```
import java.util.*;

public class Bully {
   int coordinator;
   int max_processes;
   boolean processes[];

   public Bully(int max) {
      max_processes = max;
      processes = new boolean[max_processes];
      coordinator = max;

      System.out.println("Creating processes..");
      for(int i = 0; i < max; i++) {
         processes[i] = true;
         System.out.println("P"+ (i+1) + " created");
      }
      System.out.println("Process P" + coordinator + " is the coordinator");

   }
   void displayProcesses() {
      for(int i = 0; i < max_processes; i++) {
         if(processes[i]) {
            System.out.println("P" + (i+1) + " is up");
         } else {
            System.out.println("P" + (i+1) + " is down");
         }
      }
      System.out.println("Process P" + coordinator + " is the coordinator");
   }
```

```
void upProcess(int process_id) {
    if(!processes[process_id - 1]) {
        processes[process_id - 1] = true;
        System.out.println("Process " + process_id + " is now up.");
    } else {
        System.out.println("Process " + process_id + " is already up.");
    }
}

void downProcess(int process_id) {
    if(!processes[process_id - 1]) {
        System.out.println("Process " + process_id + " is already down.");
    } else {
        processes[process_id - 1] = false;
        System.out.println("Process " + process_id + " is now down.");
    }
}

void runElection(int process_id) {
    coordinator = process_id;
    boolean keepGoing = true;

    for(int i = process_id; i < max_processes && keepGoing; i++) {
        System.out.println("Election message sent from process " + process_id + " to process " + (i+1));

        if(processes[i]) {
            keepGoing = false;
            runElection(i + 1);
        }
    }
}
```

```
public static void main(String args[]) {
    Bully bully = null;
    int max_processes = 0, process_id = 0;
    int choice = 0;
    Scanner sc = new Scanner(System.in);

    while(true) {
        System.out.println("Bully Algorithm");
        System.out.println("1. Create processes");
        System.out.println("2. Display processes");
        System.out.println("3. Up a process");
        System.out.println("4. Down a process");
        System.out.println("5. Run election algorithm");
        System.out.println("6. Exit Program");
        System.out.print("Enter your choice:- ");
        choice = sc.nextInt();

        switch(choice) {
            case 1:
                System.out.print("Enter the number of processes:- ");
                max_processes = sc.nextInt();
                bully = new Bully(max_processes);
                break;
            case 2:
                bully.displayProcesses();
                break;
            case 3:
                System.out.print("Enter the process number to up:- ");
                process_id = sc.nextInt();
                bully.upProcess(process_id);
```

```
                break;
            case 4:
                System.out.print("Enter the process number to down:- ");
                process_id = sc.nextInt();
                bully.downProcess(process_id);
                break;
            case 5:
                System.out.print("Enter the process number which will perform election:- ");
                process_id = sc.nextInt();
                bully.runElection(process_id);
                bully.displayProcesses();
                break;
            case 6:
                System.exit(0);
                break;
            default:
                System.out.println("Error in choice. Please try again.");
                break;
        }
      }
   }
}
```