### UML Diagram for the Rectangle Class

```plaintext

+----------------------+

| Rectangle |

+----------------------+

| - width: double |

| - height: double |

+----------------------+

| + Rectangle() |

| + Rectangle(width: |

| double, height: |

| double) |

+----------------------+

| + getArea(): double |

| + getPerimeter(): |

| double |

+----------------------+

```

### Java Implementation of the Rectangle Class

```java

public class Rectangle {

private double width = 1;

private double height = 1;

// No-arg constructor

public Rectangle() {

}

// Constructor with specified width and height

public Rectangle(double width, double height) {

this.width = width;

this.height = height;

}

// Method to return the area of the rectangle

public double getArea() {

return width \* height;

}

// Method to return the perimeter of the rectangle

public double getPerimeter() {

return 2 \* (width + height);

}

// Getters for width and height (optional, if needed)

public double getWidth() {

return width;

}

public double getHeight() {

return height;

}

}

```

### Test Program to Create and Display Rectangle Objects

```java

public class TestRectangle {

public static void main(String[] args) {

// Create two Rectangle objects

Rectangle rectangle1 = new Rectangle(4, 40);

Rectangle rectangle2 = new Rectangle(3.5, 35.9);

// Display the properties of the first rectangle

System.out.println("Rectangle 1:");

System.out.println("Width: " + rectangle1.getWidth());

System.out.println("Height: " + rectangle1.getHeight());

System.out.println("Area: " + rectangle1.getArea());

System.out.println("Perimeter: " + rectangle1.getPerimeter());

System.out.println();

// Display the properties of the second rectangle

System.out.println("Rectangle 2:");

System.out.println("Width: " + rectangle2.getWidth());

System.out.println("Height: " + rectangle2.getHeight());

System.out.println("Area: " + rectangle2.getArea());

System.out.println("Perimeter: " + rectangle2.getPerimeter());

}

}

```

### UML Diagram for the Stock Class

```plaintext

+----------------------+

| Stock |

+----------------------+

| - symbol: String |

| - name: String |

| - previousClosingPrice: double |

| - currentPrice: double |

+----------------------+

| + Stock(symbol: String, name: String) |

| + getChangePercent(): double |

+----------------------+

```

### Java Implementation of the Stock Class

```java

public class Stock {

private String symbol;

private String name;

private double previousClosingPrice;

private double currentPrice;

// Constructor that creates a stock with the specified symbol and name.

public Stock(String symbol, String name) {

this.symbol = symbol;

this.name = name;

}

// Getter and Setter for previousClosingPrice

public double getPreviousClosingPrice() {

return previousClosingPrice;

}

public void setPreviousClosingPrice(double previousClosingPrice) {

this.previousClosingPrice = previousClosingPrice;

}

// Getter and Setter for currentPrice

public double getCurrentPrice() {

return currentPrice;

}

public void setCurrentPrice(double currentPrice) {

this.currentPrice = currentPrice;

}

// Method to return the percentage changed from previousClosingPrice to currentPrice.

public double getChangePercent() {

return ((currentPrice - previousClosingPrice) / previousClosingPrice) \* 100;

}

}

```

### Test Program

```java

public class TestStock {

public static void main(String[] args) {

// Create a Stock object with the stock symbol ORCL, the name Oracle Corporation, and the previous closing price of 34.5.

Stock stock = new Stock("ORCL", "Oracle Corporation");

stock.setPreviousClosingPrice(34.5);

// Set a new current price to 34.35.

stock.setCurrentPrice(34.35);

// Display the price-change percentage.

System.out.printf("The price-change percentage is %.2f%%\n", stock.getChangePercent());

}

}

```