

направо. Это означает, что первый спецификатор формата соответствует первому аргументу, второй спецификатор – второму аргументу и т. д. Однако, используя порядковый номер или индекс аргумента, можно указать явное соответствие спецификатора формата аргументу.

Порядковый номер аргумента указывается за знаком % в спецификаторе формата и имеет следующий формат: **N\$**. Символ **N** обозначает порядковый номер нужного аргумента, нумерация аргументов начинается с единицы.

*/*пример #13: применение порядкового номера аргумента:*

*FormatterDemoArguments.java */*

```
package chapt07;
import java.util.Formatter;

public class FormatterDemoArguments {
    public static void main(String[] args) {
        Formatter f = new Formatter();
        Number n[] = { 4, 2.2, 3, 1.1 };
        f.format("%4$.1f %2$.1f %3$d %1$d", n[0], n[1],
                n[2], n[3]);
        System.out.println(f);
    }
}
```

В результате выполнения этого кода будет выведено:

```
1,1 2,2 3 4
```

Такой же вывод легко получить, используя метод **printf()** в виде:

```
System.out.printf("%4$.1f %2$.1f %3$d %1$d", n[0], n[1],
                n[2], n[3]);
```

Лексический анализ текста

Класс **StringTokenizer** содержит методы, позволяющие разбивать текст на лексемы, отделяемые разделителями. Набор разделителей по умолчанию: пробел, символ табуляции, символ новой строки, перевод каретки. В задаваемой строке разделителей можно указывать другие разделители, например «= ; : ».

Класс **StringTokenizer** имеет конструкторы:

```
StringTokenizer(String str);
StringTokenizer(String str, String delimiters);
StringTokenizer(String str, String delimiters,
                Boolean delimAsToken);
```

Некоторые методы:

```
String nextToken() – возвращает лексему как String объект;
boolean hasMoreTokens() – возвращает true, если одна или несколько
лексем остались в строке;
int countToken() – возвращает число лексем.
```

Класс был реализован в самой первой версии языка. Однако в настоящее время существуют более совершенные средства по обработке текстовой информации – регулярные выражения.