- 미션 컴퓨터의 부하를 가져오는 코드를 get_mission_computer_load() 메소드로 만들고 MissionComputer 클래스에 추가한다
- get_mission_computer_load() 메소드의 경우 다음과 같은 정보들을 가져 올 수 있게한다.
  - CPU 실시간 사용량
  - 메모리 실시간 사용량
- get_mission_computer_load()에 해당 결과를 JSON 형식으로 출력하는 코드를 추가한다.
- get_mission_computer_info(), get_mission_computer_load()를 호출해서 출력이 잘되는지 확인한다.
- MissionComputer 클래스를 runComputer 라는 이름으로 인스턴스화 한다.
- runComputer 인스턴스의 get_mission_computer_info(), get_mission_computer_load() 메소드를 호출해서 시스템 정보에 대한 값을 출력 할 수 있도록 한다.
- 최종적으로 결과를 mars_mission_computer.py 에 저장한다.

스토리

별다른 코딩을 한 것도 아닌데 미션 컴퓨터가 중간 중간 다운되는 현상이 일어나기 시작했다. 아직 생명유지와 관련된 기능까지 연결되어 있다면 문제가 심각해 질 것 같다. 컴퓨터의 상태를 좀 알고 싶긴한데 우주 기지에 설치된 컴퓨터라 완전히 밀봉되어 있어서 뜯어 보지도 못할 것 같다. 미션 컴퓨터의 지금 상태는 뭐가 문제인지 알 수가 없다.

지금 미션 컴퓨터의 상태를 알아보고 문제를 파악해 봐야겠는데 일단은 미션 컴퓨터 정보를 가져오는 코드를 좀 작성해서 상태를 파악해 보아겠다.

제약사항

- python에서 기본 제공되는 명령어 이외의 별도의 라이브러리나 패키지를 사용해서는 안된다.
- 단 시스템 정보를 가져오는 부분은 별도의 라이브러리를 사용 할 수 있다.
- 시스템 정보를 가져오는 부분은 예외처리가 되어 있어야 한다.
- 모든 라이브러리는 안정된 마지막 버전을 사용해야 한다.

보너스 과제

- setting.txt 파일을 만들어서 출력되는 정보의 항목을 셋팅 할 수 있도록 코드를 수정한다.