*Dt : 22/10/2022*

*Assignment-1:(Solution)*

*Convert IArithmetic application into "Anonymous InnerClass as implementation class" model*

*Note:*

*Addition,Subtraction,Multiplication,Division,ModDivision class as Anonymous*

*IArithmetic.java*

```
package test;
public interface IArithmetic {
  public abstract double calculate(int x,int y);
}
```

*DemoAnonymous4.java(MainClass)*

```
package maccess;

import java.util.*;

import test.*;

public class DemoAnonymous4 {

        public static void main(String[] args) {

    Scanner s = new Scanner(System.in);

    System.out.println("Enter the value of x:");

    int x = s.nextInt();

    System.out.println("Enter the value of y:");

    int y = s.nextInt();

    System.out.println("====Choice====");

    System.out.println("1.add\n2.sub\n3.mul\n4.div\n5.modDiv");
```

```
System.out.println("Enter the Choice:");
switch(s.nextInt())
{
case 1:
    //Addition class as Anonymous
    IArithmetic ad = new IArithmetic()
    {
            public double calculate(int x,int y)
            {
                    return x+y;
            }
    };
    System.out.println("Sum="+ad.calculate(x, y));
    break;
case 2:
    //Subtraction class as Anonymous
    IArithmetic sb = new IArithmetic()
    {
            public double calculate(int x,int y)
            {
                    return x-y;
            }
    };
    System.out.println("Sub="+sb.calculate(x, y));
```

```
    break;
case 3:
    //Multiplication class without name
    IArithmetic ml = new IArithmetic()
    {
            public double calculate(int x,int y)
            {
                    return x*y;
            }
    };
    System.out.println("Mul="+ml.calculate(x, y));
    break;
case 4:
    //Division class as Anonymous
    IArithmetic dv = new IArithmetic()
    {
            public double calculate(int x,int y)
            {
                    return (float)x/y;
            }
    };
    System.out.println("Div="+dv.calculate(x, y));
    break;
case 5:
```

```
        //ModDivision class as Anonymous
        IArithmetic md = new IArithmetic()
        {
                public double calculate(int x,int y)
                {
                        return x%y;
                }
        };
        System.out.println("ModDiv="+md.calculate(x, y));
        break;
    default:
        System.out.println("Invalid Choice...");
    }//end of switch
    s.close();
        }
}
```

---

*ClassFiles:*

*IArithmetic.class*

*DemoAnonymous4.class*

*DemoAnonymous4$1.class*

*DemoAnonymous4$2.class*

*DemoAnonymous4$3.class*

*DemoAnonymous4$4.class*

*DemoAnonymous4$5.class*

==========

*Assignment-2:*

*Convert BankTransaction application into "Anonymous InnerClass as implementation class" model*

*Note:*

*WithDraw and Deposit classes as Anonymous*

*Balance.java*

```
package test;
public class Balance {
   public double bal=2000;
   public double getBalance() {
        return bal;
   }
}
```

*Transaction.java*

```
package test;
public interface Transaction {
    public static final Balance b = new Balance();
    public abstract void process(int amt);
}
```

*CheckPinNo.java*

```
package test;
public class CheckPinNo {
     public boolean verify(int pinNo) {
      return switch(pinNo) {
      case 1111:yield true;
      case 2222:yield true;
      case 3333:yield true;
      default:yield false;
      };
```

```
        }
}
```

*BankMainClass.java(MainClass)*

```
package maccess;
import test.*;
import java.util.*;
public class BankMainClass {
        public static void main(String[] args) {
    Scanner s = new Scanner(System.in);
    int count=0;
    pqr:
    while(true) {
        System.out.println("Enter the pinNo:");
        int pinNo = s.nextInt();
        CheckPinNo cpn = new CheckPinNo();
        boolean k = cpn.verify(pinNo);
        if(k)
        {
                System.out.println("====Choice====");
                System.out.println("1.WithDraw\n2.Deposit");
                System.out.println("Enter the Choice:");
                switch(s.nextInt())
                {
                case 1:
```

```
System.out.println("Enter the amt:");
int a1 = s.nextInt();
if(a1>0 && a1%100==0)
{
        //WithDraw class as Anonymous
        Transaction wd = new Transaction()
        {
                //WithDraw Logic
          public void process(int amt) {
                if(amt<b.bal)
                {
                        System.out.println("Amt WithDrawn:"+amt);
                        b.bal=b.bal-amt;
                        System.out.println("Balance
amt:"+b.getBalance());
                        System.out.println("Transaction Completed...");
                }//end of if
                else
                {
                        System.out.println("InSufficient fund...");
                }
          }
        };
        wd.process(a1);
```

```
        }//end of if
        else
        {
                System.out.println("Invlid amt...");
        }
        break pqr;//stop the loop
case 2:
        System.out.println("Enter the amt:");
        int a2 = s.nextInt();
        if(a2>0 && a2%100==0)
        {
                //Deposit as Anonymous
                Transaction dp = new Transaction()
                {
                        //Deposit Logic
                  public void process(int amt) {
                                System.out.println("Amt deposited:"+amt);
                                b.bal=b.bal+amt;
                                System.out.println("Balance amt:"+b.getBalance());
                                System.out.println("Transaction Completed...");
                    }
                };
                dp.process(a2);
        }//end of if
```

```
                    else
                    {
                            System.out.println("Invlid amt...");
                    }
                    break pqr;//stop the loop
            default:
                    System.out.println("Invalid Choice...");
            break pqr;//stop the loop
            }//end of switch
        }//end of if
        else
        {
            System.out.println("Invalid pinNo....");
            count++;
        }

        if(count==3)
        {
            System.out.println("Transaction blocked...");
            break;//stop the loop
        }
    }//end of loop
        }
}
```

***----------***

***Balance.class***

***Transaction.class***

***CheckPinNo.class***

***BankMainClass.class(MainClass)***

***BankMainClass$1.class***

***BankMainClass$2.class***

***==========***

***Note:***

***=>"Anonymous InnerClass as Implementation class" model is modified as "LambdaExpression" in Java8 version.***

***==========***

****Imp***

***LambdaExpressions in Java:(Java8 - new feature)***

***=>The process of declaring method without method_name is known as "LambdaExpression" and which is also known as "Anonymous method".***

***structure of LambdaExpression:***

```
(para_list)->
{
 //method_body
}
```

***Note:***

***=>The abstract method of interface is attached with the LambdaExpression,***

***and the LambdaExpression is called for execution using abstract_method_name.***

***syntax:***

```
interface ITest
{
  public abstract void m1(int x);
}

ITest ob = (int x)->
        {
         //method_body
        };
```

***Ex-Program:***

***ITest.java***

```
package test;
public interface ITest {
   public abstract void m1(int x);
   public default void m2(int y) {
        System.out.println("===default m2(y)====");
        System.out.println("The value y:"+y);
   }
}
```

***LambdaExpression1.java(MainClass)***

```
package maccess;
import test.*;
```

```
public class LambdaExpression1 {
      public static void main(String[] args) {
        ITest ob = (int x)->
        {
       System.out.println("====LambdaExpresion (x)====");
       System.out.println("The value x:"+x);
        };
        ob.m1(12);//LambdaExpression call
        ob.m2(13);//default method call
      }
}
```

*o/p:*

*====LambdaExpresion (x)====*

*The value x:12*

*===default m2(y)====*

*The value y:13*

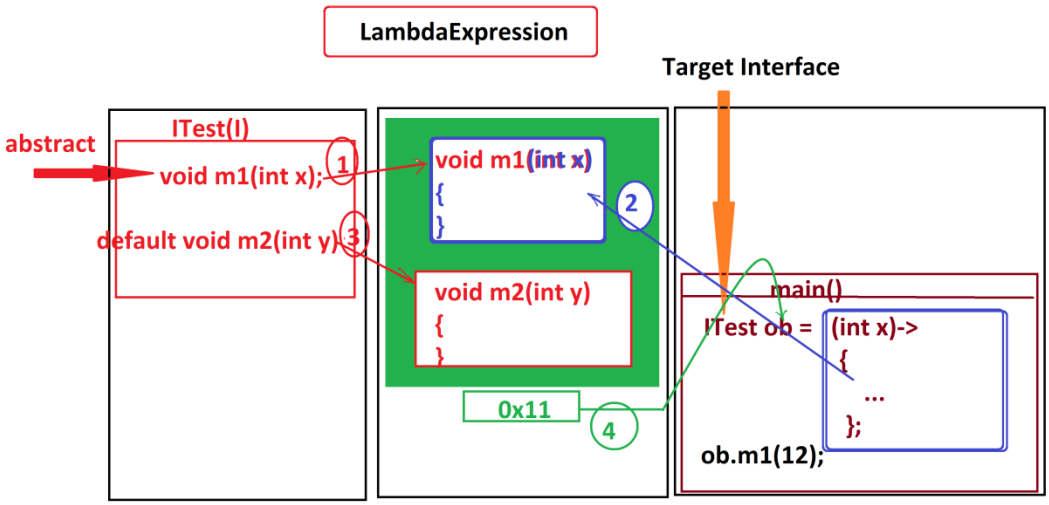
*---*

*Execution flow of above program:*

*ClassFiles:*

*ITest.class*

*LambdaExpression1.class(MainClass)*




==========

***Ex:***

***Convert IComparable Application into LambdaExpressions.***

***IComparable.java***

***LambdaExpression2.java(MainClass)***

==========

***Assignment-1:***

***Convert IArithmetic application into LambdaExpressions.***

***Assignment-2:***

***Convert BankTransaction application into LambdaExpression.***

==========