

Ex.No.2

Access Specifiers

26.08.2020

Mugilan E.S.

2019202033

1. Constructor & Destructors

Student.java

```
package lab.two;
public class Student {
    private String name;
    private int age;
    public Student() {
        this.name = "Mugilan";
        this.age = 21;
    }
    public Student(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public void getDetails() {
        System.out.println("Name of the java.lab.one.Student: " + name);
        System.out.println("Age of the java.lab.one.Student : " + age);
    }
    @Override
    protected void finalize() {
        System.out.println("Destructor Called...");
    }
}
```

Main.java

```
package lab.two;
public class Main {
    public static void main(String[] args) {
        Student stud1 = new Student();
        Student stud2 = new Student("Samantha", 30);
        stud1.getDetails();
        stud2.getDetails();
        stud1 = null; // Needed for Destructor to be called
        stud2 = null;
        System.gc();
    }
}
```

Output:

```
//Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/
Name of the java.lab.one.Student: Mugilan
Age of the java.lab.one.Student : 21
Name of the java.lab.one.Student: Samantha
Age of the java.lab.one.Student : 30
Destructor Called...
Destructor Called...

Process finished with exit code 0
```

2. Access Modifiers

Animal.java

```
package lab.two.animal;
public class Animal {
    void jump() {
        System.out.println("I can Jump");
    }
    public void display() {
        System.out.println("I am an Animal");
    }
    private void run() {
        System.out.println("Animal is running");
    }
    protected void eat() {
        System.out.println("Animal is eating");
    }
}
class Cat extends Animal {
    public static void main(String[] args) {
        Cat cat = new Cat();
        cat.jump();
        cat.display();
        // cat.run(); // 'run()' has private access in 'lab.two.animal.Animal'
        cat.eat();
    }
}
```

NewAnimal.java

```
package lab.two.newanimal;
import lab.two.animal.Animal;
public class NewAnimal extends Animal {
    public static void main(String[] args) {
        NewAnimal dog = new NewAnimal();
        // dog.jump(); // 'jump()' is not public in 'lab.two.animal.Animal'.
        // Cannot be accessed from outside package
    }
}
```

```

        dog.display();
        // dog.run(); // 'run()' has private access in 'lab.two.animal.Animal'
        dog.eat();
    }
}

```

Output:

```

/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/
I can Jump
I am an Animal
Animal is eating
Process finished with exit code 0

```

```

/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/
I am an Animal
Animal is eating
Process finished with exit code 0

```

[3, 4, 5]. Matrices

AddMatrix.java

```

package lab.two.Matrices;
import java.util.Scanner;
public class AddMatrix {
    private int rows, columns;
    private int[][] a, b, sum;
    AddMatrix() {
        this.rows = 2;
        this.columns = 2;
        setMatrices();
    }
    AddMatrix(int rows, int columns) {
        this.rows = rows;
        this.columns = columns;
        setMatrices();
    }
    private void setMatrices() {
        this.a = new int[rows][columns];
        this.b = new int[rows][columns];
        this.sum = new int[rows][columns];
    }
    private void getInputs() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the elements of the first " + rows + " x " + columns + " matrix:");
        for(int i=0; i<rows; i++) {
            for(int j=0; j<columns; j++) {
                System.out.print(i + " x " + j + " element = ");
            }
        }
    }
}

```

```

        a[i][j] = sc.nextInt();
    }
}

System.out.println("Enter the elements of the second " + rows + " x " + columns + " matrix:");
for(int i=0; i<rows; i++) {
    for(int j=0; j<columns; j++) {
        System.out.print(i + " x " + j + " element = ");
        b[i][j] = sc.nextInt();
    }
}

}

public void displayMatrices() {
    System.out.println("Getting inputs...");
    getInputs();
    addMatrices();
    System.out.println("The First Matrix:");
    for(int i=0; i<rows; i++) {
        for(int j=0; j<columns; j++) {
            System.out.print(a[i][j] + " ");
        }
        System.out.println("");
    }
    System.out.println("The Second Matrix:");
    for(int i=0; i<rows; i++) {
        for(int j=0; j<columns; j++) {
            System.out.print(b[i][j] + " ");
        }
        System.out.println("");
    }
    System.out.println("The Sum Matrix:");
    for(int i=0; i<rows; i++) {
        for(int j=0; j<columns; j++) {
            System.out.print(sum[i][j] + " ");
        }
        System.out.println("");
    }
}

private void addMatrices() {
    for(int i=0; i<rows; i++) {
        for(int j=0; j<columns; j++) {
            sum[i][j] = a[i][j] + b[i][j];
        }
    }
}

```

```

    }
}

```

SubMatrix.java

```

package lab.two.Matrices;
import java.util.Scanner;
public class SubMatrix {
    private int rows, columns;
    private int[][] a, b, diff;
    SubMatrix() {
        this(2, 2);
    }
    SubMatrix(int rows, int columns) {
        this.rows = rows;
        this.columns = columns;
        setMatrices();
    }
    private void setMatrices() {
        this.a = new int[rows][columns];
        this.b = new int[rows][columns];
        this.diff = new int[rows][columns];
    }
    private void getInputs() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the elements of the first " + rows + " x " + columns + " matrix:");
        for(int i=0; i<rows; i++) {
            for(int j=0; j<columns; j++) {
                System.out.print(i + " x " + j + " element = ");
                a[i][j] = sc.nextInt();
            }
        }
        System.out.println("Enter the elements of the second " + rows + " x " + columns + " matrix:");
        for(int i=0; i<rows; i++) {
            for(int j=0; j<columns; j++) {
                System.out.print(i + " x " + j + " element = ");
                b[i][j] = sc.nextInt();
            }
        }
    }
    private void subMatrices() {
        for(int i=0; i<rows; i++) {
            for(int j=0; j<columns; j++) {
                diff[i][j] = a[i][j] - b[i][j];
            }
        }
    }
}

```

```

    }
}

}

public void displayMatrices() {
    System.out.println("Getting inputs...");
    getInputs();
    subMatrices();
    System.out.println("The First Matrix:");
    for(int i=0; i<rows; i++) {
        for(int j=0; j<columns; j++) {
            System.out.print(a[i][j] + " ");
        }
        System.out.println("");
    }
    System.out.println("The Second Matrix:");
    for(int i=0; i<rows; i++) {
        for(int j=0; j<columns; j++) {
            System.out.print(b[i][j] + " ");
        }
        System.out.println("");
    }
    System.out.println("The Difference Matrix:");
    for(int i=0; i<rows; i++) {
        for(int j=0; j<columns; j++) {
            System.out.print(diff[i][j] + " ");
        }
        System.out.println("");
    }
}
}
}

```

IdentityMatrix.java

```

package lab.two.Matrices;
import java.util.Scanner;
public class IdentityMatrix {
    private int rows, columns;
    private boolean isIdentity = true;
    private int[][] matrix;
    IdentityMatrix() {
        this(3);
    }
    IdentityMatrix(int side) {
        this.rows = side;
    }
}

```

```

        this.columns = side;
        setMatrix();
    }

    private void setMatrix() {
        this.matrix = new int[rows][columns];
    }

    private void getInput() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the elements of the " + rows + "x" + columns + " matrix:");
        for (int i=0; i<rows; i++) {
            for(int j=0; j<columns; j++) {
                System.out.print("Element " + i + "x" + j + ": ");
                matrix[i][j] = sc.nextInt();
            }
        }
    }

    private void displayMatrix() {
        System.out.println("The Matrix: ");
        for (int i=0; i<rows; i++) {
            for(int j=0; j<columns; j++) {
                System.out.print(matrix[i][j] + " ");
            }
            System.out.println(" ");
        }
    }

    public void checkIdentity() {
        getInput();
        displayMatrix();
        for(int i=0; i<rows; i++){
            for(int j=0; j<columns; j++) {
                if (i == j && matrix[i][j] != 1) {
                    isIdentity = false;
                    break;
                }
                if(i != j && matrix[i][j] != 0) {
                    isIdentity = false;
                    break;
                }
            }
        }
        if (isIdentity) {
            System.out.println("Given matrix is a identity matrix");
        } else {
            System.out.println("Given matrix is not a identity matrix");
        }
    }
}

```



```

    }
}
}

```

Main.java

```

package lab.two.Matrices;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        System.out.println("1 - Addition\n2 - Subtraction\n3 - Check Identity");
        Scanner input = new Scanner(System.in);
        System.out.print("Enter your choice: ");
        int choice = input.nextInt();
        switch (choice) {
            case 1:
                System.out.println("Matrix Addition");
                AddMatrix addMatrix = new AddMatrix();
                addMatrix.displayMatrices();
                break;
            case 2:
                System.out.println("Matrix Subtraction");
                SubMatrix subMatrix = new SubMatrix();
                subMatrix.displayMatrices();
                break;
            case 3:
                System.out.println("Check for Identity Matrix");
                IdentityMatrix identityMatrix = new IdentityMatrix();
                identityMatrix.checkIdentity();
                break;
            default:
                System.out.println("Invalid Choice");
        }
    }
}

```

Output:

```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/3
1 - Addition
2 - Subtraction
3 - Check Identity
Enter your choice: 1
Matrix Addition
Getting inputs...
Enter the elements of the first 2 x 2 matrix:
0 x 0 element = 1
0 x 1 element = 2
1 x 0 element = 3
1 x 1 element = 4
Enter the elements of the second 2 x 2 matrix:
0 x 0 element = 4
0 x 1 element = 3
1 x 0 element = 2
1 x 1 element = 1
The First Matrix:
1 2
3 4
The Second Matrix:
4 3
2 1
The Sum Matrix:
5 5
5 5

Process finished with exit code 0
```

```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/3
1 - Addition
2 - Subtraction
3 - Check Identity
Enter your choice: 2
Matrix Subtraction
Getting inputs...
Enter the elements of the first 2 x 2 matrix:
0 x 0 element = 10
0 x 1 element = 9
1 x 0 element = 8
1 x 1 element = 7
Enter the elements of the second 2 x 2 matrix:
0 x 0 element = 1
0 x 1 element = 2
1 x 0 element = 3
1 x 1 element = 4
The First Matrix:
10 9
8 7
The Second Matrix:
1 2
3 4
The Difference Matrix:
9 7
5 3

Process finished with exit code 0
```

```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/3
1 - Addition
2 - Subtraction
3 - Check Identity
Enter your choice: 3
Check for Identity Matrix
Enter the elements of the 3x3 matrix:
Element 0x0: 1
Element 0x1: 0
Element 0x2: 0
Element 1x0: 0
Element 1x1: 1
Element 1x2: 0
Element 2x0: 0
Element 2x1: 0
Element 2x2: 1
The Matrix:
1 0 0
0 1 0
0 0 1
Given matrix is a identity matrix

Process finished with exit code 0
```

6. Inheritance

Employee.java

```
package lab.two.employee;

class Employee {
    private String name;
    Employee(String name) {
        this.name = name;
    }
    protected String getName() {
        return name;
    }
}

class SalaryEmployee extends Employee{
    private int salary;
    SalaryEmployee(String name, int salary) {
        super(name);
        this.salary = salary;
    }
    protected int getSalary() {
        return salary;
    }
    public void getDetails() {
        System.out.println("The Name of the Salaried Employee: " + getName());
        System.out.println("The Salary Amount : " + getSalary());
    }
}

class Programmer extends SalaryEmployee {
    private int age;
    Programmer(String name, int salary, int age) {
        super(name, salary);
        this.age = age;
    }
    protected int getAge() {
        return age;
    }
    public void getDetails() {
        System.out.println("The Name of the Salaried Employee: " + getName());
        System.out.println("The Salary Amount : " + getSalary());
        System.out.println("The Age of the Employee : " + getAge());
    }
}
```

Main.java

```
package lab.two.employee;

public class Main {

    public static void main(String[] args) {

        SalaryEmployee salaryEmployee = new SalaryEmployee("Aravind", 45000);
        salaryEmployee.getDetails();

        Programmer programmer = new Programmer("Mugilan", 90000, 21);
        programmer.getDetails();

    }

}
```

Output:

```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/
The Name of the Salaried Employee: Aravind
The Salary Amount      : 45000
The Name of the Salaried Employee: Mugilan
The Salary Amount      : 90000
The Age of the Employee : 21

Process finished with exit code 0
```

7. Overloading

MethodOverloading.java

```
package lab.two.overloading;

public class MethodOverloading {

    public double add(double a, double b) {

        return a + b;

    }

    public int add(int a, int b) {

        return a + b;

    }

    public double add(double a, double b, double c) {

        return a + b + c;

    }

    public int add(int a, int b, int c) {

        return a + b + c;

    }

    public double sub(double a, double b) {

        return a - b;

    }

    public int sub(int a, int b) {

        return a - b;

    }

    public double sub(double a, double b, double c) {

        return a - b - c;

    }

}
```

```

        public int sub(int a, int b, int c) {
            return a - b - c;
        }
    }
}

class Overload {
    public static void main(String[] args) {
        MethodOverloading overload = new MethodOverloading();
        System.out.println(overload.add(2, 3));
        System.out.println(overload.add(2, 3, 4));
        System.out.println(overload.add(2.4, 3.2));
        System.out.println(overload.add(2.4, 3.2, 4.6));
        System.out.println(overload.sub(5, 3));
        System.out.println(overload.sub(5, 4, 3));
        System.out.println(overload.sub(4.2, 3.2));
        System.out.println(overload.sub(5.2, 4.6, 2.6));
    }
}

```

Output:

```

/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/3
5
9
5.6
10.2
2
-2
1.0
-1.9999999999999996
Process finished with exit code 0

```

8. Overriding

Bank.java

```

package lab.two.bank;
import java.util.Scanner;
public class Bank {
    private String name;
    private double interestRate = 0.0;
    public Bank(String name) {
        this.name = name;
    }
    protected void getName() {
        System.out.println(name);
    }
    public double getCompoundInterest(double loanMoney, int year) {
        double yr = (double) year / 100;
        return loanMoney * Math.pow((1 + interestRate), yr);
    }
}

```

```

}

class SBI extends Bank {
    private double interestRate = 10.0;
    SBI() {
        super("State Bank of India");
    }
    @Override
    public double getCompoundInterest(double loanMoney, int year) {
        double yr = (double) year / 100;
        return loanMoney * Math.pow((1 + interestRate), yr);
    }
}

class IOB extends Bank {
    private double interestRate = 8.5;
    IOB() {
        super("Indian Overseas Bank");
    }
    @Override
    public double getCompoundInterest(double loanMoney, int year) {
        double yr = (double) year / 100;
        return loanMoney * Math.pow((1 + interestRate), yr);
    }
}

class Axis extends Bank {
    private double interestRate = 12.75;
    Axis() {
        super("Axis Bank");
    }
    @Override
    public double getCompoundInterest(double loanMoney, int year) {
        double yr = (double) year / 100;
        return loanMoney * Math.pow((1 + interestRate), yr);
    }
}

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the Loan Money: ");
        double loanMoney = sc.nextDouble();
        System.out.print("Enter the year: ");
        int year = sc.nextInt();
        Bank defaultBank = new Bank("Default Bank");
        defaultBank.getName();
        System.out.println(defaultBank.getCompoundInterest(loanMoney, year));
    }
}

```

```

        SBI sbi = new SBI();
        sbi.getName();
        System.out.println(sbi.getCompoundInterest(loanMoney, year));
        IOB iob = new IOB();
        iob.getName();
        System.out.println(iob.getCompoundInterest(loanMoney, year));
        Axis axis = new Axis();
        axis.getName();
        System.out.println(axis.getCompoundInterest(loanMoney, year));
    }
}

```

Output:

```

/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/
Enter the Loan Money: 80000
Enter the year: 10
Default Bank
80000.0
State Bank of India
101678.52921681126
Indian Overseas Bank
100198.76012122465
Axis Bank
103972.9237913842
Process finished with exit code 0

```

Source Code:

<https://github.com/Mugilan-Codes/java-lab-exercises>