PL ASSIGNMENT 12

Source Code:

Client.java

```
import java.util.Scanner;

class Client {

    public static void main(String[] args) {
        String a;

        Context context = new Context();
        context.setState(new ConcreteState1());

        Scanner sc = new Scanner(System.in);

        System.out.println("Current state is " +
context.getState());
        char c = 'a';
        while (c != 'a' || c != 'b') {
            System.out.println("----------------------------------
----");
            System.out.println("Enter a or b");
            a = sc.next();

            // System.out.println("\nInput is " + a);
            if (a.charAt(0) == 'a') {
                context.onA();
                c = 'a';
            } else if (a.charAt(0) == 'b') {
                context.onB();
                c = 'b';
            } else {

                System.out.println("\nInvalid String");
                break;
```

```
            }
        }
        sc.close();
    }
}
```

Context.java

```
public class Context {
    AbstractState state;

    // Concrete concrete;

    public void setState(AbstractState s) {
        this.state = s;
        this.state.context = this;
    }

    public void onA() {
        System.out.println("Context -> On A called\n");

        this.state.onA();

    }

    public void onB() {
        System.out.println("Context-> On B called\n");

        this.state.onB();

    }

    public int getState() {
        return this.state.state;
    }

}
```

AbstractState.java

```java
abstract class AbstractState {

    Context context;
    int state;

    // public AbstractState(Context c) {
    // this.context = c;
    // }

    public Context getcontext() {
        return this.context;
    }

    public void setContext(Context c) {
        this.context = c;
    }

    public abstract void onA();

    public abstract void onB();

}
```

ConcreteState1.java

```java
public class ConcreteState1 extends AbstractState {
    public ConcreteState1() {
        super();
        this.state = 1;
    }

    @Override
    public void onA() {
        this.context.setState(new ConcreteState2());
        System.out.println("State Changed to 2");
    }
```

```
    @Override
    public void onB() {
        this.context.setState(new ConcreteState3());
        System.out.println("State Changed to 3");
    }
}
```

ConcreteState2.java

```
public class ConcreteState2 extends AbstractState {

    public ConcreteState2() {
        super();
        this.state = 2;
    }

    @Override
    public void onA() {

        this.context.setState(new ConcreteState1());
        System.out.println("State Changed to 1");
    }

    @Override
    public void onB() {

        this.context.setState(new ConcreteState4());
        System.out.println("State Changed to 4");
    }

}
```

ConcreteState3.java

```
public class ConcreteState3 extends AbstractState {

    public ConcreteState3() {
```

```
        super();
        this.state = 3;
    }

    @Override
    public void onA() {

        this.context.setState(new ConcreteState4());
        System.out.println("State Changed to 4");
    }

    @Override
    public void onB() {

        this.context.setState(new ConcreteState1());
        System.out.println("State Changed to 1");
    }

}
```

ConcreteState4.java

```
public class ConcreteState4 extends AbstractState {

    public ConcreteState4() {
        super();
        this.state = 4;
    }

    @Override
    public void onA() {

        this.context.setState(new ConcreteState3());
        System.out.println("State Changed to 3");
    }

    @Override
```

```
    public void onB() {

        this.context.setState(new ConcreteState2());
        System.out.println("State Changed to 2");

    }

}
```

OUTPUT:

```
(base) darshanuttammistry@dhcp-39-27-101 PL Assignment 12 % javac Client.java
(base) darshanuttammistry@dhcp-39-27-101 PL Assignment 12 % java Client
Current state is 1
---------------------------------------
Enter a or b
b
Context-> On B called

State Changed to 3
---------------------------------------
Enter a or b
b
Context-> On B called

State Changed to 1
---------------------------------------
Enter a or b
a
Context -> On A called

State Changed to 2
---------------------------------------
Enter a or b
b
Context-> On B called

State Changed to 4
---------------------------------------
Enter a or b
a
Context -> On A called

State Changed to 3
---------------------------------------
Enter a or b
0

Invalid String
```