Factorial_Excep.java | Interfaces_Q.java | Account_Excep.java

Interfaces_Q.java > QMain > main(String[])

```java
/*Implement Interfaces - QUEUE OPERATIONS*/
import java.util.*;
interface Q
{
    void insert_rear(int item);
    int delete_front();
    void display();
}
class Queue implements Q
{
    private int q[];
    private int rear;
    private int front;
    Queue(int size)
    {
       q = new int[size];
       rear = -1;
       front = 0;
    }
    public void insert_rear(int item)
    {
        if(rear==q.length-1)
        System.out.println("Queue Overflow ");
        else
        q[++rear] = item;
    }
    public int delete_front()
    {
        if(front>rear)
        {
            System.out.println("Queue Underflow.");
            front = 0;
            rear = -1;
            return -1;
        }
        return q[front++];
    }
    public void display()
    {
        System.out.println("contents of queue :");
```

Factorial_Excep.java | Interfaces_Q.java | Account_Excep.java

Interfaces_Q.java > QMain > main(String[])

```
39      {
40          System.out.println("contents of queue :");
41          for(int i=front;i<=rear;i++)
42          System.out.print(q[i]+"  ");
43          System.out.println();
44      }
45  }
46
47  class QMain
48  {
        Run | Debug
49      public static void main(String args[])
50      {
51          Queue obj = new Queue(10);
52          int n,item;
53          Scanner sc=new Scanner(System.in);
54          while(true)
55          {
56              System.out.println("1.Insert into queue\n2.Delete from queue\n3.Display\n4.Exit");
57              n=sc.nextInt();
58              switch(n)
59              {
60                  case 1:System.out.println("enter item ");
61                  item=sc.nextInt();
62                  obj.insert_rear(item);
63                  break;
64                  case 2:item=obj.delete_front();
65                  if(item==-1)
66                  System.out.println("queue is empty");
67                  else
68                  System.out.println("deleted item : "+item);
69                  break;
70                  case 3:obj.display();
71                  break;
72                  default:System.exit(0);
73              }
74          }
75      }
76  }
```

```
C:\Users\akki\Desktop\java files>javac Interfaces_Q.java

C:\Users\akki\Desktop\java files>java QMain
1.Insert into queue
2.Delete from queue
3.Display
4.Exit
1
enter item
12
1.Insert into queue
2.Delete from queue
3.Display
4.Exit
1
enter item
34
1.Insert into queue
2.Delete from queue
3.Display
4.Exit
1
enter item
56
1.Insert into queue
2.Delete from queue
3.Display
4.Exit
3
contents of queue :
12  34  56
1.Insert into queue
2.Delete from queue
3.Display
4.Exit
2
deleted item : 12
1.Insert into queue
2.Delete from queue
3.Display
4.Exit
2
deleted item : 34
1.Insert into queue
2.Delete from queue
3.Display
```

OUTPUT DEBUG CONSOLE TERMINAL 1: cmd

```
1.Insert into queue
2.Delete from queue
3.Display
4.Exit
3
contents of queue :
12  34  56
1.Insert into queue
2.Delete from queue
3.Display
4.Exit
2
deleted item : 12
1.Insert into queue
2.Delete from queue
3.Display
4.Exit
2
deleted item : 34
1.Insert into queue
2.Delete from queue
3.Display
4.Exit
4

C:\Users\akki\Desktop\java files>
```

Factorial_Excep.java ● | Interfaces_Q.java | Account_Excep.java

Factorial_Excep.java > Factorial_Excep > Computefact(int)

```java
/*Write a Java program to compute the factorial of a number. The input value must be tested
for validity. If it is greater than 15, the method ComputeFactorial( ) should raise an
Userdefined Exception MyException with appropriate messages.*/

import java.util.*;

class MyException extends Exception
{
    int  num;
    MyException(int n)
    {
        num=n;
    }
    public String toString()
    {
        return "The input number cannot be greater than 15";
    }
}
public class Factorial_Excep
{
    int Computefact(int n) throws MyException
    {
        if(n>15)
        {
            throw new MyException(n);
        }
        else if(n==0)
        {
            return 1;
        }
        else
        {
            return n*Computefact(n-1);
        }
    }
    Run | Debug
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int n,fact;
```

Factorial_Excep.java ● | Interfaces_Q.java | Account_Excep.java

Factorial_Excep.java > Factorial_Excep > Computefact(int)

```
        else if(n==0)
        {
            return 1;
        }
        else
        {
            return n*Computefact(n-1);
        }
    }
    Run | Debug
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int n,fact;
        for(int i=0;i<2;i++)
        {
            System.out.println("Enter the number:");
            n=sc.nextInt();
            Factorial_Excep f=new Factorial_Excep();
            try
            {
                fact=f.Computefact(n);
                System.out.println("The factorial of "+n+" is "+fact);
            }
            catch(MyException e)
            {
                System.out.println("Caught Exception:"+e);
            }

        }
    }
}
```

```
C:\Users\akki\Desktop\java files>javac Factorial_Excep.java

C:\Users\akki\Desktop\java files>java Factorial_Excep
Enter the number:
5
The factorial of 5 is 120
Enter the number:
16
Caught Exception:The input number cannot be greater than 15

C:\Users\akki\Desktop\java files>
```

Factorial_Excep.java ● | Interfaces_Q.java | Account_Excep.java ×

Account_Excep.java > AccMain

```java
/*Write a Java program to create an account class. Define appropriate constructor for this
class. Implement a separate methods to display account balance and withdraw money.
Raise a user defined exception if there is an attempt to withdraw money which is greater
than the account balance. Make necessary assumptions required.*/

import java.util.*;

class MyException extends Exception
{
   double amount;
   MyException(double a)
   {
     amount = a;
   }
  public String toString()
   {
      return "Insufficient balance in your account\nYour account balance="+amount;
   }
}

class Account
{
    Scanner sc=new Scanner(System.in);
    double balance;
    int amt;
    Account(double bal)
    {
        balance=bal;
    }
    double withdraw() throws MyException
    {
        System.out.println("Enter the amount to withdraw");
        amt=sc.nextInt();
        if(balance>=amt)
        {
            balance=balance-amt;
            return balance;
        }
        else
        throw new MyException(balance);
```

Factorial_Excep.java ● Interfaces_Q.java Account_Excep.java ×

Account_Excep.java > AccMain

```
38          }
39          else
40          throw new MyException(balance);
41      }
42
43      void display()
44      {
45          System.out.println("Account Balance="+balance);
46      }
47  }
48
49  class AccMain
50  {
        Run | Debug
51      public static void main(String args[])
52      {
53          Scanner sc=new Scanner(System.in);
54          System.out.println("Enter the initial balance");
55          double b=sc.nextDouble();
56          Account obj= new Account(b);
57          while(true)
58          {
59              System.out.println("1.Withdraw\n2.Display Balance\n3.Exit");
60              System.out.println("Enter the choice");
61              int n=sc.nextInt();
62              switch(n)
63              {
64                  case 1:
65                  try
66                  {
67                      obj.withdraw();
68                  }
69                  catch(MyException e)
70                  {
71                      System.out.println(e);
72                  }
73                  break;
74                  case 2:
75                  obj.display();
76                  break;
```

Factorial_Excep.java ● | Interfaces_Q.java | Account_Excep.java ×

Account_Excep.java > AccMain

```
        double b=sc.nextDouble();
        Account obj= new Account(b);
        while(true)
        {
            System.out.println("1.Withdraw\n2.Display Balance\n3.Exit");
            System.out.println("Enter the choice");
            int n=sc.nextInt();
            switch(n)
            {
                case 1:
                try
                {
                    obj.withdraw();
                }
                catch(MyException e)
                {
                    System.out.println(e);
                }
                break;
                case 2:
                obj.display();
                break;
                default:
                System.exit(0);
            }
        }
    }
}
```

OUTPUT DEBUG CONSOLE TERMINAL 1: cmd

```
C:\Users\akki\Desktop\java files>javac Account_Excep.java

C:\Users\akki\Desktop\java files>java AccMain
Enter the initial balance
10000
1.Withdraw
2.Display Balance
3.Exit
Enter the choice
1
Enter the amount to withdraw
1000
1.Withdraw
2.Display Balance
3.Exit
Enter the choice
2
Account Balance=9000.0
1.Withdraw
2.Display Balance
3.Exit
Enter the choice
1
Enter the amount to withdraw
10000
Insufficient balance in your account
Your account balance=9000.0
1.Withdraw
2.Display Balance
3.Exit
Enter the choice
3

C:\Users\akki\Desktop\java files>
```