

**< 과제 3 – 도서 대출 반납 프로그램 >**

도서 대출 프로그램을 구현하고, 프로그램 작성 과정과 결과를 기술한 보고서를 제출한다.

- < 과제 3 >은 다음과 같이 하나의 압축 파일(.zip)로 제출한다.
- zip 파일에는 1개의 보고서(.pdf)와 1개의 소스 파일(.py)을 포함한다.
- 제출 파일명: **과제3\_학번\_이름.zip**
  - 보고서 파일명: 과제3\_학번\_이름.pdf
  - 소스 코드 파일명: 과제3\_학번\_이름.py
- 제출된 소스 파일(.py)은 적절한 주석을 포함해야 한다.
  - ▶ 주석 포함 내용: 프로그램 이름, 학번, 이름, 프로그램의 주요 코드에 대한 설명
- 제출 기한: **06월 30일 (화요일) 21:59 까지**
  - ▶ 아주Bb-과제 제출 메뉴에 생성된 과제란에 개별적으로 제출합니다.
  - ▶ 제출 후 다시 한번 업로드 한 파일 확인하기 바랍니다.
  - ▶ 지각 제출은 **07월 01일 (수요일) 14:59 까지**만 허용합니다. (지각 제출은 5점 감점, 이후 제출은 0점)
  - ▶ **COPY 한 코드나 보고서 발견 시, 두 사람 모두 0점 처리합니다.**
  - ▶ 아주 Blackboard는 표절 제출물 자동 검사 기능 'SafeAssign'을 제공합니다. 과제는 반드시 본인이 직접 작성하여 제출하기 바랍니다.

※ 주의: 수업 시간에 학습한 내용 이외의 기능을 활용한 경우 COPY로 간주함

- 채점 기준
  - ▶ 40점 만점 : 보고서 15점, 프로그램 25점 (평가 기준에 따라 추후 환산 반영됨)
  - ▶ 감점 사항
    - 제출 형식 위반 (파일 이름 등)
    - 코드의 주석 부재
    - 보고서의 항목 누락 및 설명 부족
    - 각 프로그램의 조건 누락 시
    - 지각 제출

## ◇ 도서 대출 반납 프로그램

- 아래 조건에 따른 도서 대출 반납 프로그램을 작성하세요.

### < 조건 >

1. 서비스 번호([ 1. 대출, 2. 반납 ])를 보기로 제시하고, 사용자로부터 서비스 번호를 입력받는다.
    - ❖ 보기에 없는 번호가 입력되면 "서비스 번호를 잘못 입력했습니다."를 출력하고, 서비스 번호 입력을 다시 요청한다.
  2. 대출 서비스를 선택한 경우, 대출일과 대출 권수를 입력받고, 입력받은 대출 권수만큼 대출할 도서명을 입력받는다.
    - ❖ 현재 3권을 모두 대출한 경우라면 "현재 3권 모두 대출 중입니다."를 출력하고, 서비스 번호 입력을 다시 요청한다.
- 가. 대출일 (반납일 입력도 동일)
- 대출일과 반납일은 년(정수), 월(문자열), 일(정수) 정보를 입력받으며, 스페이스로 구분하여 한 줄로 입력받는다. 예) 2020 MAY 20
  - 년(정수)과 일(정수)은 항상 올바르게 입력받는다고 가정하고, 월(문자열)에 대한 입력 오류만 처리한다.
  - 월(문자열)은 모두 영문 대문자 앞의 3글자로 입력받는다. 예를 들어, 1월은 "JAN" 만 가능하며, "Jan"나 "01" 등의 다른 문자열을 입력받으면 "월 입력이 잘못되었습니다."를 출력하고, 대출일 혹은 반납일을 다시 입력받도록 한다.
  - 년(정수), 월(문자열), 일(정수)은 리스트에 저장하여 관리한다.
- 나. 대출 권수
- 대출은 1인당 3권 이내만 가능하고, 한 번에 3권까지 모두 가능하다. 사용자로부터 원하는 대출 권수를 입력받아 처리한다.
  - 대출이 가능한 경우, 총 3권을 넘지 않도록 입력받는다. 현재 대출 중인 도서 수와 입력한 대출 권수의 합이 3권을 넘으면 "대출 가능한 권수를 초과했습니다."를 출력하고, 대출 권수를 다시 입력받도록 한다.
- 다. 대출할 도서명
- 도서명(문자열)과 대출일에 대한 리스트(년(정수), 월(문자열), 일(정수))을 딕셔너리에 저장하여 관리한다.
  - 입력한 대출 권수만큼 대출할 도서명을 연속적으로 입력받고, 모두 같은 대출일에 대출한 것으로 처리한다.

## 2020년 1학기 프로그래밍 기초 < 과제 3 >

---

- 반납 서비스를 선택한 경우, 현재 대출 현황이 출력되도록 한다. 그리고 반납일과 반납할 도서명을 입력받는다.

### 가. 대출 현황

- 현재 대출 중인 도서 수만큼 번호([번호])와 대출일(월은 정수로 출력), 도서명에 대한 정보를 출력한다.
- 현재 대출한 책이 없어서 출력할 대출 현황이 없는 경우, "반납할 도서가 없습니다."를 출력하고, 서비스 번호 입력을 다시 요청한다.

### 나. 반납일

- 반납일 입력도 대출일 입력과 동일하게 처리한다. (2-가 대출일 항목 참고)
- 대출일 입력 날짜가 반납일 입력 날짜보다 과거여야 한다.

### 다. 반납할 도서명

- 반납은 한 번에 1권만 가능하다.
- 반납할 도서명 입력에 오류는 없다고 가정(출력된 대출 현황 내의 도서명을 정확히 입력함)한다.

### 라. 대출 기간

- 1권당 가능한 대출 기간은 50일이다.
- 입력한 반납할 도서명의 대출일과 반납일 사이의 대출 기간을 계산하여 출력한다. (예를 들어, 2020년 1월 1일에 대출하고, 2020년 1월 2일에 반납하면 대출 기간은 총 1일이다.)
- 대출일과 반납일은 실행 예시와 다른 값이 입력되어도 계산되어야 한다.
- 윤년을 고려해야 하며 입력한 년이 윤년이라면 2월은 29일로 처리해야 한다.
- 계산한 대출 기간은 반납할 도서명과 대출일, 반납일의 정보(월은 정수로 출력)와 함께 출력한다.
- 1권당 50일로 정해져 있는 대출 기간을 초과한 경우, 연체일을 계산하여 출력한다.
- 마지막으로 "반납이 완료되었습니다."를 출력한다.

- 대출 기간 계산을 위해 필요한 함수(필수)를 구현하여 프로그램을 작성한다. (그 외 월(문자열) 입력 오류를 검사를 위한 함수와 월(문자열)을 숫자로 바꾸는 함수 등을 선택적으로 추가 구현할 수 있다.)

❖ 단, <과제 3>은 반드시 main() 함수와 **최소 3개 이상의 사용자 정의 함수**로 구성한다. (본인의 소스 코드에서 작성된 main() 함수와 필수로 구현해야 하는 사용자 정의 함수, 선택적으로 구현한 사용자 정의 함수들을 기반으로 함수 관계도를 작성한다.)

- 입력 데이터에 따른 결과를 제시된 실행 예시와 동일하게 구성하고 출력되도록 한다.

## 1. 보고서 (.pdf)

- 보고서의 내용은 다음 사항을 포함하되, **보고서에 기술된 내용을 바탕으로 프로그램이 작성되어야** 한다.
  - (1) 요구사항 분석 결과
  - (2) 문제 해결을 위한 주요 알고리즘 설명
    - 대출 기간 계산 과정 (**필수**)
    - 그 외 선택적으로 작성
  - (3) 함수 관계도 작성 및 설명(**필수**) <- (과제 설명 강의 영상 확인 필요)
  - (4) 실행 화면 캡처 및 결과 설명
  - (5) 결과 평가

## 2. 프로그램 (.py)

- 실행 예1 >>

```
[ 도서 대출 반납 프로그램 ]
[ 1. 대출, 2. 반납 ]
>>> 서비스 번호를 선택하세요: 1

>>> [대출일] 년, 월, 일을 입력하세요.(예: 2020 JAN 20): 2012 feb 15
월 입력이 잘못되었습니다.

>>> [대출일] 년, 월, 일을 입력하세요.(예: 2020 JAN 20): 2012 FEB 15
>>> 대출 권수 입력: 1
>>> 대출할 도서명 입력: Python

>>> 계속 하시겠습니까?(y 또는 n): y

[ 1. 대출, 2. 반납 ]
>>> 서비스 번호를 선택하세요: 2

*****현재 대출 현황*****
[1] 대출일: 2012년 2월 15일, 도서명: Python
*****

>>> [반납일] 년, 월, 일을 입력하세요.(예: 2020 MAR 10): 2012 03 15
월 입력이 잘못되었습니다.

>>> [반납일] 년, 월, 일을 입력하세요.(예: 2020 MAR 10): 2012 MAR 15
>>> 반납할 도서명 입력: Python

< Python > 도서의 대출일은 2012년 2월 15일이고, 반납일은 2012년 3월 15일입니다.
대출 기간은 총 29일입니다.

반납이 완료되었습니다.

>>> 계속 하시겠습니까?(y 또는 n): n
>>>
```

## 2020년 1학기 프로그래밍 기초 < 과제 3 >

### - 실행 예2 >>

```
[ 도서 대출 반납 프로그램 ]

[ 1. 대출, 2. 반납 ]
>>> 서비스 번호를 선택하세요: 1

>>> [대출일] 년, 월, 일을 입력하세요.(예: 2020 JAN 20): 2020 JAN 20
>>> 대출 권수 입력: 2
>>> 대출할 도서명 입력: C
>>> 대출할 도서명 입력: JAVA

>>> 계속 하시겠습니까?(y 또는 n): y

[ 1. 대출, 2. 반납 ]
>>> 서비스 번호를 선택하세요: 1

>>> [대출일] 년, 월, 일을 입력하세요.(예: 2020 JAN 20): 2020 JAN 21
>>> 대출 권수 입력: 2
대출 가능한 권수를 초과했습니다.
>>> 대출 권수 입력: 1
>>> 대출할 도서명 입력: Python

>>> 계속 하시겠습니까?(y 또는 n): y

[ 1. 대출, 2. 반납 ]
>>> 서비스 번호를 선택하세요: 2

*****현재 대출 현황*****
[1] 대출일: 2020년 1월 20일, 도서명: C
[2] 대출일: 2020년 1월 20일, 도서명: JAVA
[3] 대출일: 2020년 1월 21일, 도서명: Python
*****

>>> [반납일] 년, 월, 일을 입력하세요.(예: 2020 MAR 10): 2020 MAR 10
>>> 반납할 도서명 입력: JAVA

< JAVA > 도서의 대출일은 2020년 1월 20일이고, 반납일은 2020년 3월 10일입니다.
대출 기간은 총 50일입니다.

반납이 완료되었습니다.

>>> 계속 하시겠습니까?(y 또는 n): y

[ 1. 대출, 2. 반납 ]
>>> 서비스 번호를 선택하세요: 2

*****현재 대출 현황*****
[1] 대출일: 2020년 1월 20일, 도서명: C
[2] 대출일: 2020년 1월 21일, 도서명: Python
*****

>>> [반납일] 년, 월, 일을 입력하세요.(예: 2020 MAR 10): 2020 MAR 12
>>> 반납할 도서명 입력: Python

< Python > 도서의 대출일은 2020년 1월 21일이고, 반납일은 2020년 3월 12일입니다.
대출 기간은 총 51일입니다.

1일 연체되었습니다.
반납이 완료되었습니다.

>>> 계속 하시겠습니까?(y 또는 n): n
>>>
```

- 실행 예3 >>

```
[ 도서 대출 반납 프로그램 ]

[ 1. 대출, 2. 반납 ]
>>> 서비스 번호를 선택하세요: 3
서비스 번호를 잘못 입력했습니다.

[ 1. 대출, 2. 반납 ]
>>> 서비스 번호를 선택하세요: 2
반납할 도서가 없습니다.

[ 1. 대출, 2. 반납 ]
>>> 서비스 번호를 선택하세요: 1

>>> [대출일] 년, 월, 일을 입력하세요.(예: 2020 JAN 20): 2019 JAN 01
>>> 대출 권수 입력: 1
>>> 대출할 도서명 입력: MATLAB

>>> 계속 하시겠습니까?(y 또는 n): y

[ 1. 대출, 2. 반납 ]
>>> 서비스 번호를 선택하세요: 2

*****현재 대출 현황*****
[1] 대출일: 2019년 1월 1일, 도서명: MATLAB
*****

>>> [반납일] 년, 월, 일을 입력하세요.(예: 2020 MAR 10): 2020 JAN 01
>>> 반납할 도서명 입력: MATLAB

< MATLAB > 도서의 대출일은 2019년 1월 1일이고, 반납일은 2020년 1월 1일입니다.
대출 기간은 총 365일입니다.

315일 연체되었습니다.
반납이 완료되었습니다.

>>> 계속 하시겠습니까?(y 또는 n): y

[ 1. 대출, 2. 반납 ]
>>> 서비스 번호를 선택하세요: 1

>>> [대출일] 년, 월, 일을 입력하세요.(예: 2020 JAN 20): 2020 JAN 01
>>> 대출 권수 입력: 1
>>> 대출할 도서명 입력: R

>>> 계속 하시겠습니까?(y 또는 n): y

[ 1. 대출, 2. 반납 ]
>>> 서비스 번호를 선택하세요: 2

*****현재 대출 현황*****
[1] 대출일: 2020년 1월 1일, 도서명: R
*****

>>> [반납일] 년, 월, 일을 입력하세요.(예: 2020 MAR 10): 2021 JAN 01
>>> 반납할 도서명 입력: R

< R > 도서의 대출일은 2020년 1월 1일이고, 반납일은 2021년 1월 1일입니다.
대출 기간은 총 366일입니다.

316일 연체되었습니다.
반납이 완료되었습니다.

>>> 계속 하시겠습니까?(y 또는 n): n
>>>
```