



class CourtCase {

private String caseNumber;

private String defendantName;

private String address;

private double stolenAmount;

private boolean isOpen;

public CourtCase(String caseNumber, String defendantName, String address, double stolenAmount) {

this.caseNumber = caseNumber;

this.defendantName = defendantName;

this.address = address;

this.stolenAmount = stolenAmount;

this.isOpen = true;

}

public String getCaseNumber() {

return caseNumber;

}

public String getDefendantName() {

return defendantName;

}

public String getAddress() {

return address;

}

public double getStolenAmount() {

return stolenAmount;

}

public boolean isOpen() {

return isOpen;

}

public void closeCase() {

isOpen = false;

}

@Override

public String toString() {

return "Case Number: " + caseNumber +

"\nDefendant Name: " + defendantName +

"\nAddress: " + address +

"\nStolen Amount: $" + stolenAmount +

"\nIs Open: " + (isOpen ? "Yes" : "No");

}

}

// Class representing the application and its main functionality

class OpenCourtCasePrototype {

private CourtCase[] courtCases;

private int caseCount;

public OpenCourtCasePrototype() {

courtCases = new CourtCase[10]; // assuming a maximum of 10 cases for prototype

caseCount = 0;

}

public void addCourtCase(CourtCase courtCase) {

if (caseCount < courtCases.length) {

courtCases[caseCount] = courtCase;

caseCount++;

System.out.println("Court case added successfully.");

} else {

System.out.println("Cannot add court case. Maximum limit reached.");

}

}

public void closeCourtCase(String caseNumber) {

for (CourtCase courtCase : courtCases) {

if (courtCase != null && courtCase.getCaseNumber().equals(caseNumber)) {

courtCase.closeCase();

System.out.println("Court case closed successfully.");

return;

}

}

System.out.println("Court case not found.");

}

public void displayOpenCourtCases() {

boolean foundOpenCases = false;

System.out.println("Open Court Cases:");

for (CourtCase courtCase : courtCases) {

if (courtCase != null && courtCase.isOpen()) {

System.out.println(courtCase);

foundOpenCases = true;

}

}

if (!foundOpenCases) {

System.out.println("No open court cases found.");

}

}

}