

고급프로그래밍 1분반

3조 프로젝트: 도서관리시스템

학과	소프트웨어학과
조명	3조
조원	손명진(조장), 서유빈, 서준호, 성종석, 신현빈, 안상완
제출일	

프로젝트 경과 상황

11/24/2021 손명진: 시스템의 전체적인 틀 구현, 임시 데이터 생성, 사용할 기능 버튼 구현, 도서 목록 조회 구현

텍스트 데이터

명품 Java Programming, 황기태/김효수, 생능출판, 대여중
Doit! 자바프로그래밍입문, 박은종, 이지스퍼블리싱, 대여가능

Bookshelf 클래스

```
package team3.assignment;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;

public class Bookshelf extends JFrame {

    public Bookshelf() {
        setTitle("도서 관리 프로그램");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        Container contentPane = getContentPane();
        contentPane.setLayout(new BorderLayout());

        JPanel introPage = new JPanel();
        JLabel logo = new JLabel("3 조의 도서관리시스템 로고입니다.");
        introPage.add(logo);
        contentPane.add(introPage, BorderLayout.NORTH);

        JPanel mainPage = new JPanel();
        contentPane.add(mainPage, BorderLayout.CENTER);

        JPanel menus = new JPanel(new GridLayout(7, 1, 5, 5));
        JButton searchBook = new JButton("도서 검색");
        menus.add(searchBook);
        JButton showBooks = new JButton("도서 목록 출력");
        showBooks.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                setVisible(false);
                mainPage.removeAll();
            }
        });
    }
}
```

```
mainPage.add(new JLabel("도서명 | 저자 | 출판사 | 대여가능여부"));

try(FileReader fReader = new FileReader("bookData.txt");
Scanner bookReader = new Scanner(fReader)) {

    while (bookReader.hasNext ()) {
        mainPage.add(new JLabel(bookReader.nextLine()));
    }
} catch (FileNotFoundException ept) {
    ept.printStackTrace();
} catch (IOException ept) {
    ept.printStackTrace();
}

setSize(600, 600);
setVisible(true);
}

);
menus.add(showBooks);
JButton rentalBook = new JButton("도서 대여하기");
menus.add(rentalBook);
JButton returnBook = new JButton("도서 반납하기");
menus.add(returnBook);
JButton wearBook = new JButton("입고 도서 등록");
menus.add(wearBook);
JButton deleteBook = new JButton("기존 도서 삭제하기");
menus.add(deleteBook);
JButton exit = new JButton("시스템 종료");
menus.add(exit);
contentPane.add(menus, BorderLayout.WEST);

setSize(600, 600);
setVisible(true);
}

public static void main(String[] args) {
    Bookshelf bookshelf = new Bookshelf();
}
}
```

11/27/2021: 안상완 : 도서검색 기능 추가, 해당 기능 추가를 위한 추가적인 필드 구현

변경사항: 파일 구분자 변경:

「도서명,저자명,출판사,대여여부」에서 「도서명,저자명;출판사;대여여부」로 변경

명품 Java Programming, 황기태/김효수;생능출판;대여중
Doit! 자바프로그래밍입문, 박은종, 이지스퍼블리싱;대여가능

코드:

```
package team3.assignment;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.HashMap;
import java.util.Scanner;

/*
1- 손명진 : 사용할 기능 입력 및 도서 목록 조회
2- 안상완 : 도서 검색
3- 신현빈 : 도서 대여 및 반납
4- 서유빈 : 새로운 책 등록 및 삭제
5- 서준호 : 보고서 보완 및 스윙 디자인 다듬기
 */

public class Bookshelf extends JFrame {
    private HashMap<String, String> dataHashMap;
    public Bookshelf() {
        initiateSetting();
        setTitle("도서 관리 프로그램");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        Container contentPane = getContentPane();
        contentPane.setLayout(new BorderLayout());

        JPanel introPage = new JPanel();
        JLabel logo = new JLabel("3 조의 도서관리시스템 로고입니다.");
        introPage.add(logo);
        contentPane.add(introPage, BorderLayout.NORTH); // 여기까지
    }
}
```

```

JPanel mainPage = new JPanel(); // JFrame 의 contentPane 의 가운데에
들어갈 JPanel
contentPane.add(mainPage, BorderLayout.CENTER);

JPanel menus = new JPanel(new GridLayout(7,1,5, 5)); // 
contentPane 의 좌측에 삽입될, 버튼을 모아놓은 JPanel

JButton searchBook = new JButton("도서 검색");
searchBook.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        setVisible(false); // 이거 해줘야지 별 문제없이 제대로 안에있는
컴포넌트가 사라진게 "시각적으로 표현됨"
        mainPage.removeAll();
        JTextField jtf = new JTextField("찾는 책의 이름을 입력하세요", 35);
        mainPage.add(jtf);
        jtf.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String temp = ((JTextField)e.getSource()).getText();
                String mapValue = dataHashMap.get(temp);
                if(mapValue == null){
                    jtf.setText("해당하는 책이 없습니다.");
                }else {
                    jtf.setText(temp + ", " + mapValue);
                }
            }
        });
        setVisible(true);
    }
});
menus.add(searchBook);

JButton showBooks = new JButton("도서 목록 출력");
showBooks.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        setVisible(false); // 변경하는 과정에서 미리 안보이게 하고 다시 보이게
한다. 없어도 될듯?
        // 건의사항 : 해당 mainPage(JPanel)의 Layout 지정하는 것 / JPanel
은 기본적으로 FlowLayout 이라 옆으로 채운다
        // mainPage.setLayout(new GridLayout(0,1,0,0));
        mainPage.removeAll();
        mainPage.add(new JLabel("도서명 | 저자 | 출판사 | 대여가능여부"));

        try(FileReader fReader = new FileReader("bookData.txt");
            Scanner bookReader = new Scanner(fReader)) {

            while (bookReader.hasNext()) {
                mainPage.add(new JLabel(bookReader.nextLine()));
            }
        } catch (FileNotFoundException ept) {

```

```

        ept.printStackTrace();
    } catch (IOException ept) {
        ept.printStackTrace();
    }

    setSize(600, 600);
    setVisible(true);

}
});

menus.add(showBooks);

JButton rentalBook = new JButton("도서 대여하기");
menus.add(rentalBook);

JButton returnBook = new JButton("도서 반납하기");
menus.add(returnBook);

JButton wearBook = new JButton("입고 도서 등록");
menus.add(wearBook);

JButton deleteBook = new JButton("기존 도서 삭제하기");
menus.add(deleteBook);

JButton exit = new JButton("시스템 종료");
menus.add(exit);

contentPane.add(menus, BorderLayout.WEST);

setSize(600, 600);
setVisible(true);
}

private void initiateSetting(){ // 안상완 제작 메소드, 시작시에 파일로부터
데이터를 읽고, 해당 데이터를 분류하여 HashMap에 저장한다.
    dataHashMap = new HashMap<String, String>();
    try(FileReader fReader = new FileReader("bookData.txt"));
        Scanner bookReader = new Scanner(fReader)) {

        while (bookReader.hasNext()) {
            String[] temp = bookReader.nextLine().split(",");
            dataHashMap.put(temp[0],temp[1]);
        }
    } catch (FileNotFoundException ept) {
        ept.printStackTrace();
    } catch (IOException ept) {
        ept.printStackTrace();
    }
}

public static void main(String[] args) {
    Bookshelf bookshelf = new Bookshelf();
}
}

```

11/27/2021: 신현빈: 도서대출, 반납 기능 추가,

해당 기능 추가를 위한 추가적인 메소드, 필드 구현

변경사항:

「도서명,저자명;출판사;대여여부」에서 「도서명,저자명,출판사,대여여부」로 변경

```
명품 JavaProgramming,황기태/김효수,생능출판,대여가능  
Doit!자바프로그래밍입문,박은종,이지스퍼블리싱,대여가능
```

hashMap<String, String> 에서 hashMap<String, book> 으로 변경

```
package team3.assignment;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;
import java.util.HashMap;
import java.util.Map.Entry;
import java.util.Scanner;

/*
1- 손명진 : 사용할 기능 입력 및 도서 목록 조회
2- 안상완 : 도서 검색
3- 신현빈 : 도서 대여 및 반납
4- 서유빈 : 새로운 책 등록 및 삭제
5- 서준호 : 보고서 보완 및 스윙 디자인 다듬기
 */

public class Bookshelf extends JFrame {
    private HashMap<String, book> dataHashMap;

    public Bookshelf() {
        initiateSetting();
        setTitle("도서 관리 프로그램");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        Container contentPane = getContentPane();
        contentPane.setLayout(new BorderLayout());

        JPanel introPage = new JPanel();
        JLabel logo = new JLabel("3 조의 도서관리시스템 로고입니다.");
        introPage.add(logo);
        contentPane.add(introPage, BorderLayout.NORTH); // 여기까지

        컨텐트페인(JFrame에 JLabel을 넣은 JPanel을 NORTH에 삽입)

        JPanel mainPage = new JPanel(); // JFrame의 contentPane의 가운데에
```

```

들어갈 JPanel
    contentPane.add(mainPage, BorderLayout.CENTER);

    JPanel menus = new JPanel(new GridLayout(7,1,5, 5)); // 
contentPane 의 좌측에 삽입될, 버튼을 모아놓은 JPanel

    JButton searchBook = new JButton("도서 검색");
    searchBook.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            setVisible(false); // 이거 해줘야지 별 문제없이 제대로 안에있는
컴포넌트가 사라진게 "시각적으로 표현됨"
            mainPage.removeAll();
            JTextField jtf = new JTextField("찾는 책의 이름을 입력하세요", 35);
            mainPage.add(jtf);
            jtf.addActionListener(new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent e) {
                    String temp = ((JTextField)e.getSource()).getText();
                    book book = dataHashMap.get(temp);
                    if(book == null){
                        jtf.setText("해당하는 책이 없습니다.");
                    }else{
                        jtf.setText(book.toString());
                    }
                }
            });
            setVisible(true);
        }
    });
    menus.add(searchBook);

    JButton showBooks = new JButton("도서 목록 출력");
    showBooks.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            setVisible(false); // 변경하는 과정에서 미리 안보이게 하고 다시 보이게
한다. 없어도 될듯?
            // 건의사항 : 해당 mainPage(JPanel)의 Layout 지정하는 것 / JPanel
은 기본적으로 FLowLayout 이라 옆으로 채운다
            // mainPage.setLayout(new GridLayout(0,1,0,0));
            mainPage.removeAll();
            mainPage.add(new JLabel("도서명 | 저자 | 출판사 | 대여가능여부"));

            try(FileReader fReader = new FileReader("bookData.txt");
                Scanner bookReader = new Scanner(fReader)) {

                while (bookReader.hasNext()) {
                    mainPage.add(new JLabel(bookReader.nextLine()));
                }
            } catch (FileNotFoundException ept) {
                ept.printStackTrace();
            }
        }
    });
    menus.add(showBooks);
}

```

```

        } catch (IOException ept) {
            ept.printStackTrace();
        }

        setSize(600, 600);
        setVisible(true);
    }
}) ;
menus.add(showBooks);

JButton rentalBook = new JButton("도서 대여하기");
rentalBook.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        MainPage.removeAll();
        JTextField jtf = new JTextField("대여하실 책의 이름을
입력하세요", 35);
        MainPage.add(jtf);
        jtf.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String temp = ((JTextField)e.getSource()).getText();
                Book book = dataHashMap.get(temp);
                if(book == null) {
                    jtf.setText("해당하는 책이 없습니다.");
                } else {
                    if(book.isRental) jtf.setText("이미 대여중인
책입니다.");
                    else{
                        book.setRental(true);
                        saveData();
                        jtf.setText("대여완료되었습니다.");
                    }
                }
            }
        });
        setVisible(true);
    }
});
menus.add(rentalBook);

JButton returnBook = new JButton("도서 반납하기");
returnBook.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        MainPage.removeAll();
        JTextField jtf = new JTextField("반납하실 책의 이름을
입력하세요", 35);
        MainPage.add(jtf);
        jtf.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {

```

```

        String temp = ((JTextField)e.getSource()).getText();
        book book = dataHashMap.get(temp);
        if(book == null) {
            jtf.setText("해당하는 책이 없습니다.");
        } else {
            if(!book.Isrental) jtf.setText("대출한적이 없는
책입니다.");
            else{
                book.setRental(false);
                saveData();
                jtf.setText("반납완료되었습니다.");
            }
        }
    });
    setVisible(true);
}
);
menus.add(returnBook);

JButton wearBook = new JButton("입고 도서 등록");
menus.add(wearBook);

JButton deleteBook = new JButton("기존 도서 삭제하기");
menus.add(deleteBook);

JButton exit = new JButton("시스템 종료");
menus.add(exit);

contentPane.add(menus, BorderLayout.WEST);

setSize(600, 600);
setVisible(true);
}

public void saveData(){ // 신현빈 제작 메소드, 텍스트파일 변경
    try (FileWriter fileWriter = new FileWriter("bookData.txt")) {
        PrintWriter printWriter = new PrintWriter(fileWriter);
        for(Entry<String, book> entrySet : dataHashMap.entrySet()){
            printWriter.print(entrySet.getValue().toString() + "\n");
        }
    } catch (FileNotFoundException ept) {
        ept.printStackTrace();
    } catch (IOException ept) {
        ept.printStackTrace();
    }
}

private void initiateSetting(){ // 안상완 제작 메소드, 시작시에 파일로부터
데이터를 읽고, 해당 데이터를 분류하여 HashMap에 저장한다.
    dataHashMap = new HashMap<String, book>();
    try(FileReader fReader = new FileReader("bookData.txt")){
        Scanner bookReader = new Scanner(fReader);

        while (bookReader.hasNext()) {

```

```
        String[] temp = bookReader.nextLine().split(",");
        book b = new book(temp[0], temp[1], temp[2],
        (temp[3].equals("대여중")) ? true : false);
        dataHashMap.put(temp[0], b);
    }

} catch (FileNotFoundException ept) {
    ept.printStackTrace();
} catch (IOException ept) {
    ept.printStackTrace();
}
}

public static void main(String[] args) {
    Bookshelf bookshelf = new Bookshelf();
}
}

class book{
    private String name, writer, publisher;
    Boolean Isrental;

    public book(String name, String writer, String publisher, Boolean b) {
        this.name = name;
        this.writer = writer;
        this.publisher = publisher;
        this.Isrental = b;
    }

    public void setRental(Boolean b){
        Isrental = b;
    }

    @Override
    public String toString(){
        String s = name + "," + writer + "," + publisher + ",";
        if(Isrental) s += "대여중";
        else s += "대여가능";
        return s;
    }
}
```

11/30/2021: 서유빈: 입고 도서 등록 및 기존 도서 삭제

해당 기능 추가를 위한 추가적인 메소드, 필드 구현

```
package teamproject_4;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;
import java.util.HashMap;
import java.util.Map.Entry;
import java.util.Scanner;

/*
1- 손명진 : 사용할 기능 입력 및 도서 목록 조회
2- 안상완 : 도서 검색
3- 신현빈 : 도서 대여 및 반납
4- 서유빈 : 새로운 책 등록 및 삭제
5- 서준호 : 보고서 보완 및 스윙 디자인 담당
*/
public class Bookshelf extends JFrame {
    private HashMap<String, book> dataHashMap;

    public Bookshelf() {
        initiateSetting();
        setTitle("도서 관리 프로그램");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        Container contentPane = getContentPane();
        contentPane.setLayout(new BorderLayout());

        JPanel introPage = new JPanel();
        JLabel logo = new JLabel("3 조의 도서관리시스템 로고?입니다.");
        introPage.add(logo);
        contentPane.add(introPage, BorderLayout.NORTH); // 여기까지 컨텐트페인(JFrame에 JLabel을 넣은 JPanel을 NORTH에 삽입)

        JPanel mainPage = new JPanel(); // JFrame의 contentPane의 가운데에 들어갈 JPanel
        contentPane.add(mainPage, BorderLayout.CENTER);

        JPanel menus = new JPanel(new GridLayout(7,1,5, 5)); // contentPane의 좌측에 삽입될, 버튼을 모아놓은 JPanel

        JButton searchBook = new JButton("도서 검색");
        searchBook.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
```

```

        setVisible(false); // 이거 해줘야지 별 문제없이 제대로 안에있는 컴포넌트가 사라진게
    "시각적으로 표현됨"
        mainPage.removeAll();
        JTextField jtf = new JTextField("찾는 책의 이름을 입력하세요",35);
        mainPage.add(jtf);
        jtf.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String temp = ((JTextField)e.getSource()).getText();
                book book = dataHashMap.get(temp);
                if(book == null){
                    jtf.setText("해당하는 책이 없습니다.");
                }else {
                    jtf.setText(book.toString());
                }
            }
        });
        setVisible(true);
    });
    menus.add(searchBook);

 JButton showBooks = new JButton("도서 목록 출력");
 showBooks.addActionListener(new ActionListener() {
     @Override
     public void actionPerformed(ActionEvent e) {
         setVisible(false); // 변경하는 과정에서 미리 안보이게 하고 다시 보이게 한다. 없어도 될듯?
         // 건의사항 : 해당 mainPage(JPanel)의 Layout 지정하는 것 / JPanel 은 기본적으로
         FFlowLayout 이라 옆으로 채운다
         // mainPage.setLayout(new GridLayout(0,1,0,0));
         mainPage.removeAll();
         mainPage.add(new JLabel("도서명 | 저자 | 출판사 | 대여가능여부"));

         try(FileReader fReader = new FileReader("bookData.txt");
             Scanner bookReader = new Scanner(fReader)) {

             while (bookReader.hasNext()) {
                 mainPage.add(new JLabel(bookReader.nextLine()));
             }
         } catch (FileNotFoundException ept) {
             ept.printStackTrace();
         } catch (IOException ept) {
             ept.printStackTrace();
         }

         setSize(600, 600);
         setVisible(true);
     }
 })

```

```

    });
    menus.add(showBooks);

    JButton rentalBook = new JButton("도서 대여하기");
    rentalBook.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            setVisible(false);
            mainPage.removeAll();
            JTextField jtf = new JTextField("대여하실 책의 이름을 입력하세요",35);
            mainPage.add(jtf);
            jtf.addActionListener(new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent e) {
                    String temp = ((JTextField)e.getSource()).getText();
                    book book = dataHashMap.get(temp);
                    if(book == null){
                        jtf.setText("해당하는 책이 없습니다.");
                    }else {
                        if(book.Isrental) jtf.setText("이미 대여중인 책입니다.");
                        else{
                            book.setRental(true);
                            saveData();
                            jtf.setText("대여완료되었습니다.");
                        }
                    }
                }
            });
            setVisible(true);
        }
    });
    menus.add(rentalBook);

    JButton returnBook = new JButton("도서 반납하기");
    returnBook.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            setVisible(false);
            mainPage.removeAll();
            JTextField jtf = new JTextField("반납하실 책의 이름을 입력하세요",35);
            mainPage.add(jtf);
            jtf.addActionListener(new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent e) {
                    String temp = ((JTextField)e.getSource()).getText();
                    book book = dataHashMap.get(temp);
                    if(book == null){
                        jtf.setText("해당하는 책이 없습니다.");
                    }else {
                        if(!book.Isrental) jtf.setText("대출한적이 없는 책입니다.");
                    }
                }
            });
            setVisible(true);
        }
    });
    menus.add(returnBook);
}

```

```

        else{
            book.setRental(false);
            saveData();
            jtf.setText("반납완료되었습니다.");
        }
    }
});
setVisible(true);
}
});

menus.add(returnBook);

JButton wearBook = new JButton("입고 도서 등록"); //입고 도서 등록
wearBook.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        mainPage.removeAll();
        JTextField jtf = new JTextField("입고된 책의 제목, 저자, 출판사를 차례로 입력하세요",35);
        mainPage.add(jtf);
        jtf.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String temp = ((JTextField)e.getSource()).getText() + ", 대여가능"; //책 제목, 저자,
                출판사 입력값 가져오기
                try(FileWriter fWriter = new FileWriter("bookData.txt", true)){ //bookData에 새로운 책
                    등록
                    fWriter.write(temp);
                    fWriter.flush();
                }
                catch(IOException e1) {
                    e1.printStackTrace();
                }
            }
        });
        setVisible(true);
    }
});
menus.add(wearBook);

JButton deleteBook = new JButton("기존 도서 삭제하기"); //기존 도서 삭제
deleteBook.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        mainPage.removeAll();
        JTextField jtf = new JTextField("삭제하실 책의 제목을 입력하세요",35);
        mainPage.add(jtf);
        jtf.addActionListener(new ActionListener() {

```

```
    @Override
    public void actionPerformed(ActionEvent e) {
        String temp = ((JTextField)e.getSource()).getText();
        book book = dataHashMap.get(temp);
        if(book == null) {
            jtf.setText("해당하는 책이 없습니다.");
        }
        else {
            jtf.setText(book.toString()+"를 삭제합니다.");
            try {
                InputStream is = new FileInputStream("bookData.txt");
            }
            catch (FileNotFoundException e3) {
                e3.printStackTrace();
            }
            String result = "";
            try {
                BufferedReader br = new BufferedReader(new FileReader(new
File("bookData.txt")));
                BufferedWriter bw = new BufferedWriter(new FileWriter(new
File("newbookData.txt")));

                while((result = br.readLine()) != null) {
                    if(result.startsWith(temp)) {
                        result = result.replace(book.toString(), ""); //행 삭제
                    }
                    bw.write(result);
                    bw.flush();
                }
                bw.close();
                br.close();
                File file1 = new File("bookData.txt"); //원래 파일
                File file2 = new File("newbookData.txt"); //책을 삭제한 후의 파일
                file1.delete(); //원래 파일 삭제
                file2.renameTo(new File("bookData.txt")); //파일 이름 변경
            }
            catch(FileNotFoundException e1) {
                e1.printStackTrace();
            }
            catch(IOException e2) {
                e2.printStackTrace();
            }
            jtf.setText("삭제되었습니다.");
        }
    }

});  
setVisible(true);
```

```

        }
    });

menus.add(deleteBook);

JButton exit = new JButton("시스템 종료");
menus.add(exit);

contentPane.add(menus, BorderLayout.WEST);

setSize(600, 600);
setVisible(true);
}

public void saveData(){ // 신현빈 제작 메소드, 텍스트파일 변경
try (FileWriter fileWriter = new FileWriter("bookData.txt")){
    PrintWriter printWriter = new PrintWriter(fileWriter);
    for(Entry<String, book> entrySet : dataHashMap.entrySet()){
        printWriter.print(entrySet.getValue().toString() + "\n");
    }
} catch (FileNotFoundException ept) {
    ept.printStackTrace();
} catch (IOException ept) {
    ept.printStackTrace();
}
}

private void initiateSetting(){ // 안상완 제작 메소드, 시작시에 파일로부터 데이터를 읽고, 해당 데이터를
분류하여 HashMap에 저장한다.
dataHashMap = new HashMap<String, book>();
try(FileReader fReader = new FileReader("bookData.txt");
Scanner bookReader = new Scanner(fReader)) {

    while (bookReader.hasNext()) {
        String[] temp = bookReader.nextLine().split(",");
        book b = new book(temp[0], temp[1], temp[2], (temp[3].equals("대여중")) ? true : false);
        dataHashMap.put(temp[0], b);
    }
}

} catch (FileNotFoundException ept) {
    ept.printStackTrace();
} catch (IOException ept) {
    ept.printStackTrace();
}
}

public static void main(String[] args) {
    Bookshelf bookshelf = new Bookshelf();
}

}

class book{

```

```
private String name, writer, publisher;
Boolean Isrental;

public book(String name, String writer, String publisher, Boolean b) {
    this.name = name;
    this.writer = writer;
    this.publisher = publisher;
    this.Isrental = b;
}

public void setRental(Boolean b){
    Isrental = b;
}

@Override
public String toString(){
    String s = name + "," + writer + "," + publisher + ",";
    if(Isrental) s += "대여중";
    else s += "대여가능";
    return s;
}
}
```

2021/12/04 서준호 : 폰트 변경 및 버튼 디자인 수정

```
package expr;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;
import java.util.HashMap;
import java.util.Map.Entry;
import java.util.Scanner;

/*
1- 손명진 : 사용할 기능 입력 및 도서 목록 조회
2- 안상완 : 도서 검색
3- 신현빈 : 도서 대여 및 반납
4- 서유빈 : 새로운 책 등록 및 삭제
5- 서준호 : 보고서 보완 및 스윙 디자인 담당
 */

public class Bookshelf extends JFrame {
    private HashMap<String, book> dataHashMap;
    Font font = new Font("함초롱바탕", Font.BOLD, 17); // 폰트와 글씨 크기를
    새로 지정하는 폰트 삽입

    public Bookshelf() {

        initiateSetting();
        setTitle("도서 관리 프로그램");
        setSize(580, 800);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        Container contentPane = getContentPane();
        contentPane.setLayout(new BorderLayout());

        JPanel introPage = new JPanel();
        add(introPage);
        contentPane.add(introPage, BorderLayout.NORTH); // 여기까지
        컨텐트페인(JFrame에 JLabel을 넣은 JPanel을 NORTH에 삽입)

        JPanel mainPage = new JPanel(); // JFrame의 contentPane의 가운데에
        들어갈 JPanel
        contentPane.add(mainPage, BorderLayout.CENTER);

        JPanel menus = new JPanel(new GridLayout(7, 1, 5, 5)); // contentPane의 좌측에 삽입될, 버튼을 모아놓은 JPanel

        JButton searchBook = new JButton("도서 검색");

        searchBook.setFont(font); // 버튼 디자인
        searchBook.setBorderPainted(false);
        searchBook.setOpaque(true);
        searchBook.setBackground(new Color(255, 158, 0));

        searchBook.addActionListener(new ActionListener() {
```

```

@Override
public void actionPerformed(ActionEvent e) {
    setVisible(false); // 이거 해줘야지 별 문제없이 제대로 안에있는
컴포넌트가 사라진게 "시각적으로 표현됨"
    MainPage.removeAll();
    JTextField jtf = new JTextField("찾는 책의 이름을 입력하세요",
35);
    MainPage.add(jtf);
    jtf.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            String temp = ((JTextField) e.getSource()).getText();
            book book = dataHashMap.get(temp);
            if (book == null) {
                jtf.setText("해당하는 책이 없습니다.");
            } else {
                jtf.setText(book.toString());
            }
        }
    });
    setVisible(true);
}
});
menus.add(searchBook);

JButton showBooks = new JButton("도서 목록 출력");

showBooks.setFont(font);
showBooks.setBorderPainted(false);
showBooks.setOpaque(true);
showBooks.setBackground(new Color(255, 158, 0));

showBooks.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        setVisible(false); // 변경하는 과정에서 미리 안보이게 하고 다시 보이게
한다. 없어도 될듯?
        // 건의사항 : 해당 MainPage(JPanel)의 Layout 지정하는 것 / JPanel
은 기본적으로 FFlowLayout 이라 옆으로 채운다
        // MainPage.setLayout(new GridLayout(0,1,0,0));
        MainPage.removeAll();
        MainPage.add(new JLabel("도서명 | 저자 | 출판사 | 대여가능여부"));

        try (FileReader fReader = new FileReader("bookData.txt");
             Scanner bookReader = new Scanner(fReader)) {

            while (bookReader.hasNext()) {
                MainPage.add(new JLabel(bookReader.nextLine()));
            }
        } catch (FileNotFoundException ept) {
            ept.printStackTrace();
        } catch (IOException ept) {
            ept.printStackTrace();
        }

        setSize(580, 800);
        setVisible(true);
    }
});

```

```

        }
    });
    menus.add(showBooks);

    JButton rentalBook = new JButton("도서 대여하기");
    rentalBook.setFont(font);
    rentalBook.setBorderPainted(false);
    rentalBook.setOpaque(true);
    rentalBook.setBackground(new Color(255, 158, 0));

    rentalBook.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            setVisible(false);
            MainPage.removeAll();
            JTextField jtf = new JTextField("대여하실 책의 이름을 입력하세요",
35);
            MainPage.add(jtf);
            jtf.addActionListener(new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent e) {
                    String temp = ((JTextField) e.getSource()).getText();
                    Book book = dataHashMap.get(temp);
                    if (book == null) {
                        jtf.setText("해당하는 책이 없습니다.");
                    } else {
                        if (book.isRental) jtf.setText("이미 대여중인
책입니다.");
                    } else {
                        book.setRental(true);
                        saveData();
                        jtf.setText("대여완료되었습니다.");
                    }
                }
            });
            setVisible(true);
        }
    });
    menus.add(rentalBook);

    JButton returnBook = new JButton("도서 반납하기");
    returnBook.setFont(font);
    returnBook.setBorderPainted(false);
    returnBook.setOpaque(true);
    returnBook.setBackground(new Color(255, 158, 0));

    returnBook.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            setVisible(false);
            MainPage.removeAll();
            JTextField jtf = new JTextField("반납하실 책의 이름을 입력하세요",
35);
            MainPage.add(jtf);
            jtf.addActionListener(new ActionListener() {

```

```

@Override
public void actionPerformed(ActionEvent e) {
    String temp = ((JTextField) e.getSource()).getText();
    book book = dataHashMap.get(temp);
    if (book == null) {
        jtf.setText("해당하는 책이 없습니다.");
    } else {
        if (!book.Isrental) jtf.setText("대출한적이 없는
책입니다.");
        else {
            book.setRental(false);
            saveData();
            jtf.setText("반납완료되었습니다.");
        }
    }
}
setVisible(true);
})
);
menus.add(returnBook);

JButton wearBook = new JButton("입고 도서 등록"); //입고 도서 등록

wearBook.setFont(font);
wearBook.setBorderPainted(false);
wearBook.setOpaque(true);
wearBook.setBackground(new Color(255, 158, 0));

wearBook.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
    setVisible(false);
    mainPage.removeAll();
    JTextField jtf = new JTextField("입고된 책의 제목, 저자, 출판사를
차례로 입력하세요", 35);
    mainPage.add(jtf);
    jtf.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            String temp = ((JTextField) e.getSource()).getText() +
", 대여가능"; //책 제목, 저자, 출판사 입력값 가져오기
            try (FileWriter fWriter = new
FileWriter("bookData.txt", true)) { //bookData에 새로운 책 등록
                fWriter.write(temp);
                fWriter.flush();
            } catch (IOException e1) {
                e1.printStackTrace();
            }
        }
    });
    setVisible(true);
}
});
menus.add(wearBook);

JButton deleteBook = new JButton("기존 도서 삭제"); //기존 도서 삭제

```

```

deleteBook.setFont(font);
deleteBook.setBorderPainted(false);
deleteBook.setOpaque(true);
deleteBook.setBackground(new Color(255, 158, 0));

deleteBook.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        mainPage.removeAll();
        JTextField jtf = new JTextField("삭제하실 책의 제목을 입력하세요",
35);
        mainPage.add(jtf);
        jtf.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String temp = ((JTextField) e.getSource()).getText();
                book book = dataHashMap.get(temp);
                if (book == null) {
                    jtf.setText("해당하는 책이 없습니다.");
                } else {
                    jtf.setText(book.toString() + "를 삭제합니다.");
                    try {
                        InputStream is = new
FileInputStream("bookData.txt");
                    } catch (FileNotFoundException e3) {
                        e3.printStackTrace();
                    }
                    String result = "";
                    try {
                        BufferedReader br = new BufferedReader(new
FileReader(new File("bookData.txt")));
                        BufferedWriter bw = new BufferedWriter(new
FileWriter(new File("newbookData.txt")));
                        while ((result = br.readLine()) != null) {
                            if (result.startsWith(temp)) {
                                result = result.replace(book.toString(),
 ""));
                            }
                            bw.write(result);
                            bw.flush();
                        }
                        bw.close();
                        br.close();
                        File file1 = new File("bookData.txt"); //원래 파일
                        File file2 = new File("newbookData.txt"); //책을
삭제한 후의 파일
                        file1.delete(); //원래 파일 삭제
                        file2.renameTo(new File("bookData.txt")); //파일
이름 변경
                    } catch (FileNotFoundException e1) {
                        e1.printStackTrace();
                    } catch (IOException e2) {
                        e2.printStackTrace();
                    }
                    jtf.setText("삭제되었습니다.");
                }
            }
        });
    }
});

```

```

        }

    }

}) ;
setVisible(true) ;
}
}) ;
menus.add(deleteBook) ;
JButton exit = new JButton("시스템 종료") ;

exit.setFont(font) ;
exit.setBorderPainted(false) ;
exit.setOpaque(true) ;
exit.setBackground(new Color(255, 158, 0)) ;

menus.add(exit) ;

contentPane.add(menus, BorderLayout.WEST) ;

setSize(580, 800) ;
setVisible(true) ;

exit.addActionListener(new ActionListener() { // 서준호 메소드 보완,
종료 기능 구현
    @Override
    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
}) ;

}

public void saveData() { // 신현빈 제작 메소드, 텍스트파일 변경
    try (FileWriter fileWriter = new FileWriter("bookData.txt")) {
        PrintWriter printWriter = new PrintWriter(fileWriter);
        for (Entry<String, book> entrySet : dataHashMap.entrySet()) {
            printWriter.print(entrySet.getValue().toString() + "\n");
        }
    } catch (FileNotFoundException ept) {
        ept.printStackTrace();
    } catch (IOException ept) {
        ept.printStackTrace();
    }
}

private void initiateSetting() { // 안상완 제작 메소드, 시작시에 파일로부터
데이터를 읽고, 해당 데이터를 분류하여 HashMap에 저장한다.
    dataHashMap = new HashMap<String, book>();
    try (FileReader fReader = new FileReader("bookData.txt"));
        Scanner bookReader = new Scanner(fReader)) {

        while (bookReader.hasNext()) {
            String[] temp = bookReader.nextLine().split(",");

```

```

        book b = new book(temp[0], temp[1], temp[2],
(temp[3].equals("대여중")) ? true : false);
        dataHashMap.put(temp[0], b);
    }

} catch (FileNotFoundException ept) {
    ept.printStackTrace();
} catch (IOException ept) {
    ept.printStackTrace();
}
}

public static void main(String[] args) {
    Bookshelf bookshelf = new Bookshelf();
}
}

class book{
    private String name, writer, publisher;
    Boolean Isrental;

    public book(String name, String writer, String publisher, Boolean b) {
        this.name = name;
        this.writer = writer;
        this.publisher = publisher;
        this.Isrental = b;
    }

    public void setRental(Boolean b){
        Isrental = b;
    }

    @Override
    public String toString(){
        String s = name + "," + writer + "," + publisher + ",";
        if(Isrental) s += "대여중";
        else s += "대여가능";
        return s;
    }
}

```

2021 / 12 / 05 : 안상완

코드 정리, 새로운 책 등록 및 삭제 보완, 레이아웃 설정 등 마무리

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;
import java.util.HashMap;
import java.util.Map.Entry;
import java.util.Scanner;

```

```

/*
1- 손명진 : 사용할 기능 입력 및 도서 목록 조회
2- 안상완 : 도서 검색
3- 신현빈 : 도서 대여 및 반납
4- 서유빈 : 새로운 책 등록 및 삭제
5- 서준호 : 보고서 보완 및 스윙 디자인 다듬기
 */

public class Bookshelf extends JFrame {
    private HashMap<String, book> dataHashMap;
    private Font font = new Font("함초롱바탕", Font.BOLD, 17); // 폰트와 글씨
크기를 새로 지정하는 폰트 삽입
    private JButton searchBook, showBooks, rentalBook, returnBook,
wearBook, deleteBook, exit;
    private JPanel mainPage;
    // Color buttonColor = new Color(255, 158, 0);
// JButton[] jButtons = {searchBook, showBooks, rentalBook, returnBook,
wearBook, deleteBook, exit};
    public static void main(String[] args) {
        new Bookshelf();
    }
    public Bookshelf() {
        GUISetUp();
    }
    public void GUISetUp() {

        initiateSetting(); // HashMap 생성 및 초기화
        initiateComponentCondition(); // Bookshelf 클래스의 필드 버튼들 생성 및
설정
        ComponentListener(); // 컴포넌트 (버튼들)에 리스너 지정

        setTitle("도서 관리 프로그램");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        Container contentPane = getContentPane();
        contentPane.setLayout(new BorderLayout());

        JPanel introPage = new JPanel();
        contentPane.add(introPage, BorderLayout.NORTH); // 여기까지
컨텐트페인 (JFrame에 JLabel을 넣은 JPanel을 NORTH에 삽입)

        mainPage = new JPanel(); // JFrame의 contentPane의 가운데에 들어갈
JPanel
        contentPane.add(mainPage, BorderLayout.CENTER);

        JPanel menus = new JPanel(new GridLayout(7, 1, 5, 5)); // 
contentPane의 좌측에 삽입될, 버튼을 모아놓은 JPanel

        contentPane.add(menus, BorderLayout.WEST);
        menus.add(searchBook);

```

```

        menus.add(showBooks);
        menus.add(rentalBook);
        menus.add(returnBook);
        menus.add(wearBook);
        menus.add(deleteBook);
        menus.add(exit);

        setSize(700, 800);
        setVisible(true);
    }

    public void initiateComponentCondition() {
        searchBook = new JButton("도서 검색");
        showBooks = new JButton("도서 목록 출력");
        rentalBook = new JButton("도서 대여하기");
        returnBook = new JButton("도서 반납하기");
        wearBook = new JButton("입고 도서 등록"); //입고 도서 등록
        deleteBook = new JButton("기존 도서 삭제"); //기존 도서 삭제
        exit = new JButton("시스템 종료");

        JButton[] jButtons = {searchBook, showBooks, rentalBook, returnBook,
wearBook, deleteBook, exit};

        for(JButton buttonComponent : jButtons) {
            buttonComponent.setFont(font);
            buttonComponent.setBorderPainted(false);
            buttonComponent.setOpaque(true);
            buttonComponent.setBackground(new Color(255, 158, 0));
        }
    }

    public void ComponentListener() {
        searchBook.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                MainPage.removeAll();
                JPanel jPanel = new JPanel();
                jPanel.setLayout(new BoxLayout(jPanel,BoxLayout.Y_AXIS));
                JTextField jTextField = new JTextField("찾는 책의 이름을
입력하세요", 30);
                JTextArea jTextArea = new JTextArea("",10,30);
                jTextField.setFont(font);
                jTextArea.setFont(font);
                jPanel.add(jTextField);
                jPanel.add(new JScrollPane(jTextArea));
                MainPage.add(jPanel);
                jTextField.addActionListener(new ActionListener() {
                    @Override
                    public void actionPerformed(ActionEvent e) {
                        String temp = ((JTextField) e.getSource()).getText();
                        book book = dataHashMap.get(temp);
                        if (book == null) {
                            jTextArea.setText("해당하는 책이 없습니다.");
                        } else {
                            jTextArea.setText(book.toString());
                        }
                    }
                });
            }
        });
    }
}

```

```
        }
    });
    revalidate();
    repaint();
}
}

showBooks.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        MainPage.removeAll();
        JPanel jPanel = new JPanel();
        jPanel.setLayout(new BoxLayout(jPanel,BoxLayout.Y_AXIS));

        JLabel temp;
        temp = new JLabel("도서명 | 저자 | 출판사 | 대여가능여부");
        temp.setAlignmentX(CENTER_ALIGNMENT);
        jPanel.add(temp);
        temp.setFont(font);

        try (FileReader fReader = new FileReader("bookData.txt");
             Scanner bookReader = new Scanner(fReader)) {

            while (bookReader.hasNext()) {
                temp = new JLabel(bookReader.nextLine());
                temp.setFont(font);
                temp.setAlignmentX(CENTER_ALIGNMENT);
                jPanel.add(temp);
            }
        } catch (FileNotFoundException ept) {
            ept.printStackTrace();
        } catch (IOException ept) {
            ept.printStackTrace();
        }
        MainPage.add(jPanel);
        revalidate();
        repaint();
    }
});

rentalBook.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {

        MainPage.removeAll();
        JTextField jtf = new JTextField("대여하실 책의 이름을 입력하세요",
35);
        MainPage.add(jtf);
        jtf.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String temp = ((JTextField) e.getSource()).getText();
                book book = dataHashMap.get(temp);
                if (book == null) {
                    jtf.setText("해당하는 책이 없습니다.");
                }
            }
        });
    }
});
```

```

        } else {
            if (book.Isrental) jtf.setText("이미 대여중인
책입니다.");
        }
    }
}
);
revalidate();
repaint();
}
);

returnBook.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
mainPage.removeAll();
JTextField jtf = new JTextField("반납하실 책의 이름을 입력하세요",
35);
mainPage.add(jtf);
jtf.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
String temp = ((JTextField) e.getSource()).getText();
book book = dataHashMap.get(temp);
if (book == null) {
jtf.setText("해당하는 책이 없습니다.");
} else {
if (!book.Isrental) jtf.setText("대출한적이 없는
책입니다.");
else {
book.setRental(false);
saveData();
jtf.setText("반납완료되었습니다.");
}
}
}
});
revalidate();
repaint();
}
);

wearBook.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
mainPage.removeAll();
JTextField jtf = new JTextField("입고된 책의 제목, 저자, 출판사를
차례로 입력하세요", 35);
mainPage.add(jtf);
jtf.addActionListener(new ActionListener() {
@Override

```



```

PrintWriter("bookData.txt")) {
            for(String hashKey : dataHashMap.keySet()) {
                printWriter.println(dataHashMap.get(hashKey));
            }
        } catch (FileNotFoundException fileNotFoundException) {
            jtf.setText("FileNotFoundException 발생");
        }
        jtf.setText("정상적으로 \"" + temp + "\"를 삭제했습니다.");
    }
}

}); revalidate(); repaint();
}
});

exit.addActionListener(new ActionListener() { // 서준호 메소드 보완,
종료 기능 구현
@Override
public void actionPerformed(ActionEvent e) {
    System.exit(0);
}
});
}

public void saveData() { // 신현빈 제작 메소드, 텍스트파일 변경
try (FileWriter fileWriter = new FileWriter("bookData.txt")) {
    PrintWriter printWriter = new PrintWriter(fileWriter);
    for (Entry<String, book> entrySet : dataHashMap.entrySet()) {
        printWriter.print(entrySet.getValue().toString() + "\n");
    }
} catch (FileNotFoundException ept) {
    ept.printStackTrace();
} catch (IOException ept) {
    ept.printStackTrace();
}
}

private void initiateSetting() { // 안상완 제작 메소드, 시작시에 파일로부터
데이터를 읽고, 해당 데이터를 분류하여 HashMap에 저장한다.
dataHashMap = new HashMap<String, book>();
try (FileReader fReader = new FileReader("bookData.txt"));
Scanner bookReader = new Scanner(fReader)) {

    while (bookReader.hasNext()) {
        String[] temp = bookReader.nextLine().split(",");
        book b = new book(temp[0], temp[1], temp[2],
(temp[3].equals("대여중")) ? true : false);
        dataHashMap.put(temp[0], b);
    }
} catch (FileNotFoundException ept) {
    ept.printStackTrace();
} catch (IOException ept) {
}
}

```

```
        ept.printStackTrace();
    }
}
class book{
    private String name, writer, publisher;
    Boolean Isrental;

    public book(String name, String writer, String publisher, Boolean b) {
        this.name = name;
        this.writer = writer;
        this.publisher = publisher;
        this.Isrental = b;
    }

    public void setRental(Boolean b){
        Isrental = b;
    }

    @Override
    public String toString(){
        String s = name + "," + writer + "," + publisher + ",";
        if(Isrental) s += "대여중";
        else s += "대여가능";
        return s;
    }
}
```