

설계 보고서

컴퓨터 공학부
20153243 허준녕

구현 완료된 인터페이스의 모습입니다.

UI 프레임 구성은

레이블과 엔트리, 그리고 추가 삭제 수정 저장 열기 버튼 부를 top 프레임으로 잡고
중간 번호순, 이름순, 점수내림차순, 점수오름차순을 mid 프레임으로
아래 두 개의 텍스트 위젯을 btm 프레임으로 지정했습니다.

```
ui = UI(window)
datahandler = DataHandler(ui)
filehandler = FileHandler(datahandler,ui)

#버튼을 초기화 합니다
ui.init_btn(datahandler,filehandler)
window.mainloop()
```

먼저 인터페이스 구현을 소개하면

Initializer 에서 프레임과 레이블, 엔트리, 텍스트 위젯을 생성했습니다.

버튼 위젯의 커맨드는 흐름 상 인터페이스 생성 시에 참조 하기엔
무리가 있어, 버튼 객체 생성은 메소드로 분리하여 진행하였습니다.

탑 프레임에서 레이블 부분은 변수를 참조하지 않는 단순 객체 생성을 for loop 을 이용해
생성했고

entry 는 코드 효율성을 위해 리스트로 묶어 for loop 을 이용해 생성했습니다.

전반적인 프로그램 구조와 흐름에 대해 설명하겠습니다.

먼저 모듈은 main, interface, io, data 의 4 개로 구성 했으며 data 와 io 는 핵심 알고리즘과 동작이 포함되어 있어 testDataHandler, testFileHandler 2 개의 UnitTest 모듈을 추가로 작성했습니다.

main 모듈은 각 모듈의 클래스 객체를 생성하여 UI 객체나 dataHandler 객체 등을 필요한 클래스 생성자나 메소드에 분배해주는 역할을 합니다.

Interface 모듈은 UI 클래스를 담고 있으며 자세한 설명은 1Page 에서 설명합니다.

data 모듈은 데이터 관리를 담당하는 dataHandler 클래스를 가집니다.

인터페이스 Entry 로 부터 값을 가져와야 할 때가 많아 , 클래스 멤버변수가 UI 객체를 참조해 접근을 용이하게 합니다.

핵심 데이터 관리를 위한 자료구조는 딕셔너리를 사용했으며 번호를 key 로, 이름과 점수를 리스트로 묶어 value 로 사용했습니다.

메소드는 setIndex, add, delete, changeName , changeScore, sort 관련 등이 있습니다.

setIndex 메소드는 자료를 추가할시 번호부여를 관리합니다. 삭제를 하여 번호가 비거나 새로운 파일을 읽어왔을 때 데이터 번호가 비거나 달라지므로 적절한 빈 위치를 찾아 리턴합니다.

Add 메소드는 이름과 점수를 입력 받아 딕셔너리에 추가합니다.

Delete 메소드는 번호를 입력 받아 딕셔너리로부터 제거합니다.

sort 의 경우 딕셔너리 자료구조의 비 규칙성에 의해 딕셔너리를 직접 정렬하여도 정렬하기 전과 같게 되어 직접 데이터를 정렬하지 않고 메소드를 이용해 튜플화한 후 Text 창에 출력만 하게 했습니다.

io 모듈은 파일 입출력을 담당하는 fileHandler 클래스를 가집니다. 클래스 내 원할 한 코드 진행을 위해 UI 객체와 dataHandler 객체를 클래스 멤버변수가 참조합니다. dataHandler 클래스 객체를 참조하는 변수가 있으므로 데이터를 담는 딕셔너리를 쉽게 참조할 수 있습니다.

메소드는 inFile,outFile 메소드가 있습니다.

inFile 메소드는 딕셔너리를 초기화 한 후 파일을 읽어와 딕셔너리를 읽어온 데이터로 채워줍니다 .

outFile 메소드는 데이터 딕셔너리를 입력한 파일이름으로 텍스트 문서를 만들어 데이터를 저장합니다.