

Programming in Python

과제 프로젝트 - 설계



2016년 8월, 국민대학교 컴퓨터공학부

최상위 수준의 설계

- 몇 개의 모듈을 어떤 식으로 나누어 구현하면 좋을지?
 - 적어도 UI 를 담당하는 모듈과 (스크립트 형태로 실행될 top-level 프로그램) 데이터 모델은 분리
- 각 모듈의 기능적 구분은 어느 수준에서 이루어져야 좋은가?
 - 이 경우는 매우 단순하기 때문에 직관적으로 구분지을 수 있을 것
- 테스트는 어떻게 할 것인가?
 - 테스트 케이스를 미리 정의해 두는 것이 효과적이면서도 효율적임

단위 테스트의 중요성

```
def func(x, y):  
    if x > 0:  
        return x + y  
    else:  
        rerun x - y
```

```
if __name__ == '__main__':  
    assert f(2, 7) == 9, 'fail for (2, 7)'  
    assert f(3, 5) == 8, 'fail for (2, 7)'  
    ...  
    assert f(100, 32) == 132, 'fail for (100, 32)'
```

```
for i in range(1000):  
    for j in range(1000):  
        assert f(i, j) == i + j, 'fail for (%d, %d)' % (i, j)
```

UI 설계

tk

이름: 점수: 추가

번호: 삭제

파일이름: 저장

파일이름: 열기

번호순 이름순 점수내림차순 점수오름차순

- 몇 개의 프레임을 이용할 것인가?
- 각 위젯의 폭
 - 이름: 20
 - 점수: 7
 - 번호: 5
 - 작은 버튼: 5
 - 큰 버튼: 15
 - 데이터 출력창: 75 X 10
 - 상태 출력창: 75 X 1

UI 요구사항 정리

- 프로그램의 기능적 요구사항으로부터 도출
 - 각 UI 컴포넌트가 목적하는 기능을 달성하기 위해서는, 어떠한 데이터가 어떤 모양으로 제공되어야 하는가?
- 그렇다면, 데이터 모델에서 어떤 종류의 클래스들이 있어야 하며, 어떤 기능을 제공해야 하는가?
- 예
 - 데이터가 추가되면 출력창에 추가해야 하는데, 이 때 데이터 모델로부터 받아야 하는 리턴 타입은 무엇으로 하면 가장 효과적일까?
 - 데이터가 삭제되면 출력창에서도 없어져야 하는데, 이것은 어떻게 처리해야 할까?

데이터 모델 설계

- 어떤 클래스들을 정의할 것인가?
 - 각 클래스에 포함될 속성들은 무엇이 있는가?
 - 각 클래스가 제공해야 하는 메서드들은 무엇인가?
 - 이 메서드들은 어떤 형태로 데이터를 돌려주어야 하는가? (앞 페이지의 고려사항으로부터)
 - 메서드들을 구현하는 데 이용할 수 있는 자료들은 무엇인가?
 - 예: 파일 입출력을 위해서는?
 - 단위 테스트 케이스에는 어떤 것들이 포함되어야 하는가?
- 예
 - 내부 데이터 구조는 무엇을 선택하는 것이 좋을까? - **리스트!**
 - 추가, 삭제, 정렬 등이 용이
 - 그런데, 무엇의 리스트?
 - 서로 다른 기준으로 데이터를 정렬하는 방식은 무엇이 좋은가?

주의사항

- 설계 단계에 충분한 시간과 노력을 들이지 않고 구현에 덤비면
 - 나중에 낭패를 보게 되는 경우가 허다함
- 우리가 취할 흐름은:
 - 문제를 충분히 생각해 본다. (너무 단순하지만)
 - 발생할 수 있는 상황을 정리해 보고, 각 상황에 어떻게 대처할지 (코드로) 주의깊게 살펴본다.
 - 미심쩍은 부분이 있으면, Python interactive session 을 이용하여 확인해 둔다.
 - 확인한 코드 구조는 메모 등을 통하여 나중에 활용할 수 있도록 한다.
 - 코드의 구조를 머리 속에 그려 본다.
 - 스케치, 다이어그램, 표, 메모 등을 통하여 나중에 알아볼 수 있도록 기록해 둔다.
 - 프로그램의 규모가 커질수록 개발 노트를 잘 적는 것은 큰 도움이 된다!

설계 시작

Q & A