Write a Java program to create a class called "Building" with attributes for address, number of floors, and total area. Create subclasses "ResidentialBuilding" and "CommercialBuilding" that add specific attributes like number of apartments for residential and office space for commercial buildings. Implement a method to calculate the total rent for each subclass.

Sample Solution:

Java Code:

Building.java

```
// Import necessary packages
import java.util.ArrayList;
import java.util.List;

// Define the Building class
class Building {
    // Attributes for the Building class
    String address;
    int numberOfFloors;
    double totalArea;

    // Constructor for the Building class
    public Building(String address, int numberOfFloors, double totalArea) {
        this.address = address;
        this.numberOfFloors = numberOfFloors;
        this.totalArea = totalArea;
    }

    // Method to display basic information about the building
    public void displayInfo() {
        System.out.println("Address: " + address);
        System.out.println("Number of Floors: " + numberOfFloors);
        System.out.println("Total Area: " + totalArea + " sq meters");
    }
}
```

Explanation:

The above Java code defines a "Building class" with three attributes: address, numberOfFloors, and totalArea. It includes a constructor to initialize these attributes and a method displayInfo() to print the building's information. The import statements at the top are prepared for possible use of ArrayList and List classes, although they are not utilized in the provided code segment.

ResidentialBuilding.java

```
// Define the ResidentialBuilding class that extends Building
class ResidentialBuilding extends Building {
    // Additional attribute for ResidentialBuilding
    int numberOfApartments;
    double rentPerApartment;

    // Constructor for the ResidentialBuilding class
    public ResidentialBuilding(String address, int numberOfFloors, double t
        super(address, numberOfFloors, totalArea); // Call the constructor
        this.numberOfApartments = numberOfApartments;
        this.rentPerApartment = rentPerApartment;
    }

    // Method to calculate total rent for ResidentialBuilding
    public double calculateTotalRent() {
        return numberOfApartments * rentPerApartment;
    }

    // Override the displayInfo method to include additional details
    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Number of Apartments: " + numberOfApartments);
        System.out.println("Rent per Apartment: $" + rentPerApartment);
        System.out.println("Total Rent: $" + calculateTotalRent());
    }
}
```

Explanation:

The above Java code defines a "ResidentialBuilding class" that extends the Building class. It adds two additional attributes: numberOfApartments and rentPerApartment. The constructor initializes these attributes along with those inherited from Building. The method calculateTotalRent() computes the total rent based on the number of apartments and the rent

per apartment. The displayInfo() method is overridden to include details specific to residential buildings, in addition to the information from the "Building class".

CommercialBuilding.java

```
// Define the CommercialBuilding class that extends Building
class CommercialBuilding extends Building {
    // Additional attribute for CommercialBuilding
    double officeSpace; // in square meters
    double rentPerSquareMeter;

    // Constructor for the CommercialBuilding class
    public CommercialBuilding(String address, int numberOfFloors, double to
        super(address, numberOfFloors, totalArea); // Call the constructor
        this.officeSpace = officeSpace;
        this.rentPerSquareMeter = rentPerSquareMeter;
    }

    // Method to calculate total rent for CommercialBuilding
    public double calculateTotalRent() {
        return officeSpace * rentPerSquareMeter;
    }

    // Override the displayInfo method to include additional details
    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Office Space: " + officeSpace + " sq meters");
        System.out.println("Rent per Square Meter: $" + rentPerSquareMeter)
        System.out.println("Total Rent: $" + calculateTotalRent());
    }
}
```

Explanation:

The above Java code defines a "CommercialBuilding class" that extends the Building class. It adds two additional attributes: officeSpace (in square meters) and rentPerSquareMeter. The constructor initializes these attributes along with those inherited from Building. The method calculateTotalRent() computes the total rent based on the office space and rent per square meter. The displayInfo() method is overridden to include details specific to commercial buildings, in addition to the information from the Building class.

Main.java

```
// Main class to test the Building, ResidentialBuilding, and CommercialBuil
public class Main {
    public static void main(String[] args) {
        // Create an instance of ResidentialBuilding
        ResidentialBuilding residentialBuilding = new ResidentialBuilding('

        // Create an instance of CommercialBuilding
        CommercialBuilding commercialBuilding = new CommercialBuilding("10

        // Display information about the residential building
        System.out.println("Residential Building Info:");
        residentialBuilding.displayInfo();

        // Display information about the commercial building
        System.out.println("\nCommercial Building Info:");
        commercialBuilding.displayInfo();
    }
}
```

Explanation:

The above Java code defines the Main class to test the "Building", "ResidentialBuilding", and "CommercialBuilding" classes. In the main method, it performs the following actions:

- Creates an instance of ResidentialBuilding with specified attributes.
- Creates an instance of CommercialBuilding with specified attributes.
- Prints information about the residential building using the displayInfo() method.
- Prints information about the commercial building using the displayInfo() method.

The code demonstrates how to instantiate and utilize the "ResidentialBuilding" and "CommercialBuilding" classes.

Output:

```
Residential Building Info:
Address: 99 ABC Street.
Number of Floors: 10
Total Area: 2500.0 sq meters
Number of Apartments: 20
Rent per Apartment: $1000.0
```

```
Total Rent: $20000.0

Commercial Building Info:
Address: 100 PQR Business Avenue.
Number of Floors: 15
Total Area: 4500.0 sq meters
Office Space: 3000.0 sq meters
Rent per Square Meter: $20.0
Total Rent: $60000.0
```

Java Code Editor: