Week 2 assignment

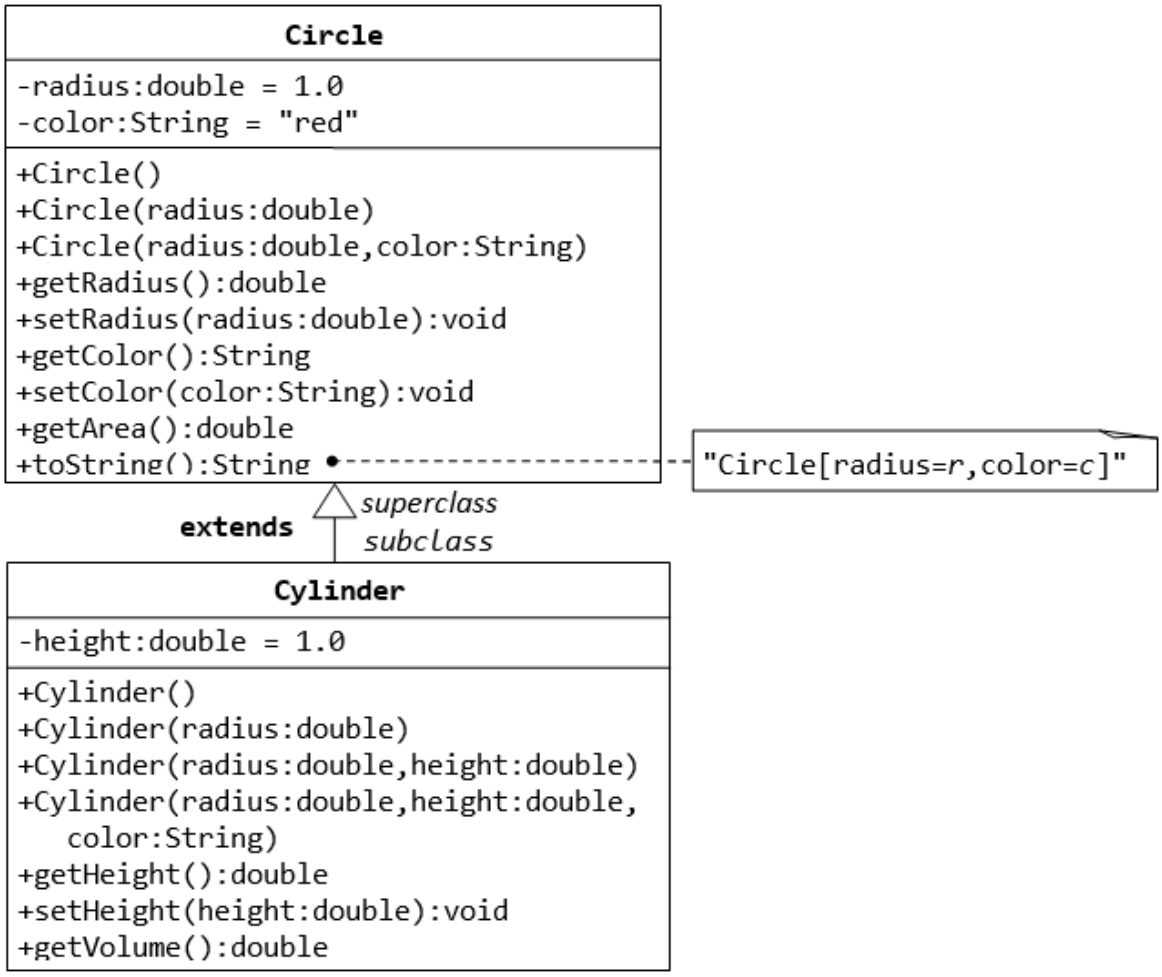
Classes and objects, inheritance, polymorphism and exceptions

---

Assignment #1

Inheritance and polymorphism

Create classes `Circle` and `Cylinder` as shown in the *UML* diagram below:



In the `main()` function of a Program class, create an array of `Circle` references with the initialization shown below:

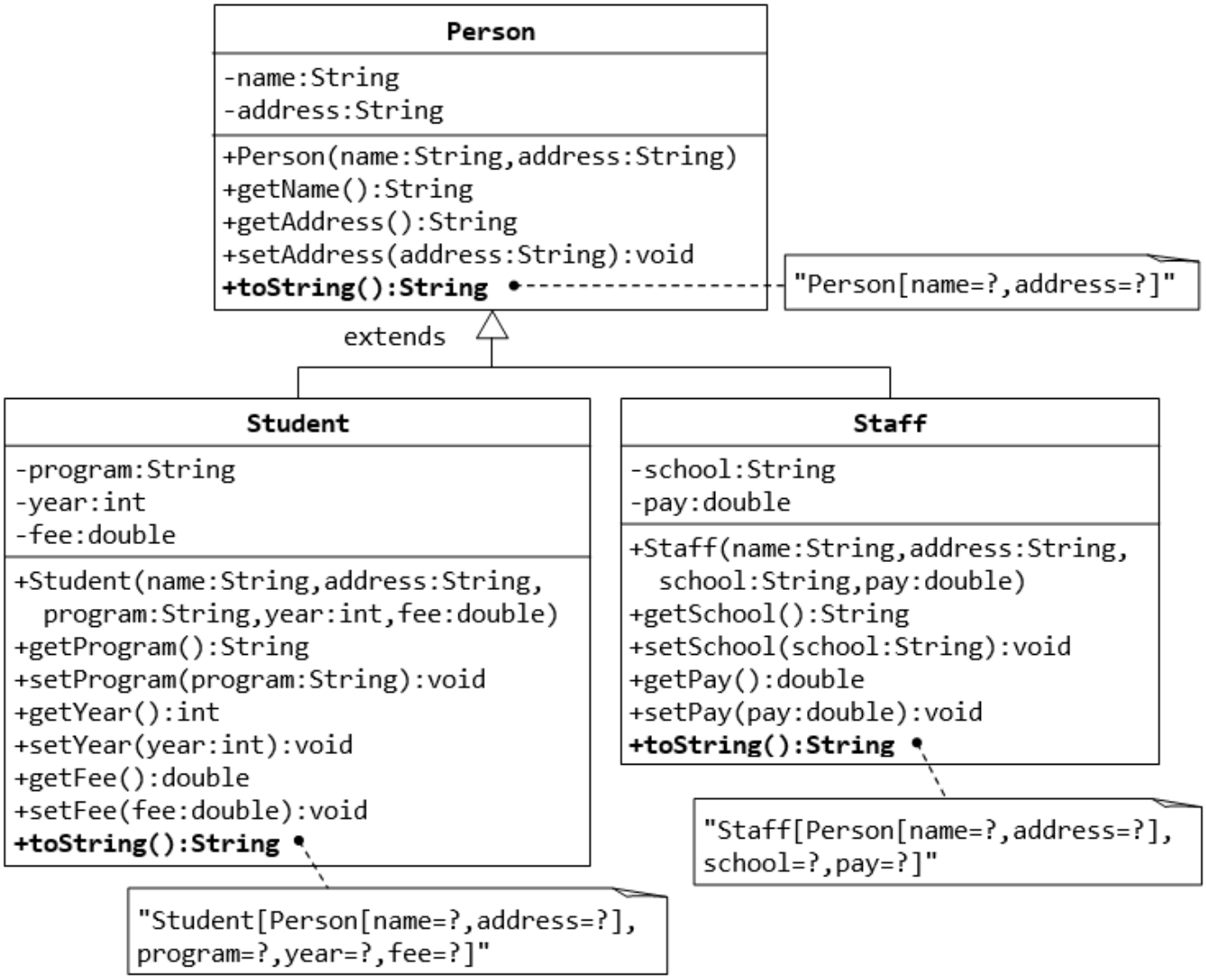
```
Circle[] circles = {
        new Cylinder(12.34),
        new Cylinder(12.34, 10.0),
        new Cylinder(12.34, 10.0, "blue")
};
```

Print the area of the circular region of each cylinder along with the volume of the same.

Assignment #2

Classes, inheritance and polymorphism

Create the classes `Person`, `Student`, and `Staff` as shown in the *UML* diagram below:



In the `main()` function of a Program class, create an array of `Person` references with the initialisation shown below:

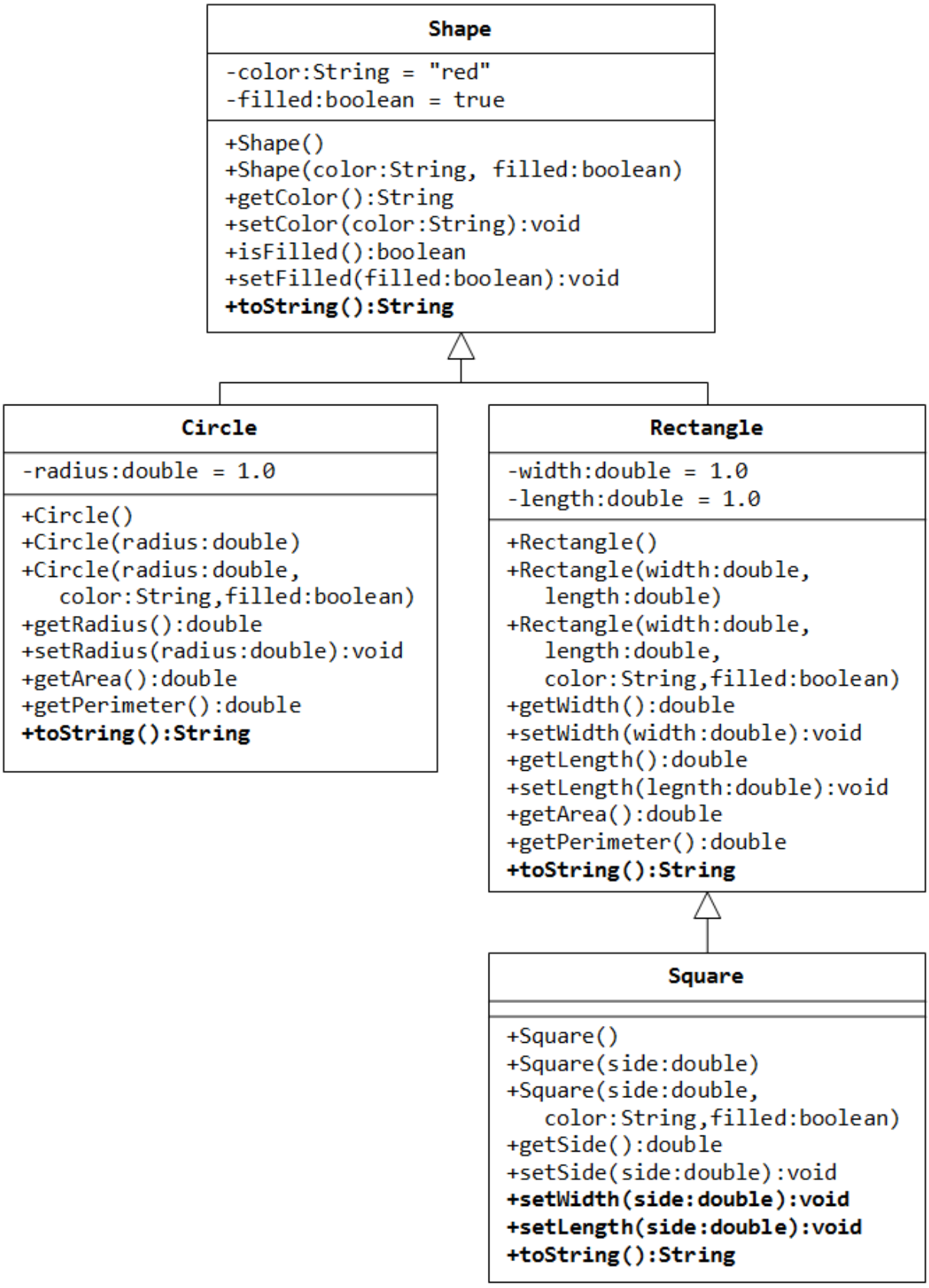
```
Person[] people = {
        new Student("Shyam", "Bangalore, Karnataka", "Java fundamentals", 2010, 4500.0
        new Staff("Anand", "Bangalore, Karnataka", "Delhi Public school", 35000.0),
        new Staff("Umesh", "Bangalore, Karnataka", "National Public school", 42000.0),
        new Student("Suresh", "Hassan, Karnataka", "Java fundamentals", 2012, 4750.0),
        new Student("Kiran", "Vasco, Goa", "ReactJS", 2017, 12500.0)
};
```

Print the details of all `Person` objects (using the `toString()`).

Assignment #3

Classes, inheritance and polymorphism

Create the classes `Shape`, `Circle`, `Rectangle`, and `Square` as shown in the *UML* diagram below:



The `toString` function of the above classes should return text as given below:

|Classname|Sample return value from toString()| |---|---| |Shape|A Shape with color of xxx and filled/Not filled| |Circle|A Circle with radius=xxx, which is a subclass of yyy (where yyy is the output of the toString() method from the superclass)| |Rectangle|A Rectangle with width=xxx and length=zzz, which is a subclass of yyy (where yyy is the output of the toString() method from the superclass)| |Square|A Square with side=xxx, which is a subclass of yyy (where yyy is the output of the toString() method from the superclass)|

In the `main()` method of a Program class, create an array of 10 `Shape` references containing a mixture of `Circle`, `Rectangle` and `Square` objects of different dimensions. Using a loop, print the `perimeter` and `area` for all of them.

Assignment #4

Exception handling

You are working on a Java application that involves processing large amounts of data. As part of your work, you need to write code that can handle various exceptions that may occur during the processing of data.

Your task is to write a Java program that reads data from a file and performs various calculations on the data. Your program should be able to handle any exceptions that may occur during the processing of the data.

The program should read input data from a text file named "data.txt", which contains a series of numbers (one number per line). The program should calculate the average of these numbers and display the result on the console.

However, there are several potential exceptions that your program may encounter while reading and processing the data. Specifically, your program should be able to handle the following exceptions:

1. FileNotFoundException: If the data.txt file cannot be found, the program should display an error message indicating that the file could not be found.
2. NumberFormatException: If the data.txt file contains invalid data (i.e., non-numeric characters), the program should display an error message indicating that the data is invalid.
3. IOException: If there is an error while reading the data from the file, the program should display an error message indicating that there was a problem reading the data.
4. ArithmeticException: If there is an error while calculating the average (i.e., dividing by zero), the program should display an error message indicating that there was a problem calculating the average.

To handle these exceptions, your program should use try-catch blocks. The program should catch each exception individually and display an appropriate error message on the console.

Your program should have the following features:

1. A main method that reads input data from a file and calculates the average.
2. Error handling using try-catch blocks.
3. Display of appropriate error messages on the console.
4. The ability to handle multiple types of exceptions.
5. The ability to handle large amounts of input data.

Note: You can assume that the input data is always valid, except for the exceptions mentioned above.
