UNIVERSITY INSTITUTE OF ENGINEERING

Subject Name –Project Based Learning In Java

Subject Code: 23CSH-304

Submitted To: **Submitted By:**

Faculty Name: - Deep Prakash Gupta **Name:** Harjit Singh

UID: 23BCS10849

Project Report: Development of Java Programs for String Analysis, Matrix Operations, and Basic Banking System

Aim: Develop Java programs to analyze strings, perform matrix operations, and implement basic banking system functionality.

Easy Level

Objective: To understand string manipulation in Java.

Procedure

Step1: Prompt the user to enter a string.
Step2: Traverse each character in the string.
Step3: Classify each character using conditions:

- If the character is a vowel (a, e, i, o, u), increment the vowel count.
- If it is a consonant (alphabetic and not a vowel), increment the consonant count.
- If it is a digit (0–9), increment the digit count.
- If it is none of the above and not a space, it is a special character.

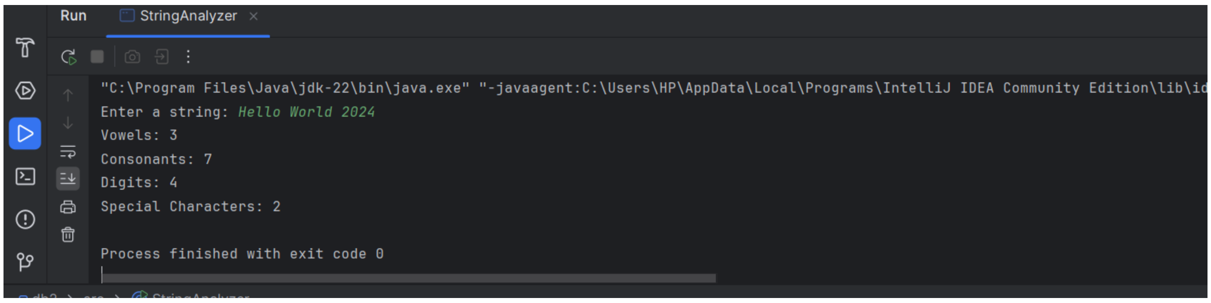
Step4: Print the counts of vowels, consonants, digits, and special characters.

Code

```
import java.util.Scanner;

public class StringAnalyzer {

  public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);

    System.out.print("Enter a string: ");

    String str = sc.nextLine();

    int vowels = 0, consonants = 0, digits = 0, special = 0;

    for (char c : str.toCharArray()) {

      if (Character.isLetter(c)) {

        char lc = Character.toLowerCase(c);

        if (lc == 'a' || lc == 'e' || lc == 'i' || lc == 'o' || lc == 'u') {

          vowels++;
```

```
                } else {
                    consonants++;
                }
            } else if (Character.isDigit(c)) {
                digits++;
            } else {
                special++;
            }
        }
        System.out.println("Vowels: " + vowels);
        System.out.println("Consonants: " + consonants);
        System.out.println("Digits: " + digits);
        System.out.println("Special Characters: " + special);
    }
}
```

Output



```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Users\HP\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\id
Enter a string: Hello World 2024
Vowels: 3
Consonants: 7
Digits: 4
Special Characters: 2

Process finished with exit code 0
```

Medium Level

Objective: Understand multidimensional array manipulation and matrix operation validation.

Procedure

Step1: Accept input from the user for two matrices (2D arrays).
Step2: Check that the dimensions of matrices are valid for the desired operations:

- For addition/subtraction: dimensions must be equal.
- For multiplication: columns of Matrix A = rows of Matrix B.

Step3: Use nested loops to perform:

- Addition: result[i][j] = matrixA[i][j] + matrixB[i][j]
- Subtraction: result[i][j] = matrixA[i][j] - matrixB[i][j]
- Multiplication: result[i][j] = sum(matrixA[i][k] * matrixB[k][j])
  Step4: Display the resulting matrices.

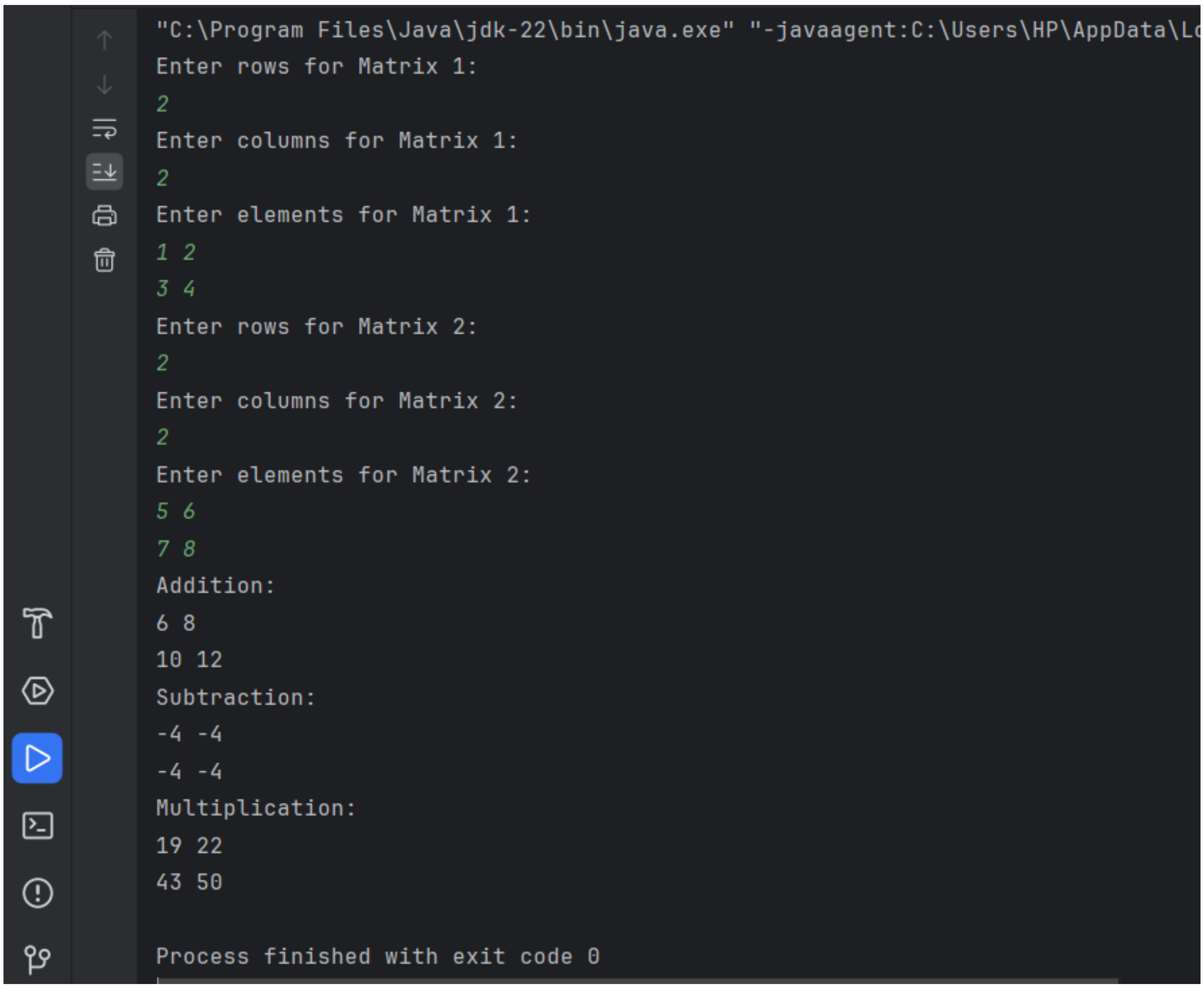
Code

```
import java.util.Scanner;

public class MatrixOperations {

  public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);


    // Matrix 1

    System.out.println("Enter rows for Matrix 1:");

    int r1 = sc.nextInt();

    System.out.println("Enter columns for Matrix 1:");

    int c1 = sc.nextInt();

    int[][] mat1 = new int[r1][c1];

    System.out.println("Enter elements for Matrix 1:");

    for (int i = 0; i < r1; i++) {

      for (int j = 0; j < c1; j++) {

        mat1[i][j] = sc.nextInt();

      }

    }


    // Matrix 2

    System.out.println("Enter rows for Matrix 2:");

    int r2 = sc.nextInt();

    System.out.println("Enter columns for Matrix 2:");

    int c2 = sc.nextInt();
```

```
int[][] mat2 = new int[r2][c2];
System.out.println("Enter elements for Matrix 2:");
for (int i = 0; i < r2; i++) {
    for (int j = 0; j < c2; j++) {
        mat2[i][j] = sc.nextInt();
    }
}

// Addition
if (r1 == r2 && c1 == c2) {
    System.out.println("Addition:");
    for (int i = 0; i < r1; i++) {
        for (int j = 0; j < c1; j++) {
            System.out.print((mat1[i][j] + mat2[i][j]) + " ");
        }
        System.out.println();
    }
} else {
    System.out.println("Addition not possible");
}

// Subtraction
if (r1 == r2 && c1 == c2) {
    System.out.println("Subtraction:");
    for (int i = 0; i < r1; i++) {
        for (int j = 0; j < c1; j++) {
            System.out.print((mat1[i][j] - mat2[i][j]) + " ");
        }
        System.out.println();
```

```
            }
        } else {
            System.out.println("Subtraction not possible");
        }

        // Multiplication
        if (c1 == r2) {
            System.out.println("Multiplication:");
            for (int i = 0; i < r1; i++) {
                for (int j = 0; j < c2; j++) {
                    int sum = 0;
                    for (int k = 0; k < c1; k++) {
                        sum += mat1[i][k] * mat2[k][j];
                    }
                    System.out.print(sum + " ");
                }
                System.out.println();
            }
        } else {
            System.out.println("Multiplication not possible");
        }
    }
}
```

OUTPUT:



```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Users\HP\AppData\L
Enter rows for Matrix 1:
2
Enter columns for Matrix 1:
2
Enter elements for Matrix 1:
1 2
3 4
Enter rows for Matrix 2:
2
Enter columns for Matrix 2:
2
Enter elements for Matrix 2:
5 6
7 8
Addition:
6 8
10 12
Subtraction:
-4 -4
-4 -4
Multiplication:
19 22
43 50

Process finished with exit code 0
```

Hard Level

Objective: Apply object-oriented programming concepts in a practical system.

Procedure

Step1: Define a BankAccount class with fields like name, account number, and balance.
Step2: Implement methods for:

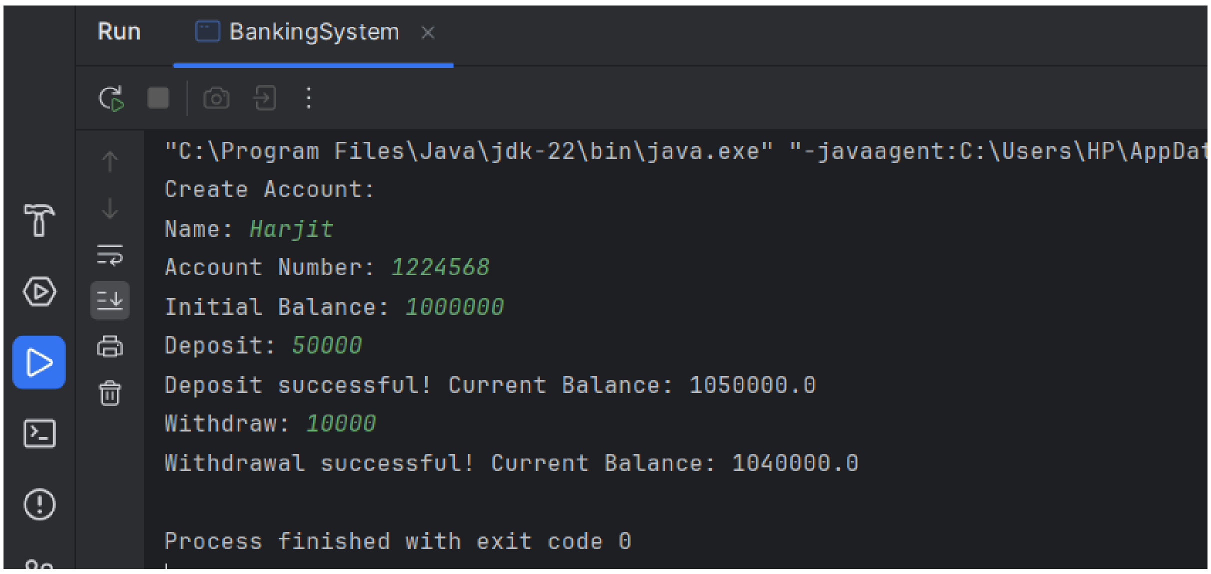
- deposit(double amount): Adds amount to balance.
- withdraw(double amount): Checks balance before subtracting.

Step3: In the main program, create a new account by taking user input.
Step4: Allow the user to perform deposit and withdrawal operations.
Step5: Display appropriate messages and updated balances.

Code

```
import java.util.Scanner;
class BankAccount {
    private String name;
    private String accountNumber;
    private double balance;

    public BankAccount(String name, String accountNumber, double balance) {
        this.name = name;
        this.accountNumber = accountNumber;
        this.balance = balance;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposit successful! Current Balance: " + balance);
        } else {
            System.out.println("Invalid deposit amount");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful! Current Balance: " + balance);
        } else if (amount > balance) {
            System.out.println("Error: Insufficient funds. Current Balance: " + balance);
        } else {
            System.out.println("Invalid withdrawal amount");
        }
    }
}

public class BankingSystem {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Create Account:");
        System.out.print("Name: ");
        String name = sc.nextLine();
        System.out.print("Account Number: ");
        String accountNumber = sc.nextLine();
        System.out.print("Initial Balance: ");
        double balance = sc.nextDouble();
        BankAccount account = new BankAccount(name, accountNumber, balance);
        sc.nextLine(); // Consume newline

        System.out.print("Deposit: ");
        double depositAmount = sc.nextDouble();
        account.deposit(depositAmount);

        System.out.print("Withdraw: ");
        double withdrawAmount = sc.nextDouble();
        account.withdraw(withdrawAmount);
    }
}
```

OUTPUT



```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Users\HP\AppDat
Create Account:
Name: Harjit
Account Number: 1224568
Initial Balance: 1000000
Deposit: 50000
Deposit successful! Current Balance: 1050000.0
Withdraw: 10000
Withdrawal successful! Current Balance: 1040000.0

Process finished with exit code 0
```