Practice Applications
on
Predicate Functional Interface

Program to display names starts with 'K' by using Predicate

```java
import java.util.function.Predicate;
class Test
{
    public static void main(String[] args)
    {
        String[] names={"Sunny","Kajal","Mallika","Katrina","Kareena"};
        Predicate<String> startsWithK=s->s.charAt(0)=='K';
        System.out.println("The Names starts with K are:");
        for(String s: names)
        {
            if(startsWithK.test(s))
            {
                System.out.println(s);
            }
        }
    }
}
```

Output:

```
The Names starts with K are:
Kajal
Katrina
Kareena
```

Program for User Authentication by using Predicate

```java
import java.util.function.Predicate;
import java.util.Scanner;
class User
{
   String username;
   String pwd;
   User(String username,String pwd)
   {
      this.username=username;
      this.pwd=pwd;
   }
}
class Test
{
   public static void main(String[] args)
   {
      Predicate<User> p = u->u.username.equals("durga")&& u.pwd.equals("java");
      Scanner sc= new Scanner(System.in);
      System.out.println("Enter User Name:");
      String username=sc.next();
      System.out.println("Enter Password:");
      String pwd=sc.next();
      User user=new User(username,pwd);
      if(p.test(user))
      {
         System.out.println("Valid user and can avail all services");
      }
      else
      {
         System.out.println("invalid user you cannot avail services");
      }
   }
}
```

```
D:\durgaclasses>java Test
Enter User Name:
durga
Enter Password:
java
Valid user and can avail all services

D:\durgaclasses>java Test
Enter User Name:
ravi
Enter Password:
java
```

invalid user you cannot avail services

Employee Management Application

```
import java.util.function.Predicate;
import java.util.ArrayList;
class Employee
{
    String name;
    String designation;
    double salary;
    String city;
    Employee(String name,String designation,double salary,String city)
    {
        this.name=name;
        this.designation=designation;
        this.salary=salary;
        this.city=city;
    }
    public String toString()
    {
        String s=String.format("[%s,%s,%.2f,%s]",name,designation,salary,city);
        return s;
    }
    public boolean equals(Object obj)
    {
        Employee e=(Employee)obj;
        if(name.equals(e.name)&&designation.equals(e.designation)&&salary==e.salary && city==e.city)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}
class Test
{
    public static void main(String[] args)
    {
        ArrayList<Employee> list= new ArrayList<Employee>();
        populate(list);

        Predicate<Employee> p1=emp->emp.designation.equals("Manager");
        System.out.println("Managers Information:");
        display(p1,list);
```

```
        Predicate<Employee> p2=emp->emp.city.equals("Bangalore");
        System.out.println("Bangalore Employees Information:");
        display(p2,list);

        Predicate<Employee> p3=emp->emp.salary<20000;
        System.out.println("Employees whose slaray <20000 To Give Increment:");
        display(p3,list);

        System.out.println("All Managers from Bangalore city for Pink Slip:");
        display(p1.and(p2),list);

        System.out.println("Employees Information who are either Managers or salary <20000");
        display(p1.or(p2),list);

        System.out.println("All Employees Information who are not managers:");
        display(p1.negate(),list);

        Predicate<Employee> isCEO=Predicate.isEqual(new Employee("Durga","CEO",30000,"Hyderabad"));

        Employee e1=new Employee("Durga","CEO",30000,"Hyderabad");
        Employee e2=new Employee("Sunny","Manager",20000,"Hyderabad");
        System.out.println(isCEO.test(e1));//true
        System.out.println(isCEO.test(e2));//false

    }
    public static void populate(ArrayList<Employee> list)
    {
        list.add(new Employee("Durga","CEO",30000,"Hyderabad"));
        list.add(new Employee("Sunny","Manager",20000,"Hyderabad"));
        list.add(new Employee("Mallika","Manager",20000,"Bangalore"));
        list.add(new Employee("Kareena","Lead",15000,"Hyderabad"));
        list.add(new Employee("Katrina","Lead",15000,"Bangalore"));
        list.add(new Employee("Anushka","Developer",10000,"Hyderabad"));
        list.add(new Employee("Kanushka","Developer",10000,"Hyderabad"));
        list.add(new Employee("Sowmya","Developer",10000,"Bangalore"));
        list.add(new Employee("Ramya","Developer",10000,"Bangalore"));
    }
    public static void display(Predicate<Employee> p,ArrayList<Employee> list)
    {
        for (Employee e: list )
        {
            if(p.test(e))
            {
                System.out.println(e);
            }
        }
```

```
        System.out.println("**************************************************");
    }
}
```

Output:

```
Managers Information:
[Sunny,Manager,20000.00,Hyderabad]
[Mallika,Manager,20000.00,Bangalore]
**************************************************
Bangalore Employees Information:
[Mallika,Manager,20000.00,Bangalore]
[Katrina,Lead,15000.00,Bangalore]
[Sowmya,Developer,10000.00,Bangalore]
[Ramya,Developer,10000.00,Bangalore]
**************************************************
Employees whose slaray <20000 To Give Increment:
[Kareena,Lead,15000.00,Hyderabad]
[Katrina,Lead,15000.00,Bangalore]
[Anushka,Developer,10000.00,Hyderabad]
[Kanushka,Developer,10000.00,Hyderabad]
[Sowmya,Developer,10000.00,Bangalore]
[Ramya,Developer,10000.00,Bangalore]
**************************************************
All Managers from Bangalore city for Pink Slip:
[Mallika,Manager,20000.00,Bangalore]
**************************************************
Employees Information who are either Managers or salary <20000
[Sunny,Manager,20000.00,Hyderabad]
[Mallika,Manager,20000.00,Bangalore]
[Katrina,Lead,15000.00,Bangalore]
[Sowmya,Developer,10000.00,Bangalore]
[Ramya,Developer,10000.00,Bangalore]
**************************************************
All Employees Information who are not managers:
[Durga,CEO,30000.00,Hyderabad]
[Kareena,Lead,15000.00,Hyderabad]
[Katrina,Lead,15000.00,Bangalore]
[Anushka,Developer,10000.00,Hyderabad]
[Kanushka,Developer,10000.00,Hyderabad]
[Sowmya,Developer,10000.00,Bangalore]
[Ramya,Developer,10000.00,Bangalore]
**************************************************
true
false
```