

Meeting Agenda

Date: November 2, 2024

Time: 7.30 pm

Platform: Google Meet

Attendees: Sandhiya Suresh, Dharani G, Deepak Kumar, Hari Priya

1. Introduction

- Importance of understanding different approaches to palindrome checks in Java.

2. Discussion Topics

- Code Review of Two Palindrome Implementations:
- Implementation Using == Operator:
- Explanation of how the == operator compares object references in Java.
- Example code :

```
java
public class Main {
    public static void main(String[] args) {
        String str = "radar";
        String reversed = new StringBuilder(str).reverse().toString();
        if (str == reversed) {
            System.out.println(str + " is a Palindrome.");
        } else {
            System.out.println(str + " is not a Palindrome.");
        }
    }
}
```

- Discussion on potential pitfalls of using == for string comparison.
- Implementation Using .equals():
- Explanation of the .equals() method for content comparison.
- Example code snippet:

```
java
public class Main {
    public static void main(String[] args) {
        String str = "radar";
        String reversed = new StringBuilder(str).reverse().toString();
        if (str.equals(reversed)) {
            System.out.println(str + " is a Palindrome.");
        } else {
            System.out.println(str + " is not a Palindrome.");
        }
    }
}
```

- Benefits of using .equals() over ==.
- User Input Implementation with Scanner:
- Example code snippet:

```
java
import java.util.Scanner;

public class PalindromeChecker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter a string: ");
        String input = scanner.nextLine();

        String reversed = new StringBuilder(input).reverse().toString();

        if (input.equals(reversed)) {
            System.out.println(input + " is a palindrome.");
        } else {
            System.out.println(input + " is not a palindrome.");
        }
    }
}
```

- Explanation:
- 1. The program prompts the user to enter a string.

2. It reverses the input string using `StringBuilder`.
3. It compares the original string with the reversed string using `.equals()`.
4. If they match, it prints that the string is a palindrome.
 - Why Use `.equals()` Instead of `==`:
 - In Java:
 - `==` checks for reference equality (i.e., whether both objects point to the same memory location).
 - `.equals()` checks for content equality (i.e., whether the values of the objects are equal).
 - For strings, we want to check content equality, so we use `.equals()`. Using `==` would return incorrect results.

4. Example Output

```
text
Enter a string:
madam
madam is a palindrome.
```

- Java Array Merging Problem:
- I practiced a coding problem on LeetCode that involved merging two sorted integer arrays.
- The task was to merge `nums1` and `nums2` into a single sorted array stored in `nums1`.
- Here is the code I implemented:

```
java
public class Solution {
    public void merge(int[] nums1, int m, int[] nums2, int n) {
        int i = m - 1; // Pointer for nums1
        int j = n - 1; // Pointer for nums2
        int k = m + n - 1; // Pointer for the last position in nums1

        while (i >= 0 && j >= 0) {
            if (nums1[i] > nums2[j]) {
                nums1[k--] = nums1[i--];
            } else {
                nums1[k--] = nums2[j--];
            }
        }

        while (j >= 0) {
            nums1[k--] = nums2[j--];
        }
    }

    public static void main(String[] args) {
        Solution solution = new Solution();

        // Test Case
        int[] nums1 = {1, 2, 3, 0, 0, 0};
        int m = 3;
        int[] nums2 = {2, 5, 6};
        int n = 3;

        solution.merge(nums1, m, nums2, n);

        // Print merged array for verification
        System.out.print("Merged array: ");
        for (int num : nums1) {
            System.out.print(num + " ");
        }
    }
}
```

- Outcome:
- Successfully merged the arrays and verified the output.
- The expected output for the test case was [1, 2, 2, 3, 5, 6].