

미션: 도서 관리 시스템 구축

*개요

최종 목표: 도서관의 책 대여 및 반납을 관리하는 시스템을 구축하고, 대여 통계를 파일로 저장하는 프로그램을 완성합니다.

핵심 요구사항:

- 책 정보를 담는 클래스 구조 설계
- 대여/반납 기능 구현
- 사용자 입력 처리 및 예외 처리
- 대여 기록을 파일로 저장하고 불러오기
- 통계 정보 출력 기능

제공 데이터:

도서 목록:

- 파이썬 완전정복 (저자: 김코딩, ISBN: 9781234567890, 재고: 3권)
- 데이터 과학 입문 (저자: 이데이터, ISBN: 9781234567891, 재고: 2권)
- 웹 개발의 모든 것 (저자: 박웹개, ISBN: 9781234567892, 재고: 5권)
- AI 프로그래밍 (저자: 최인공, ISBN: 9781234567893, 재고: 1권)
- 알고리즘 기초 (저자: 정알고, ISBN: 9781234567894, 재고: 4권)

(필요에 따라 자유롭게 설정)

미션 1: 데이터 구조 설계하기

도서 정보를 효율적으로 관리하기 위한 데이터 구조를 설계하세요.

요구사항:

- 책의 제목, 저자, ISBN, 재고 수량을 저장할 수 있어야 함
- Book 클래스를 설계
- 최소 5권의 책 정보를 저장하는 리스트 생성
- 책 정보를 출력하는 함수 작성

미션 2: 대여 및 반납 기능 구현하기

도서 대여와 반납 기능을 구현하세요.

요구사항:

- 책 제목으로 대여할 수 있는 **borrow_book()** 함수
- 재고가 0이면 대여 불가 처리
- 대여 시 재고 감소, 반납 시 재고 증가
- **return_book()** 함수 구현
- 예외 처리: 존재하지 않는 책 제목 입력 시

힌트:

- 리스트에서 특정 책을 찾는 방법 고려
- 조건문으로 재고 확인
- **try-except**로 예외 상황 처리

미션 3: 대여 기록 관리하기

대여 기록을 저장하고 관리하는 기능을 추가하세요.

요구사항:

- 대여 기록을 저장하는 리스트 또는 딕셔너리 생성
- 대여 시 기록 추가 (책 제목, 대여 날짜/시간)
- 대여 기록을 조회하는 함수
- **datetime** 모듈 활용

힌트:

- **datetime** 모듈의 **datetime.now()** 사용
- 딕셔너리로 **{'title': '책제목', 'date': '날짜'}** 형태 저장
- 리스트에 여러 대여 기록 딕셔너리 추가

미션 4: 파일 저장 및 불러오기

대여 기록과 현재 재고 상태를 파일로 저장하고 불러오는 기능을 구현하세요.

요구사항:

- 대여 기록을 텍스트 파일로 저장
- 현재 책 재고 상태를 파일로 저장
- 프로그램 시작 시 파일에서 데이터 불러오기
- 파일이 없을 때 예외 처리

힌트:

- `open()` 함수와 `with` 문 사용
- `FileNotFoundException` 예외 처리
- 문자열 포맷팅으로 데이터 정리

미션 5: 메뉴 시스템 및 통계 기능

사용자가 쉽게 사용할 수 있는 메뉴 시스템과 통계 기능을 구현하세요.

요구사항:

- 반복문을 사용한 메뉴 시스템
- 도서 목록 보기, 2. 대여하기, 3. 반납하기, 4. 대여 기록, 5. 통계 보기, 6. 종료
- 통계: 총 대여 횟수, 가장 많이 대여된 책
- 잘못된 입력에 대한 예외 처리

힌트:

- `while True` 반복문과 `break` 사용
- 딕셔너리로 책별 대여 횟수 카운트
- `max()` 함수로 최대값 찾기

추가 도전 과제!

* 회원 관리 기능: 대여자 정보를 저장하고 관리

회원 정보에 대한 클래스 **Member** 정의

회원 등록, 조회, 목록 기능

회원 기반 대여 및 반납 기능 구현

회원 정보 기반 대여 기록 관리하기

* 연체 확인: 대여일로부터 **7일** 이상 지난 책 찾기

* 검색 기능: 저자나 제목으로 책 검색