NAME- Harshita Totala ROLL NO- SEITA14 DIV- IT A

SUBJECT- Object Oriented Programming

ASSIGNMENT NO. 7

Problem Statement:Implement a generic program using any collection class to count the number of elements in a collection that have a specific property such as even numbers, odd number, prime number and palindromes.

Objectives: To learn the concept templates and generic programming

Theory:

1. Java Generic methods

- Syntax to declare class
- Instance variable in Java
- Method in Java
- 'new' keyword in Java

2. Generic classes

- multiple Type parameters

3. Advantages of Generics:

Generic Types

✓ Generic type represents classes, interfaces and methods in a type safe manner

✓ Generic types can act on any type of data

✓ All Generic types are subclasses of Object class, it acts on Objects only

✓ Generic types act on advanced data type only

✓ It is not possible to create an object to Generic type itself

✓ Using generic types, we can avoid casting in many cases Generic Class:

When we create a class with an instance variable to store an Integer object, it can be used to store Integer type data only

We cannot use that instance variable to store a Float class object or a String type Object To store different types of data into a class, we have to write the same class again and again by changing the data type of the variables. This can be avoided using a generic class A generic class represents a class that is type-safe. This means a generic class can act upon any data type Generic classes and generic interfaces are also called „parameterized types" because they use a parameter that determines which data type they should work upon

Generic Method: We can make a method alone as generic method by writing the generic

parameter before the method return type as:

returntypemethodname ()

{

Method code;

}

eg: void display_data () { Method body; }

Generic Interface:

It is possible to develop an interface using generic type concept. The general form of generic

interface looks like:

interface interface_name

{ //method that accepts any object return_typemethod_name (T object_name); }

Here, T represents any data type which is used in the interface.

We can write an implementation class for the above interface as: class class_name implements

interface_name

{ public return_typemethod_name (T object_name)

{ //provide body of the method }

}

Program:

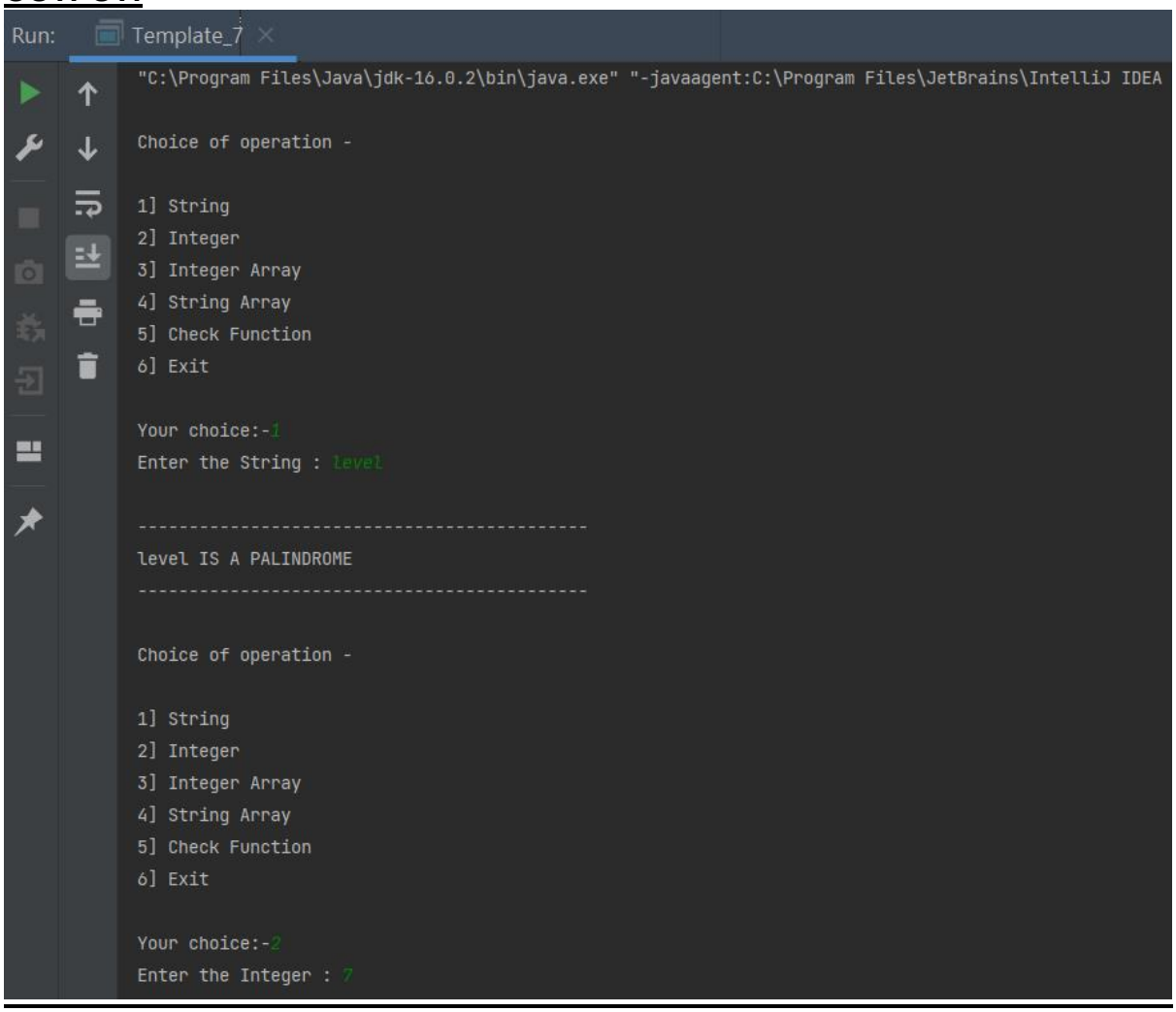
```
import java.util.Objects;
import java.util.Scanner;

class Number{

    public static < T > void integer_arr( )
    {
        Scanner sc =new Scanner(System.in);
        System.out.print("Enter size of array:- ");
        int n=sc.nextInt();

        Object[] arr=new Object[n];

        System.out.print("Enter Array Elements :");
```

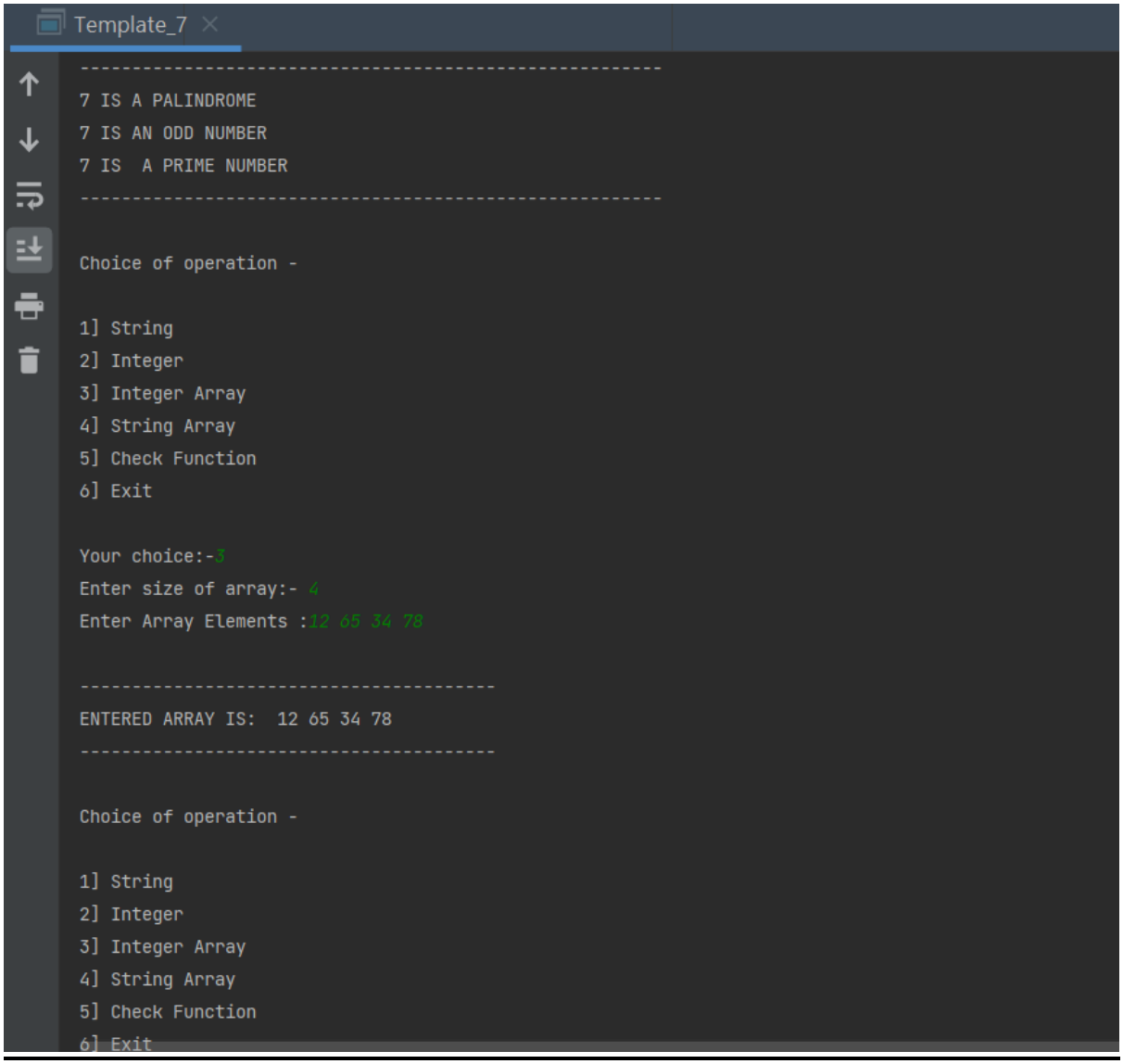
```
        for(int i =0; i<n ;i++) {
            arr[i]=sc.next();
        }
        System.out.println("\n----------------------------------------");
        System.out.print("ENTERED ARRAY IS: ");
        for(int i =0; i<n ;i++) {
            System.out.print(" "+arr[i]);
        }
        System.out.println("\n----------------------------------------");

    }

    public static < T > void Palindrome(T s)
    {
        String s1 =(String)s;
        s1=s1.toLowerCase();

        String  reverse = "";
        int length = s1.length();

        for ( int i = length-1 ; i >=0; i-- )
            reverse = reverse + s1.charAt(i);

        if (s1.equals(reverse))
            System.out.println(s+" IS A PALINDROME");
        else
            System.out.println(s+" IS NOT A PALINDROME");
    }



    public static <T>void even_odd(T a)
    {
        if((int)a%2==0)
            System.out.println(a+" IS AN EVEN NUMBER");

        else
            System.out.println(a+" IS AN ODD NUMBER");
    }



    public static <T>void prime_no(T a)
    {
        int flag=0,n=2;
        while(n<= (int)a/2)
        {
            if ((int) a % 2 == 0) {
                flag++;
                break;
            }
            n++;
        }
                if(flag==1) {
                    System.out.println(a + " IS NOT A PRIME NUMBER");
                }
                else{
                    System.out.println(a+" IS  A PRIME NUMBER");
                }
            }
```

```
    public static <T>void check_func(T s) {

        try {
            int x = Integer.parseInt((String) s);
            System.out.println("\n-----------------------------------------
----------------------------------------");
            System.out.println("WE CAN PERFORM PALINDROME, INTEGER ARRAY,
PRIME , EVEN_ODD NUMBER");
            System.out.println("-----------------------------------------
----------------------------------------");
        }

        catch (NumberFormatException e) {
            System.out.println("\n-----------------------------------------
----------------------------------------");
            System.out.println("WE CAN PERFORM STRING ARRAY, PALINDROME");
            System.out.println("-----------------------------------------
----------------------------------------");
        }
    }
}



public class Template_7 {

    public static void main(String[] args) {
        String s;
        int ch;
        Scanner sc =new Scanner(System.in);
        do{
            System.out.println("\nChoice of operation -");
            System.out.println("\n1] String\n2] Integer\n3] Integer Array
\n4] String Array\n5] Check Function\n6] Exit");
            System.out.print("\nYour choice:-");
            ch =sc.nextInt();


            switch(ch) {

                case 1:
                    System.out.print("Enter the String : ");
                    s =sc.next();
                    System.out.println("\n-----------------------------------
-----------");
                    Number.Palindrome(s);
                    System.out.println("-----------------------------------
---------");
                    break;

                case 2:
                    System.out.print("Enter the Integer : ");
                    s =sc.next();
                    System.out.println("\n-----------------------------------
------------------------");
                    Number.Palindrome(s);
                    Number.even_odd(Integer.parseInt(s));
                    Number.prime_no(Integer.parseInt(s));
```

```
                    System.out.println("-----------------------------------
----------------------");
                    break;


                case 3:

                case 4:
                    Number.integer_arr();
                    break;

                case 5:
                    System.out.print("Enter the String : ");
                    s =sc.next();
                    Number.check_func(s);
                    break;

                case 6:
                    System.out.println("Program Exited !!!\n");
                    break;

                default:
                    System.out.println("Wrong Choice! Enter again...\n");

            }
        }while(ch!=6);
    }

}
```

OUTPUT:

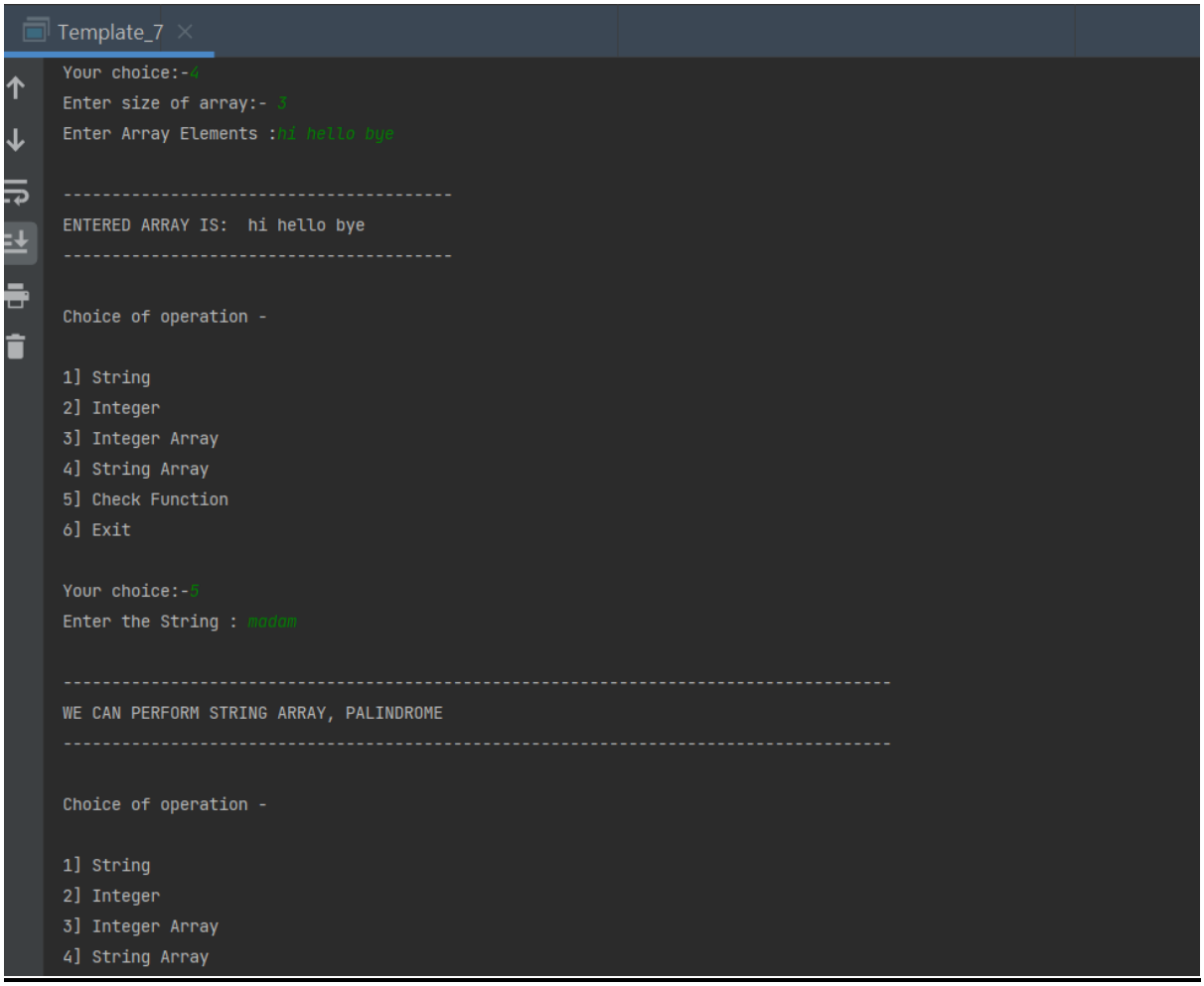

Run:
Template_7
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
Choice of operation -
1] String
2] Integer
3] Integer Array
4] String Array
5] Check Function
6] Exit
Your choice:-1
Enter the String : level
level IS A PALINDROME
Choice of operation -
1] String
2] Integer
3] Integer Array
4] String Array
5] Check Function
6] Exit
Your choice:-2
Enter the Integer : 7

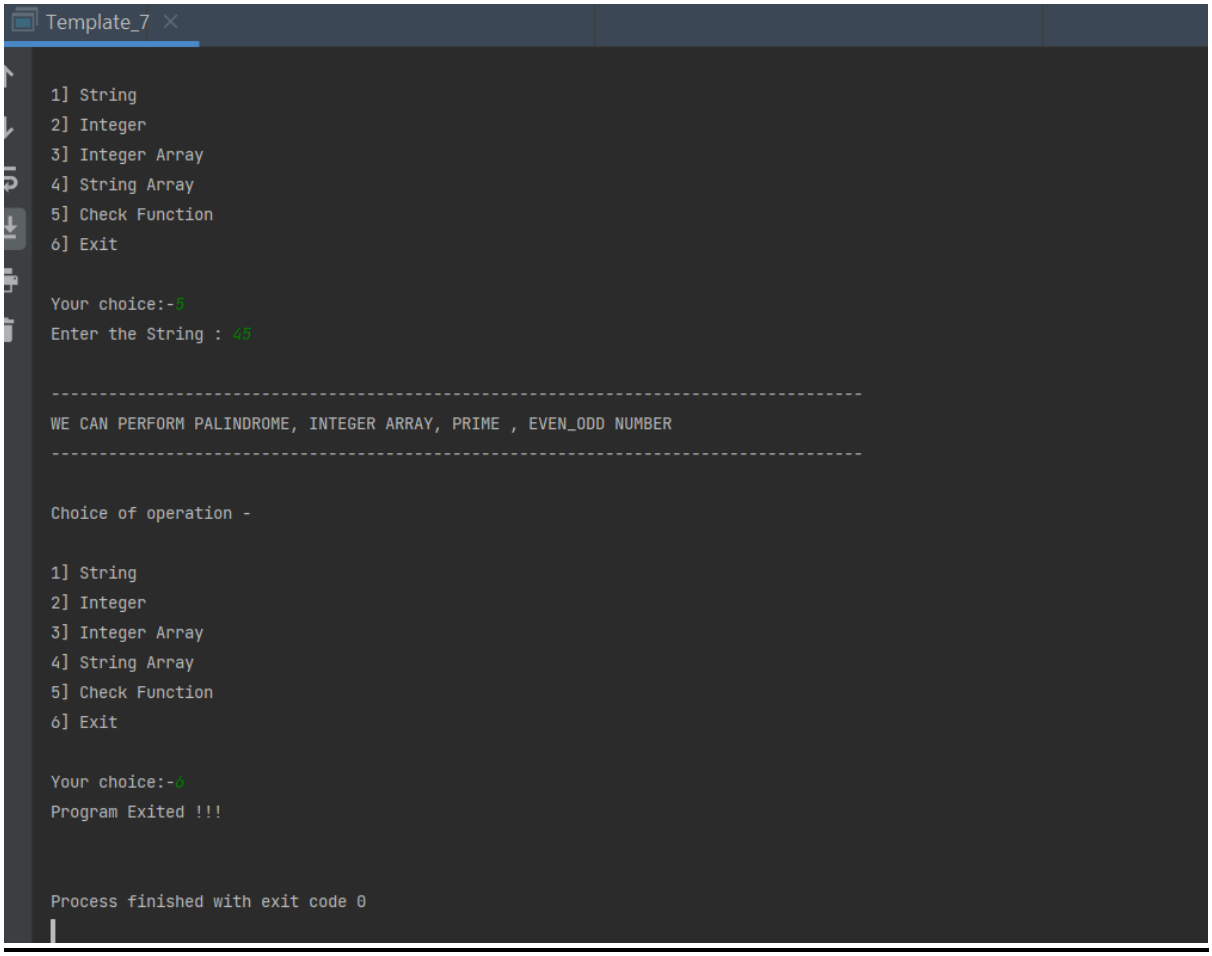

Template_7
---
7 IS A PALINDROME
7 IS AN ODD NUMBER
7 IS A PRIME NUMBER

Choice of operation -
1] String
2] Integer
3] Integer Array
4] String Array
5] Check Function
6] Exit
Your choice:-3
Enter size of array:- 4
Enter Array Elements :12 65 34 78

ENTERED ARRAY IS: 12 65 34 78

Choice of operation -
1] String
2] Integer
3] Integer Array
4] String Array
5] Check Function
6] Exit


Template_7
Your choice:-4
Enter size of array:- 3
Enter Array Elements :hi hello bye
ENTERED ARRAY IS: hi hello bye
Choice of operation -
1] String
2] Integer
3] Integer Array
4] String Array
5] Check Function
6] Exit
Your choice:-5
Enter the String : madam
WE CAN PERFORM STRING ARRAY, PALINDROME
Choice of operation -
1] String
2] Integer
3] Integer Array
4] String Array



Conclusion: Here, we have successfully implemented the concept of templates and generic programming.