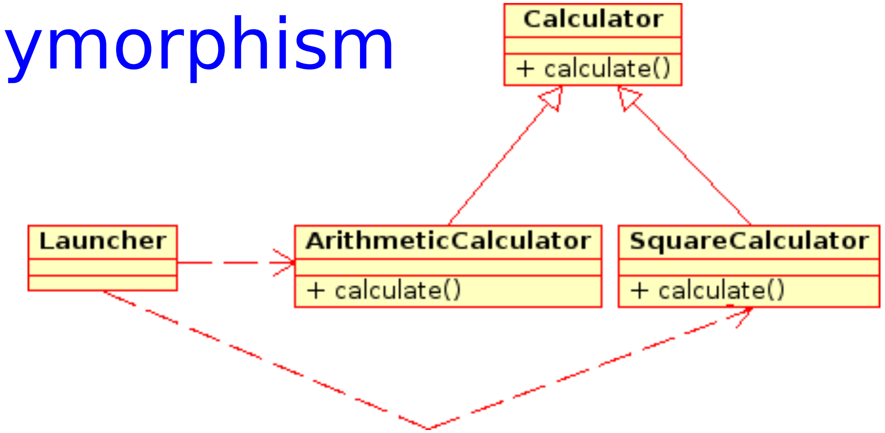
Container Elements and Polymorphism



```
import java.util.*;

public class Launcher {

    public static void main(String[] s) {
        double a = 3.0;
        double b = 2.0;
        String o = "+";
        double r = 0.0;
        Calculator c1 = new ArithmeticCalculator();
        Calculator c2 = new SquareCalculator();
        List<Calculator> l = new ArrayList<Calculator>();
        l.add(c1);
        l.add(c2);
        int j = 0;
        Calculator c = null;
        while (true) {
            if (j >= l.size()) {
                break;
            }
            c = l.get(j);
            r = c.calculate(a, b, o);
            System.out.println("The result of calculator number: " + j + "is: " + r);
            j++;
        }
    }
}
```

Summary

- data structure: composed (static), container (dynamic)
- [primitive data types do not have a structure]
- methods: create, insert, size, access, remove, clear
- Java collection framework: collections (set, list), maps

Contents

Introduction

Style Guide

Logging

File and Directory

Data Structure

Stream

Object Serialisation and Persistence

Object Cloning

Date and Time

XML Processing

Concurrency

Meta Programming and Reflexion

Language Binding

Java Native Interface (JNI)

Example

```
import java.io.*;

public class Launcher {

    public static void main(String[] args) throws Exception {

        FileReader in = new FileReader("testin.txt");
        FileWriter out = new FileWriter("testout.txt");
        int c = in.read();

        while (c != -1) {

            System.out.print((char) c);
            out.write(c);

            c = in.read();
        }

        // while ((c = in.read()) != -1) {

        in.close();
        out.close();
    }
}
```

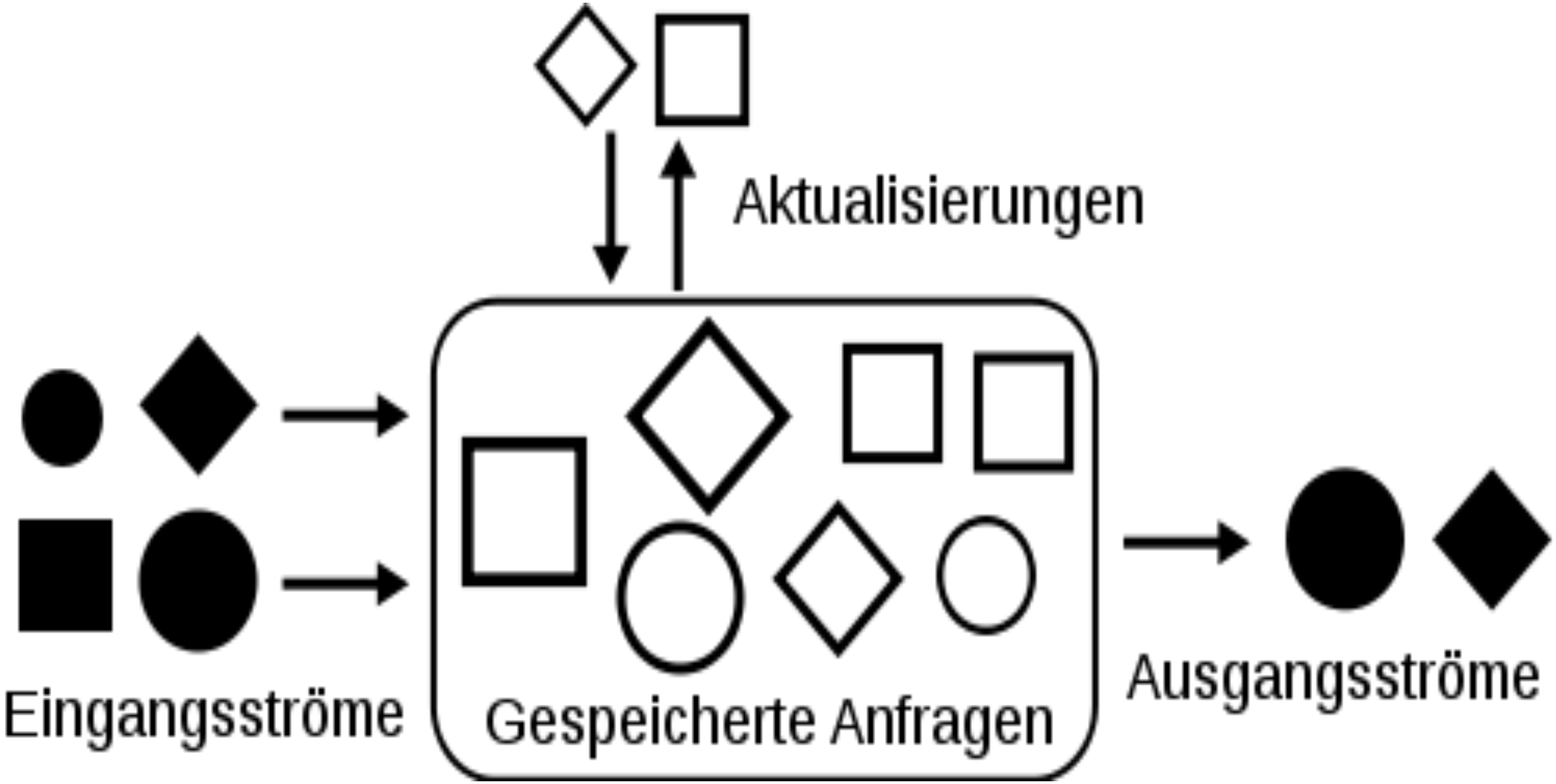
Standard Input / Output Channels (Streams)

- System.in: standard input channel
- System.out: standard output channel
- System.err: error output channel

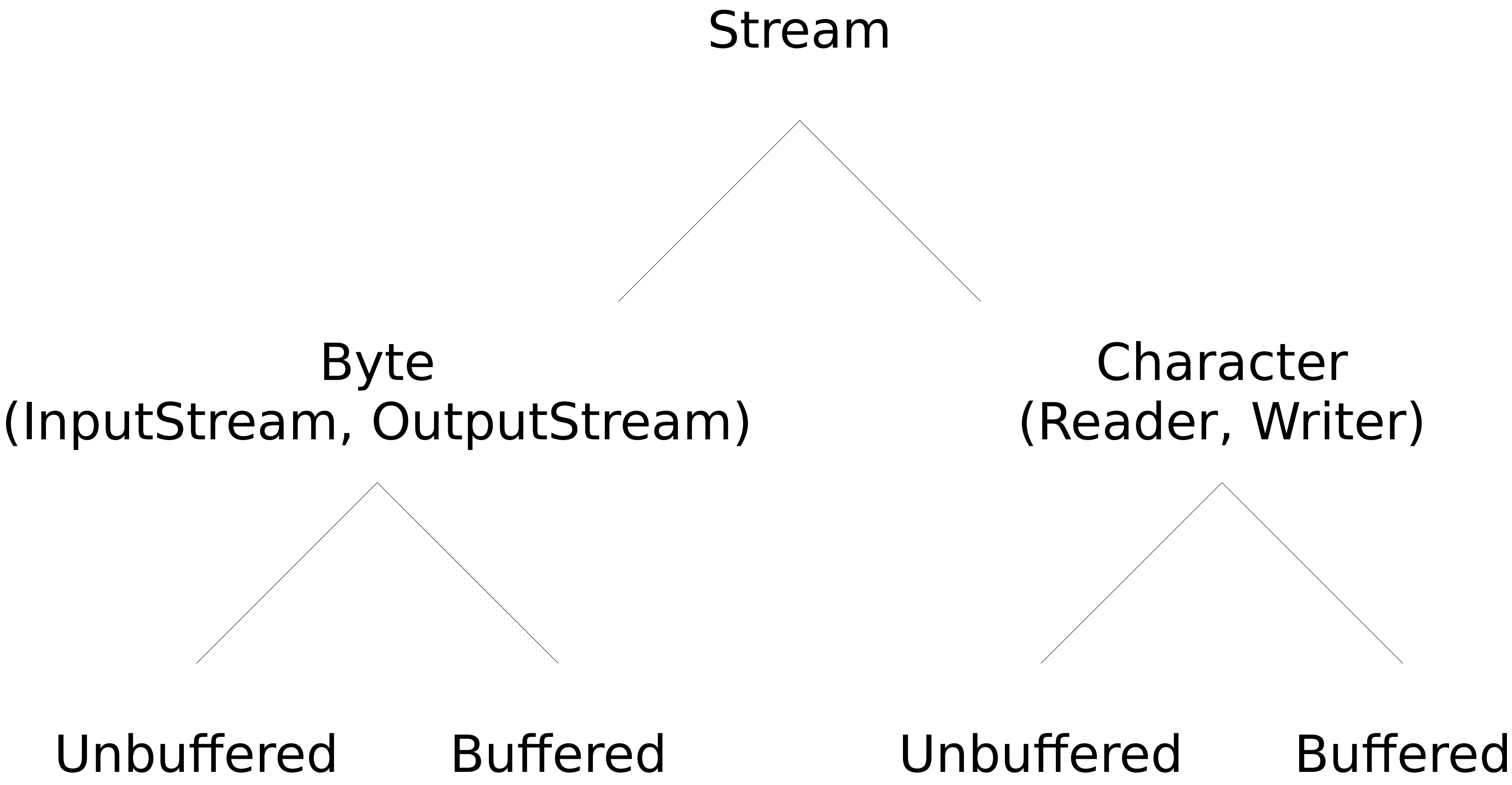
```
System.out.println("I am a standard message");

// Printed in red colour by Eclipse IDE
System.err.println("I am an error message");
```

Stream


Aktualisierungen
Eingangsströme
Gespeicherte Anfragen
Ausgangsströme

Systematics



Stream

Byte
(InputStream, OutputStream)

Character
(Reader, Writer)

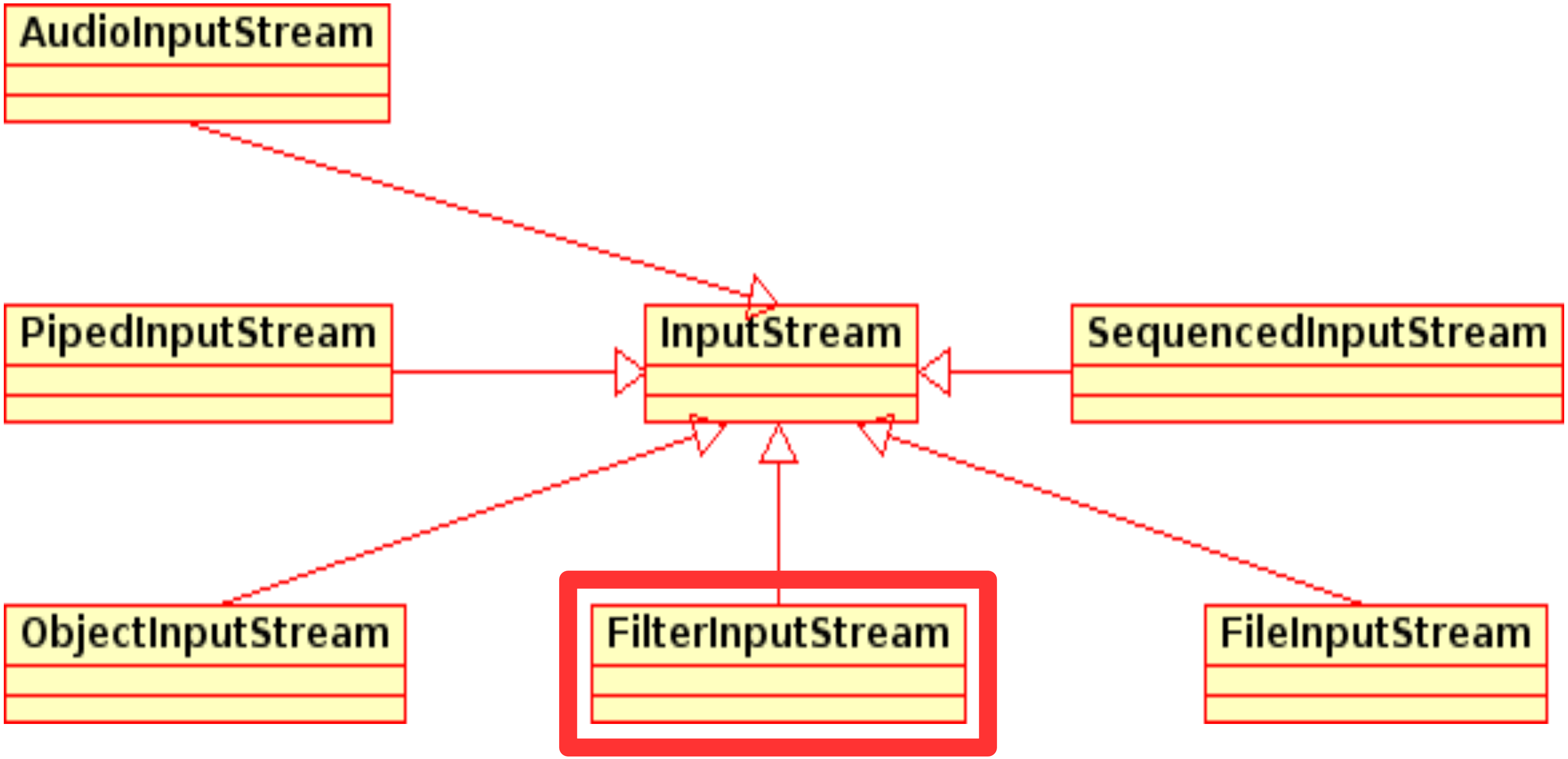
Unbuffered

Buffered

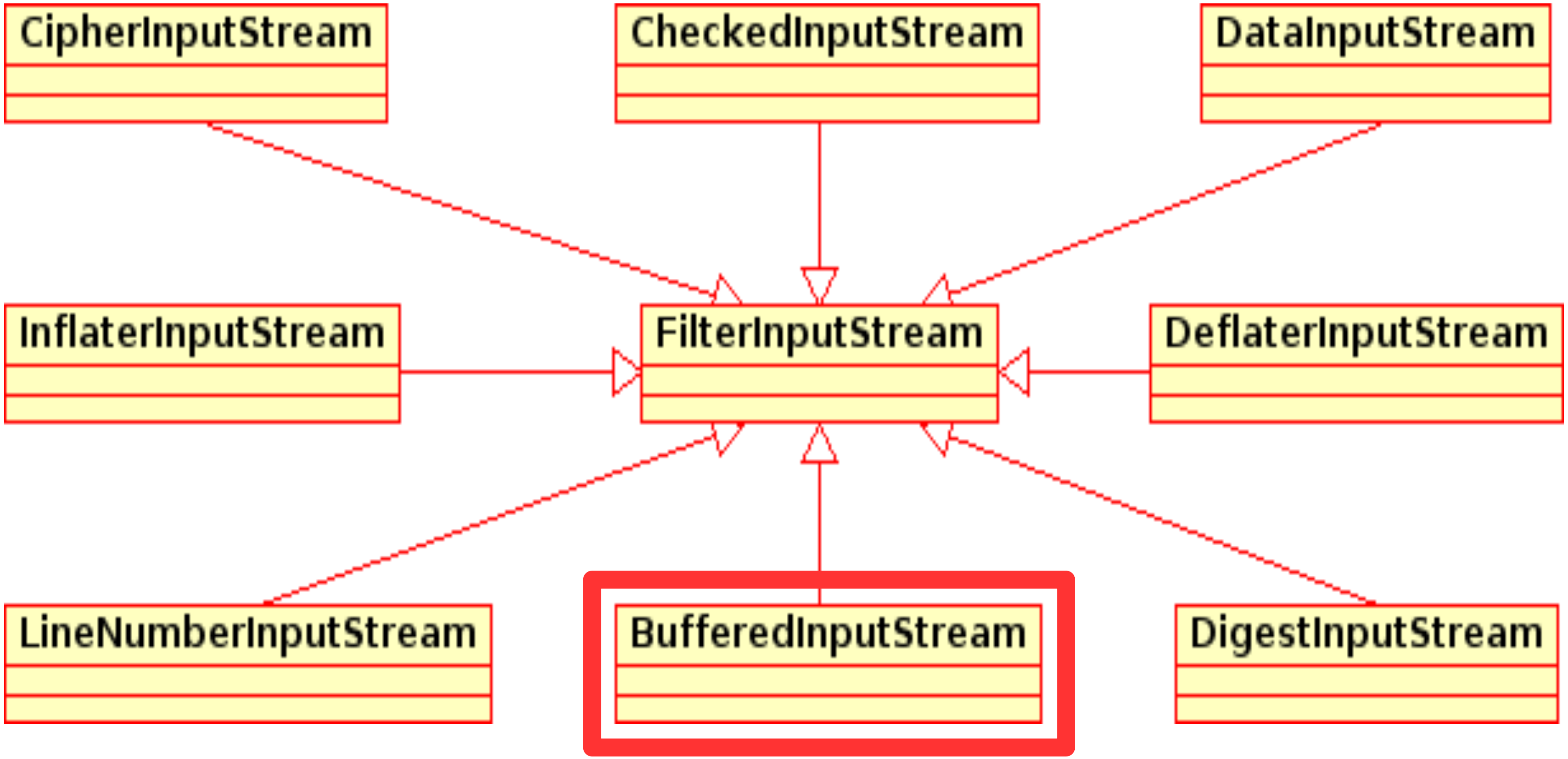
Unbuffered

Buffered

Input Byte Stream



Filter Input Byte Stream


CipherInputStream
CheckedInputStream
DataInputStream
InflaterInputStream
FilterInputStream
DeflaterInputStream
LineNumberInputStream
BufferedInputStream
DigestInputStream