

SWE 4202: OBJECT ORIENTED CONCEPTS I LAB

LAB_10: Wrapper Classes

PREPARED BY :

FARZANA TABASSUM || LECTURER
farzana@iut-dhaka.edu

ZANNATUN NAIM SRISTY || LECTURER
zannatunnaim@iut-dhaka.edu

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ISLAMIC UNIVERSITY OF TECHNOLOGY

APRIL 22, 2024

1 Wrapper Classes

The wrapper classes in Java are used to convert primitive types (*int*, *char*, *float*, etc) into corresponding objects. This is necessary because in Java, everything is an object, and objects are required in certain situations, such as when a method requires an object as a parameter or when you need to store multiple values in a single structure like an *ArrayList* or a *HashMap*.

Primitive Data Type	Wrapper Class
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
boolean	Boolean

Table 1: Primitive Data Types and their Corresponding Wrapper Class

1.1 Autoboxing and Unboxing

Autoboxing:

The automatic conversion of primitive types to the object of their corresponding wrapper classes is known as autoboxing. For example – conversion of *int* to *Integer*, *long* to *Long*, *double* to *Double*, etc.

```
1 import java.util.ArrayList;
2
3 class Main {
4     public static void main(String[] args) {
5
6         // create primitive types
7         int a = 5;
8         double b = 5.65;
9
10        //converts into wrapper objects
11        Integer aObj = Integer.valueOf(a);
12        Double bObj = b; //automatic conversion
13
14        if(aObj instanceof Integer) {
15            System.out.println("An object of Integer is created.");
16        }
17
18        if(bObj instanceof Double) {
19            System.out.println("An object of Double is created.");
20        }
21
22        ArrayList<Integer> arrayList = new ArrayList<Integer>();
23
24        // Autoboxing because ArrayList stores only objects
25        arrayList.add(25);
26
27        // printing the values from object
28        System.out.println(arrayList.get(0));
29    }
30 }
```

Output:

```
An object of Integer is created.  
An object of Double is created.  
25
```

Unboxing:

It is just the reverse process of autoboxing. Automatically converting an object of a wrapper class to its corresponding primitive type is known as unboxing. For example – conversion of Integer to int, Long to long, Double to double, etc.

```
1 import java.util.ArrayList;  
2  
3 class Main {  
4     public static void main(String[] args) {  
5  
6         // creates objects of wrapper class  
7         Integer aObj = Integer.valueOf(23);  
8         Double bObj = Double.valueOf(5.55);  
9  
10        // converts into primitive types  
11        int a = aObj.intValue();  
12        double b = bObj; //automatic conversion  
13  
14        System.out.println("The value of a: " + a);  
15        System.out.println("The value of b: " + b);  
16  
17        ArrayList<Integer> arrayList = new ArrayList<Integer>();  
18        arrayList.add(24);  
19  
20        // unboxing because get method returns an Integer object  
21        int num = arrayList.get(0);  
22  
23        // printing the values from primitive data types  
24        System.out.println(num);  
25    }  
26 }
```

Output:

```
The value of a: 23  
The value of b: 5.55  
24
```

2 Problem Statement 01

Your task involves managing online book shop. Each book is represented by a Book object containing attributes such as title, author, and genre. The genre is defined using an enum called Genre, which includes options like FICTION, SCIENCE_FICTION, MYSTERY, ROMANCE and others. After a book is added to the shop, you can add ratings from the main method. There is another enum called MenuOption which allows user to perform some actions.

You have to convert the **if_else** statement to **Switch cases**.

```
1 private static Scanner scanner = new Scanner(System.in);  
2  
3 public static void main(String[] args) {  
4     MenuOption selectedOption;  
5  
6     do {  
7         displayMenu();
```

```
8      int option = scanner.nextInt();
9      selectedOption = MenuOption.values()[option - 1];
10
11     if (selectedOption == MenuOption.ADD_NEW_BOOK) {
12         addNewBook();
13     } else if (selectedOption == MenuOption.SEARCH_FOR_BOOK) {
14         searchForBookByTitle("To Kill a Mockingbird");
15     } else if (selectedOption == MenuOption.DISPLAY_ALL_BOOKS) {
16         displayAllBooks();
17     } else if (selectedOption == MenuOption.ADD_RATINGS) {
18         addBookRatings("To Kill a Mockingbird");
19     } else if (selectedOption == MenuOption.EXIT) {
20         System.out.println("Exiting program.");
21     } else {
22         System.out.println("Invalid option");
23     }
24 } while (selectedOption != MenuOption.EXIT);
25
26 // Close scanner to release resources
27 scanner.close();
28 }
29
30 private static void displayMenu() {
31     System.out.println("Choose an option:");
32     System.out.println("1. Add a new book");
33     System.out.println("2. Search for a book");
34     System.out.println("3. Display all books");
35     System.out.println("4. Add ratings to a book");
36     System.out.println("5. Exit");
37 }
```

Final Output:

Displaying all books:

Book 1:

Title: To Kill a Mockingbird

Author: Harper Lee

Genre: FICTION

Average Ratings: 4.0

Book 2:

Title: The Great Gatsby

Author: F Scott Fitzgerald

Genre: ROMANCE

Average Ratings: 4.2