**PROGRAM NO : 1**

**AIM:** Define a class ‘product’ with data members pcode, pname and price. Create 3 objects of the class and find the product having the lowest price.

**ALGORITHM:**

Step 1: Start.

Step 2: Define a class having name Product and members as pcode,pname and price.

Step 3: Declare three objects in the class and add the values of each data members into objects.

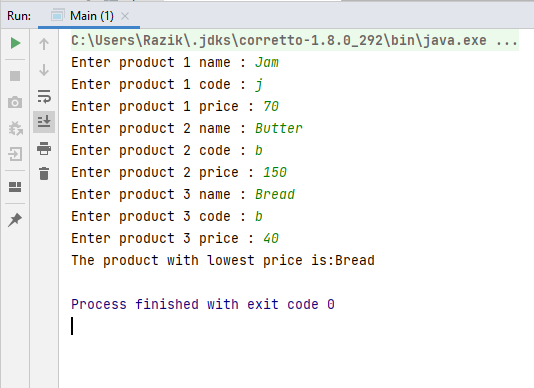
Step 4: Using if condition check which object has the lowest price and print it.

Step 5: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.Scanner;  public class Main {  public static void main(String[] args) {  //Driver program  Scanner scan = new Scanner(System.in);  Product[] products = new Product[3];  for(int i=0;i<3;i++)  {  int count = i+1;  products[i] = new Product();  System.out.print("Enter product "+count+" name : ");  products[i].pname = scan.nextLine();  System.out.print("Enter product "+count+" code : ");  products[i].pcode = scan.nextLine();  System.out.print("Enter product "+count+" price : ");  products[i].price = Integer.parseInt(scan.nextLine());  }  scan.close();  if(products[0].price< products[1].price && products[0].price< products[2].price) {  System.out.println("The product with lowest price is:"+ products[0].pname);  }  if(products[1].price<products[0].price && products[1].price<products[2].price) {  System.out.println("The product with lowest price is:"+products[1].pname);  }  if(products[2].price<products[0].price && products[2].price<products[1].price) {  System.out.println("The product with lowest price is:"+products[2].pname);  }  }  }  package com.lab\_cycles.co1.q1;  import java.util.Scanner;  public class Main {  public static void main(String[] args) {  //Driver program  Scanner scan = new Scanner(System.in);  Product[] products = new Product[3];  for(int i=0;i<3;i++)  {  int count = i+1;  products[i] = new Product();  System.out.print("Enter product "+count+" name : ");  products[i].pname = scan.nextLine();  System.out.print("Enter product "+count+" code : ");  products[i].pcode = scan.nextLine();  System.out.print("Enter product "+count+" price : ");  products[i].price = Integer.parseInt(scan.nextLine());  }  scan.close();  if(products[0].price< products[1].price && products[0].price< products[2].price) {  System.out.println("The product with lowest price is:"+ products[0].pname);  }  if(products[1].price<products[0].price && products[1].price<products[2].price) {  System.out.println("The product with lowest price is:"+products[1].pname);  }  if(products[2].price<products[0].price && products[2].price<products[1].price) {  System.out.println("The product with lowest price is:"+products[2].pname);  }  }  } |
| Product.java | public class Product {  public String pcode;  public String pname;  public int price;  } |

**OUTPUT:**

****

**PROGRAM NO : 2**

**AIM:** Read 2 matrices from the console and perform matrix addition.

**ALGORITHM :**

Step 1: Start.

Step 2: Define a class having name AddMatrix.

Step 3: Read row number(m),column number (n) and initialize the  double dimensional arrays mat1[][],mat2[][],res[][] with same row number ,column number.

Step 4: Store the first matrix elements into the two-dimensional array matrix mat1[][] using two for loops. i indicates row number, j indicates column index. Similarly second matrix elements in to mat2[][].

Step 5: Add the two matrices using for loop.

for i=0 to i<m

for j=0 to j<n

mat1[i][j] + mat2[i][j] and store it in to the matrix res[i][j] .

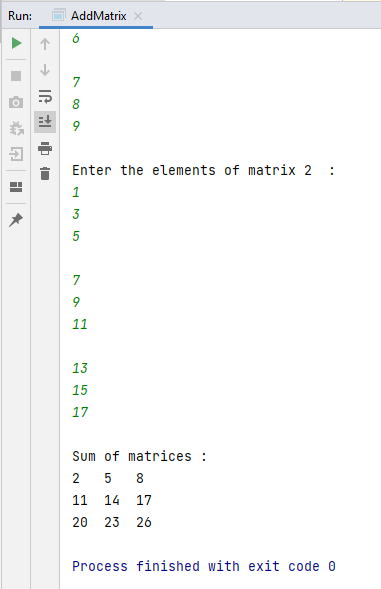
Step 6: Print sum of matrices res[i][j].

Stop 7: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| AddMatrix  .java | import java.util.Scanner;  public class AddMatrix {  public static void main(String args[])  {  int row, col,i,j;  Scanner in = new Scanner(System.in);  System.out.println("Enter the number of rows : ");  row = in.nextInt();  System.out.println("Enter the number columns : ");  col = in.nextInt();  int mat1[][] = new int[row][col];  int mat2[][] = new int[row][col];  int res[][] = new int[row][col];  System.out.println("Enter the elements of matrix 1 : ");  for ( i= 0 ; i < row ; i++ )  {  for ( j= 0 ; j < col ;j++ )  mat1[i][j] = in.nextInt();  System.out.println();  }  System.out.println("Enter the elements of matrix 2 : ");  for ( i= 0 ; i < row ; i++ )  {  for ( j= 0 ; j < col ;j++ )  mat2[i][j] = in.nextInt();  System.out.println();  }  for ( i= 0 ; i < row ; i++ )  for ( j= 0 ; j < col ;j++ )  res[i][j] = mat1[i][j] + mat2[i][j] ;  System.out.println("Sum of matrices : ");  for ( i= 0 ; i < row ; i++ )  {  for ( j= 0 ; j < col ;j++ )  System.out.print(res[i][j]+"\t");  System.out.println();  }  }  } |

**OUTPUT:**

****

**PROGRAM NO : 3**

**AIM:** Add complex numbers.

**ALGORITHM:**

Step 1: Start.

Step 2: Define a class having name ComplexNumber and data members are real and imaginary number.

Step 3: Define a function ComplexNumber and add values to variables.

Step 4 : Define a function ComplexNumber sum to add complex number using 3rd ComplexNumber object and return the value.

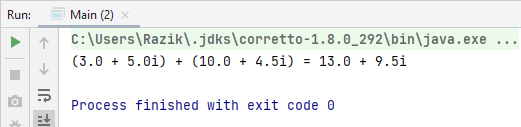
Step 5: Print the sum value.

Step 6: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| ComplexNumber. java | public class ComplexNumber {  double real,imaginary;  ComplexNumber(double real,double imaginary)  {  this.real = real;  this.imaginary = imaginary;  }  ComplexNumber addComplexNumber(ComplexNumber complexNumber1,ComplexNumber complexNumber2)  {  double real = complexNumber1.real+ complexNumber2.real;  double imaginary = complexNumber1.imaginary+complexNumber2.imaginary;  ComplexNumber complexNumber = new ComplexNumber(real,imaginary);  return complexNumber;  }  String displayComplexNumber()  {  return this.real+" + "+this.imaginary+"i";  }  } |
| Main.java | public class Main {  public static void main(String[] args) {  ComplexNumber complexNumber1 = new ComplexNumber(3,5);  ComplexNumber complexNumber2 = new ComplexNumber(10,4.5);  ComplexNumber complexNumber3 = new ComplexNumber(0,0);  complexNumber3 = complexNumber3.addComplexNumber(complexNumber1,complexNumber2);  System.out.println("("+complexNumber1.displayComplexNumber()+") + ("+complexNumber2.displayComplexNumber()+") = "+complexNumber3.displayComplexNumber());  }  } |

**OUTPUT:**



**PROGRAM NO : 4**

**AIM:** Read a matrix from the console and check whether it is symmetric or not.

**ALGORITHM:**

Step 1: Start.

Step 2 : Read row number,column number and initialize the  double dimensional array with same row number ,column number.

Step 3 : Store the first matrix elements into the two-dimensional array matrix using two for loops. i indicates row number, j indicates column index.

Step 4: Check whether the matrix is symmetric or not.

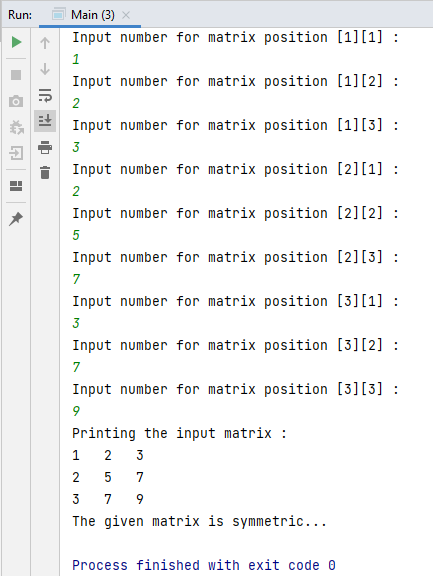
Step 5: Print the symmetric matrix or if not.

Step 6: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.Scanner;  public class Main  {  public static void main(String[] args)  {  Scanner sc = new Scanner(System.in);  System.out.println("Enter the no. of rows : ");  int rows = sc.nextInt();  System.out.println("Enter the no. of columns : ")  int cols = sc.nextInt();  int matrix[][] = new int[rows][cols];  System.out.println("Enter the elements :");  for (int i = 0; i < rows; i++)  {  for (int j = 0; j < cols; j++)  {  System.out.println("Input number for matrix position ["+(i+1)+"]["+(j+1)+"] : ");  matrix[i][j] = sc.nextInt();  }  }  System.out.println("Printing the input matrix :");  for (int i = 0; i < rows; i++)  {  for (int j = 0; j < cols; j++)  {  System.out.print(matrix[i][j]+"\t");  }  System.out.println();  }  //Checking the input matrix for symmetric  if(rows != cols)  {  System.out.println("The given matrix is not a square matrix, so it can't be symmetric.");  }  else  {  boolean symmetric = true;  for (int i = 0; i < rows; i++)  {  for (int j = 0; j < cols; j++)  {  if(matrix[i][j] != matrix[j][i])  {  symmetric = false;  break;  }  }  }  if(symmetric)  {  System.out.println("The given matrix is symmetric...");  }  else  {  System.out.println("The given matrix is not symmetric...");  }  }  sc.close();  }  } |

**OUTPUT:**



**PROGRAM NO : 5**

**AIM:** Create CPU with attribute price. Create inner class Processor (no. of cores, manufacturer) and static nested class RAM (memory, manufacturer). Create an object of CPU and print information of Processor and RAM.

**ALGORITHM :**

Step 1: Start.

Step 2: Define a class cpu with data member price and class processor.

Step 3: Class processor contain data members no\_cores,manufacturer and a nested class RAM.

Step 4: class RAM contain memory and manufacturer as data members.

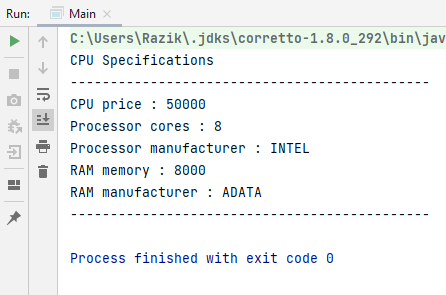
Step 5: Create objects in corresponding classes and display it’s details.

Step 6: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CPU.java | public class CPU {  int price;  Processor processor = new Processor();  RAM ram = new RAM();  static class Processor  {  int cores;  String manufacturer;  }  static class RAM  {  int memory;  String manufacturer;  }  CPU(int cpuPrice,int cores,String cpuManufacturer,int memory,String ramManufacturer){  this.price = cpuPrice;  this.processor.cores = cores;  this.processor.manufacturer = cpuManufacturer;  this.ram.memory = memory;  this.ram.manufacturer = ramManufacturer;  }  void printCPUSpecifications()  {  System.out.println("CPU Specifications");  System.out.println("---------------------------------------------");  System.out.println("CPU price : "+this.price);  System.out.println("Processor cores : "+this.processor.cores);  System.out.println("Processor manufacturer : "+this.processor.manufacturer);  System.out.println("RAM memory : "+this.ram.memory);  System.out.println("RAM manufacturer : "+this.ram.manufacturer);  System.out.println("---------------------------------------------");  }  } |
| Main.java | public class Main {  public static void main(String[] args) {  CPU cpu1 = new CPU(50000, 8, "INTEL", 8000, "ADATA");  cpu1.printCPUSpecifications();  }  } |

**OUTPUT:**

****

**PROGRAM NO : 6**

**AIM:** Program to Sort strings.

**ALGORITHM :**

Step 1: Start

Step 2: Select the first element of the list (i.e., Element at first position in the list).

Step 3: Compare the selected element with all the other elements in the list.

Step 4: In every comparision, if any element is found smaller than the selected element (for Ascending order), then both are swapped.

Step 5: Repeat the same procedure with element in the next position in the list till the entire list is sorted.

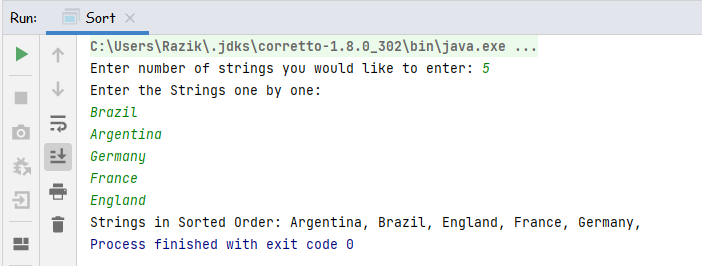
Step 6: Stop

.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Sort.java | import java.util.Scanner;  public class Sort {  public static void main(String[] args) {  int count;  String temp;  Scanner scan = new Scanner(System.in);  //User will be asked to enter the count of strings  System.out.print("Enter number of strings you would like to enter: ");  count = scan.nextInt();  String[] str = new String[count];  Scanner scan2 = new Scanner(System.in);  //User is entering the strings and they are stored in an array  System.out.println("Enter the Strings one by one: ");  for(int i = 0; i < count; i++)  {  str[i] = scan2.nextLine();  }  scan.close();  scan2.close();  //Sorting the strings  for (int i = 0; i < count; i++)  {  for (int j = i + 1; j < count; j++) {  if (str[i].compareTo(str[j])>0)  {  temp = str[i];  str[i] = str[j];  str[j] = temp;  }  }  }  //Displaying the strings after sorting them based on alphabetical order  System.out.print("Strings in Sorted Order: ");  for (int i = 0; i <= count - 1; i++)  {  System.out.print(str[i] + ", ");  }  }  } |

**OUTPUT:**



**PROGRAM NO : 7**

**AIM:** Search an element in an array.

**ALGORITHM :**

Step 1: Start

Step 2: Check each element in the given list with the string provided by the user.

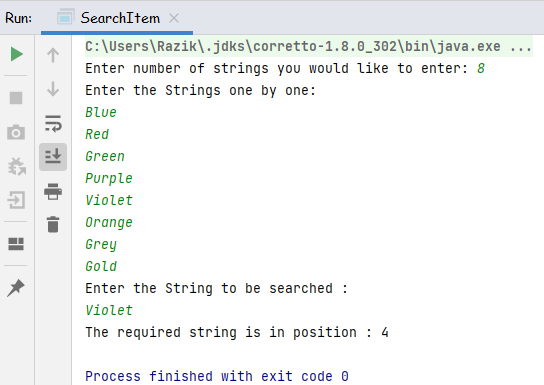
Step 3: If string is found, display the position of the string found, else display string not found.

Step : Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| SearchItem.java | import java.util.Scanner;  public class SearchItem {  public static void main(String[] args) {  int count;  String temp;  Scanner scan = new Scanner(System.in);  Scanner scan2 = new Scanner(System.in);  Scanner scan3 = new Scanner(System.in);  //User will be asked to enter the count of strings  System.out.print("Enter number of strings you would like to enter: ");  count = scan.nextInt();  String[] str = new String[count];  //User is entering the strings and they are stored in an array  System.out.println("Enter the Strings one by one: ");  for(int i = 0; i < count; i++)  {  str[i] = scan2.nextLine();  }  System.out.println("Enter the String to be searched : ");  String searchString = scan3.nextLine();  for (int i = 0; i < count; i++)  {  if(str[i].equals(searchString)){  System.out.println("The required string is in position : "+(i));  System.exit(0);  }  }  scan.close();  scan2.close();  scan3.close();  System.out.println("Could not find required string in Array.");  }  } |

**OUTPUT:**



**PROGRAM NO : 8**

**AIM:** Perform string manipulations

**ALGORITHM :**

Step 1: Start

Step 2: Take the strings provided by the user and concatenate them.

Step 3: Display the combined string with lower case.

Step 3: Display the combined string with upper case.

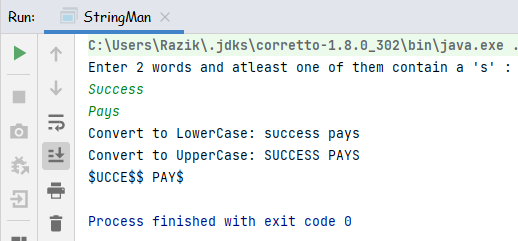
Step 4: Display the combined string after replacing all the ‘s’ & ‘S’ characters with ‘$’ character.

Step 5: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| StringMan.java | import java.util.Scanner;  public class SearchItem {  public static void main(String[] args) {  int count;  String temp;  Scanner scan = new Scanner(System.in);  Scanner scan2 = new Scanner(System.in);  Scanner scan3 = new Scanner(System.in);  //User will be asked to enter the count of strings  System.out.print("Enter number of strings you would like to enter: ");  count = scan.nextInt();  String[] str = new String[count];  //User is entering the strings and they are stored in an array  System.out.println("Enter the Strings one by one: ");  for(int i = 0; i < count; i++)  {  str[i] = scan2.nextLine();  }  System.out.println("Enter the String to be searched : ");  String searchString = scan3.nextLine();  for (int i = 0; i < count; i++)  {  if(str[i].equals(searchString)){  System.out.println("The required string is in position : "+(i));  System.exit(0);  }  }  scan.close();  scan2.close();  scan3.close();  System.out.println("Could not find required string in Array.");  }  } |

**OUTPUT:**



**PROGRAM NO : 9**

**AIM:** Program to create a class for Employee having attributes eNo, eName eSalary. Read n employ information and Search for an employee given eNo, using the concept of Array of Objects.

**ALGORITHM :**

Step 1: Start

Step 2: Search the ‘eNo’ attribute of the list of Employee Objects for the ‘eNo’ provided by the user.

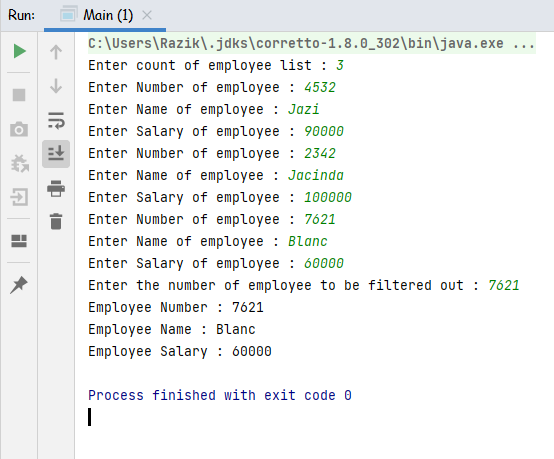
Step 3: If user provided ‘eNo’ is found inside the Employee object list, display the details of the corresponding employee.

Step 4: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Employee.java | public class Employee {  public String eNo;  public String eName;  public int eSalary;  public Employee(String eNo,String eName,int eSalary)  {  this.eNo = eNo;  this.eName = eName;  this.eSalary = eSalary;  }  } |
| Main.java | import java.util.Scanner;  public class Main {  public static void main(String[] args) {  Scanner scan = new Scanner(System.in);  System.out.print("Enter count of employee list : ");  int count = Integer.parseInt(scan.nextLine());  Employee[] employees = new Employee[count];  for (int i=0;i<count;i++)  {  System.out.print("Enter Number of employee : ");  String eNo = scan.nextLine();  System.out.print("Enter Name of employee : ");  String eName = scan.nextLine();  System.out.print("Enter Salary of employee : ");  int eSalary = Integer.parseInt(scan.nextLine());  employees[i] = new Employee(eNo,eName,eSalary);  }  System.out.print("Enter the number of employee to be filtered out : ");  String search = scan.nextLine();  for (int i=0;i<count;i++){  if(employees[i].eNo.equals(search))  {  System.out.println("Employee Number : "+employees[i].eNo);  System.out.println("Employee Name : "+employees[i].eName);  System.out.println("Employee Salary : "+employees[i].eSalary);  }  }  }  } |

**OUTPUT:**



**PROGRAM NO : 10**

**AIM:** Area of different shapes using overloaded functions.

**ALGORITHM :**

Step 1: Start

Step 2: Define the main class

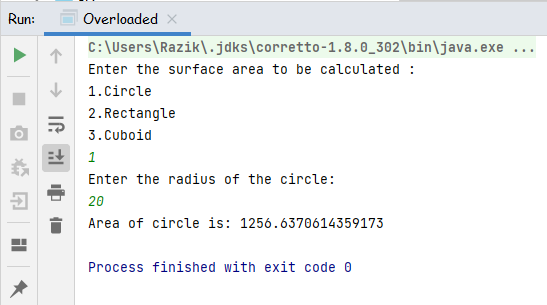
Step 3: Define methods with the same methodname that performs the area operation for each shape

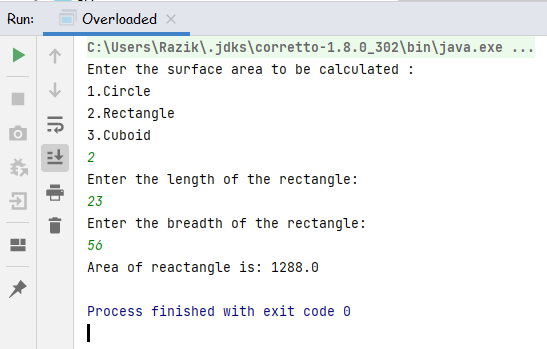
Step 4: Display the areas of each shapes.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Areas.java | public class Areas {  public void area(float length,float breadth)  {  System.out.println("Area of reactangle is: "+length\*breadth);  }  public void area(float radius)  {  System.out.println("Area of circle is: "+(radius\*radius\*Math.PI));  }  public void area(float length,float breadth, float height)  {  float area = 2\*((length\*breadth)+(breadth\*height)+(length\*height));  System.out.println("Area of cuboid is: "+area);  }  } |
| Overloaded.java | import java.util.Scanner;  public class Overloaded {  public static void main(String[] args) {  float a,b,c;  int choice;  Scanner scanner=new Scanner(System.in);  Areas areas = new Areas();  System.out.println("Enter the surface area to be calculated : ");  System.out.println("1.Circle");  System.out.println("2.Rectangle");  System.out.println("3.Cuboid");  choice = scanner.nextInt();  if(choice==1)  {  System.out.println("Enter the radius of the circle: ");  a=scanner.nextInt();  areas.area(a);  }  else if(choice==2)  {  System.out.println("Enter the length of the rectangle: ");  a=scanner.nextInt();  System.out.println("Enter the breadth of the rectangle: ");  b=scanner.nextInt();  areas.area(a,b);  }  else if(choice==3)  {  System.out.println("Enter the length of the cuboid: ");  a=scanner.nextInt();  System.out.println("Enter the breadth of the cuboid: ");  b=scanner.nextInt();  System.out.println("Enter the height of the cuboid: ");  c=scanner.nextInt();  areas.area(a,b,c);  }  else  {  System.out.println("Invalid Choice");  }  }  } |

**OUTPUT:**





**PROGRAM NO : 11**

**AIM:** Create a class ‘Employee’ with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create another class ‘Teacher’ that inherit the properties of class employee and contain its own data members department, Subjects taught and constructors to initialize these data members and also include display function to display all the data members. Use array of objects to display details of N teachers.

**ALGORITHM :**

Step 1: Start

Step 2: create class “employee” with the provided data members and define the constructors

Step 3: create another class “Teachers” that performs inheritance of employee class and define constructors for the same

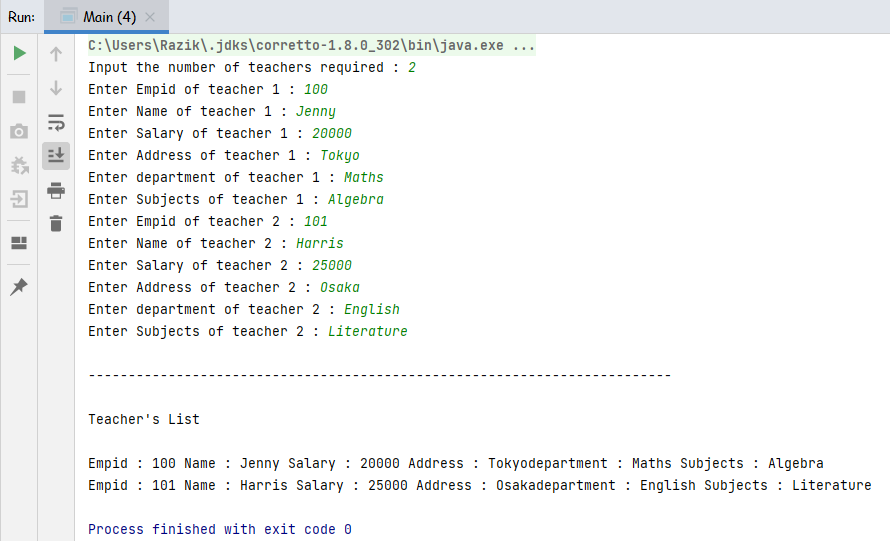
Step 4: create an array of objects in the corresponding class

Step 5: Display the details for the number of teachers provided.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Employee.java | public class Employee {  String Empid;  String Name;  String Salary;  String Address;  Employee(String Empid,String Name,String Salary,String Address)  {  this.Empid = Empid;  this.Name = Name;  this.Salary = Salary;  this.Address = Address;  }  } |
| Teacher.java | public class Teacher extends Employee {  String department;  String Subjects;  Teacher(String Empid, String Name, String Salary, String Address,String department,String Subjects) {  super(Empid, Name, Salary, Address);  this.department = department;  this.Subjects = Subjects;  }  void displayTeacherDetails()  {  System.out.println("Empid : "+this.Empid+" Name : "+this.Name+" Salary : "+this.Salary+" Address : "+this.Address+"department : "+this.department+" Subjects : "+this.Subjects);  }  } |
| Main.java | import java.util.Scanner;  public class Main {  public static void main(String[] args)  {  int count;  Scanner scanner = new Scanner(System.in);  System.out.print("Input the number of teachers required : ");  count = scanner.nextInt();  Teacher[] teachers = new Teacher[count];  for(int i=0;i<count;i++) {  int j = i+1;  System.out.print("Enter Empid of teacher "+j+" : ");  String Empid = scanner.next();  System.out.print("Enter Name of teacher "+j+" : ");  String Name = scanner.next();  System.out.print("Enter Salary of teacher "+j+" : ");  String Salary = scanner.next();  System.out.print("Enter Address of teacher "+j+" : ");  String Address = scanner.next();  System.out.print("Enter department of teacher "+j+" : ");  String department = scanner.next();  System.out.print("Enter Subjects of teacher "+j+" : ");  String Subjects = scanner.next();  teachers[i] = new Teacher(Empid, Name, Salary, Address, department, Subjects);  }  System.out.println("\n-------------------------------------------------------------------------\n");  System.out.println("Teacher's List \n");  for(int i=0;i<count;i++) {  teachers[i].displayTeacherDetails();  }  }  } |

**OUTPUT:**



**PROGRAM NO : 12**

**AIM:** Write a user defined exception class to authenticate the user name and password.

**ALGORITHM :**

Step 1: Start

Step 2: create class “person” with the provided data members and define the constructors

Step 3: create another class “employee” that performs inheritance of person class and another class “teacher” that further inherits the properties of its former class

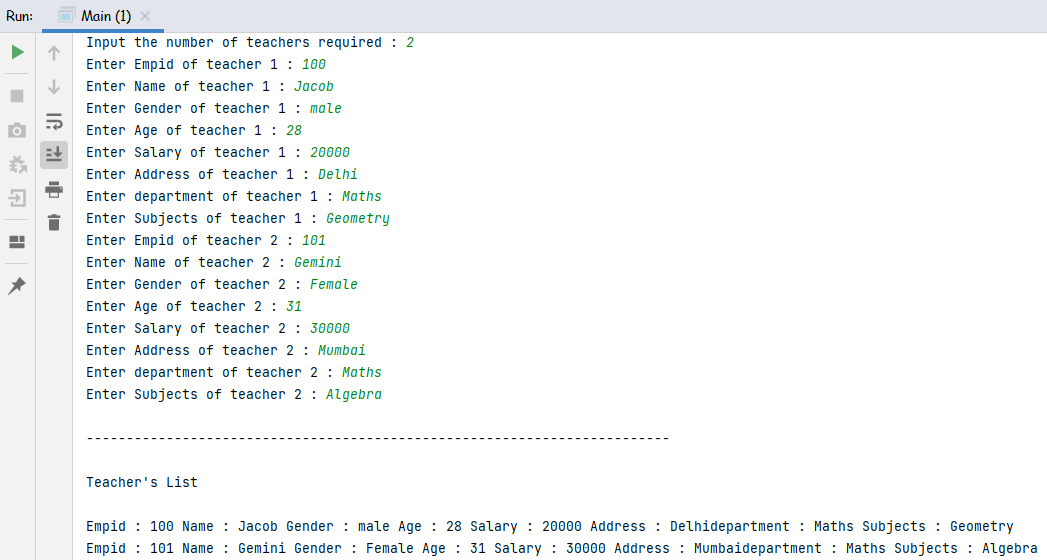
Step 4: create an array of objects in the corresponding class

Step 5: Display the details for the number of teachers provided.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Employee.java | public class Employee extends Person{  String Empid;  String Salary;  Employee(String Name,String Gender,String Address,String Age,String Empid,String Salary)  {  super(Name,Gender,Address,Age);  this.Empid = Empid;  this.Salary = Salary;  }  } |
| Teacher.java | public class Teacher extends Employee {  String department;  String Subjects;  Teacher(String Name,String Gender,String Address,String Age,String Empid,String Salary,String department,String Subjects) {  super(Name,Gender,Address,Age,Empid,Salary);  this.department = department;  this.Subjects = Subjects;  }  void displayTeacherDetails()  {  System.out.println("Empid : "+this.Empid+" Name : "+this.Name+" Gender : "+this.Gender+" Age : "+this.Age+" Salary : "+this.Salary+" Address : "+this.Address+"department : "+this.department+" Subjects : "+this.Subjects);  }  } System.out.println("Empid : "+this.Empid+" Name : "+this.Name+" Salary : "+this.Salary+" Address : "+this.Address+"department : "+this.department+" Subjects : "+this.Subjects);  }  } |
| Person.java | public class Person {  String Name;  String Gender;  String Address;  String Age;  Person(String Name,String Gender,String Address,String Age){  this.Name = Name;  this.Gender = Gender;  this.Address = Address;  this.Age = Age;  }  } |
| Main.java | public class Main {  public static void main(String[] args)  {  int count;  Scanner scanner = new Scanner(System.in);  System.out.print("Input the number of teachers required : ");  count = scanner.nextInt();  Teacher[] teachers = new Teacher[count];  for(int i=0;i<count;i++) {  int j = i+1;  System.out.print("Enter Empid of teacher "+j+" : ");  String Empid = scanner.next();  System.out.print("Enter Name of teacher "+j+" : ");  String Name = scanner.next();  System.out.print("Enter Gender of teacher "+j+" : ");  String Gender = scanner.next();  System.out.print("Enter Age of teacher "+j+" : ");  String Age = scanner.next();  System.out.print("Enter Salary of teacher "+j+" : ");  String Salary = scanner.next();  System.out.print("Enter Address of teacher "+j+" : ");  String Address = scanner.next();  System.out.print("Enter department of teacher "+j+" : ");  String department = scanner.next();  System.out.print("Enter Subjects of teacher "+j+" : ");  String Subjects = scanner.next();  teachers[i] = new Teacher(Name,Gender,Address,Age,Empid,Salary,department,Subjects);  }  System.out.println("\n-------------------------------------------------------------------------\n");  System.out.println("Teacher's List \n");  for(int i=0;i<count;i++) {  teachers[i].displayTeacherDetails();  }  }  } |

**OUTPUT:**



**PROGRAM NO : 13**

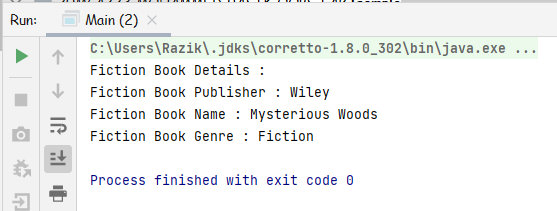
**AIM:** Find the average of N positive integers, raising a user defined exception for each negative input.

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Book.java | public class Book extends Publisher{  String book = "Mysterious Woods";  } |
| Fiction.java | public class Fiction extends Book{  String genre = "Fiction";  } |
| Literature.java | public class Literature extends Book{  String genre = "Literature";  } |
| Publisher.java | public class Publisher {  String publisher = "Wiley";  } |
| Main.java | public class Main {  public static void main(String[] args) {  Literature literatureBook = new Literature();  Fiction fictionBook = new Fiction();  System.out.println("Fiction Book Details : ");  System.out.println("Fiction Book Publisher : "+fictionBook.publisher);  System.out.println("Fiction Book Name : "+fictionBook.book);  System.out.println("Fiction Book Genre : "+fictionBook.genre);  }  } |

**OUTPUT:**



**PROGRAM NO : 14**

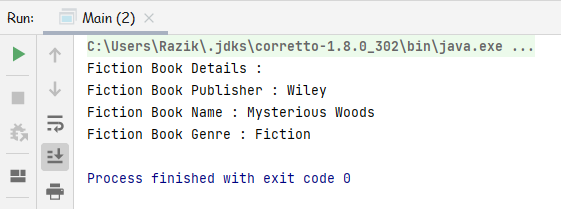
**AIM:** Find the average of N positive integers, raising a user defined exception for each negative input.

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Student.java | public class Student {  int maths = 85;  int science = 72;  int english = 88;  int socialScience = 70;  } |
| Sports.java | public class Sports extends Student {  String sport = "Football";  int goals = 2;  int assists = 1;  int minutesPlayed = 81;  int grace = 20;  } |
| Result.java | public class Result extends Sports {  public void displayInfo()  {  System.out.println("Academic Result");  System.out.println("---------------");  System.out.println("Maths : "+this.maths);  System.out.println("Science : "+this.science);  System.out.println("English : "+this.english);  System.out.println("Social Science : "+this.socialScience);  System.out.println("Sports Grace : "+this.grace);  System.out.println("---------------");  System.out.println("\n");  System.out.println("Sports Result");  System.out.println("-------------");  System.out.println("Sport : "+this.sport);  System.out.println("Goals : "+this.goals);  System.out.println("Assists : "+this.assists);  System.out.println("Minutes Played "+this.minutesPlayed);  System.out.println("-------------");  }  } |
| Main.java | public class Main {  public static void main(String[] args) {  Result result = new Result();  result.displayInfo();  }  } |

**OUTPUT:**



**PROGRAM NO : 15**

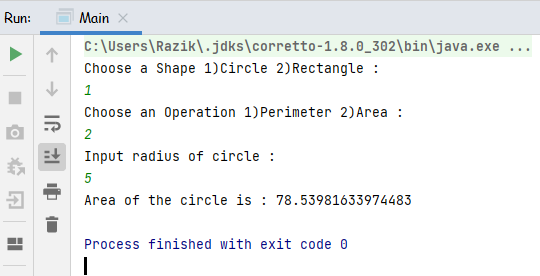
**AIM:** Create an interface having prototypes of functions area() and perimeter(). Create two classes Circle and Rectangle which implements the above interface. Create a menu driven program to find area and perimeter of objects.

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Circle.java | public class Circle implements Shape{  int radius;  Scanner scanner = new Scanner(System.in);  public void perimeter() {  System.out.println("Input radius of circle : ");  radius = scanner.nextInt();  String perimeter = Double.toString(Math.PI\*radius\*2);  System.out.println("Circumference of the circle is : "+perimeter);  }  public void area() {  System.out.println("Input radius of circle : ");  radius = scanner.nextInt();  String area = Double.toString(Math.PI\*radius\*radius);  System.out.println("Area of the circle is : "+area);  }  } |
| Rectangle.java | import java.util.Scanner;  public class Rectangle implements Shape{  int length;  int breadth;  Scanner scanner = new Scanner(System.in);  public void perimeter() {  System.out.println("Input length of rectangle : ");  length = scanner.nextInt();  System.out.println("Input breadth of rectangle : ");  length = scanner.nextInt();  String perimeter = Double.toString(2\*(length+breadth));  System.out.println("Perimeter of the rectangle is : "+perimeter);  }  public void area() {  System.out.println("Input length of rectangle : ");  length = scanner.nextInt();  System.out.println("Input breadth of rectangle : ");  length = scanner.nextInt();  String area = Double.toString(length\*breadth);  System.out.println("Area of the rectangle is : "+area);  }  } |
| Shape.java | public interface Shape {  public void perimeter();  public void area();  } |
| Main.java | import java.util.Scanner;  public class Main {  public static void main(String[] args){  Scanner scanner = new Scanner(System.in);  int shape,operation;  System.out.println("Choose a Shape 1)Circle 2)Rectangle : ");  shape = scanner.nextInt();  System.out.println("Choose an Operation 1)Perimeter 2)Area : ");  operation = scanner.nextInt();  if(shape==1){  Circle circle = new Circle();  if(operation==1){  circle.perimeter();  }  else if(operation==2)  {  circle.area();  }  else {  System.out.println("Operation code.");  }  }  else if(shape==2)  {  Rectangle rectangle = new Rectangle();  if(operation==1){  rectangle.perimeter();  }  else if(operation==2)  {  rectangle.area();  }  else {  System.out.println("Operation code :");  System.exit(0);  }  }  else {  System.out.println("Incorrect Shape code.");  }  }  } |

**OUTPUT:**



**PROGRAM NO : 16**

**AIM:** Prepare bill with the given format using calculate method from interface :

Order No.

Date :

Product Id Name Quantity unit price Total

-----------------------------------------

101 A 2 25 50

102 B 1 100 100

-----------------------------------------

Net. Amount 150

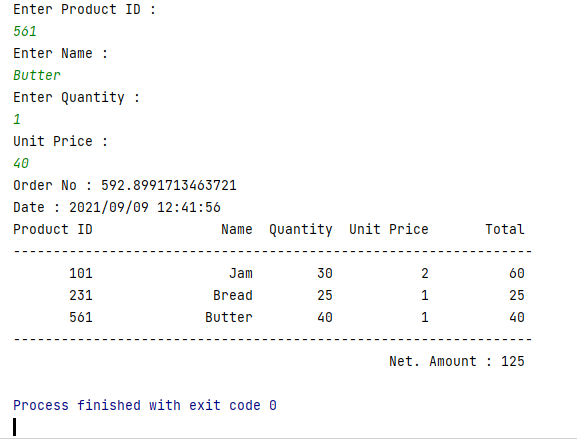
**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Bill.java | public interface Bill {  String productId = "";  String productName="";  int unitPrice = 0;  int quantity = 0;  int total = 0;  public void printBillItem();  public void printBillHeader();  public void printBillFooter(int billTotal);  } |
| ProductBill.java | import java.time.format.DateTimeFormatter;  import java.time.LocalDateTime;  public class ProductBill implements Bill {  String productId = "";  String productName="";  int unitPrice = 0;  int quantity = 0;  int total = 0;  ProductBill(String productId,String productName,int unitPrice,int quantity){  this.productId = productId;  this.productName = productName;  this.unitPrice = unitPrice;  this.quantity = quantity;  this.total = unitPrice\*quantity;  }  public void printBillHeader() {  System.out.println("Order No : " + Math.random() \* 1000);  DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:ss");  LocalDateTime now = LocalDateTime.now();  System.out.println("Date : " + dtf.format(now));  System.out.println("Product ID Name Quantity Unit Price Total ");  System.out.println("-----------------------------------------------------------------");  }  public void printBillItem()  {  System.out.format("%10s%20s%10d%12d%12d \n",this.productId,this.productName,this.unitPrice,this.quantity,this.total);  }  public void printBillFooter(int billTotal)  {  System.out.println("-----------------------------------------------------------------");  System.out.format("%64s \n","Net. Amount : "+billTotal);  }  } |
| Main.java | import java.util.Scanner;  public class Main {  public static void main(String[] args)  {  Scanner scanner = new Scanner(System.in);  System.out.println("Input Number of items in Bill : ");  int count = scanner.nextInt();  ProductBill[] productBill=new ProductBill[count];  int billTotal=0;  for(int i=0;i<count;i++) {  System.out.println("Enter Product ID : ");  String productId = scanner.next();  System.out.println("Enter Name : ");  String name = scanner.next();  System.out.println("Enter Quantity : ");  int qty = scanner.nextInt();  System.out.println("Unit Price : ");  int up = scanner.nextInt();  productBill[i]=new ProductBill(productId,name,up,qty);  }  if(count>0){  productBill[0].printBillHeader();  for(int i=0;i<count;i++) {  productBill[i].printBillItem();  billTotal += productBill[i].total;  }  productBill[0].printBillFooter(billTotal);  }  }  } |

**OUTPUT:**





**PROGRAM NO : 17**

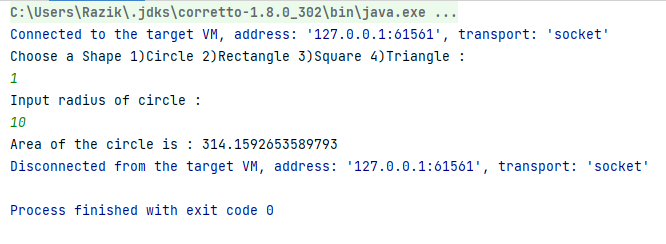
**AIM:** Create a Graphics package that has classes and interfaces for figures Rectangle, Triangle, Square and Circle. Test the package by finding the area of these figures.

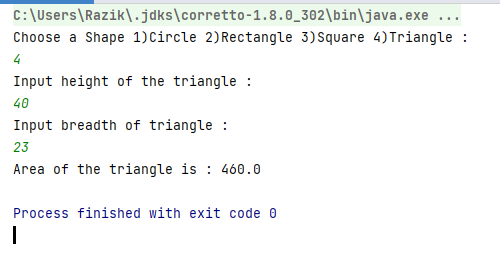
**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Graphics/  Circle.java | import java.util.Scanner;  public class Circle implements Shape {  int radius;  Scanner scanner = new Scanner(System.in);  public void area() {  System.out.println("Input radius of circle : ");  radius = scanner.nextInt();  String area = Double.toString(Math.PI\*radius\*radius);  System.out.println("Area of the circle is : "+area);  }  } |
| Graphics/  Rectangle.java | import java.util.Scanner;  public class Rectangle implements Shape {  int length;  int breadth;  Scanner scanner = new Scanner(System.in);  public void area() {  System.out.println("Input length of rectangle : ");  length = scanner.nextInt();  System.out.println("Input breadth of rectangle : ");  length = scanner.nextInt();  String area = Double.toString(length\*breadth);  System.out.println("Area of the rectangle is : "+area);  }  } |
| Graphics/  Shape.java | public interface Shape {  public void area();  } |
| Graphics/  Square.java | import java.util.Scanner;  public class Square {  int side;  Scanner scanner = new Scanner(System.in);  public void area() {  System.out.println("Input side length of square : ");  side = scanner.nextInt();  String area = Double.toString(side\*side);  System.out.println("Area of the square : "+area);  }  } |
| Graphics/  Triangle.java | import java.util.Scanner;  public class Triangle {  int height;  int breadth;  Scanner scanner = new Scanner(System.in);  public void area() {  System.out.println("Input height of the triangle : ");  height = scanner.nextInt();  System.out.println("Input breadth of triangle : ");  breadth = scanner.nextInt();  String area = Double.toString((height\*breadth)/2f);  System.out.println("Area of the triangle is : "+area);  }  } |
| Main.java | import com.lab\_cycles.co4.q1.Graphics.Circle;  import com.lab\_cycles.co4.q1.Graphics.Rectangle;  import com.lab\_cycles.co4.q1.Graphics.Square;  import com.lab\_cycles.co4.q1.Graphics.Triangle;  import java.util.Scanner;  public class Main {  public static void main(String[] args){  Scanner scanner = new Scanner(System.in);  int shape;  System.out.println("Choose a Shape 1)Circle 2)Rectangle 3)Square 4)Triangle : ");  shape = scanner.nextInt();  if(shape==1){  Circle circle = new Circle();  circle.area();  }  else if(shape==2)  {  Rectangle rectangle = new Rectangle();  rectangle.area();  }  else if(shape==3)  {  Square square = new Square();  square.area();  }  else if(shape==4)  {  Triangle triangle = new Triangle();  triangle.area();  }  else {  System.out.println("Incorrect Shape code.");  }  }  } |

**OUTPUT:**





**PROGRAM NO : 18**

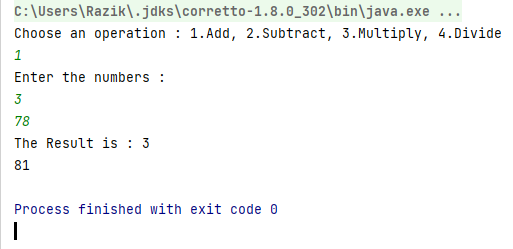
**AIM:** Create an Arithmetic package that has classes and interfaces for the 4 basic arithmetic operations. Test the package by implementing all operations on two given numbers.

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Arithmetic/  AdditionOperation.java | public class AdditionOperation implements ArithmeticOperation {  public int operateNumbers(int number1,int number2)  {  return number1+number2;  }  } |
| Arithmetic/  ArithmeticOperation.java | public interface ArithmeticOperation {  public int operateNumbers(int number1,int number2);  } |
| Arithmetic/  DivisionOperator.java | public class DivisionOperator implements ArithmeticOperation{  public int operateNumbers(int number1,int number2)  {  return number1\*number2;  }  } |
| Arithmetic/  MultiplicationOperator.java | public class MultiplicationOperator implements ArithmeticOperation{  public int operateNumbers(int number1,int number2)  {  return number1\*number2;  }  } |
| Arithmetic/  SubtractionOperation.java | public class SubtractionOperation implements ArithmeticOperation{  public int operateNumbers(int number1,int number2)  {  return number1-number2;  }  } |
| Main.java | import com.lab\_cycles.co4.q2.Arithmetic.AdditionOperation;  import com.lab\_cycles.co4.q2.Arithmetic.DivisionOperator;  import com.lab\_cycles.co4.q2.Arithmetic.MultiplicationOperator;  import com.lab\_cycles.co4.q2.Arithmetic.SubtractionOperation;  import java.util.Scanner;  public class Main {  public static void main(String[] args)  {  Scanner scanner = new Scanner(System.in);  System.out.println("Choose an operation : 1.Add, 2.Subtract, 3.Multiply, 4.Divide");  int choice = scanner.nextInt();  System.out.println("Enter the numbers : ");  int number1 = scanner.nextInt();  int number2= scanner.nextInt();  System.out.println("The Result is : 3");  switch (choice){  case 1:  AdditionOperation additionOperation = new AdditionOperation();  System.out.println(additionOperation.operateNumbers(number1,number2));  break;  case 2:  SubtractionOperation subtractionOperation = new SubtractionOperation();  System.out.println(subtractionOperation.operateNumbers(number1,number2));  break;  case 3:  MultiplicationOperator multiplicationOperator = new MultiplicationOperator();  System.out.println(multiplicationOperator.operateNumbers(number1,number2));  break;  case 4:  DivisionOperator divisionOperator = new DivisionOperator();  System.out.println(divisionOperator.operateNumbers(number1,number2));  break;  default:  System.out.println("Invalid Code");  }  }  } |

**OUTPUT:**



**PROGRAM NO : 19**

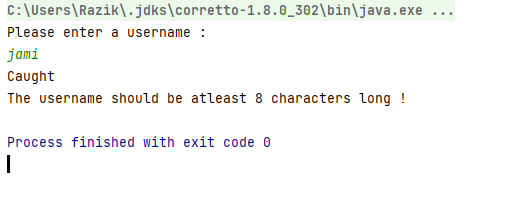
**AIM:** Write a user defined exception class to authenticate the user name and password.

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| UserException.java | public class UserException extends Exception {  public UserException(String s)  {  // Call constructor of parent Exception  super(s);  }  } |
| Main.java | import java.util.Scanner;  public class Main  {  public static void main(String args[])  {  Scanner scanner = new Scanner(System.in);  String username,password;  try  {  System.out.println("Please enter a username : ");  username = scanner.nextLine();  if(username.equals("")){  throw new UserException("Username not provided !");  }  if(username.length()<8){  throw new UserException("The username should be atleast 8 characters long !");  }  System.out.println("Please enter a password : ");  password = scanner.nextLine();  if(password.equals("")){  throw new UserException("Password not provided !");  }  if(password.length()<6){  throw new UserException("The password should be atleast 6 characters long !");  }  System.out.println("User Data Accepted");  }  catch (UserException ex)  {  System.out.println("Caught");  System.out.println(ex.getMessage());  }  }  } |

**OUTPUT:**



**PROGRAM NO : 20**

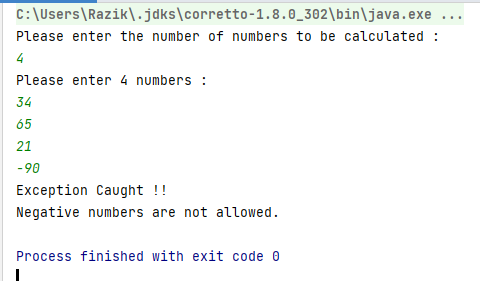
**AIM:** Find the average of N positive integers, raising a user defined exception for each negative input.

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| NegativeNumberException.java | public class NegativeNumberException extends Exception {  public NegativeNumberException(String s)  {  // Call constructor of parent Exception  super(s);  }  } |
| Main.java | import java.util.Scanner;  public class Main {  public static void main(String args[]) {  Scanner scanner = new Scanner(System.in);  int count;  System.out.println("Please enter the number of numbers to be calculated : ");  count = scanner.nextInt();  int[] numbers = new int[count];  try {  System.out.println("Please enter " + count + " numbers : ");  for (int i = 0; i < count; i++) {  int num = scanner.nextInt();  numbers[i] = num;  if (num < 0) {  throw new NegativeNumberException("Negative numbers are not allowed.");  }  }  int sum =0;  for (int i = 0; i < count; i++) {  sum += numbers[i];  }  float average = (float)sum/count;  System.out.println("Average of given numbers is : "+average);  } catch (NegativeNumberException ex) {  System.out.println("Exception Caught !!");  System.out.println(ex.getMessage());  }  }  } |

**OUTPUT:**



**PROGRAM NO : 21**

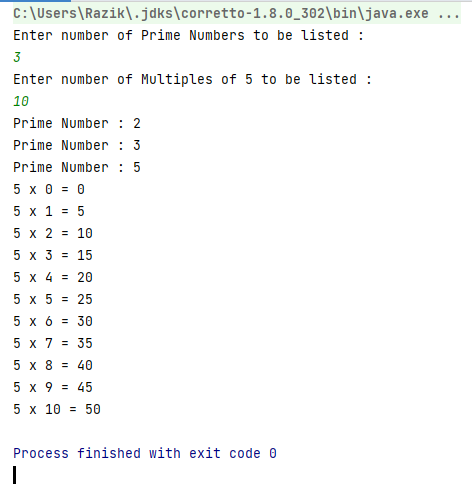
**AIM:** Define 2 classes; one for generating multiplication table of 5 and other for displaying first N prime numbers. Implement using threads. (Thread class)

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| PrimeNumbers.java | import java.util.Scanner;  public class PrimeNumbers extends Thread {  int count;  int primeCount = 0;  PrimeNumbers(int count){  this.count=count;  }  public void run() {  for(int i=1;primeCount<count;i++){  boolean isPrime=true;  if(i==1)  {  i++;  }  for (int j=2;j<i;j++){  if(i%j==0){  isPrime = false;  break;  }  }  if (isPrime){  System.out.println("Prime Number : "+i);  primeCount++;  }  }  }  } |
| MultiTable5.java | public class MultiTable5 extends Thread {  int count;  MultiTable5(int count){  this.count=count;  }  public void run() {  for (int i=0;i<=count;i++){  System.out.println("5 x "+i+" = "+(5\*i));  }  }  } |
| Main.java | import java.util.Scanner;  public class Main {  public static void main(String[] args)  {  Scanner scanner = new Scanner(System.in);  int primeCount,multipleCount;  System.out.println("Enter number of Prime Numbers to be listed : ");  primeCount = scanner.nextInt();  System.out.println("Enter number of Multiples of 5 to be listed : ");  multipleCount = scanner.nextInt();  PrimeNumbers thread1 = new PrimeNumbers(primeCount);  thread1.start();  MultiTable5 thread2 = new MultiTable5(multipleCount);  thread2.start();  }  } |

**OUTPUT:**



**PROGRAM NO : 22**

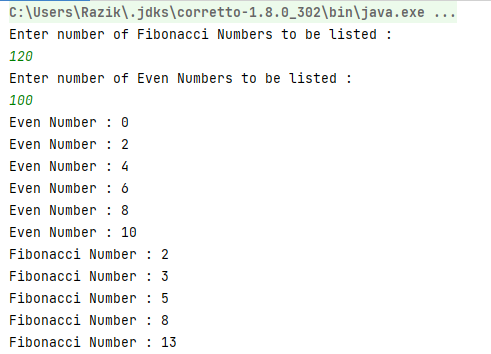
**AIM:** Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface).

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| EvenSeries.java | public class EvenSeries implements Runnable {  int count;  int[] fibonacciSeries;  EvenSeries(int count){this.count=count;}  public void run() {  for (int i=0;i<count;i++)  {  if(i%2==0)  {  System.out.println("Even Number : "+i);  }  }  }    } |
| FibonacciSeries.java | public class FibonacciSeries implements Runnable{  int count;  long[] fibonacciSeries;  FibonacciSeries(int count){this.count=count;this.fibonacciSeries=new long[count];}  public void run() {  this.fibonacciSeries[0]=0;  this.fibonacciSeries[1]=1;  this.fibonacciSeries[2]=1;  for (int i=3;i<count;i++)  {  this.fibonacciSeries[i]= this.fibonacciSeries[(i-1)]+ this.fibonacciSeries[(i-2)];  if(i>91) {  break;  }  System.out.println("Fibonacci Number : " + this.fibonacciSeries[i]);  }  }  } |
| Main.java | import java.util.Scanner;  public class Main {  public static void main(String[] args)  {  Scanner scanner = new Scanner(System.in);  int fiboCount,evenCount;  System.out.println("Enter number of Fibonacci Numbers to be listed : ");  fiboCount = scanner.nextInt();  System.out.println("Enter number of Even Numbers to be listed : ");  evenCount = scanner.nextInt();  FibonacciSeries fibonacciSeries = new FibonacciSeries(fiboCount);  EvenSeries evenSeries = new EvenSeries(evenCount);  Thread thread1 = new Thread(evenSeries);  Thread thread2 = new Thread(fibonacciSeries);  thread1.start();  thread2.start();  }  } |

**OUTPUT:**



**PROGRAM NO : 23**

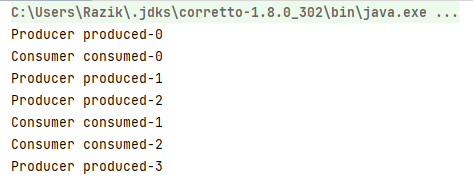
**AIM:** Producer/Consumer using ITC

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.LinkedList;  public class Main {  public static void main(String[] args) throws InterruptedException {  final PC pc = new PC();  Thread t1 = new Thread(new Runnable() {  @Override  public void run() {  try {  pc.produce();  } catch (InterruptedException e) {  e.printStackTrace();  }  }  });  Thread t2 = new Thread(new Runnable() {  @Override  public void run() {  try {  pc.consume();  } catch (InterruptedException e) {  e.printStackTrace();  }  }  });  t1.start();  t2.start();  t1.join();  t2.join();  }  public static class PC {  LinkedList<Integer> list = new LinkedList<>();  int capacity = 2;  public void produce() throws InterruptedException {  int value = 0;  while (true) {  synchronized (this) {  while (list.size() == capacity)  wait();  System.out.println("Producer produced-"  + value);  list.add(value++);  notify();  Thread.sleep(1000);  }  }  }  public void consume() throws InterruptedException {  while (true) {  synchronized (this) {  while (list.size() == 0)  wait();  int val = list.removeFirst();  System.out.println("Consumer consumed-"  + val);  notify();  Thread.sleep(1000);  }  }  }  }  } this.count=count;  }  public void run() {  for(int i=1;primeCount<count;i++){  boolean isPrime=true;  if(i==1)  {  i++;  }  for (int j=2;j<i;j++){  if(i%j==0){  isPrime = false;  break;  }  }  if (isPrime){  System.out.println("Prime Number : "+i);  primeCount++;  }  }  }  } |

**OUTPUT :**



**PROGRAM NO : 24**

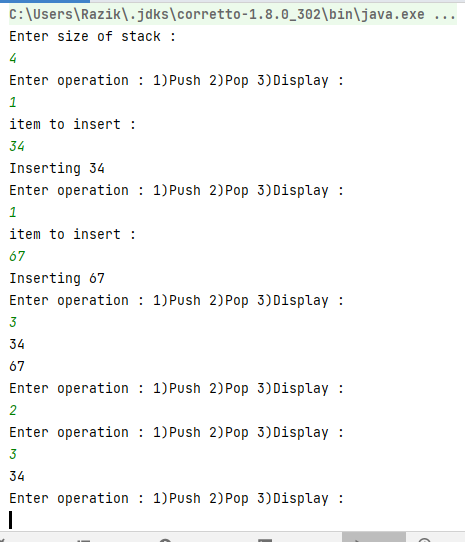
**AIM:** Program to create a generic stack and do the Push and Pop operations.

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.Scanner;  class Main{  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  System.out.println("Enter size of stack : ");  int count = scanner.nextInt();  GenericStack stack = new GenericStack(count);  while (true) {  System.out.println("Enter operation : 1)Push 2)Pop 3)Display : ");  int choice = scanner.nextInt();  switch (choice) {  case 1:  System.out.println("item to insert :");  int item = scanner.nextInt();  stack.push(item);  break;  case 2:  stack.pop();  break;  case 3:  stack.printStack();  }  }  }  } |
| GenericStack.java | class GenericStack {  private int[] arr;  private int top;  private int capacity;  // Creating a stack  GenericStack(int size) {  arr = new int[size];  capacity = size;  top = -1;  }  // Add elements into stack  public void push(int x) {  if (isFull()) {  System.out.println("OverFlow");  }  else{  System.out.println("Inserting " + x);  arr[++top] = x;  }  }  // Remove element from stack  public int pop() {  if (isEmpty()) {  System.out.println("STACK EMPTY");  return -1;  }  else {  return arr[top--];  }  }  // Utility function to return the size of the stack  public int size() {  return top + 1;  }  // Check if the stack is empty  public Boolean isEmpty() {  return top == -1;  }  // Check if the stack is full  public Boolean isFull() {  return top == capacity - 1;  }  public void printStack() {  for (int i = 0; i <= top; i++) {  System.out.println(arr[i]);  }  }  } |

**OUTPUT :**



**PROGRAM NO : 25**

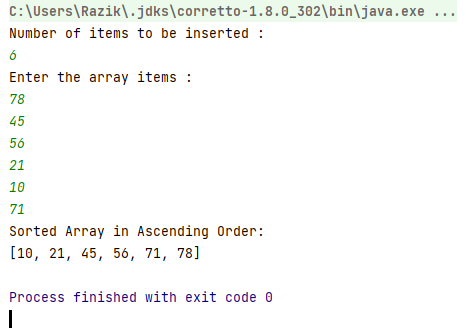
**AIM:** Using generic method perform Bubble sort.

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.Arrays;  import java.util.Scanner;  public class Main {  static void bubbleSort(int array[]) {  int size = array.length;  for (int i = 0; i < size - 1; i++)  for (int j = 0; j < size - i - 1; j++)  if (array[j] > array[j + 1]) {  int temp = array[j];  array[j] = array[j + 1];  array[j + 1] = temp;  }  }  public static void main(String args[]) {  Scanner scanner = new Scanner(System.in);  System.out.println("Number of items to be inserted : ");  int count = scanner.nextInt();  int[] data = new int[count];  System.out.println("Enter the array items : ");  for(int i=0;i<count;i++)  {  data[i] = scanner.nextInt();  }  Main.bubbleSort(data);  System.out.println("Sorted Array in Ascending Order:");  System.out.println(Arrays.toString(data));  }  } |

**OUTPUT :**



**PROGRAM NO : 26**

**AIM:** Maintain a list of Strings using ArrayList from collection framework, perform built-in operations.

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.\*;  public class Main {  public static void main(String[] args) {  // Creating ArrayList of type "String" which means we can only add "String" elements  ArrayList<String> fruits = new ArrayList<String>();  //adding elements to an ArrayList  fruits.add("Pomegranate");  fruits.add("Lemon");  fruits.add("Avocado");  fruits.add("Durian");  fruits.add("Watermelon");  fruits.add(3, "Orange");  // Displaying elements  System.out.println("\n ORIGINAL LIST:");  System.out.println("-----------------------------------------------------");  for(String str : fruits)  System.out.printf(str+" ");  //Remove elements from ArrayList  fruits.remove("Avocado");  fruits.remove(2);  // Displaying elements  System.out.println("\n-----------------------------------------------------");  System.out.println("\n\nARRAYLIST AFTER REMOVAL OF ELEMENTS:");  System.out.println("-------------------------------------");  for(String str : fruits )  System.out.printf(str+" ");  //Updating the ArrayList  fruits.set(3,"Guava");  System.out.println("\n-------------------------------------");  System.out.println("\n\n ARRAYLIST AFTER UPDATION:");  System.out.println("-------------------------------------");  for(String str : fruits )  System.out.printf(str+" ");  System.out.println("\n-------------------------------------");  //Sorting the ArrayList  Collections.sort(fruits);  System.out.println("\n\n ARRAYLIST AFTER SORTING:");  System.out.println("-------------------------------------");  for (String str : fruits)  System.out.printf(str+" ");  // Checks whether the object is in the ArrayList  System.out.println("\n------------------------------------------");  System.out.println("\nApple is in the List- "+ fruits.contains("Durian"));  System.out.println("Strawberry is in the List- "+fruits.contains("Strawberry"));  //Size of the ArrayList  System.out.println("\n------------------------------------------");  System.out.println("\nSIZE OF THE ARRAYLIST: "+ fruits.size());  //returns the object of list which is present at the specified index  System.out.println("\n------------------------------------------");  System.out.println("\n\nOBJECT AT INDEX 2: "+ fruits.get(2));  // removing all the elements of the ArrayList  fruits.clear();  System.out.println("\nARRAYLIST AFTER Clear(): "+ fruits);  }  } |

**OUTPUT :**

ORIGINAL LIST:

-----------------------------------------------------

Pomegranate Lemon Avocado Orange Durian Watermelon

-----------------------------------------------------

ARRAYLIST AFTER REMOVAL OF ELEMENTS:

-------------------------------------

Pomegranate Lemon Durian Watermelon

-------------------------------------

ARRAYLIST AFTER UPDATION:

-------------------------------------

Pomegranate Lemon Durian Guava

-------------------------------------

ARRAYLIST AFTER SORTING:

-------------------------------------

Durian Guava Lemon Pomegranate

------------------------------------------

Apple is in the List- true

Strawberry is in the List- false

------------------------------------------

SIZE OF THE ARRAYLIST: 4

------------------------------------------

OBJECT AT INDEX 2: Lemon

ARRAYLIST AFTER Clear(): []

Process finished with exit code 0

**PROGRAM NO : 27**

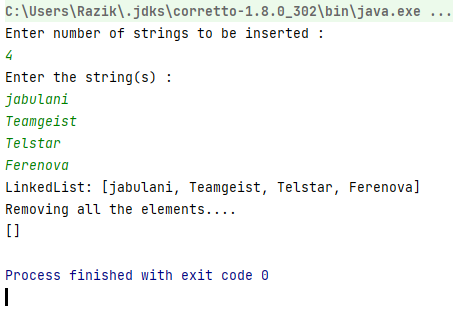
**AIM:** Program to remove all the elements from a linked list

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.\*;  public class Main {  public static void main(String[] args) {  int n;  String data;  LinkedList<String> linkedList = new LinkedList<String>();  System.out.println("Enter number of strings to be inserted : ");  Scanner scanner = new Scanner(System.in);  n = scanner.nextInt();  System.out.println("Enter the string(s) : ");  scanner.nextLine();  for (int i = 0; i < n; i++) {  data = scanner.nextLine();  linkedList.add(data);  }  System.out.println("LinkedList: " + linkedList);  System.out.println("Removing all the elements....");  linkedList.clear();  System.out.println(linkedList);  }  } |

**OUTPUT :**



**PROGRAM NO : 28**

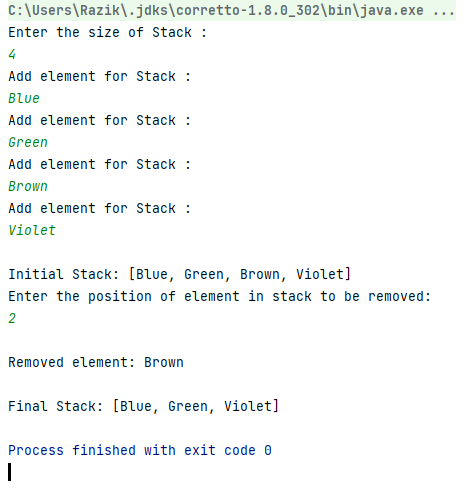
**AIM:** Program to remove an object from the Stack when the position is passed as parameter.

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.Scanner;  import java.util.Stack;  public class Main {  public static void main(String[] args) {  Stack<String> stack = new Stack<String>();  Scanner scanner=new Scanner(System.in);  System.out.println("Enter the size of Stack : ");  int num=scanner.nextInt();  for(int i =0;i<num;i++)  {  System.out.println("Add element for Stack : ");  String str=scanner.next();  stack.add(str);  }  System.out.println();  System.out.println("Initial Stack: " + stack);  System.out.println("Enter the position of element in stack to be removed: ");  int pos=scanner.nextInt();  String rem = stack.remove(pos);  System.out.println("\nRemoved element: "+ rem);  System.out.println("\nFinal Stack: " + stack);  }  } |

**OUTPUT :**



**PROGRAM NO : 29**

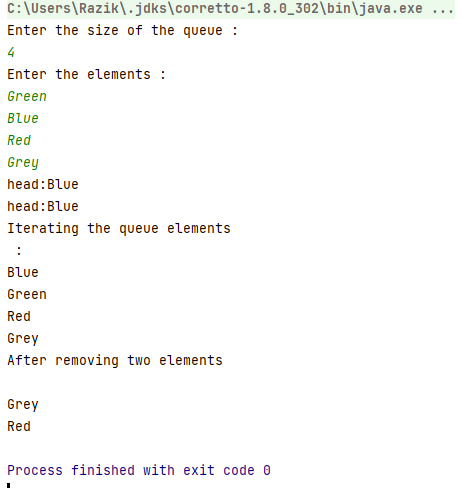
**AIM:** Program to demonstrate the creation of queue object using the PriorityQueue Class

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.PriorityQueue;  import java.util.Scanner;  public class Main {  public static void main(String[] args)  {  PriorityQueue<String> queue=new PriorityQueue<String>();  Scanner scanner=new Scanner(System.in);  System.out.println("Enter the size of the queue : ");  int n=scanner.nextInt();  System.out.println("Enter the elements : ");  for(int i =0;i<n;i++)  {  String st=scanner.next();  queue.add(st);  }  System.out.println("head:"+queue.element());  System.out.println("head:"+queue.peek());  System.out.println("Iterating the queue elements\n : ");  for (String value : queue) {  System.out.println(value);  }  queue.remove();  queue.poll();  System.out.println("After removing two elements \n");  for (String s : queue) {  System.out.println(s);  }  }  } |

**OUTPUT :**



**PROGRAM NO : 30**

**AIM:** Program to demonstrate the addition and deletion of elements in deque

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.\*;  public class Main {  public static void main(String[] args) {  int ch, data;  Deque<Integer> dq = new LinkedList<Integer>();  Scanner sc = new Scanner(System.in);  do  {  System.out.println("\n\_\_\_\_\_\_MENU\_\_\_\_\_\_");  System.out.println("1.Insert the element at first");  System.out.println("2.Insert the element at last");  System.out.println("3.Delete the element at first");  System.out.println("4.Delete the element at last");  System.out.println("5.Display");  System.out.println("6.Exit");  System.out.println("\nEnter the choice(1-6):");  ch = sc.nextInt();  sc.nextLine();  switch(ch)  {  case 1: System.out.println("Enter the element to be inserted at first:");  data = sc.nextInt();  dq.addFirst(data);  break;  case 2: System.out.println("Enter the element to be inserted at last:");  data = sc.nextInt();  dq.addLast(data);  break;  case 3: System.out.println("Element deleted from the first position");  dq.removeFirst();  break;  case 4: System.out.println("Element deleted from the last position");  dq.removeLast();  break;  case 5: System.out.println("Elements:");  System.out.println(dq);  break;  case 6: System.exit(0);  break;  default:System.out.println("Invalid choice...");  }  }while(true);  }  } |

**OUTPUT :**

\_\_\_\_\_\_MENU\_\_\_\_\_\_

1.Insert the element at first

2.Insert the element at last

3.Delete the element at first

4.Delete the element at last

5.Display

6.Exit

Enter the choice(1-6):

1

Enter the element to be inserted at first:

34

\_\_\_\_\_\_MENU\_\_\_\_\_\_

1.Insert the element at first

2.Insert the element at last

3.Delete the element at first

4.Delete the element at last

5.Display

6.Exit

Enter the choice(1-6):

1

Enter the element to be inserted at first:

56

\_\_\_\_\_\_MENU\_\_\_\_\_\_

1.Insert the element at first

2.Insert the element at last

3.Delete the element at first

4.Delete the element at last

5.Display

6.Exit

Enter the choice(1-6):

1

Enter the element to be inserted at first:

100

\_\_\_\_\_\_MENU\_\_\_\_\_\_

1.Insert the element at first

2.Insert the element at last

3.Delete the element at first

4.Delete the element at last

5.Display

6.Exit

Enter the choice(1-6):

5

Elements:

[100, 56, 34]

\_\_\_\_\_\_MENU\_\_\_\_\_\_

1.Insert the element at first

2.Insert the element at last

3.Delete the element at first

4.Delete the element at last

5.Display

6.Exit

Enter the choice(1-6):

4

Element deleted from the last position

\_\_\_\_\_\_MENU\_\_\_\_\_\_

1.Insert the element at first

2.Insert the element at last

3.Delete the element at first

4.Delete the element at last

5.Display

6.Exit

Enter the choice(1-6):

5

Elements:

[100, 56]

\_\_\_\_\_\_MENU\_\_\_\_\_\_

1.Insert the element at first

2.Insert the element at last

3.Delete the element at first

4.Delete the element at last

5.Display

6.Exit

Enter the choice(1-6):

6

Process finished with exit code 0

**PROGRAM NO : 31**

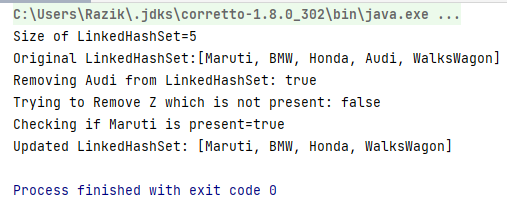
**AIM:** Program to demonstrate the creation of Set object using the LinkedHashset Class.

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.LinkedHashSet;  public class Main {  public static void main(String[] args) {  LinkedHashSet<String> linkedset = new LinkedHashSet<String>();  // Adding element to LinkedHashSet  linkedset.add("Maruti");  linkedset.add("BMW");  linkedset.add("Honda");  linkedset.add("Audi");  linkedset.add("Maruti"); //This will not add new element as Maruti already exists  linkedset.add("WalksWagon");  System.out.println("Size of LinkedHashSet=" + linkedset.size());  System.out.println("Original LinkedHashSet:" + linkedset);  System.out.println("Removing Audi from LinkedHashSet: " + linkedset.remove("Audi"));  System.out.println("Trying to Remove Z which is not present: "  + linkedset.remove("Z"));  System.out.println("Checking if Maruti is present=" + linkedset.contains("Maruti"));  System.out.println("Updated LinkedHashSet: " + linkedset);  }  } |

**OUTPUT :**



**PROGRAM NO : 32**

**AIM:** Write a Java program to compare two hash sets.

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.\*;  public class Main {  public static void main(String[] args) {  int n;  String str;  HashSet<String> set1= new HashSet<String>();  System.out.println("HashSet 1");  System.out.println("Enter No. of countries:");  Scanner sc=new Scanner(System.in);  n=sc.nextInt();  System.out.println("Enter the name of countries:");  Scanner sc1=new Scanner(System.in);  for(int i=0;i<n;i++) {  str=sc1.nextLine();  set1.add(str);  }  System.out.println("HashSet 2");  HashSet<String> set2= new HashSet<String>();  System.out.println("Enter No. of countries:");  n=sc.nextInt();  System.out.println("Enter the name of countries:");  for(int i=0;i<n;i++) {  str=sc1.nextLine();  set2.add(str);  }  System.out.println("Set1:"+set1);  System.out.println("Set2:"+set2);  HashSet<String> a= new HashSet<String>(set1);  a.addAll(set2);  System.out.println("Union of country set:"+a);  HashSet<String> b= new HashSet<String>(set1);  b.retainAll(set2);  System.out.println("Intersection of country set:"+b);  HashSet<String> c= new HashSet<String>(set1);  c.removeAll(set2);  System.out.println("Difference of country set:"+c);  }  } |

**OUTPUT :**



**PROGRAM NO : 33**

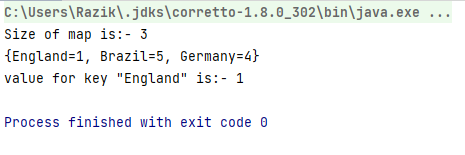
**AIM:** Program to demonstrate the working of Map interface by adding, changing and removing elements.

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | // Java program to illustrate HashMap class of java.util  // package  // Importing HashMap class  import java.util.HashMap;  // Main class  public class Main {  // Main driver method  public static void main(String[] args)  {  // Create an empty hash map by declaring object  // of string and integer type  HashMap<String, Integer> map = new HashMap<>();  // Adding elements to the Map  // using standard add() method  map.put("Germany", 4);  map.put("England", 1);  map.put("Brazil", 5);  // Print size and content of the Map  System.out.println("Size of map is:- "  + map.size());  // Printing elements in object of Map  System.out.println(map);  // Checking if a key is present and if  // present, print value by passing  // random element  if (map.containsKey("England")) {  // Mapping  Integer a = map.get("England");  // Printing value fr the corresponding key  System.out.println("value for key"  + " \"England\" is:- " + a);  }  }  } |

**OUTPUT :**



**PROGRAM NO : 34**

**AIM:** Program to Convert HashMap to TreeMap.

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.\*;  public class Main {  public static void main(String[] args) {  // TODO Auto-generated method stub  Map<Integer,String> hm=new LinkedHashMap<>();  hm.put(1,"England");  hm.put(1,"Spain");  hm.put(2,"France");  hm.put(5,"Brazil");  hm.put(2,"Argentina");  System.out.println("HashMap:"+hm);  Map<Integer,String> tm=new TreeMap<>(hm);  System.out.println("\n \*\*\*\*\*\*\* Convert HashMap to TreeMap \*\*\*\*\*\*\*\n");  System.out.println("TreeMap:"+tm);  }  } |

**OUTPUT :**

