프로그래밍 언어 응용
2조

서은빈 김준희 최서연 하태형

문제 1 - 1

```java
package Practice;

import java.util.ArrayList;

public class 문제1 {
    private static Scanner sc = new Scanner(System.in);
    private static List<String> wordlist = new ArrayList<String>();


    public static void sort (boolean isAsend) {
        // 오름차순 >> true
        // 내림차순 >> false
        if(isAsend == true) {
            Collections.sort(wordlist); // 리스트를 오름차순
        }else{
            Collections.sort(wordlist);
            Collections.reverse(wordlist);  // 오름차순 시킨 뒤에 역배치한다.
        }
    }

    public static void add(String word) {
        wordlist.add(word);
    }
    public static void show() {
        if(wordlist.isEmpty()) {
            System.out.println("저장된 단어가 없습니다. 새로운 단어를 추가해주세요");
```

```java
    public static void show() {
        if(wordlist.isEmpty()) {
            System.out.println("저장된 단어가 없습니다. 새로운 단어를 추가해주세요");
            return;
        }
        // 향상된 for 문을 이용하여 배열을 처음부터 끝까지 실행한다.
        for(String el : wordlist) {
            System.out.println(el);
        }
    }

    public static void main(String[] args) {

        int n = 0;
        while(true) {
            System.out.println("-----------MENU-----------");
            System.out.println("1 추가");
            System.out.println("2 정렬");
            System.out.println("3 확인");
            System.out.println("4 종료");
            System.out.println("-----------MENU-----------");
            System.out.print("번호 : ");
            n = sc.nextInt();
```

sort() 메서드

: 컬렉션을 사용하여 정렬을 한다

boolean값을 인자로 받으며, true 일 때는 오름차순, false 일 때는 내림차순을 한다.

내림차순은 오름차순을 한 뒤, 배열을 뒤집어 내림차순으로 만든다.

add() 메서드

: 리스트에 새로운 String(단어)을 추가한다.

show() 메서드

: list를 println을 통해 보여준다.

향상된 for문을 이용하여 list 처음부터 끝까지 반복해서 보여준다.

만약, list에 내용이 없다면 오류 메세지를 출력하고 메서드를 종료한다.

문제 1 - 2

```
n = sc.nextInt();

switch (n) {

case 1 :
    System.out.print("단어 입력 : ");
    String word = sc.next();
    add(word);
    break;

case 2 :
    System.out.print("오름차순 여부 (1: 오름차순, 0: 내림차순)");
    int no = sc.nextInt();
    if(no == 1) {
        sort(true);
    }else {
        sort(false);
    }
    break;

case 3 :
    show();
    break;

case 4 :
    System.out.println("종료합니다");
    sc.close();
    System.exit(-1);
    break;

default :
    System.out.println("다시 입력 하세요");
}
```

문제 1 결과

```
----------MENU----------
1 추가
2 정렬
3 확인
4 종료
----------MENU----------
번호 : 1
단어 입력 : barnana
----------MENU----------
1 추가
2 정렬
3 확인
4 종료
----------MENU----------
번호 : 1
단어 입력 : orange
----------MENU----------
1 추가
2 정렬
3 확인
4 종료
----------MENU----------
번호 : 1
단어 입력 : apple
----------MENU----------
1 추가
2 정렬
3 확인
4 종료
----------MENU----------
번호 : 3
barnana
orange
```

```
번호 : 3
barnana
orange
apple
----------MENU----------
1 추가
2 정렬
3 확인
4 종료
----------MENU----------
번호 : 2
오름차순 여부 (1: 오름차순, 0: 내림차순)1
----------MENU----------
1 추가
2 정렬
3 확인
4 종료
----------MENU----------
번호 : 3
apple
barnana
orange
----------MENU----------
1 추가
2 정렬
3 확인
4 종료
----------MENU----------
번호 : 2
오름차순 여부 (1: 오름차순, 0: 내림차순)0
----------MENU----------
1 추가
```

```
----------MENU----------
번호 : 3
apple
barnana
orange
----------MENU----------
1 추가
2 정렬
3 확인
4 종료
----------MENU----------
번호 : 2
오름차순 여부 (1: 오름차순, 0: 내림차순)0
----------MENU----------
1 추가
2 정렬
3 확인
4 종료
----------MENU----------
번호 : 3
orange
barnana
apple
----------MENU----------
1 추가
2 정렬
3 확인
4 종료
----------MENU----------
번호 : 4
종료합니다
```

sc.close();
: 스캐너를 닫아 리소스 해제한다.

문제 2-1

PUBLIC CLASS BOOKDTO 작성

- 전체 생성자
- toString
- getter and setter

```
// constructor
public BookDto(Long bookCode, String bookName, String publisher, String isbn) {
    super();
    this.bookCode = bookCode;
    this.bookName = bookName;
    this.publisher = publisher;
    this.isbn = isbn;
}

// getter and setter
public Long getBookCode() {
    return bookCode;
}

public void setBookCode(Long bookCode) {
    this.bookCode = bookCode;
}
```

```
public String getBookName() {
    return bookName;
}

public void setBookName(String bookName) {
    this.bookName = bookName;
}

public String getPublisher() {
    return publisher;
}

public void setPublisher(String publisher) {
    this.publisher = publisher;
}

public String getIsbn() {
    return isbn;
}

public void setIsbn(String isbn) {
    this.isbn = isbn;
}

// toString
@Override
public String toString() {
    return "BookDto [ bookCode = " + bookCode + ", bookName = " + bookName + ", publisher = " + publisher + ", isbn = " + isbn + " ]";
}
```

문제 2 - 2

```java
public static void conn() throws SQLException, ClassNotFoundException {
    // DB 연결 코드
    Class.forName("com.mysql.cj.jdbc.Driver");
    System.out.println("Driver Loading Success...");
    conn = DriverManager.getConnection(url, id, pw);
    System.out.println("DB CONNECTED...");

}
```

DB 연결 진행

```java
public static List<BookDto> selectAll() throws SQLException {
    // 전체 조회
    pstmt = conn.prepareStatement("select * from tbl_book");
    rs = pstmt.executeQuery();

    List<BookDto> allBook = new ArrayList();
    if (rs != null) {
        while (rs.next()) {
            Long bookCode = rs.getLong("bookCode");
            String bookName = rs.getString("bookName");
            String publisher = rs.getString("publisher");
            String isbn = rs.getString("isbn");
            allBook.add(new BookDto(bookCode, bookName, publisher, isbn));
        }
    }
    return allBook;
}
```

SELECTALL()
tbl_book 테이블에서 모든 정보를 조회합니다
새로운 allBook 리스트를 생성하고
조회한 데이터를 BookDto 객체로 변환하여 저장 한뒤
allBook을 리턴합니다

문제 2-3

```java
public static BookDto select(Long bookCode) throws SQLException {
    // 단건 조회
    pstmt = conn.prepareStatement("select * from tbl_book where bookCode = ?");
    pstmt.setLong(1, bookCode);
    rs = pstmt.executeQuery();

    if (rs.next()) {
        String bookName = rs.getString("bookName");
        String publisher = rs.getString("publisher");
        String isbn = rs.getString("isbn");

        return new BookDto(bookCode, bookName, publisher, isbn);
    }
    return null;
}

public static int insertBook(BookDto bookDto) throws SQLException {
    // 내용 입력
    pstmt = conn.prepareStatement("insert into tbl_book values(?,?,?,?)");
    pstmt.setLong(1, bookDto.getBookCode());
    pstmt.setString(2, bookDto.getBookName());
    pstmt.setString(3, bookDto.getPublisher());
    pstmt.setString(4, bookDto.getIsbn());

    return pstmt.executeUpdate();
}
```

```java
public static int updateBook(BookDto bookDto) throws SQLException {
    // 내용 수정
    String query = "update tbl_book set bookName = ?, publisher = ?, isbn = ? where bookCode = ?";
    pstmt = conn.prepareStatement(query);

    pstmt.setString(1, bookDto.getBookName());
    pstmt.setString(2, bookDto.getPublisher());
    pstmt.setString(3, bookDto.getIsbn());
    pstmt.setLong(4, bookDto.getBookCode());

    return pstmt.executeUpdate();
}

public static int deleteBook(Long bookCode) throws SQLException {
    // 내용 삭제
    pstmt = conn.prepareStatement("delete from tbl_book where bookCode = ?");
    pstmt.setLong(1, bookCode);

    return pstmt.executeUpdate();
}
```

SELECT()
BOOKCODE로 위치를 지정해
해당 행의 요소를 출력한다

INSERT()
GETTER로 각 변수에 접근해
요소를 불러와 입력한다

UPDATEBOOK()
GETTER로 각 변수에 접근해
요소를 불러와 입력한다

DELETEBOOK()
행을 특정 가능한 인자를 전달해
해당 행을 삭제한다

문제 2-4

MAIN 작성

TX

- setAutoCommit(false) 설정
- 코드 묶음이 끝나는 곳에서 .commit() 설정

자동 커밋을 비활성화 해
수동 커밋이 될 시점까지
저장이 되지 않도록 설정

- .rollback()

중간에 예외 상황 발생 시
코드 묶음이 원래 상태로
돌아 가도록 롤백 설정

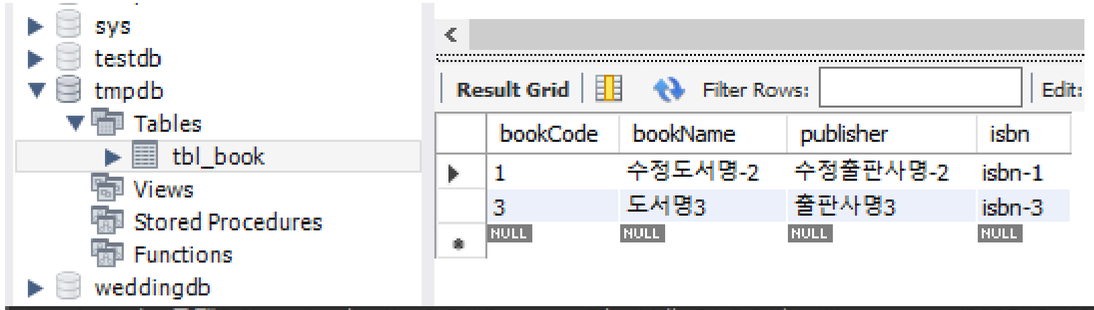
자원 해제

사용한 자원을 역순으로 해제해
반납한다

```java
public static void main(String[] args) {
    try {
        // DB Conn
        conn();

        // Tx start
        conn.setAutoCommit(false); // 자동 커밋 비활성화

        // Insert
        insertBook(new BookDto(1L, "도서명1", "출판사명1", "isbn-1"));
        insertBook(new BookDto(2L, "도서명2", "출판사명2", "isbn-2"));
        insertBook(new BookDto(3L, "도서명3", "출판사명3", "isbn-3"));

        // SelectAll
        List<BookDto> allBook = selectAll();
        System.out.println("selectAll : ");
        allBook.forEach(el -> System.out.println(el));

        // Select
        BookDto dto = select(1L);
        System.out.println("select : " + dto);

        // Update
        dto.setBookName("수정도서명-2");
        dto.setPublisher("수정출판사명-2");
        int r1 = updateBook(dto);
        if (r1 > 0)
            System.out.println("수정완료 : " + r1);

        // Delete
        int r2 = deleteBook(2L);
        if (r2 > 0)
            System.out.println("삭제완료 : " + r2);

        // Tx End
        conn.commit(); // 트랜잭션 커밋

    } catch (Exception e) {
        // Tx RollbackAll
        try {
            conn.rollback(); // 오류 발생시 롤백
        } catch (SQLException e1) {
            e1.printStackTrace(); // 롤백 오류 출력
        }
        e.printStackTrace(); // 메인 오류 출력
    } finally {
        // 자원해제
        try {
            rs.close();
        } catch (Exception e2) {}
        try {
            pstmt.close();
        } catch (Exception e2) {}
        try {
            conn.close();
        } catch (Exception e2) {}
    }
```

문제 2-5

출력 결과



Result Grid | Filter Rows: | Edit:

bookCode	bookName	publisher	isbn
1	수정도서명-2	수정출판사명-2	isbn-1
3	도서명3	출판사명3	isbn-3
NULL	NULL	NULL	NULL

```
<terminated> 문제2 [Java Application] C:₩Program Files₩jdk-21.0.2₩bin₩javaw.exe  (2025. 3. 19. 오후 2:25:58 – 오후 2:25:59) [pid: 6620]
Driver Loading Success...
3월 19, 2025 2:25:59 오후 io.opentelemetry.api.GlobalOpenTelemetry maybeAutoConfigureAndS
INFO: AutoConfiguredOpenTelemetrySdk found on classpath but automatic configuration is d
DB CONNECTED...
selectAll :
BookDto [ bookCode = 1, bookName = 도서명1, publisher = 출판사명1, isbn = isbn-1 ]
BookDto [ bookCode = 2, bookName = 도서명2, publisher = 출판사명2, isbn = isbn-2 ]
BookDto [ bookCode = 3, bookName = 도서명3, publisher = 출판사명3, isbn = isbn-3 ]
select : BookDto [ bookCode = 1, bookName = 도서명1, publisher = 출판사명1, isbn = isbn-1 ]
수정완료 : 1
삭제완료 : 1
```