Question 1

01.1

Strings are used to store an array of characters. This is a data type but not a primitive data type. It is a class that is given by the java standard library.

01.2

Java String pool is a special memory technique to store string literals in a efficient way. When we first create a string, program will first check our string pool for that string literal and if there is it will set a reference to that instead of creating a new one.

01.3

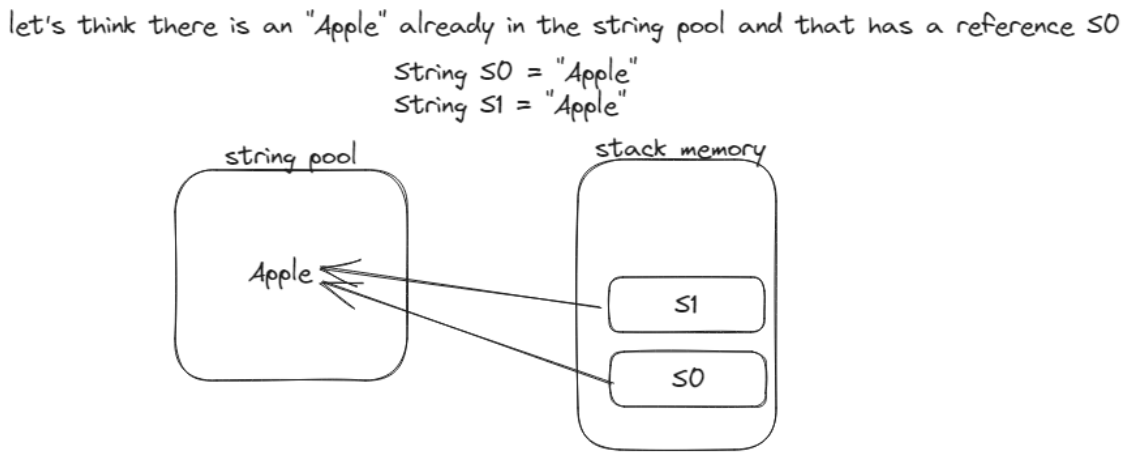
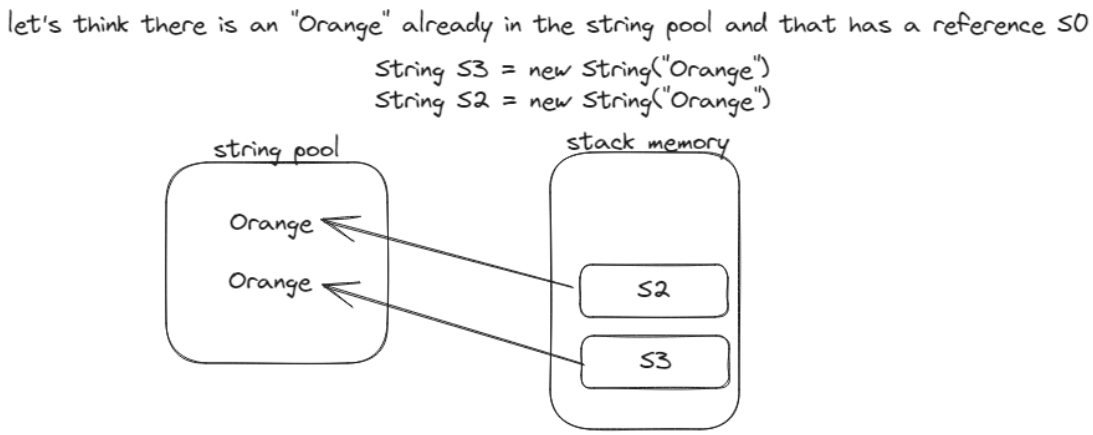


Figure 1 Scenario 1



So, for creating strings with string literals (like "Apple" and "Orange" in this examples), the most suitable way is scenario 1. It leverages the string pool, allowing multiple variables to share the same memory space when they have the same string literal content, which optimizes memory usage.

01.4

- text.length() - 17
- text.substring(5) - Galle Branch
- text.indexOf('B') - 2
- text.charAt(5) - G
- text.toUpperCase() - NIBM GALLE BRANCH
- text.replaceAll("Branch", "") - NIBM Galle

01.5

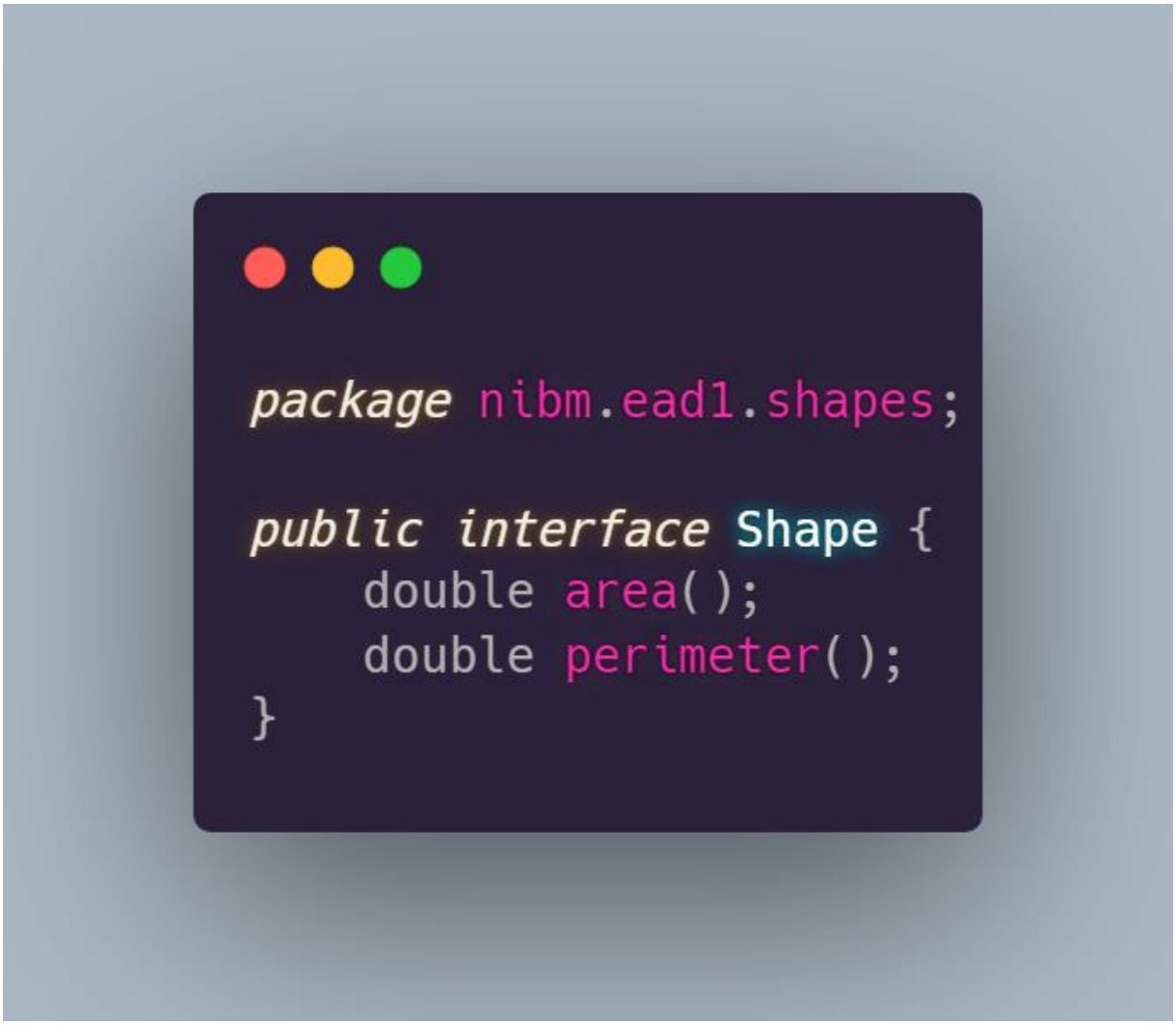
```
int wordCount = paragraph.split(" ").length;

int alphaCount = paragraph.replaceAll(" ", "").length();
```

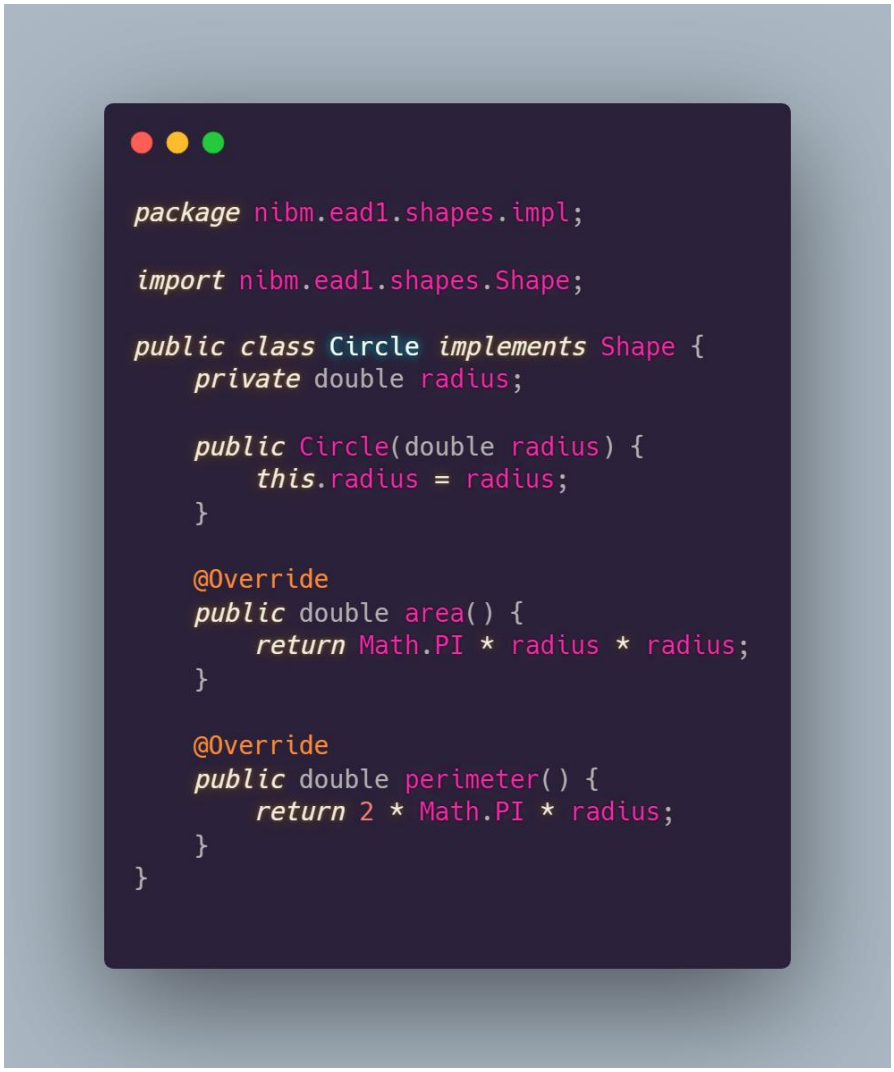
Question 2

02.1

Is concept that allows a class to inherit properties and behaviors from another class. Here the new class will be called child class and the class we get those properties will be called as the parent class.

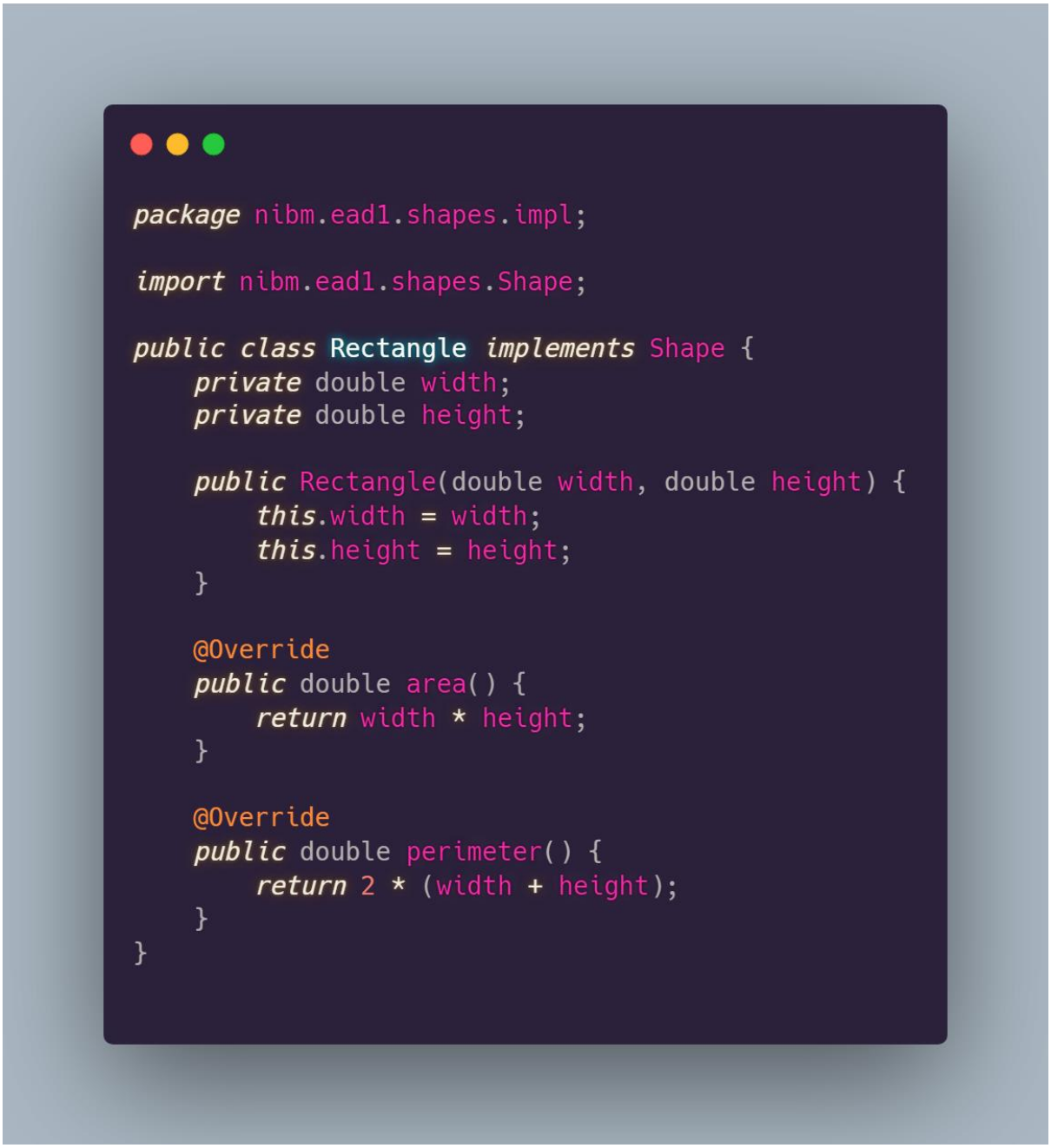
02.3


package nibm.ead1.shapes;
public interface Shape {
double area();
double perimeter();
}



```java
package nibm.ead1.shapes.impl;

import nibm.ead1.shapes.Shape;

public class Circle implements Shape {
    private double radius;

    public Circle(double radius) {
        this.radius = radius;
    }

    @Override
    public double area() {
        return Math.PI * radius * radius;
    }

    @Override
    public double perimeter() {
        return 2 * Math.PI * radius;
    }
}
```

```java
package nibm.ead1.shapes.impl;

import nibm.ead1.shapes.Shape;

public class Triangle implements Shape {
    private double base;
    private double height;

    public Triangle(double base, double height) {
        this.base = base;
        this.height = height;
    }

    @Override
    public double area() {
        return 0.5 * base * height;
    }

    @Override
    public double perimeter() {
        // Assuming all legs are equal (equilateral triangle)
        return 3 * base;
    }
}
```



```java
package nibm.ead1.shapes.impl;

import nibm.ead1.shapes.Shape;

public class Rectangle implements Shape {
    private double width;
    private double height;

    public Rectangle(double width, double height) {
        this.width = width;
        this.height = height;
    }

    @Override
    public double area() {
        return width * height;
    }

    @Override
    public double perimeter() {
        return 2 * (width + height);
    }
}
```