기본 문서를 작성

README .md 작성

-프로젝트 개요

-프로그램 설정 및 실행 지침.

-종속성 및 모듈 사용 방법의 예를 포함하여 모든 하위 모듈의 역할 및 책임에 대한 간략한  
설명입니다.

가상 환경을 이용해 라이브러리를 이용  
requirements.txt을 이용해 라이브러리 관리 pip freeze > requirements.txt

Python 예외처리 구현

1. 잘못된 파일, 잘못된 텍스트
2. Os.walk를 사용해 디렉터리에서 수집 가능한 파일을 자동으로 검색

해당 예외처리들은 사람이 읽을 수 있는 메시지 포함

Python 모듈 제작

각 모듈 디렉토리에는 적절한 init.py 파일 포함

기본적인 객체 지향 데이터 구조를 구현하는 파이썬 클래스 코드를 제공할게요

이 코드에는 QuoteMode 객체를 정의하는 클래스가 포함되어 있습니다. QuoteMode 클래스에는 본문(body)과 저자(author)를 위한 텍스트 필드가 있으며, 해당 클래스는 클래스를 인스턴스화하고 모델 내용을 "본문 텍스트" - 저자 형식으로 출력하기 위해 필요한 메서드들을 오버라이드합니다.

파이썬에서 추상 기본 클래스(ABC)를 사용하는 시점과 이 디자인 패턴을 코드에 구현하는 방법을 식별합니다.

프로젝트에는 IngestorInterface라는 추상 기본 클래스가 포함되어 있습니다. 이 클래스는 다음을 정의합니다:

파일 유형이 해당 Ingestor 클래스와 호환되는지 확인하기 위한 완전한 classmethod 메서드.

파일 내용을 구문 분석하여(예: 각 행을 분할) Quote 객체로 출력하기 위한 추상 메서드.

추상 메서드는 구현되지 않은 클래스에서만 정의되어야 합니다.

팁: Classmethods는 클래스 변수에 접근할 수 있으며, 이는 하위 클래스에서 재정의될 수 있습니다.

기본 파일 라이브러리를 사용하여 텍스트 파일을 수집합니다.

프로젝트에는 TextIngestor 클래스가 있습니다.

이 클래스는 IngestorInterface를 상속합니다.

이 클래스는 Text 파일을 구문 분석하기 위해 정의된 추상 메서드 시그니처를 완료하기 위해 어떠한 서드파티 라이브러리에도 의존하지 않습니다.

parse 메서드는 유효한 QuoteModel을 반환합니다.

라이브러리를 사용하여 DOCX 파일을 수집합니다 python-docx.

프로젝트에는 DocxIngestor 클래스가 있습니다.

이 클래스는 IngestorInterface 클래스를 상속합니다.

이 클래스는 DOCX 파일을 구문 분석하기 위해 python-docx 라이브러리에 의존합니다. 이 라이브러리를 사용하여 추상 메서드 시그니처를 완료합니다.

parse 메서드는 유효한 QuoteModel을 반환합니다.

라이브러리를 사용하여 CSV 파일을 수집합니다 pandas.

프로젝트에는 CSVIngestor 클래스가 있습니다.

이 클래스는 IngestorInterface를 상속합니다.

CSV 파일을 구문 분석하기 위해 이 클래스는 pandas 라이브러리에 의존합니다. 이 라이브러리를 사용하여 추상 메서드 시그니처를 완료합니다.

parse 메서드는 유효한 QuoteModel을 반환합니다.

Python에서 전략 객체 디자인 패턴을 사용하여 클래스 상속을 구현하고 DRY (Don't Repeat Yourself) 원칙을 적용합니다.

DocxIngestor, PDFIngestor, CSVIngestor 및 TextIngestor 클래스는 IngestorInterface 추상 기본 클래스를 구현해야 합니다.

공통 책임을 가진 메서드들은 부모 클래스에서 완전히 정의됩니다.

여러 클래스 간에 중복되는 코드가 없도록 합니다.

모든 Ingestor들은 하나의 인터페이스를 제공하기 위해 모든 Ingestor를 캡슐화하는 main Ingestor 클래스로 패키징되어야 합니다.

양식의 맨 위

라이브러리를 사용하여 Pillow기본 이미지 작업을 수행합니다.

프로젝트는 MemeGenerator다음과 같은 책임을 가진 모듈을 정의합니다.

* 디스크에서 파일 로드
* 입력 화면 비율을 유지하면서 최대 너비 500px로 크기를 조정하여 이미지를 변환합니다.
* 이미지의 임의 위치에 본문과 작성자가 포함된 캡션(문자열 입력)을 이미지에 추가합니다.

클래스는 PillowJPEG/PNG 파일과 함께 작동하도록 정의된 불완전한 메서드 서명을 완성하기 위해 라이브러리에 의존합니다.

밈을 만들기 위한 메소드 서명은 다음과 같아야 합니다.make\_meme(self, img\_path, text, author, width=500) -> str #generated image path

init 메소드는 생성된 이미지 **를** 저장할 위치에 대한 필수 인수를 취해야 합니다: \_\_init\_\_(self, output\_dir...).

CLI 실행을 위해 Python 인수 변수를 사용합니다.

프로젝트에는 , , 및 메서드를 main.py사용하여 임의 캡션 이미지를 생성하는 파일이 포함되어 있습니다.ImageCaptionerDocxIngestorPDFIngestorCSVIngestor

프로그램은 명령줄에서 실행 가능해야 합니다.

이 프로그램은 세 가지 선택적 인수를 사용합니다.

* 문자열 인용 본문
* 문자열 인용 작성자
* 이미지 경로

프로그램은 생성된 이미지에 대한 경로를 반환합니다.  
인수가 정의되지 않은 경우 무작위 선택이 사용됩니다.

flask및 를 사용하여 웹 리소스와 인터페이스합니다 requests.

프로젝트는 Flask 앱 시작 코드를 완성합니다 app.py. 파일에 나열된 모든 @TODO 작업이 완료되었습니다.

app.pyQuote Engine모듈과 모듈을 사용하여 Meme Generator임의의 캡션 이미지를 생성합니다.

app.py패키지를 사용하여 requests사용자가 제출한 URL에서 이미지를 가져옵니다.

플라스크 서버가 오류 없이 실행됩니다.