```
//Keyur Ved, Michael Osorio, Gregory Goldshteyn, Matthew Stypulkoski
#include <typeinfo>
using namespace std;
public class DumbPhoneContext
{
  private State state;

  public DumbPhone()
  {
    state = new StateStandby();
  }

  public DumbPhone(State s)
  {
    state = s;
  }

  public void handleButton(int button)
  {
    state.handleButton(this, button);
  }

  public void handleSlider(int direction)
  {
    state.handleSlider(this, direction);
  }
  public void handleTouch()
  {
    //If state is of type Standby, open an application and go to
application mode
  }
}

public abstract class State
{
  public:
    static const int SEND_BUTTON = 0;
    static const int END_BUTTON = 1;
    static const int UP = 0;
    static const int DOWN = 1;
    abstract State handleButton(StateContext context, int button);
    abstract State handleSlider(StateContext context, int direction);
    abstract State handleTouch(StateContext context);
}

public class StateStandby extends State
{
  public StateStandby()
  {
    // Construct a new StateStandby object
  }
  public State handleButton(StateContext context, int button)
  {
     if( button == SEND_BUTTON ){
           //Switch to call state
     }
     return;
  }
  public State handleSlider(StateContext context, int direction)
  {
     if(direction == UP){
           //Increase ringer
     }
     if(direction == DOWN){
           //Decrease ringer
     }
     else{
           //DO nothing
     }
     return context;
  }
  public State handleTouch(StateContext context)
  {
     //If state is of type Standby, open an application and go to
application mode
  }
}

public class StateApp extends State
{
     public StateApp()
     {
       // Construct a new StateApp object
     }
     public State handleButton(StateContext context, int button)
     {
           if(button == END){
               //State switch to standby
           }
           return context;
     }
     public void handleSlider(StateContext context, int direction)
     {
           if (direction == UP) {
           //If Slider moved up, scroll display up
           }
           if (direction == DOWN) {
           //If Slider moved down, scroll display down
           }
     }
}
public class StateCall extends State
{
     public StateCall()
     {
       //Construct a new StateCall Object
     }
     public void handleButton(StateContext context, int button)
     {
           if (button == SND_Button){
               //Do Nothing
           }
           if(button == END_Button) {
               //Switch state to StandBy
       //If SND pressed then do nothing
       //If END pressed then switch state to StandBy
     }
     public void handleSlider(StateContext context, int direction)
     }
           if (direction == UP) {
               //Volume increases
           }
           if (direction == DOWN) {
               //Volume decreases
           }
       //If Slider moved up then increase volume
       //If Slider moved down then decrease volume
     }
```