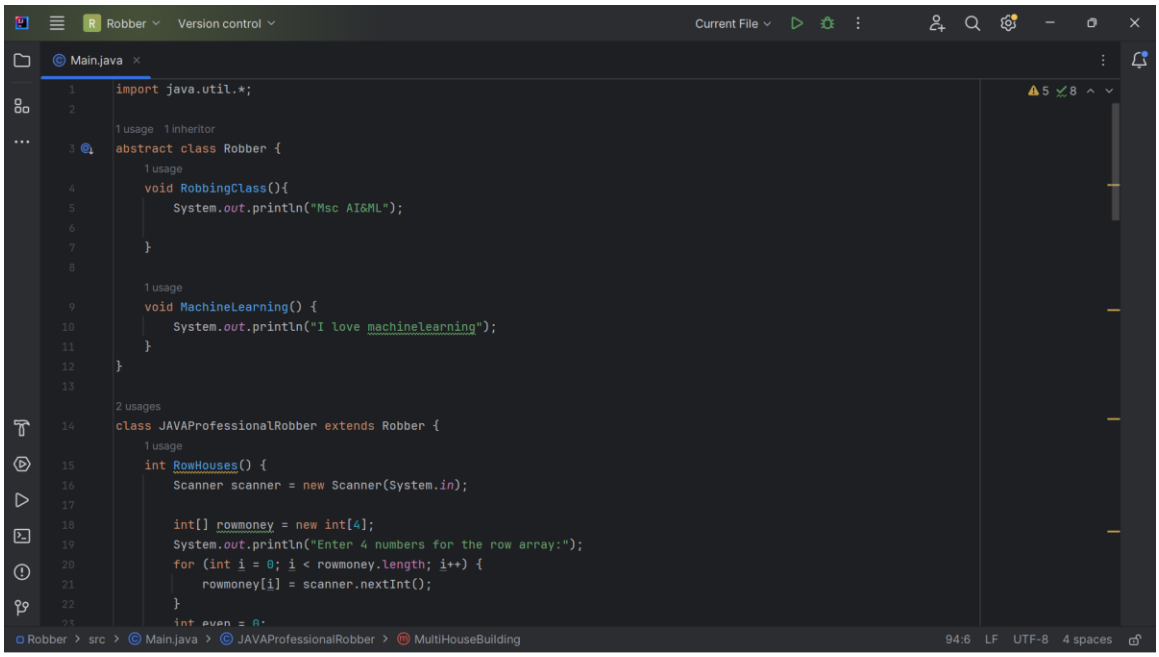
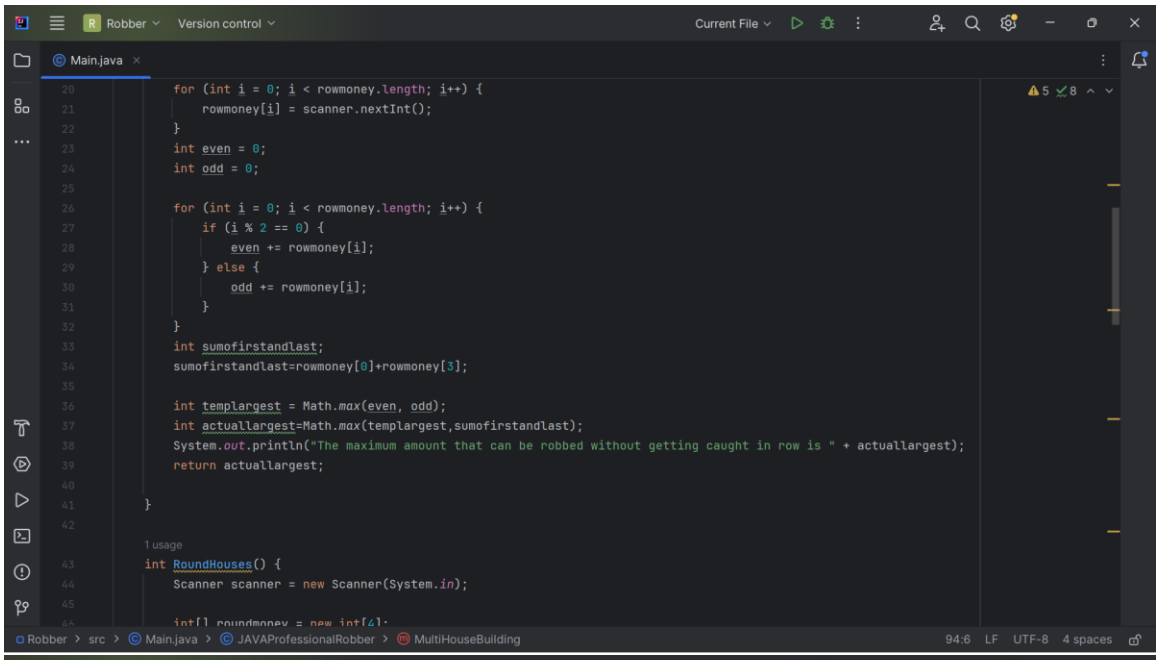
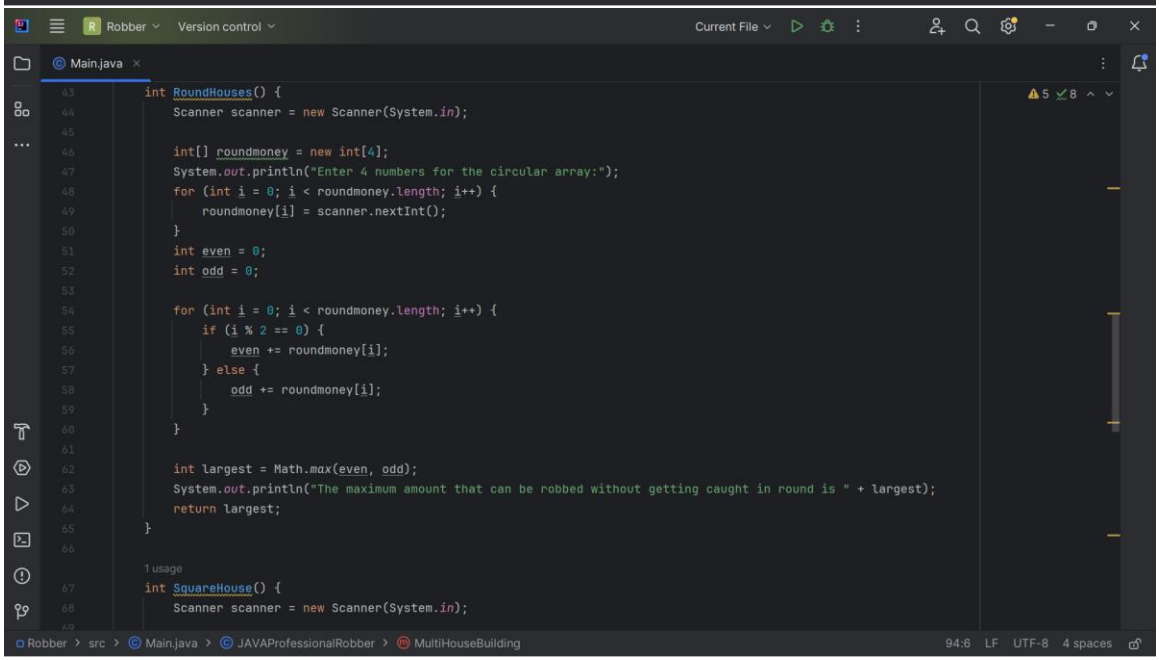
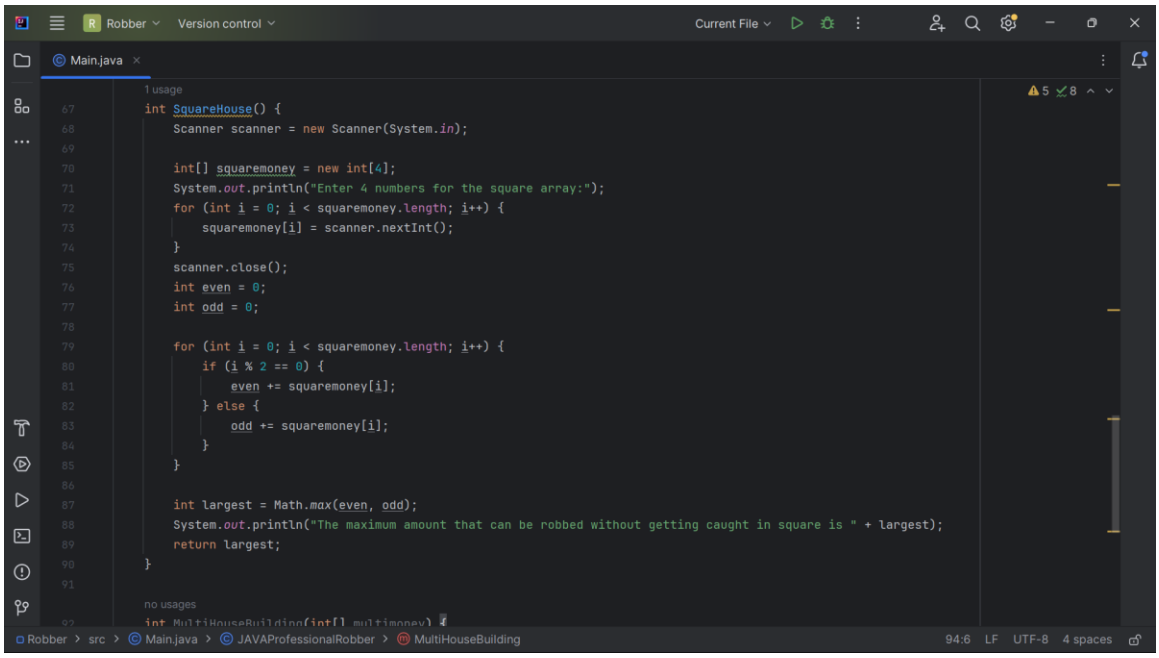

Robber
Version control
Current File
Main.java
import java.util.*;
1 usage 1 inheritor
abstract class Robber {
1 usage
void RobbingClass(){
System.out.println("Msc AI&ML");
}
1 usage
void MachineLearning() {
System.out.println("I love machinelearning");
}
}
2 usages
class JAVAProfessionalRobber extends Robber {
1 usage
int RowHouses() {
Scanner scanner = new Scanner(System.in);
int[] rowmoney = new int[4];
System.out.println("Enter 4 numbers for the row array:");
for (int i = 0; i < rowmoney.length; i++) {
rowmoney[i] = scanner.nextInt();
}
Robber › src › Main.java › JAVAProfessionalRobber › MultiHouseBuilding
94:6 LF UTF-8 4 spaces

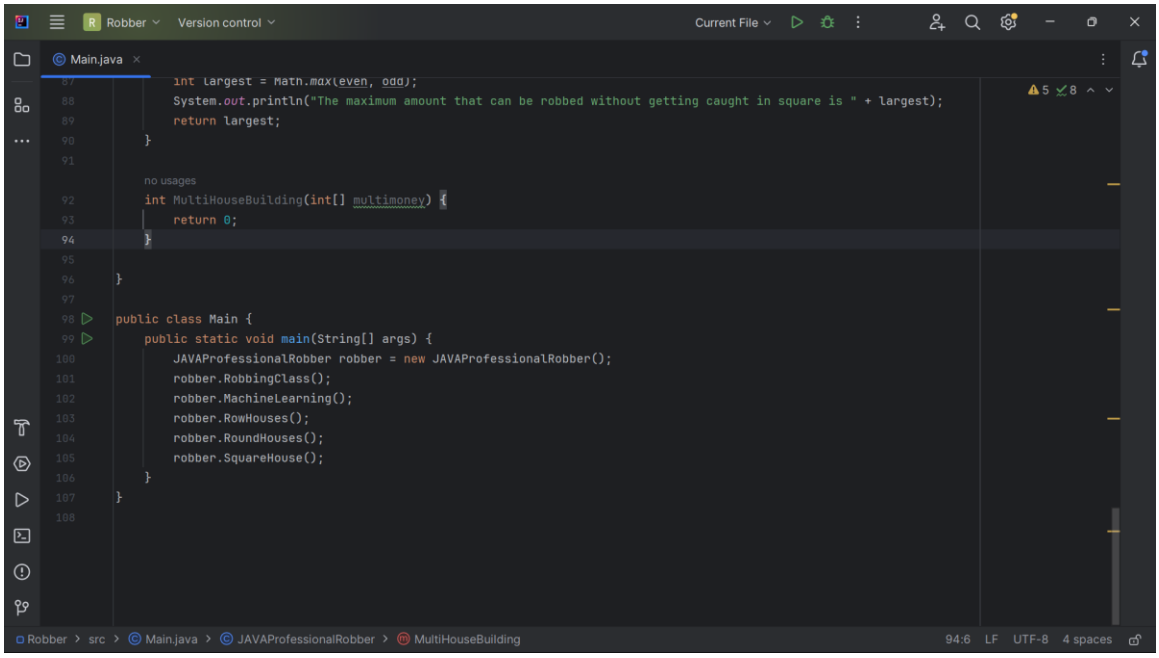


```
        for (int i = 0; i < rowmoney.length; i++) {
            rowmoney[i] = scanner.nextInt();
        }
        int even = 0;
        int odd = 0;

        for (int i = 0; i < rowmoney.length; i++) {
            if (i % 2 == 0) {
                even += rowmoney[i];
            } else {
                odd += rowmoney[i];
            }
        }
        int sumofirstandlast;
        sumofirstandlast=rowmoney[0]+rowmoney[3];

        int templargest = Math.max(even, odd);
        int actuallargest=Math.max(templargest,sumofirstandlast);
        System.out.println("The maximum amount that can be robbed without getting caught in row is " + actuallargest);
        return actuallargest;

    }

    1 usage
    int RoundHouses() {
        Scanner scanner = new Scanner(System.in);
```

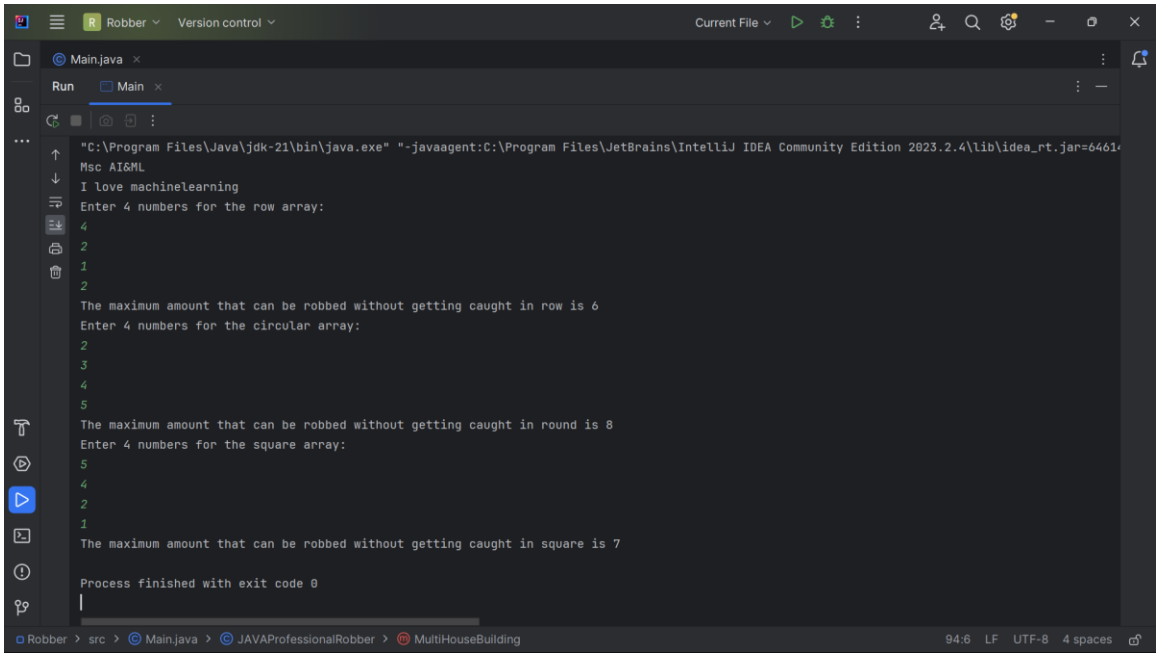


```
    int RoundHouses() {
        Scanner scanner = new Scanner(System.in);

        int[] roundmoney = new int[4];
        System.out.println("Enter 4 numbers for the circular array:");
        for (int i = 0; i < roundmoney.length; i++) {
            roundmoney[i] = scanner.nextInt();
        }
        int even = 0;
        int odd = 0;

        for (int i = 0; i < roundmoney.length; i++) {
            if (i % 2 == 0) {
                even += roundmoney[i];
            } else {
                odd += roundmoney[i];
            }
        }

        int largest = Math.max(even, odd);
        System.out.println("The maximum amount that can be robbed without getting caught in round is " + largest);
        return largest;
    }

    1 usage
    int SquareHouse() {
        Scanner scanner = new Scanner(System.in);
```


Robber
Version control
Current File
Main.java
1 usage
int SquareHouse() {
Scanner scanner = new Scanner(System.in);
int[] squaremoney = new int[4];
System.out.println("Enter 4 numbers for the square array:");
for (int i = 0; i < squaremoney.length; i++) {
squaremoney[i] = scanner.nextInt();
}
scanner.close();
int even = 0;
int odd = 0;
for (int i = 0; i < squaremoney.length; i++) {
if (i % 2 == 0) {
even += squaremoney[i];
} else {
odd += squaremoney[i];
}
}
int largest = Math.max(even, odd);
System.out.println("The maximum amount that can be robbed without getting caught in square is " + largest);
return largest;
}
no usages
Robber › src › Main.java › JAVAProfessionalRobber › MultiHouseBuilding
94:6 LF UTF-8 4 spaces


Robber
Version control
Current File
Main.java
int largest = Math.max(even, odd);
System.out.println("The maximum amount that can be robbed without getting caught in square is " + largest);
return largest;
}
no usages
int MultiHouseBuilding(int[] multimoney) {
return 0;
}
}
public class Main {
public static void main(String[] args) {
JAVAProfessionalRobber robber = new JAVAProfessionalRobber();
robber.RobbingClass();
robber.MachineLearning();
robber.RowHouses();
robber.RoundHouses();
robber.SquareHouse();
}
}
Robber › src › Main.java › JAVAProfessionalRobber › MultiHouseBuilding
94:6 LF UTF-8 4 spaces

OUTPUT:


Robber
Version control
Current File
Main.java
Run
Main
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.2.4\lib\idea_rt.jar=6461
Msc AI&ML
I love machinelearning
Enter 4 numbers for the row array:
4
2
1
2
The maximum amount that can be robbed without getting caught in row is 6
Enter 4 numbers for the circular array:
2
3
4
5
The maximum amount that can be robbed without getting caught in round is 8
Enter 4 numbers for the square array:
5
4
2
1
The maximum amount that can be robbed without getting caught in square is 7
Process finished with exit code 0
Robber
src
Main.java
JAVAProfessionalRobber
MultiHouseBuilding
94:6
LF
UTF-8
4 spaces