과제 #2

M3239.002300_003 데이터 사이언스 특강

Due: 2021년 10월 12일 23시 59분

1 문제 1: 스프레드시트

Microsoft Excel과 유사하게 데이터를 저장하고 읽을 수 있는 스프레드시트를 class Spreadsheet로 구현하라. Spreadsheet의 제약조건은 아래와 같다.

- 스프레드시트는 10개의 row, 10개의 column으로 구성된다. Microsoft Excel과 유사하게 column은 알파벳(A, B, ..., J, 혹은 a, b, ..., j)으로, row는 정수(1, 2, 3, ..., 10)로 indexing한다.
- 스프레드시트의 각 셀은 **int, bool, string**을 저장할 수 있다. 이 때 string은 whitespace 문자(\n, \t, space 등)를 포함하지 않는다고 가정한다.
- 생성자는 별도의 입력을 받지 않는다.
- Spreadsheet object를 print할 경우, Spreadsheet에 저장된 내용이 출력되어야 한다. 이 때 한 row 내의 element 사이는 comma(,) 한 개로 구분하고, row 사이는 개행(\n) 한 개로 구분하도록 하여 출력한다. 출력을 보기 좋게 만들기 위해 임의로 whitespace를 추가하여도 된다(예시 참고).

Spreadsheet class는 set_value, get_value method를 지원하며, 각각의 기능은 아래와 같다.

- **set_value(idx, value)**

  idx가 가리키는 스프레드시트의 셀 한 개에 value 값을 입력한다. idx는 string object로, Microsoft Excel과 유사하게 알파벳 + 정수 조합으로 구성된다 (예: 'A7'). value는 int, bool, 혹은 string이다. 해당 셀에 값이 이미 있을 경우 이를 덮어쓴다.

  올바르지 않은 값이 인자로 들어올 경우 적합한 Python built-in exception을 발생시켜야 한다. idx, value가 올바르지 않은 타입일 경우 TypeError를 발생시킨다. idx 값이 올바른 포맷이 아니거나 범위를 벗어날 경우(예: 'A5C', 'C0', 'F150') IndexError를 발생시킨다.

- **get_value(idx)**

  idx가 가리키는 스프레드시트의 셀 한 개에 저장된 값을 반환한다. 값을 입력하지 않은 셀을 참조할 경우 None을 반환한다. 올바르지 않은 idx가 입력될 경우 set_value method에서 설명한 바와 동일하게 Exception을 발생시킨다.

입출력 예시는 아래와 같다.

```
spdsta@login:~/HW2$ python3 -i prob1.py
>>> sheet1 = Spreadsheet()
>>> print(sheet1)
      ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,

>>> sheet1.set_value("A3", 5)
>>> sheet1.set_value("C1", True)
>>> sheet1.set_value("F5", "hello")
>>> print(sheet1)
      ,       , True  ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,
5     ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       , hello ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,

>>> sheet1.set_value("C1", "world")
>>> print(sheet1)
      ,       , world ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,
5     ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       , hello ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,       ,

>>> sheet1.get_value("C1")
'world'
>>> sheet1.get_value("J10")
>>> try:
...     sheet1.get_value("A4C")
... except IndexError:
...     print("Invalid index")
...
Invalid index
>>> sheet2 = Spreadsheet()
>>> sheet2.get_value("A3")
>>>
```

2 문제 2: 저장 가능한 스프레드시트

스프레드시트의 현재 상태를 파일에 저장하고, 추후 불러올 수 있도록 기능을 제공하는 class PermanentSpreadsheet를 구현하라. PermanentSpreadsheet는 Spreadsheet를 상속한 class로 구현한다. 추가로 구현해야 할 method는 아래와 같다.

- **export_sheet(filename)**

  현재 PermanentSpreadsheet object의 상태를 filename 파일에 저장한다. 포맷은 무관하다. 동일한 이름의 파일이 이미 있을 경우, 해당 파일을 덮어쓴다. filename은 string object이며, 다른 타입의 입력이 들어올 경우 TypeError를 생성한다.

- **import_sheet(filename)**

  현재 PermanentSpreadsheet object에 저장된 내용을 모두 버리고, 이전에 export_sheet method로 filename 파일에 저장해 둔 PremanentSpeadsheet를 다시 불러온다.

두 method 모두 파일 처리와 관련하여 문제에서 정의하지 않은 에러를 처리할 필요는 없다. 예를 들어 invalid한 filename을 사용하는 경우, import_sheet 함수 실행 시 filename 파일이 없는 경우 등은 테스트하지 않는다. 입출력 예시는 아래와 같다.

```
spdsta@login:~/HW2$ python3 -i prob2.py
>>> sheet = PermanentSpreadsheet()
>>> sheet.set_value("A1", 5)
>>> sheet.set_value("B2", True)
>>> sheet.set_value("C3", "True")
>>> print(sheet)
5     ,       ,       ,       ,       ,       ,       ,       ,
      , True  ,       ,       ,       ,       ,       ,       ,
      ,       , True  ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,

>>> sheet.export_sheet("sheet.dump")
>>> quit()
spdsta@login:~/HW2$ python3 -i prob2.py
>>> sheet = PermanentSpreadsheet()
>>> sheet.import_sheet("sheet.dump")
>>> print(sheet)
5     ,       ,       ,       ,       ,       ,       ,       ,
      , True  ,       ,       ,       ,       ,       ,       ,
      ,       , True  ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,
      ,       ,       ,       ,       ,       ,       ,       ,

>>>
```

3 문제 3: 고급 스프레드시트

다른 셀 한 개를 참조하는 lambda function을 특정 셀에 입력할 수 있도록 기능을 제공하는 class SmartSpreadsheet를 구현하라. SmartSpreadsheet는 Spreadsheet를 상속한 class로 구현한다. 추가로 구현해야할 method는 아래와 같다.

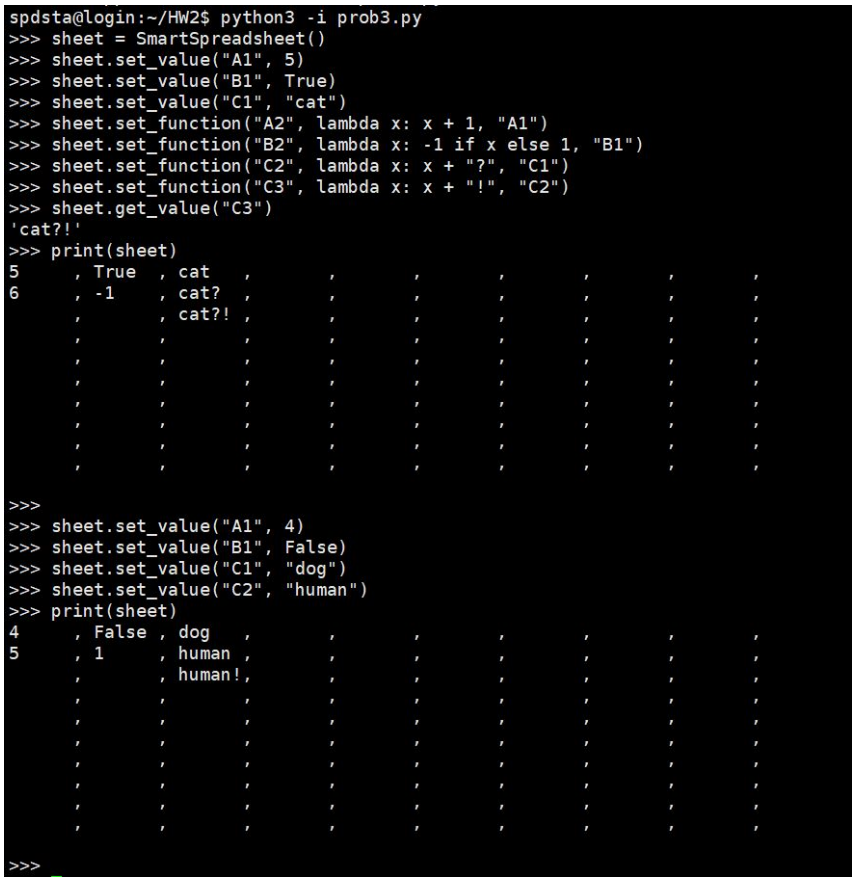
- **set_function(idx, function, operand_idx)**

  idx가 가리키는 스프레드시트의 셀 한 개에 함수 function을 입력한다. function은 입력 값 한 개를 받는 python lambda function이다. operand_idx는 function의 인자로 들어갈 셀의 index이다 (예: 'B3'). 함수 실행 이후 get_value(idx)를 실행하거나 SmartSpreadsheet object 자체를 print할 경우, operand_idx 셀 값을 function에 인자로 넣어 얻은 계산 결과를 idx 셀 값으로 사용한다.

함수 구현 시 아래와 같은 사항을 유의하여야 한다.

- 셀 참조는 Microsoft Excel과 마찬가지로, 참조하는 셀의 값이 변하면 수식 계산 결과도 바뀌어야 한다. 즉 operand_idx 셀의 값을 바꾼 후 idx 셀의 값을 확인하면 바뀐 값을 반영한 계산 결과를 반환해야 한다.
- operand_idx 셀도 다른 셀을 참조하는 셀일 수 있음을 유의하여야 한다.
- 입력한 lambda function은 **int, bool, 혹은 string**을 입력으로 받아 **int, bool, 혹은 string**을 반환하는 함수라고 가정한다. lambda function이 잘못된 타입의 인자/반환값을 사용하거나 Exception을 발생시키는 경우는 없다고 가정한다.
- 순환 참조는 일어나지 않는다고 가정한다. 예를 들어 A3이 A1의 값을 참조하고 A1이 A3을 참조하는 경우는 없다고 가정한다.
- 문제 2와 문제 3은 별개이다. SmartSpreadsheet에 대해 import_sheet, export_sheet method를 구현할 필요는 없다.

입출력 예시는 아래와 같다.



4 구현 및 제출

실습 서버의 실습 계정 홈 디렉토리의 HW2 디렉토리에 prob1.py, prob2.py, prob3.py를 작성하여 저장한다. 각각의 파일에는 아래와 같은 내용이 구현되어 있어야 한다.

파일	설명
prob1.py	Spreadsheet class 구현
prob2.py	PermanentSpreadsheet class 구현
prob3.py	SmartSpreadsheet class 구현

제출 시 아래와 같은 사항을 유의하라.

- 제출 파일에는 **반드시 class를 정의하는 코드만 포함되어야 하며, 테스트 코드 등이 포함되어서는 안 된다**. 과제 채점 시 실행 예시 스크린샷에서 보인 바와 유사한 방법으로 기계적인 채점을 수행할 예정이다. 이 때 제출한 코드에 불필요한 내용이 포함되어 채점이 방해될 경우 해당 과제가 0점 처리될 수 있다. 단, class에 추가로 method를 구현하여 내부 구현에서 사용하는 것은 괜찮다.
- PermanentSpreadsheet, SmartSpreadsheet class는 반드시 Spreadsheet class를 상속하도록 구현하여야 한다. 이를 만족하지 않을 경우 과제가 크게 감점될 수 있다. 상속을 위해 아래 예시와 같이 prob1.py 파일에서 Spreadsheet class를 import하여 사용할 수 있다.

```
fropm prob1 import Spreadsheet

class PermanentSpreadsheet(Spreadsheet):
    ...
```

- 과제 1번과 마찬가지로 기본 Python 문법에서 제공하는 기능만을 사용하여 과제를 구현하여야 하며, **prob1.py 파일 외에 타 모듈을 import하여 사용하는 것은 허용하지 않는다**.
- 과제 조건 중 **Exception 처리를 해야 하는 경우**와 **없다고 가정한 경우**를 잘 구분하여 구현하여야 한다. 과제 문서에 명시적으로 Exception 처리를 요구한 경우, 해당 case를 조교가 채점 시 사용할 수 있다. 없다고 가정한 경우들은 채점에서 사용하지 않는다. 과제 문서의 설명이 명확하지 않다고 판단되면 조교 메일 혹은 게시판에 질문할 것.
- 채점 시 Spreadsheet object print가 깔끔한지(예: align이 잘 되어있는지)의 여부는 점수에 반영되지 않는다.
- **이번 과제부터는 별도의 언급이 없으면 과제를 메일로 제출하지 않는다**. 실습 서버의 HW2 디렉토리에 파일을 저장하면, 과제 제출 시간에 자동으로 해당 파일을 복사해 간다. 파일명을 다른 이름으로 수정할 경우 (예: p1.py) 과제가 제출되지 않을 수 있으니 유의하여야 한다.