```java
import java.util.Scanner;

abstract class Robber {
    abstract int RowHouses(int[] a);

    abstract int RoundHouses(int[] b);

    abstract int SquareHouses(int[] c);

    abstract int MultiHouseBuilding(int[][] d);

    void RobbingClass() {
        System.out.println("MScAI&ML");
    }

    void MachineLearning() {
        System.out.println("I love MachineLearning.");
    }
}

class JAVAProfessionalRobber extends Robber {
    @Override
    int RowHouses(int[] a) {
        int len = a.length;
        if (len == 0) {
            return 0;
        }
        if (len == 1) {
            return a[0];
        }
        int[] d = new int[len];
        d[0] = a[0];
        d[1] = Math.max(a[0], a[1]);
        for (int i = 2; i < len; i++) {
```

```
            d[i] = Math.max(d[i - 1], d[i - 2] +
a[i]);
        }
        return d[len - 1];
    }

    @Override
    int RoundHouses(int[] b) {
        int len = b.length;
        if (len == 0) {
            return 0;
        }
        if (len == 1) {
            return b[0];
        }
        int robFirst = helper(b, 0, len - 2);
        int robSecond = helper(b, 1, len - 1);
        return Math.max(robFirst, robSecond);
    }

    private int helper(int[] b, int start, int end) {
        int len = end - start + 1;
        if (len == 0) {
            return b[start];
        }
        int[] d = new int[len];
        d[0] = b[start];
        d[1] = Math.max(b[start], b[start + 1]);

        for (int i = 2; i < len; i++) {
            d[i] = Math.max(d[i - 1], d[i - 2] +
b[start + i]);
        }
        return d[len - 1];
```

```
    }

    @Override
    int SquareHouses(int[] c) {
        int n = c.length;
        if (n == 0) {
            return 0;
        }
        if (n == 1) {
            return c[0];
        }

        int[] dp = new int[n];
        dp[0] = c[0];
        dp[1] = Math.max(c[0], c[1]);

        for (int i = 2; i < n; i++) {
            dp[i] = Math.max(dp[i - 1], dp[i - 2] +
c[i]);
        }

        return dp[n - 1];
    }

    @Override
    int MultiHouseBuilding(int[][] d) {
        int len = d.length;
        if (len == 0) {
            return 0;
        }
        if (len == 1) {
            return Math.max(d[0][0], d[0][1]);
        }
```

```
        int[] dp = new int[len];
        dp[0] = Math.max(d[0][0], d[0][1]);
        dp[1] = Math.max(d[1][0], d[1][1]);

        for (int i = 2; i < len; i++) {
            dp[i] = Math.max(dp[i - 1], dp[i - 2] +
Math.max(d[i][0], d[i][1]));
        }

        return dp[len - 1];
    }

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        JAVAProfessionalRobber javaRobber = new
JAVAProfessionalRobber();
        javaRobber.RobbingClass();
        javaRobber.MachineLearning();

        // User input for RowHouses
        System.out.println("Enter the number of houses
for RowHouses:");
        int n = scanner.nextInt();
        System.out.println("Enter the amount of money
in each house:");
        int[] moneyInRowHouses = new int[n];

        for (int i = 0; i < n; i++) {
            moneyInRowHouses[i] = scanner.nextInt();
        }
        int maxAmountRowHouses =
javaRobber.RowHouses(moneyInRowHouses);
```

```
        System.out.println("Max amount robbed from row
houses: " + maxAmountRowHouses);

        // User input for RoundHouses
        System.out.println("Enter the number of houses
for RoundHouses:");
        n = scanner.nextInt();
        int[] moneyInRoundHouses = new int[n];
        System.out.println("Enter the amount of money
in each house:");
        for (int i = 0; i < n; i++) {
            moneyInRoundHouses[i] = scanner.nextInt();
        }
        int maxAmountRoundHouses =
javaRobber.RoundHouses(moneyInRoundHouses);
        System.out.println("Max amount robbed from
round houses: " + maxAmountRoundHouses);

        // User input for Square houses
        System.out.println("Enter the number of houses
for SquareHouses:");
        n = scanner.nextInt();
        int[] moneyInSquareHouses = new int[n];
        System.out.println("Enter the amount of money
in each house:");
        for (int i = 0; i < n; i++) {
            moneyInSquareHouses[i] =
scanner.nextInt();
        }
        int maxAmountSquareHouses =
javaRobber.SquareHouses(moneyInSquareHouses);
        System.out.println("Max amount robbed from
square houses: " + maxAmountSquareHouses);
```

```
        //user inputed for MultiHouse
        System.out.println("Enter the number of house
types for MultiHouseBuilding:");
        int types = scanner.nextInt();
        int[][] moneyInMultiTypeBuilding = new
int[types][];
        for (int t = 0; t < types; t++) {
            System.out.println("Enter the number of
houses for type " + (t + 1) + ":");
            n = scanner.nextInt();
            moneyInMultiTypeBuilding[t] = new int[n];
            System.out.println("Enter the amount of
money in each house for type " + (t + 1) + ":");
            for (int i = 0; i < n; i++) {
                moneyInMultiTypeBuilding[t][i] =
scanner.nextInt();
            }
        }
        int maxAmountMultiTypeBuilding =
javaRobber.MultiHouseBuilding(moneyInMultiTypeBuilding
);
       System.out.println("Max amount robbed from
square houses: " + maxAmountMultiTypeBuilding);
        scanner.close();
    }
}
}
```

```
MScAI&ML
I love MachineLearning.
Enter the number of houses for RowHouses:
4
Enter the amount of money in each house:
23 45 55 19
Max amount robbed from row houses: 78
Enter the number of houses for RoundHouses:
4
Enter the amount of money in each house:
20 90 71 50
Max amount robbed from round houses: 140
Enter the number of houses for SquareHouses:
4
Enter the amount of money in each house:
10 12 19 10
Max amount robbed from square houses: 29
```