

Collections

By

Dr. Tabbasum Naz

Collection

- *A collection* — sometimes called a container — is simply an object that groups multiple elements into a single unit.
- Collections are used to store, retrieve, manipulate, and communicate aggregate data.
- Typically, they represent data items that form a natural group, such as a poker hand (a collection of cards), a mail folder (a collection of letters), or a telephone directory (a mapping of names to phone numbers).

Benefits of the Java Collections Framework

- Reduces programming effort
- Increases program speed and quality
- Fosters software reuse
- Many more

```
import java.util.*;
```

```
public class CollectionDemo {
```

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

```
        List a1 = new ArrayList();
```

```
        a1.add("Zara");
```

```
        a1.add("Mahnaz");
```

```
        a1.add("Ayan");
```

```
        System.out.println(" ArrayList Elements");
```

```
        System.out.print("\t" + a1);
```

```
    }
```

```
}
```

Output

ArrayList Elements

[Zara, Mahnaz, Ayan]

```

import java.util.List;
class Student{

    static int counterSDP, counterDDP;
    double calculateFee(){
        System.out.println("fee is not defined");
        return 0.0;
    }

    public static void calculateFee(List<Student> student){
        for(Student i: student){
            if(i instanceof SingleDegreeStudent)
            {
                System.out.println("Fee of Single Degree Student is : "+i.calculateFee());
                counterSDP++;
            }
            else
            {
                System.out.println("Fee of DDP Student is : "+i.calculateFee());
                counterDDP++;
            }
        }
    }
}

```

```

static void printNumberOfStudents()
{
    System.out.println("Number of SDP Student="+counterSDP);
    System.out.println("Number of DDP Student="+counterDDP);
}
}

```

```
class SingleDegreeStudent extends Student{  
    private int noOfcreditHours;  
  
    public SingleDegreeStudent (int noOfcreditHours){  
        this.noOfcreditHours=noOfcreditHours;  
    }  
  
    double calculateFee()  
    {  
        return noOfcreditHours * 6000 + 45000;  
    }  
}
```

```
class DDPStudent extends Student{  
    private int noOfcreditHours;  
  
    public DDPStudent (int noOfcreditHours){  
        this.noOfcreditHours=noOfcreditHours;  
    }  
  
    double calculateFee()  
    {  
        return noOfcreditHours * 10000 + 90000 ;  
    }  
}
```



```
import java.util.ArrayList;
import java.util.List;

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

        List<Student> stu = new ArrayList<Student>();

        stu.add(new SingleDegreeStudent(135));
        stu.add(new SingleDegreeStudent(130));

        stu.add(new DDPStudent(135));
        stu.add(new DDPStudent(130));
        stu.add(new DDPStudent(140));

        double sumOfFees = 0.0;
        for(Student i: stu){
            sumOfFees += i.calculateFee();
        }
        System.out.println( "Sum of Fees = " + sumOfFees);

        Student.calculateFee(stu);
        Student.printNumberOfStudents();

    }
}
```

OUTPUT

Sum of Fees = 6000000.0

Fee of Single Degree Student is : 855000.0

Fee of Single Degree Student is : 825000.0

Fee of DDP Student is : 1440000.0

Fee of DDP Student is : 1390000.0

Fee of DDP Student is : 1490000.0

Number of SDP Student=2

Number of DDP Student=3

THANK YOU