# **Experiment 4**

**1.   You are required to develop a Java program that implements an interface to perform basic arithmetic operations.**

**The interface, named "ArithmeticOperations," should define four methods: add, subtract, multiply, and divide.**

**The program should also include a class named "Calculator" that implements the "ArithmeticOperations" interface.**

**The "add" method should take two integers as parameters and return their sum. The "subtract" method should take two integers as parameters and return their difference.**

**The "multiply" method should take two integers as parameters and return their product. The "divide" method should take two integers as parameters and return their quotient as a floating-point number.**

**In the "Calculator" class, implement the methods defined in the "ArithmeticOperations" interface.**

**Create an instance of the "Calculator" class and demonstrate the usage of each method by performing arithmetic operations on different pairs of numbers.**

**Ensure that the program is user-friendly and provides clear instructions for inputting the numbers and displaying the results.**

# **Program:**

import java.util.Scanner;

interface ArithmaticOperations{

    public int add(int a,int b);

    public int subtract(int a,int b);

    public int multiply(int a,int b);

    public float divide(int a,int b);

}

class Calculator implements ArithmaticOperations{

    public int add(int a,int b){

        return a+b;

    }

    public int subtract(int a, int b){

        return a-b;

    }

    public int multiply(int a,int b){

        return a\*b;

    }

    public float divide(int a,int b){

        return a/b;

    }

}

class exp4\_1{

    public static void main(String args[]){

        Calculator c1=new Calculator();

        Scanner sc= new Scanner(System.in);

        int a,b,choice;

        do{

            System.out.println("\nEnter Choice");

            System.out.println("\n1.Add\n2.Subtract\n3.Multiply\n4.Divide\n5.Exit");

            choice=sc.nextInt();

            switch(choice){

                case 1:

                    System.out.println("Enter two numbers");

                    a=sc.nextInt();

                    b=sc.nextInt();

                    int sum=c1.add(a,b);

                    System.out.print("\nSum:"+sum);

                    break;

                case 2:

                    System.out.println("Enter two numbers");

                    a=sc.nextInt();

                    b=sc.nextInt();

                    int subtraction=c1.subtract(a,b);

                    System.out.print("\nSubtraction:"+subtraction);

                    break;

                case 3:

                    System.out.println("Enter two numbers");

                    a=sc.nextInt();

                    b=sc.nextInt();

                    int multiplication=c1.multiply(a,b);

                    System.out.print("\nMultiplication:"+multiplication);

                    break;

                case 4:

                    System.out.println("Enter two numbers");

                    a=sc.nextInt();

                    b=sc.nextInt();

                    float division=c1.divide(a,b);

                    System.out.print("\nDivision:"+division);

                    break;

                case 5:

                    System.out.println("Exiting....");

                    break;

                default:

                    System.out.println("Enter valid choice");

            }

        }while(choice!=5);

    }

}

# **Output:**

Enter Choice

1.Add

2.Subtract

3.Multiply

4.Divide

5.Exit

1

Enter two numbers

5 4

Sum:9

Enter Choice

1.Add

2.Subtract

3.Multiply

4.Divide

5.Exit

2

Enter two numbers

56

56

Subtraction:0

Enter Choice

1.Add

2.Subtract

3.Multiply

4.Divide

5.Exit

3

Enter two numbers

34 23

Multiplication:782

Enter Choice

1.Add

2.Subtract

3.Multiply

4.Divide

5.Exit

4

Enter two numbers

34 2

Division:17.0

Enter Choice

1.Add

2.Subtract

3.Multiply

4.Divide

5.Exit

5

Exiting....

**2.   Problem Statement: Design an interface in Java to calculate the area of different shapes.**

**Create an interface named "Shape" that includes the following methods:**

**calculateArea(): This method should return the area of the shape as a double value. Next, implement three classes that implement the "Shape" interface:**

**Circle: This class should have a constructor that takes the radius of the circle as a parameter and implements the calculateArea() method to calculate and return the area of the circle. Rectangle: This class should have a constructor that takes the length and width of the rectangle as parameters and implements the calculateArea() method to calculate and return the area of the rectangle. Triangle: This class should have a constructor that takes the base and height of the triangle as parameters and implements the calculateArea() method to calculate and return the area of the triangle.**

**Write a Java program that demonstrates the usage of the interface and the implemented classes. It should create objects of the Circle, Rectangle, and Triangle classes, call the calculateArea() method for each object, and display the calculated areas.**

# **Program:**

import java.util.Scanner;

interface Shape{

    double calculateArea();

}

class Circle implements Shape{

    private

    double radius;

    public Circle(double radius){

        this.radius=radius;

    }

    public double calculateArea(){

        return 3.14\*radius\*radius;

    }

}

class Rectangle implements Shape{

    private

    double length;

    double breadth;

    public Rectangle(double length,double breadth){

        this.length=length;

        this.breadth=breadth;

    }

     public double calculateArea(){

        return length\*breadth;

     }

}

class Triangle implements Shape{

    private

    double base;

    double height;

     public Triangle(double base,double height){

        this.base=base;

        this.height=height;

     }

     public double calculateArea(){

        return 0.5\*base\*height;

     }

}

class exp4\_2{

    public static void main(String args[]){

        Scanner sc=new Scanner(System.in);

        int choice;

        double a,b;

        double area;

        do{

            System.out.println("\nEnter Choice");

            System.out.println("\n1.Area of Circle\n2.Area of Rectangle\n3.Area of a Triangle\n4.Exit");

            choice=sc.nextInt();

            switch(choice){

                case 1:

                    System.out.println("Enter radius");

                    a=sc.nextInt();

                    Circle c=new Circle(a);

                    area=c.calculateArea();

                    System.out.print("\nArea of Circle:"+area);

                    break;

                case 2:

                    System.out.println("Enter Length and breadth ");

                    a=sc.nextInt();

                    b=sc.nextInt();

                    Rectangle r=new Rectangle(a,b);

                    area=r.calculateArea();

                    System.out.print("\nArea of Rectangle:"+area);

                    break;

                case 3:

                    System.out.println("Enter base and height");

                    a=sc.nextInt();

                    b=sc.nextInt();

                    Triangle t=new Triangle(a,b);

                    area=t.calculateArea();

                    System.out.print("\nArea of a Triangle:"+area);

                    break;

                case 4:

                    System.out.println("Exiting....");

                    break;

                default:

                    System.out.println("Enter valid choice");

            }

        }while(choice!=4);

    }

}

# **Output:**

Enter Choice

1.Area of Circle

2.Area of Rectangle

3.Area of a Triangle

4.Exit

1

Enter radius

7

Area of Circle:153.86

Enter Choice

1.Area of Circle

2.Area of Rectangle

3.Area of a Triangle

4.Exit

2

Enter Length and breadth

7 8

Area of Rectangle:56.0

Enter Choice

1.Area of Circle

2.Area of Rectangle

3.Area of a Triangle

4.Exit

3

Enter base and height

6 10

Area of a Triangle:30.0

Enter Choice

1.Area of Circle

2.Area of Rectangle

3.Area of a Triangle

4.Exit

4

Exiting....

**3. Problem Statement: Create a Java program that implements an interface called "Shape" to represent different geometric shapes. The interface should contain the following methods:**

**calculateArea(): This method should be used to calculate the area of the shape and return it as a double.**

**calculatePerimeter(): This method should be used to calculate the perimeter of the shape and return it as a double.**

**Create three classes that implement the "Shape" interface: "Circle", "Rectangle", and "Triangle". Each class should provide its own implementation for the calculateArea() and calculatePerimeter() methods.**

**Write a main program that demonstrates the usage of these classes. Create objects of each class, set appropriate values for their dimensions, and display the calculated area and perimeter of each shape.**

# **Program:**

import java.util.Scanner;

interface Shape{

    double calculateArea();

    double calculatePerimeter();

}

class Circle implements Shape{

    private

    double radius;

    public Circle(double radius){

        this.radius=radius;

    }

    public double calculateArea(){

        return 3.14\*radius\*radius;

    }

    public double calculatePerimeter(){

        return 2\*3.14\*radius;

    }

}

class Rectangle implements Shape{

    private

    double length;

    double breadth;

    public Rectangle(double length,double breadth){

        this.length=length;

        this.breadth=breadth;

    }

     public double calculateArea(){

        return length\*breadth;

     }

     public double calculatePerimeter(){

        return 2\*(length+breadth);

     }

}

class Triangle implements Shape{

    private

    double base;

    double height;

     public Triangle(double base,double height){

        this.base=base;

        this.height=height;

     }

     public double calculateArea(){

        return 0.5\*base\*height;

     }

     public double calculatePerimeter(){

        double hypotenus\_square= base\*height;

        double hypotenus=Math.sqrt(hypotenus\_square);

        return (base+height+ hypotenus);

     }

}

class exp4\_3{

    public static void main(String args[]){

        Scanner sc=new Scanner(System.in);

        int choice;

        double a,b;

        double area,perimeter;

        do{

            System.out.println("\n\nEnter Choice");

            System.out.println("1.Area and perimeter of Circle\n2.Area and perimeter of Rectangle\n3.Area of and perimeter a Triangle\n4.Exit");

            choice=sc.nextInt();

            switch(choice){

                case 1:

                    System.out.println("Enter radius");

                    a=sc.nextInt();

                    Circle c=new Circle(a);

                    area=c.calculateArea();

                    perimeter=c.calculatePerimeter();

                    System.out.print("\nArea of Circle:"+area);

                    System.out.print("\nPerimeter of Circle:"+perimeter);

                    break;

                case 2:

                    System.out.println("Enter Length and breadth ");

                    a=sc.nextInt();

                    b=sc.nextInt();

                    Rectangle r=new Rectangle(a,b);

                    area=r.calculateArea();

                    perimeter=r.calculatePerimeter();

                    System.out.print("\nArea of Rectangle:"+area);

                    System.out.print("\nPerimeter  of Rectangle:"+perimeter );

                    break;

                case 3:

                    System.out.println("Enter base and height");

                    a=sc.nextInt();

                    b=sc.nextInt();

                    Triangle t=new Triangle(a,b);

                    area=t.calculateArea();

                    perimeter=t.calculatePerimeter();

                    System.out.print("\nArea of a Triangle:"+area);

                    System.out.print("\nPerimeter of a Triangle:"+perimeter);

                    break;

                case 4:

                    System.out.println("Exiting....");

                    break;

                default:

                    System.out.println("Enter valid choice");

            }

        }while(choice!=4);

    }

}

# **Output:**

Enter Choice

1.Area and perimeter of Circle

2.Area and perimeter of Rectangle

3.Area of and perimeter a Triangle

4.Exit

1

Enter radius

7

Area of Circle:153.86

Perimeter of Circle:43.96

Enter Choice

1.Area and perimeter of Circle

2.Area and perimeter of Rectangle

3.Area of and perimeter a Triangle

4.Exit

2

Enter Length and breadth

5 6

Area of Rectangle:30.0

Perimeter of Rectangle:22.0

Enter Choice

1.Area and perimeter of Circle

2.Area and perimeter of Rectangle

3.Area of and perimeter a Triangle

4.Exit

3

Enter base and height

6 10

Area of a Triangle:30.0

Perimeter of a Triangle:23.745966692414832

Enter Choice

1.Area and perimeter of Circle

2.Area and perimeter of Rectangle

3.Area of and perimeter a Triangle

4.Exit

4

Exiting....

**4.   Problem Statement: Implement an interface-based solution in Java to manage a library system.Create an interface called LibraryItem with the following methods:**

**void checkOut(): Marks the library item as checked out.**

**void checkIn(): Marks the library item as checked in.**

**boolean isAvailable(): Returns true if the library item is available for checkout, false otherwise.**

**Implement two classes that represent different types of library items: Book and DVD.**

**Both classes should implement the LibraryItem interface.**

**Each class should have appropriate instance variables and methods to represent the item's unique properties.**

**For example, the Book class can have instance variables like title, author, and ISBN, along with getters and setters.**

**Create a Library class that manages a collection of library items.**

**The Library class should have methods to add items to the library, remove items from the library, and search for items by title or author.**

**The library should also keep track of the items that are currently checked out.**

**Implement a LibraryApp class with a main method that demonstrates the usage of the library system.**

**The main method should create instances of books and DVDs, add them to the library, and perform various operations such as checking out and checking in items.**

**Display appropriate messages to indicate the success or failure of each operation.**

**Your task is to design and implement the classes and interfaces mentioned above, ensuring proper encapsulation and adherence to object-oriented principles. Use appropriate data structures and algorithms to efficiently manage the library system.**

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Scanner;

interface LibraryItem {

  void checkOut();

  void checkIn();

  boolean isAvailable();

  String getTitle();

}

class Book implements LibraryItem {

    String title;

    String author;

    String ISBN;

    int price;

    boolean isAvailable = true; // Track availability status

    public Book(String title, String author, String ISBN, int price) {

        this.title = title;

        this.author = author;

        this.ISBN = ISBN;

        this.price = price;

    }

    public String getTitle() {

        return title;

    }

    public String getAuthor() {

        return author;

    }

    public String getISBN() {

        return ISBN;

    }

    public int getPrice() {

        return price;

    }

    @Override

    public void checkOut() {

        if (isAvailable) {

            System.out.println("Book \"" + title + "\" by " + author + " checked out successfully!");

            isAvailable = false;

        } else {

            System.out.println("Sorry, book \"" + title + "\" is currently unavailable.");

        }

    }

    @Override

    public void checkIn() {

        if (!isAvailable) {

            System.out.println("Book \"" + title + "\" by " + author + " checked in successfully!");

            isAvailable = true;

        } else {

            System.out.println("Book \"" + title + "\" by " + author + " is already available.");

        }

    }

    @Override

    public boolean isAvailable() {

        return isAvailable;

    }

}

class DVD implements LibraryItem {

    String title;

    String director;

    String genre;

    String language;

    String releaseDate;

    String duration;

    boolean isAvailable = true; // Track availability status

    public DVD(String title, String director, String genre, String language, String releaseDate, String duration) {

        this.title = title;

        this.director = director;

        this.genre = genre;

        this.language = language;

        this.releaseDate = releaseDate;

        this.duration = duration;

    }

    public String getTitle() {

        return title;

    }

    public String getLanguage() {

        return language;

    }

    public String getDirector(){

        return director;

    }

    @Override

    public void checkOut() {

        if (isAvailable) {

            System.out.println("DVD \"" + title + "\" by " + director + " checked out successfully!");

            isAvailable = false;

        } else {

            System.out.println("Sorry, DVD \"" + title + "\" is currently unavailable.");

        }

    }

    @Override

    public void checkIn() {

        if (!isAvailable) {

            System.out.println("DVD \"" + title + "\" by " + director + " checked in successfully!");

            isAvailable = true;

        } else {

            System.out.println("DVD \"" + title + "\" by " + director + " is already available.");

        }

    }

    @Override

    public boolean isAvailable() {

        return isAvailable;

    }

}

class Library {

    private ArrayList<Book> books = new ArrayList<>();

    private ArrayList<DVD> dvds = new ArrayList<>();

    private HashMap<String, LibraryItem> checkedOutItems = new HashMap<>();

    // ... existing methods ...

    // Added methods for managing checked out items

    public void displayCheckedOutItems() {

        // Display both books and DVDs

        displayCheckedOutBooks();

        displayCheckedOutDVDs();

    }

    public void displayCheckedOutBooks() {

        System.out.println("Books currently checked out:");

        for (LibraryItem item : checkedOutItems.values()) {

            if (item instanceof Book) {

                Book book = (Book) item;

                System.out.println(book.getTitle() + " by " + book.getAuthor());

            }

        }

    }

    public void displayCheckedOutDVDs() {

        System.out.println("DVDs currently checked out:");

        for (LibraryItem item : checkedOutItems.values()) {

            if (item instanceof DVD) {

                DVD dvd = (DVD) item;

                System.out.println(dvd.getTitle() + " by " + dvd.getDirector());

            }

        }

    }

    public Book findCheckedOutBookByTitle(String title) {

        for (LibraryItem item : checkedOutItems.values()) {

            if (item instanceof Book && item.getTitle().equalsIgnoreCase(title)) {

                return (Book) item;

            }

        }

        return null;

    }

    public DVD findCheckedOutDVDByTitle(String title) {

        for (LibraryItem item : checkedOutItems.values()) {

            if (item instanceof DVD && item.getTitle().equalsIgnoreCase(title)) {

                return (DVD) item;

            }

        }

        return null;

    }

    public void removeBook(Book book) {

        checkedOutItems.remove(book.getTitle());

        books.remove(book);

    }

    public void removeDVD(DVD dvd) {

        checkedOutItems.remove(dvd.getTitle());

        dvds.remove(dvd);

    }

    public void addBook(Book book) {

        books.add(book);

      }

      public void addDVD(DVD dvd) {

        dvds.add(dvd);

      }

}

class exp4\_4 {

    public static void main(String[] args) {

        Library library = new Library();  // Create an instance of the Library

        Scanner sc = new Scanner(System.in);

        Book[] books = new Book[5];   // Changed the array size to 5

        books[0] = new Book("Can't Hurt Me", "David Goggins", "9780062857218", 500);

        books[1] = new Book("Mrutyunjaya", "Shivaji Sawant", "9782253059177", 857);

        books[2] = new Book("Shyamchi Aai", "Sane Guruji", "9788184565186", 292);

        books[3] = new Book("How to Read a Book", "Mortimer J. Adler", "9780671212094", 426);

        books[4] = new Book("Yugandhar", "Shivaji Sawant", "9788184987984", 650);

//Similar changes to store DVDs in correct manner

DVD[] dvds = new DVD[4];

        dvds[0] = new DVD("Atomic Habits", "James Clear", "Motivational", "English", "16 October 2018", "5 Hour 35 Minute");

        dvds[1] = new DVD("The Ultimate Zig Ziglar Library", "Zig Ziglar", "Self-Help", "English", "25 April 2016", "13 hour 55 min");

        dvds[2] = new DVD("Shrimad Bhagavad Gita", "Om Swami", "Spirituality", "English", "14 October 2021", "Duration not Available");

        dvds[3] = new DVD("Dopamine Detox", "Joshua Alexander", "Self-Improvement", "English", "26 August 2021", "1 hour 5 min");

        int choice;

        do {

            System.out.println("\n\nWhat do you want?");

            System.out.println("1. Get Book\n2. Get DVD\n3. Return Book\n4. Return DVD\n5. Exit");

            choice = sc.nextInt();

            sc.nextLine();  // Consume newline character after integer input

            switch (choice) {

                case 1:

                    // Display available books

                    System.out.println("Available Books:");

                    for (Book book : books) {

                        if (book != null && book.isAvailable()) {

                            System.out.println(book.getTitle() + " by " + book.getAuthor());

                        }

                    }

                    // Take input for the book the user wants

                    System.out.print("Enter the title of the book you want: ");

                    String bookTitle = sc.nextLine();

                    // Find the selected book in the array

                    Book selectedBook = null;

                    for (Book book : books) {

                        if (book != null && book.getTitle().equalsIgnoreCase(bookTitle)) {

                            selectedBook = book;

                            break;

                        }

                    }

                    if (selectedBook != null && selectedBook.isAvailable()) {

                        selectedBook.checkOut();

                        library.addBook(selectedBook);  // Add the book to the library

                    } else {

                        System.out.println("Book is unavailable or not found.");

                    }

                    break;

                case 2:

                    // Display available DVDs

                    System.out.println("Available DVDs:");

                    for (DVD dvd : dvds) {

                        if (dvd != null && dvd.isAvailable()) {

                            System.out.println(dvd.getTitle() + " by " + dvd.getDirector());

                        }

                    }

                    // Take input for the DVD the user wants

                    System.out.print("Enter the title of the DVD you want: ");

                    String dvdTitle = sc.nextLine();

                    // Find the selected DVD in the array

                    DVD selectedDVD = null;

                    for (DVD dvd : dvds) {

                        if (dvd != null && dvd.getTitle().equalsIgnoreCase(dvdTitle)) {

                            selectedDVD = dvd;

                            break;

                        }

                    }

                    if (selectedDVD != null && selectedDVD.isAvailable()) {

                        selectedDVD.checkOut();

                        library.addDVD(selectedDVD);  // Add the DVD to the library

                    } else {

                        System.out.println("DVD is unavailable or not found.");

                    }

                    break;

                    // Code for menu and user input

                    case 3:

                        // Return a book to the library

                        System.out.println("Books currently checked out:");

                        library.displayCheckedOutBooks();  // Call the method on the library instance

                        System.out.print("Enter the title of the book to return: ");

                        String returnBookTitle = sc.nextLine();

                        Book returnBook = library.findCheckedOutBookByTitle(returnBookTitle);  // Call the method on the library instance

                        if (returnBook != null) {

                            returnBook.checkIn();

                            library.removeBook(returnBook);  // Call the method on the library instance

                        } else {

                            System.out.println("Book not found in the checked out items.");

                        }

                        break;

                    case 4:

                        // Return a DVD to the library

                        System.out.println("DVDs currently checked out:");

                        library.displayCheckedOutDVDs();  // Call the method on the library instance

                        System.out.print("Enter the title of the DVD to return: ");

                        String returnDVDTitle = sc.nextLine();

                        DVD returnDVD = library.findCheckedOutDVDByTitle(returnDVDTitle);  // Call the method on the library instance

                        if (returnDVD != null) {

                            returnDVD.checkIn();

                            library.removeDVD(returnDVD);  // Call the method on the library instance

                        } else {

                            System.out.println("DVD not found in the checked out items.");

                        }

                        break;

                case 5:

                    System.out.println("Exiting...");

                    break;

                default:

                    System.out.println("Enter valid choice");

            }

        }while(choice!=5);

    }

}

# **Output:**

What do you want?

1. Get Book

2. Get DVD

3. Return Book

4. Return DVD

5. Exit

1

Available Books:

Can't Hurt Me by David Goggins

Mrutyunjaya by Shivaji Sawant

Shyamchi Aai by Sane Guruji

How to Read a Book by Mortimer J. Adler

Yugandhar by Shivaji Sawant

Enter the title of the book you want: can't hurt me

Book "Can't Hurt Me" by David Goggins checked out successfully!

What do you want?

1. Get Book

2. Get DVD

3. Return Book

4. Return DVD

5. Exit

2

Available DVDs:

Atomic Habits by James Clear

The Ultimate Zig Ziglar Library by Zig Ziglar

Shrimad Bhagavad Gita by Om Swami

Dopamine Detox by Joshua Alexander

Enter the title of the DVD you want: atomic habits

DVD "Atomic Habits" by James Clear checked out successfully!

What do you want?

1. Get Book

2. Get DVD

3. Return Book

4. Return DVD

5. Exit

5

Exiting...