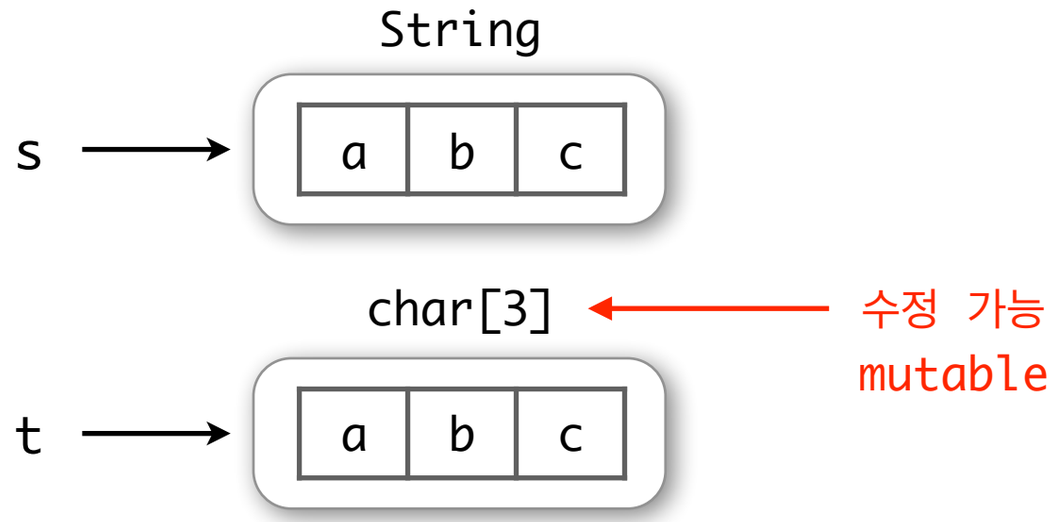
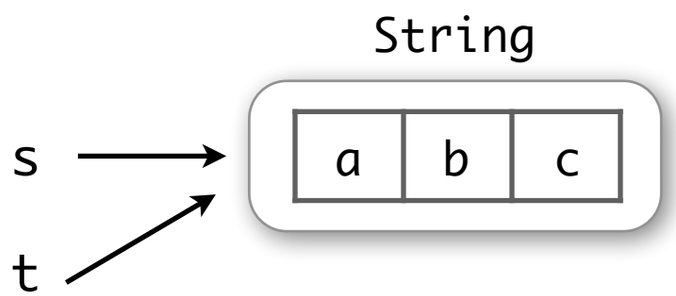
텍스트 및 파일 처리

문자열String은 수정불가Immutable

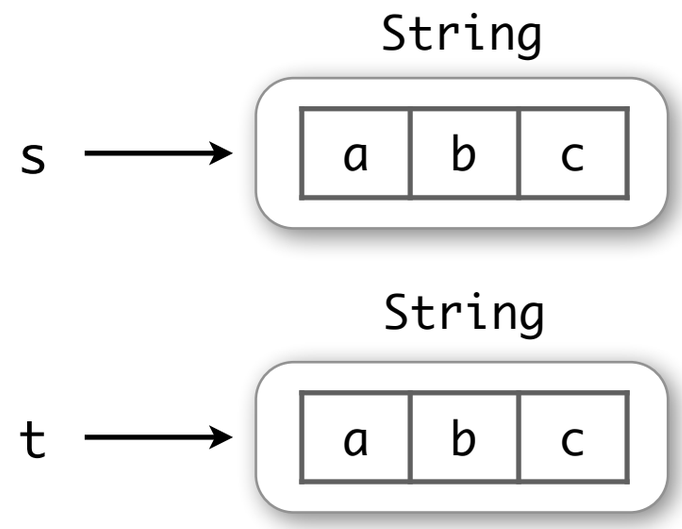
```
String s = "abc";
char[] t = new char[3];
t[0] = 'a';
t[1] = 'b';
t[2] = 'c';
System.out.println(s == t);
System.out.println(s.equals(t));
```



```
String s = "abc";
String t = "abc";
System.out.println(s == t);
System.out.println(s.equals(t));
```

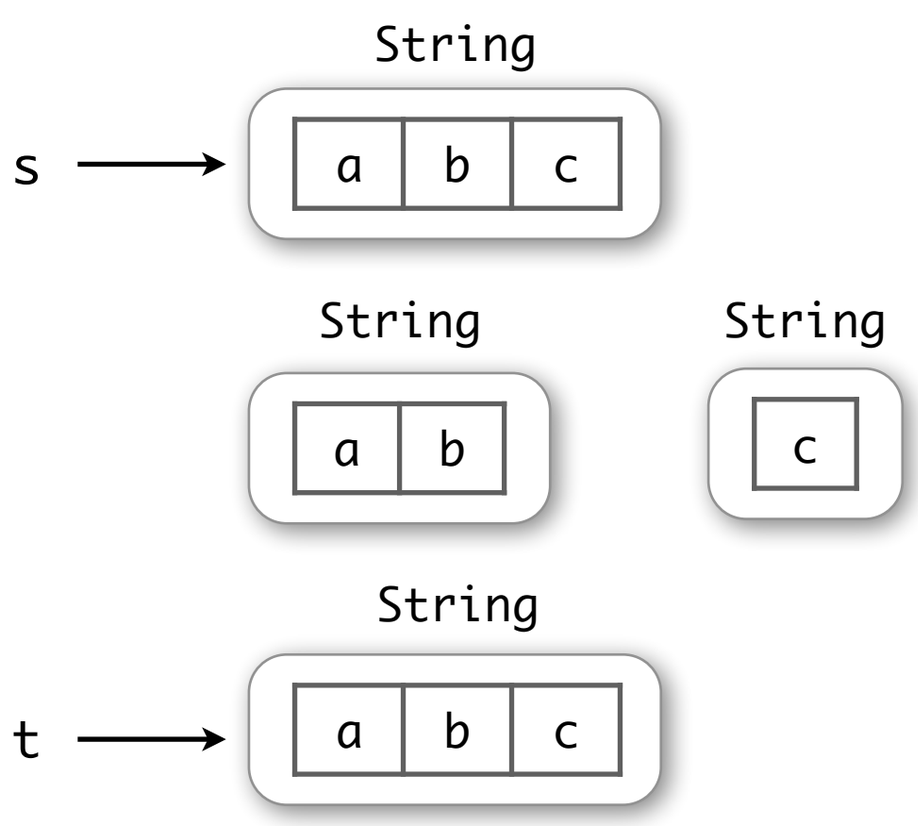


```
String s = "abc";
String t = new String("abc");
System.out.println(s == t);
System.out.println(s.equals(t));
```

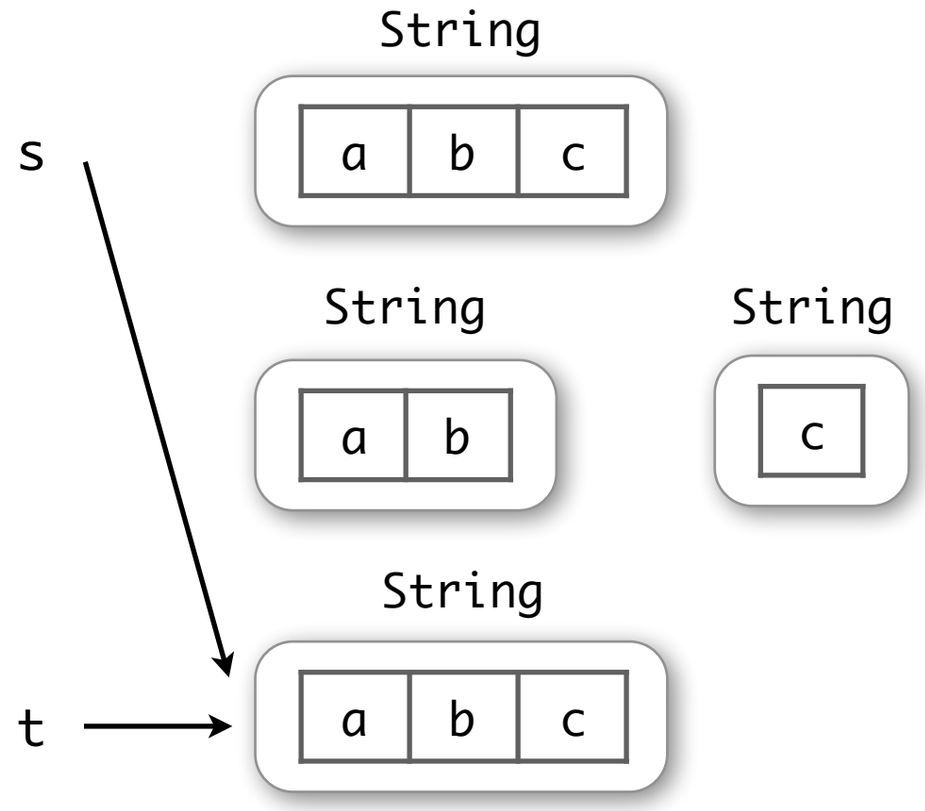


문자열String은 수정불가Immutable

```
String s = "abc";
String t = new String("ab") + "c";
System.out.println(s == t);
System.out.println(s.equals(t));
```




```
String s = "abc";
String t = new String("ab") + "c";
s = t;
System.out.println(s == t);
System.out.println(s.equals(t));
```



class StringTokenizer

java.util

문자열을 토큰 단위로 쪼개주는 메소드 장착

class	StringTokenizer	기능
Constructor	new StringTokenizer (String text, String delimiter)	text에서 토큰을 뽑아내는 객체를 생성, delimiter 문자열에 있는 문자는 모두 토큰을 구별하는 분리문자 역할을 함
methods	String nextToken()	토크나이저 객체가 갖고 있는 텍스트를 검사하여 선두에 있는 구분자(delimiter)를 제거하고 다음 구분자 또는 텍스트의 끝에 이르기 까지의 문자를 모은 문자열을 리턴
	String nextToken (String new_delimiters)	nextToken()과 동일하게 작동하나, new_delimiters를 구분자로 사용한다
	boolean hasMoreTokens()	토큰이 남아있으면 true, 남아있지 않으면 false를 리턴
	int countTokens()	남아있는 토큰의 개수를 리턴

```
String s = "경기도 안산시 상록구";
StringTokenizer t = new StringTokenizer(s, " ");
String province = t.nextToken();
System.out.println(province);
String city = t.nextToken();
System.out.println(city);
String district = t.nextToken();
System.out.println(district);
```

```
String s = "$24.99";
StringTokenizer t = new StringTokenizer(s, "$.");
System.out.println(t.nextToken());
System.out.println(t.nextToken());
```

직렬 파일 Sequential Files

`java.io`

- **파일 file** - 보조 기억장치에 영구 저장되어 있는 심벌의 시퀀스(나열)
  - 문자 파일 (= 텍스트 파일) : 키보드 문자의 시퀀스 (바이트 단위로 저장)
  - 바이너리 파일 : 0과 1의 시퀀스

직렬 파일에 쓰기

```
import java.io.*;

public class Test {
    public static void main(String[] args) throws Exception {
        FileWriter writer = new FileWriter("poem.txt"); // 쓸 용도로 파일 만들고 열기
        PrintWriter outfile = new PrintWriter(writer);
        outfile.println("가을이 오면...");
        outfile.println("학기가 저물고,");
        outfile.println("시험이 끝나면서");
        outfile.println("겨울이 온다.");
        outfile.close(); // 파일 닫기
    }
}
```

직렬 파일에 쓰기

```
import java.io.*;

public class Test {
   public static void main(String[] args) throws Exception {
      FileWriter writer = new FileWriter("poem.txt"); // 쓸 용도로 파일 만들고 열기
      PrintWriter outfile = new PrintWriter(writer);
      outfile.println("가을이 오면...");
      outfile.println("학기가 저물고,");
      outfile.println("시험이 끝나면서");
      outfile.println("겨울이 온다.");
      outfile.close(); // 파일 닫기
   }
}
```

직렬 파일에 이어 쓰기

```
public class Test {
   public static void main(String[] args) throws Exception {
      FileWriter writer = new FileWriter("poem.txt", true); // 이어 쓸 용도로 파일 열기
      PrintWriter outfile = new PrintWriter(writer);
      outfile.println("방학이 되면...");
      outfile.println("뭘 할까?");
      outfile.close(); // 파일 닫기
   }
}
```

직렬 파일에서 읽어오기

```
import java.io.*;
import javax.swing.*;

public class Test {
   public static void main(String[] args) throws IOException {
      String file_name = JOptionPane.showInputDialog("읽을 파일 이름을 쓰세요.");
      FileReader reader = new FileReader(file_name); // 읽을 용도로 파일 열기
      BufferedReader infile = new BufferedReader(reader);
      FileWriter writer = new FileWriter(file_name + ".out"); // 쓸 용도로 파일 열기
      PrintWriter outfile = new PrintWriter(writer);
      while (infile.ready()) {
         String s = infile.readLine();
         outfile.println(s);
      }
      outfile.close(); // 파일 모두 닫기
      infile.close();
   }
}
```

표준 입출력 Standard Input/Output

Standard Output	`System.out`
Standard Input	`System.in`

```java
import java.io.*;

public class Test {
    public static void main(String[] args) throws IOException {
        InputStreamReader reader = new InputStreamReader(System.in);
        BufferedReader keyboard = new BufferedReader(reader);
        System.out.println("도 단위를 입력해주세요.");
        String province = keyboard.readLine();
        System.out.println("시 단위를 입력해주세요.");
        String city = keyboard.readLine();
        System.out.println("구 단위를 입력해주세요.");
        String district = keyboard.readLine();
        System.out.print("입력한 주소는 \"");
        System.out.println(province + " " + city + " " + district + "\" 입니다.");
    }
}
```

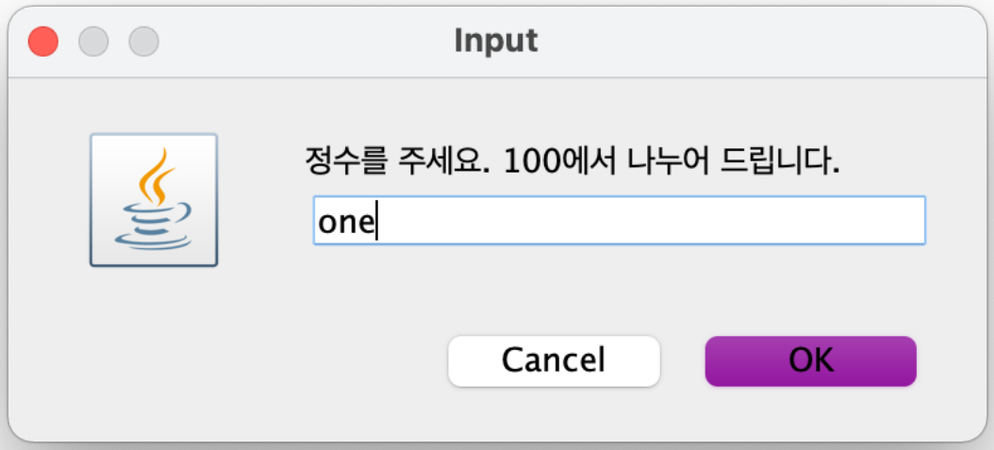
사례 학습

예외 처리

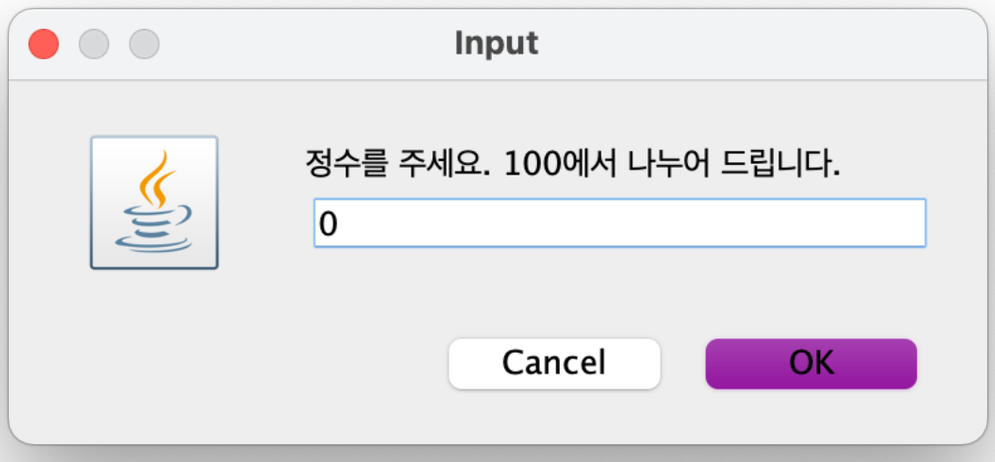
100에서 나누기

```
import javax.swing.*;

public class Test {
    public static void main(String[] args) {
        int n = readInt();
        String answer = 100 + " 나누기 " + Integer.toString(n) + " = " + (100 / n);
        JOptionPane.showMessageDialog(null, answer);
    }

    private static int readInt() {
        String input = JOptionPane.showInputDialog("정수를 주세요. 100에서 나누어 드립니다.");
        int n = Integer.parseInt(input.trim());
        return n;
    }
}
```

실행 중 예외 상황 (= 실행 오류) 발생

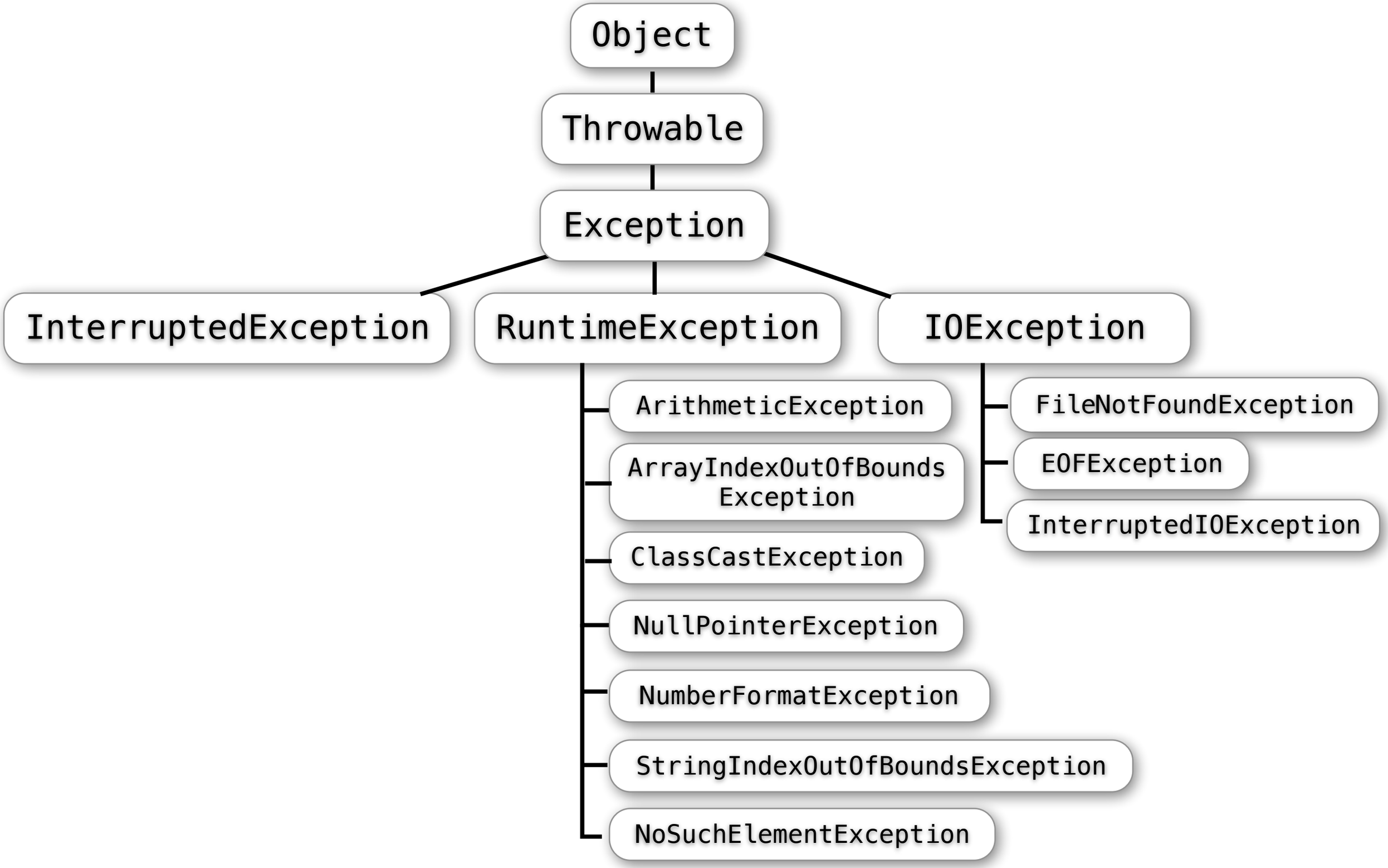



```
Exception in thread "main" java.lang.NumberFormatException: For input string: "one"
        at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:68)
        at java.base/java.lang.Integer.parseInt(Integer.java:652)
        at java.base/java.lang.Integer.parseInt(Integer.java:770)
        at DivideIntoTwelve.readAnInt(DivideIntoTwelve.java:12)
        at DivideIntoTwelve.main(DivideIntoTwelve.java:6)
```




```
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at DivideIntoTwelve.main(DivideIntoTwelve.java:7)
```

Exceptions Are Objects



예외 처리

예외 발생 블록

예외 처리 블록

```
try {

}
catch (Exception e) {

}
```

예외 타입

변수

Exception 객체가 처리 가능한 메소

class	Throwable	기능
methods	String getMessage()	발생한 오류를 설명하는 메시지 리턴
	String toString()	예외 객체의 간판 문자열 리턴 예외 클래스 이름과 오류를 설명하는 메시지를 통상 포함
	paintStackTrace()	오류가 발생한 지점과 이에 도달하기 전의 메소드 호출 지점을 콘솔창에 프린트

NumberFormatException 예외 처리

```
import javax.swing.*;

public class Test {
    public static void main(String[] args) {
        int n = readInt();
        String answer = "100에서 나누기 " + Integer.toString(n) + " = " + (100 / n);
        JOptionPane.showMessageDialog(null, answer);
    }

    private static int readInt() {
        int n;
        String input = JOptionPane.showInputDialog("정수를 주세요. 100에서 나누어 드립니다.");
        try {
            n = Integer.parseInt(input.trim());
        }
        catch (NumberFormatException e){
            JOptionPane.showMessageDialog(null, e.getMessage() + " 정수가 아닙니다.");
            n = readInt(); // 재시도
        }
        return n;
    }
}
```

ArithmeticException 예외 처리

```
import javax.swing.*;

public class Test {
    public static void main(String[] args) {
        int n = readInt();
        try {
            String answer = "100에서 나누기 " + Integer.toString(n) + " = " + (100 / n);
            JOptionPane.showMessageDialog(null, answer);
        }
        catch (ArithmeticException e) {
            JOptionPane.showMessageDialog(null, e.getMessage() + " : 0으로 나눌 수 없습니다.");
        }
    }

    private static int readInt() {
        int n;
        String input = JOptionPane.showInputDialog("정수를 주세요. 100에서 나누어 드립니다.");
        try {
            n = Integer.parseInt(input.trim());
        }
        catch (NumberFormatException e){
            JOptionPane.showMessageDialog(null, e.getMessage() + " 정수가 아닙니다.");
            n = readInt(); // 재시도
        }
        return n;
    }
}
```

예외 처리를 장착한 입력 전용 DialogReader

```
import javax.swing.*;

public class DialogReader {

    public String readString(String prompt) {
        return JOptionPane.showInputDialog(prompt);
    }

    public int readInt(String prompt) {
        int n;
        String input = readString(prompt);
        try {
            n = Integer.parseInt(input.trim());
        }
        catch (NumberFormatException e){
            JOptionPane.showMessageDialog(null, e.getMessage() + " 정수가 아닙니다.");
            n = readInt(prompt);
        }
        return n;
    }

    public double readDouble(String prompt) {
        double n;
        String input = readString(prompt);
        try {
            n = Double.parseDouble(input.trim());
        }
        catch (NumberFormatException e){
            JOptionPane.showMessageDialog(null, e.getMessage() + " 실수가 아닙니다.");
            n = readDouble(prompt); // 재시도
        }
        return n;
    }

}
```

```java
import javax.swing.*;

public class Test {
    public static void main(String[] args) {
        int n = new DialogReader().readInt("정수를 주세요. 100에서 나누어 드립니다.");
        try {
            String answer = "100에서 나누기 " + Integer.toString(n) + " = " + (100 / n);
            JOptionPane.showMessageDialog(null, answer);
        }
        catch (ArithmeticException e) {
            JOptionPane.showMessageDialog(null, e.getMessage() + " : 0으로 나눌 수 없습니다.");
        }
    }
}
```

사례 학습

시급 처리
Payroll

입력 파일

이름,근무시간,시급

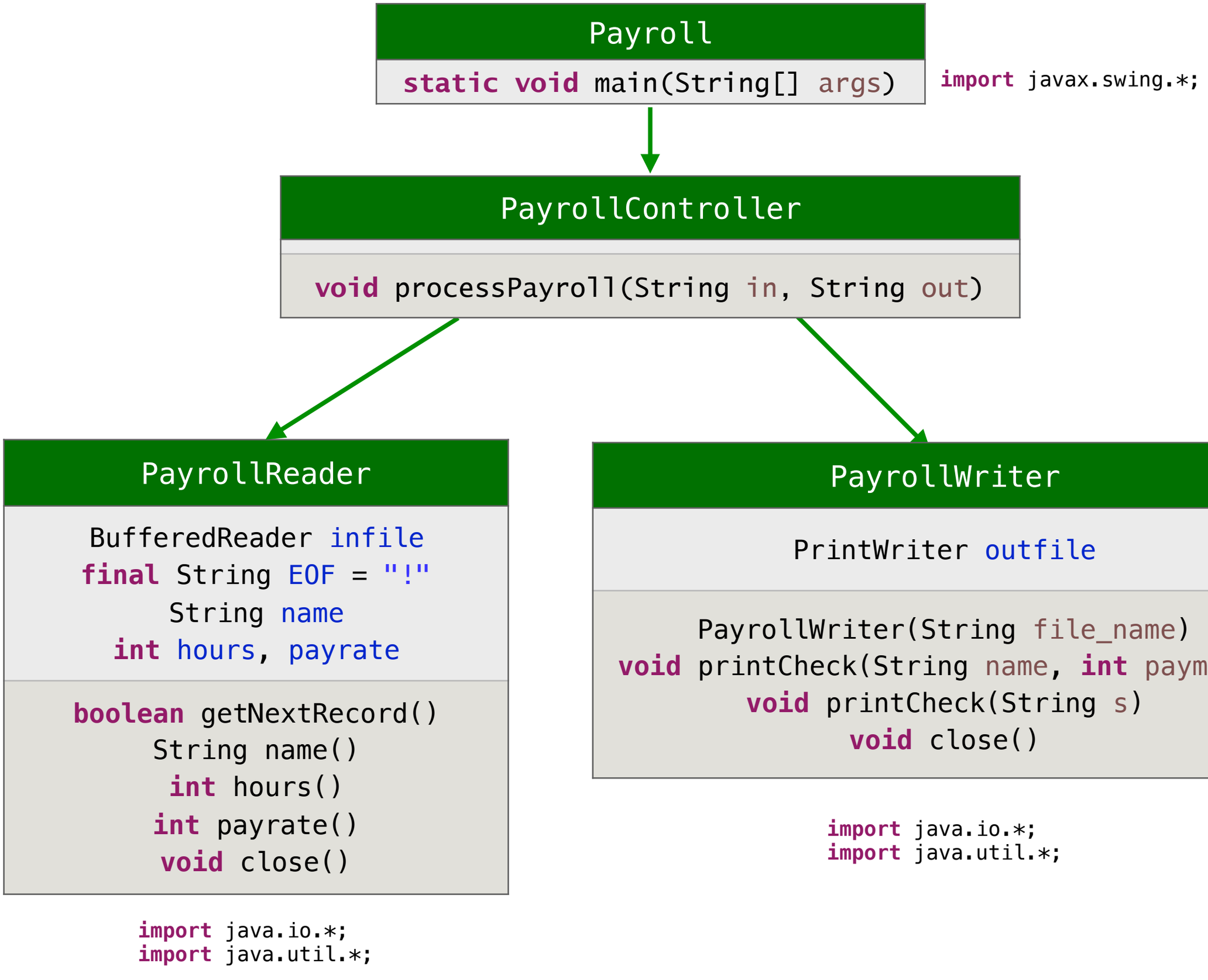
```
도경구,24,8000
양준혁,40,12000
모지환,36,10000
!
```

출력 파일

이름,급여

```
도경구,192000
양준혁,480000
모지환,360000
!
```

Class Diagram



class	PayrollReader	기능
methods	`boolean getNextRecord()`	파일에서 다음 레코드를 읽는다. 제대로 읽었으면 true, 아니면 false를 리턴
	`String name()`	읽은 레코드에서 직원의 이름을 리턴
	`int hours()`	읽은 레코드에서 근무 시간을 리턴
	`payrate()`	읽은 레코드에서 시급을 리턴
	`close()`	파일을 닫음

class	PayrollWriter	기능
methods	`printCheck(String name, int payment)`	파일에 name과 payment를 한 줄에 쓴다.
	`printCheck(String s)`	파일에 s를 쓴다
	`close()`	파일을 닫음