```java
package use_case_controller;

import java.util.ArrayList;

/**
 * Class SearchCasesController to manage functionality of searching Cases.
 * @author Daria Vekic (Student ID: 586661)
 *
 */
public class SearchCasesController {

    /**
     * Method to count the number of Criminal Cases in the List.
     * @param cases the List to be searched through.
     * @return crimCount the number of Criminal Cases in the List.
     */
    public String countCrimCases(ArrayList<Case> cases) {
        int crimCount = 0;
        for(int i = 0; i<cases.size(); i++) {
            if(cases.get(i).getType().toString().equals("CRI")) {
                crimCount++;
            } //endif
        } //endfor
        return Integer.toString(crimCount);
    } //end method countCrimCases

    /**
     * Method to count the number of Immigration Cases in the List.
     * @param cases the List to be searched through.
     * @return immCount the number of Immigration Cases in the List.
     */
    public String countImmCases(ArrayList<Case> cases) {
        int immCount = 0;
        for(int i = 0; i<cases.size(); i++) {
            if(cases.get(i).getType().toString().equals("IMM")) {
                immCount++;
            } //endif
        } //endfor
        return Integer.toString(immCount);
    } //end method countImmCases

    /**
     * Method to count the number of Personal Injury Cases in the List.
     * @param cases the List to be searched through.
     * @return pInjCount the number of Personal Injury Cases in the List.
     */
    public String countPICases(ArrayList<Case> cases) {
        int pInjCount = 0;
        for(int i = 0; i<cases.size(); i++) {
            if(cases.get(i).getType().toString().equals("PIN")) {
                pInjCount++;
            } //endif
        } //endfor
        return Integer.toString(pInjCount);
    } //end method countPICases

    /**
     * Method to retrieve a single Case using a given case number.
     * @param clients the List of Clients through which a Case is retrieved.
     * @param caseNum the reference number of Case to be retrieved.
     * @return the Client to which this Case is attached; null otherwise.
     */
    public Client findCase(ArrayList<Client> clients, String caseNum) {
        Iterator<Client> iterator = clients.iterator();
        while(iterator.hasNext()) {
            Client client = iterator.next(); //store current iterator value in client
            if(client.getCASE().getCASE_REF_NUM().equals(caseNum)) //if the Case Ref Num is found
                return client; //return the iterator value
            } //end while
        return null;
    } //end method findCase

    /**
     * Method to output data of a single Case.
     * @param clients the List of Clients through which a Case is retrieved.
     * @param caseNum the reference number of Case to be displayed.
     * @param caseNumLbl the Label to display Case number.
     * @param titleLbl the Label to display Case title.
     * @param typeLbl the Label to display Case type.
     * @param solLbl the Label to display Employee name.
     * @param descLbl the Label to display Case description.
     * @param nameLbl the Label to display Client name.
     * @param phoneLbl the Label to display Client phone number.
     * @param addressLbl the Label to display Client address.
     */
    public void listCase(ArrayList<Client> clients, String caseNum, JLabel caseNumLbl,
            JLabel titleLbl, JLabel typeLbl, JLabel solLbl, JLabel descLbl,
            JLabel nameLbl, JLabel phoneLbl, JLabel addressLbl) {
        //find the data
        Client client = findCase(clients, caseNum);
        //update labels to output info
        caseNumLbl.setText(client.getCASE().getCASE_REF_NUM());
        titleLbl.setText(client.getCASE().getTitle());
        typeLbl.setText(client.getCASE().getType().toString());
        solLbl.setText(client.getCASE().getSOLICITOR().getFullName());
        descLbl.setText(client.getCASE().getDescription());
        nameLbl.setText(client.getFullName());
        phoneLbl.setText(client.getPhoneNum());
        addressLbl.setText(client.getAddressLineOne() + ", " + client.getAddressLineTwo() + ", " + client.getPostcode());
    } //end method displayCase

    /**
     * Method that retrieves and outputs various Case data for every Case in the List.
     * Outputs to text area for testing purposes.
     * @param tArea used to output data to a given text area.
     * @param clients the List to retrieve data from.
     */
    public void viewAllCases(JTextArea tArea, ArrayList<Client> clients) {
        //System.out.println("Caseload");
        try {
            for(int i=0; i<clients.size(); i++) {
                tArea.append("\n\nCase Reference Number : " + clients.get(i).getCASE().getCASE_REF_NUM());
                tArea.append("\nCase Type : " + clients.get(i).getCASE().getType());
                tArea.append("\nSolicitor Name : " + clients.get(i).getCASE().getSOLICITOR().getFullName());
                tArea.append("\nClient Name : " + clients.get(i).getFullName());
                tArea.append("\nClient Phone No. : " + clients.get(i).getPhoneNum());
                tArea.append("\nClient Postcode : " + clients.get(i).getPostcode());
                tArea.append("\nCase Description : " + clients.get(i).getCASE().getDescription().replaceAll("\n", " "));
            } //endfor
        } //end try
        catch (IndexOutOfBoundsException i) {
            JOptionPane.showMessageDialog(null, "Sorry, no Cases to display.", "Caseload Empty", JOptionPane.INFORMATION_MESSAGE);
        } //end catch
    } //end method viewAllCases
} //end class SearchCasesController
```