



# **BBM 487 – SOFTWARE ENGINEERING LABORATORY**

## **LIBRARY BOOK LOAN SYSTEM**

### **Coding Standards**

#### **GROUP II**

**Özlem DEMİRTAŞ            21327901**

**Umut ÖZTÜRK            21328394**

**Rahmi Berk ŞEFKATLİ    21427402**

Coding standards define a programming style. A coding standard does not usually concern itself with wrong or right in a more abstract sense. It is simply a set of rules and guidelines for the formatting of source code.

### Common Aspects of a Coding Standard:

- Naming Conventions
- Declarations
- Indentation
- Statements
- Comments

The coding standards and naming conventions in the Library Book Loan System are per the Java coding standards.

**Example codes from our project according to the standards** are seen below.

#### 1. Naming Conventions

##### Classes

Class names in our project are nouns and in mixed case the first letter of each internal word capitalized.

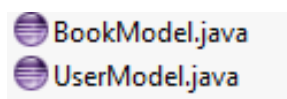


Figure 1 Class name examples

##### Methods

Method names in our project are verbs and in mixed case the first letter is lowercase and the first letter of each internal word capitalized.

```
public void getBookByCondition(String condition,String value)
        public void returnBook()
        private void initialize()
```

Figure 2 Method name examples

## Variables

In our project the first letter of the variable name is lowercase and the first letter of each internal word is capitalized.

Also, variable names are chosen according to intent of its use.

```
private int id;  
private String username;  
private String password;  
private String name;  
private String surname;  
private String email;  
private boolean isLibrarian;
```

Figure 3 Variable name examples

## 2. Declarations

### Number per line

One variable declaration per line is tried to use for readability.

```
private int barcode_no;  
private String name;  
private String author_name;  
private int printing_year;  
private int number_of_pages;  
private int count;
```

Figure 4 Declaration example

### Placement

Generally, declarations are put only at the beginning of blocks.

### Initialization

Local variables are initialized where they're declared.

```

Connection con=null;
PreparedStatement st=null;
ResultSet rs=null;

String url="jdbc:mysql://localhost:3306/tutorial";
String username="root";
String password="umut";

try{
    Class.forName("com.mysql.jdbc.Driver");
    con=DriverManager.getConnection(url,username,password);
    String parameter="";
    if(condition.equals("Author")){
        parameter="author_name = '"+value+"'";
    }else if(condition.equals("Name")){
        parameter="name = '"+value+"'";
    }else{
        parameter="barcode_no = '"+Integer.parseInt(value)+"'";
    }
}

```

Figure 5 Placement and initialization example

## Class and Interface Declarations

There is no space between a method name and parenthesis.

Open brace is placed at the end of the declaration statement.

```

public class BookModel {
    private int barcode_no;
    private String name;
    private String author_name;
    private int printing_year;
    private int number_of_pages;
    private int count;

    public BookModel(int barcode_no, String name, String author_name, int printing_year, int number_of_pages, int count) {
        this.barcode_no = barcode_no;
        this.name = name;
        this.author_name = author_name;
        this.printing_year = printing_year;
        this.number_of_pages = number_of_pages;
        this.count = count;
    }

    public String toString(){
        return name + ", " + author_name + ", " + barcode_no;
    }
}

```

Figure 6 Class declaration example

### 3. Indentation

#### Line Length

In our project, line lengths are tried to keep the under 80 characters.

### 4. Statements

#### Simple Statements

Each line contains one statement.

```
selectedBook = (BookModel)source.getSelectedValue();
if(selectedBook== null){
    return;
}
```

Figure 7 Statement example

#### Compound Statements

Braces are used to control the structure and to add statements easily.

```
isInStock = checkStock(selectedBook);
if(isInStock){
    textArea.setText("Chosen book is available, you may proceed to checkout.");
    btnAddNameTo.setEnabled(false);
    btnSelfcheckout.setEnabled(true);
}else{
    textArea.setText("Chosen book is currently loaned to someone else, you may add your name to waiting list.");
    btnAddNameTo.setEnabled(true);
    btnSelfcheckout.setEnabled(false);
}
```

Figure 8 Compound statement example

#### return Statements

Generally, return value is not used with parentheses.

```
public Object getElementAt(int index) {
    return values[index];
}
```

Figure 9 return Statement example

### if statements

In our project, even if block contains one statement, braces are used.

```
if(condition==null || value==null){
    return;
}
```

Figure 10 if statement example

### try-catch statements

```
try {
    CreateMember window = new CreateMember();
    window.frame.setVisible(true);
} catch (Exception e) {
    e.printStackTrace();
}
```

Figure 11 try-catch statement example

## 5. Comments

### Block Comments

In our project, block comments are generally used to describe the method.

```
/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                CreateMember window = new CreateMember();
                window.frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}
```

Figure 12 Block comment example

```
/**  
 * Create the application.  
 */  
public CreateMember() {  
    initialize();  
}
```

Figure 13 Block comment example

### Single-line Comments

Single-line comments are preceded by a blank line.

```
JFrame selfCheckoutFr = selfCheckout.getFrame();  
selfCheckoutFr.setVisible(true);  
  
/*if the book is not in the stock*/  
if(selectedBook == null){  
    return;  
}
```

Figure 14 Single line comment example