Experiment 7 Date: 19/09/2023

Roll number: 22z215

Overloading in Java

1. Method overloading by changing the number of arguments. Create two methods, the first method sum() will perform addition of two numbers and second method sum() will perform addition of three numbers by overloading concept.

Aim:

To create two methods, the first method sum() will perform addition of two numbers and second method sum() will perform addition of three numbers by overloading concept.

Code:

```
// Method overloading by changing the number of arguments:
// Create two methods, the first method sum() will perform addition of two
numbers and second method sum() will perform addition of three numbers by
overloading concept.

package Exp7;

import java.util.Scanner;

public class Q1 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        float a, b, c;
        System.out.print("Enter a num: ");
        a = input.nextFloat();
        System.out.print("Enter a num: ");
        b = input.nextFloat();
        System.out.print("Enter a num: ");
        c = input.nextFloat();

        System.out.println("Sum of a and b is " + new Dummy().add(a, b));
        System.out.println("Sum of a, b and c is " + new Dummy().add(a, b, c));
        input.close();
    }

}

class Dummy {
    public float add(float a, float b){
        return a + b;
    }
```

```
    public float add(float a, float b, float c){
        return a + b + c;
    }
}
```

Output:

```
Enter a num: 1
Enter a num: 2
Enter a num: 3
Sum of a and b is 3.0
Sum of 1, 2 and 3 is 6.0
```

---

2. Method overloading by changing data type of parameters. Create two methods having the same name but will differ in the data type of parameters. The first method sub() will receive two integer arguments and the second method sub() will receive two double arguments.

Aim:

To create two methods having the same name but will differ in the data type of parameters. The first method sub() will receive two integer arguments and the second method sub() will receive two double arguments.

Code:

```
// Method overloading by changing data type of parameters
// Create two methods having the same name but will differ in the data type of
parameters.
// The first method sub() will receive two integer arguments and the second
method sub() will receive two double arguments.

package Exp7;

import java.util.Scanner;

public class Q2 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        double a, b;
        System.out.print("Enter a num: ");
        a = input.nextDouble();
        System.out.print("Enter a num: ");
        b = input.nextDouble();

        System.out.println("Difference between a and b is " + new
Dummy1().sub((int) a, (int) b));
```

```
        System.out.println("Difference between a and b is " + new
Dummy1().sub(a, b));
        input.close();
    }
}

class Dummy1 {
    public int sub(int a, int b){
        return a - b;
    }
    public double sub(double a, double b){
        return a - b;
    }
}
```

Output:

```
Enter a num: 4
Enter a num: 2
Difference between a and b is 2
Difference between a and b is 2.0
```

---

3. Method overloading by changing data type of parameters. Create two methods having the same name but will differ in the data type of parameters. The first method sub() will receive two integer arguments and the second method sub() will receive two double arguments.

Aim:

To create two methods having the same name but will differ in the data type of parameters. The first method sub() will receive two integer arguments and the second method sub() will receive two double arguments.

Code:

```
// Method overloading by changing sequence of data type of parameters
// The method multiply() must be overloaded based on the sequence of data type
of parameters.
// The overloaded methods have to calculate multiplication of two numbers based
on the sequence of argument types while calling methods at runtime by JVM.

package Exp7;

import java.util.Scanner;

public class Q3 {
    public static void main(String[] args) {
```

```
        Scanner input = new Scanner(System.in);
        int a;
        float b;
        System.out.print("Enter an int: ");
        a = input.nextInt();
        System.out.print("Enter a float: ");
        b = input.nextFloat();

        System.out.println("a * b is " + new Dummy2().mul(a, b));
        System.out.println("b * a is " + new Dummy2().mul(b, a));
        input.close();
    }
}

class Dummy2 {
    public float mul(int a, float b){
        return a * b;
    }
    public float mul(float a, int b){
        return a * b;
    }
}
```

Output:

```
Enter an int: 1111
Enter a float: 13.28
a * b is 14754.08
b * a is 14754.08
```

Result:

All the programs are executed and the output are verified.
