

## Code Exercise

The text file `in.properties` (pairs key=value) contains some sequence of lines.

Let the element with key `index $i$`  has the value  $j$  ( $i=1..n$ ,  $j=1..m$  – natural numbers without leading zeros). It determines the key element with a name `value $ij$`  (`index $i$  =  $j$` ) you should work with.

Calculate:

1. the sum of values of elements that have the key `value $ij$`  (see examples below).

2. the number of index elements with “errors” ( $i, j$  – not natural numbers; no correct value-element for correct index element).

The example of the file `in1.properties`:      Results:

<code>index1 = 3</code>	<code>sum = 8.24</code>
<code>value11=qw</code>	<code>error-lines = 3</code>
<code>value12=4</code>	
<code>value13 = 5.1</code>	
<code>value14=2.7</code>	
<code>index2=15</code>	
<code>value21=</code>	
<code>value22=</code>	
<code>value23=</code>	
<code>value24=k</code>	
<code>value25=5</code>	
<code>index3=1</code>	
<code>value31= 3.14</code>	
<code>value32=fgh</code>	
<code>value33=5</code>	
<code>index4=0</code>	
<code>value41=</code>	
<code>value42=e1</code>	
<code>value43=2</code>	
<code>value44=3</code>	
<code>index5=b</code>	
<code>value51=d</code>	
<code>value52=e</code>	

The example of the file `in2.properties`:

Results:

<code>value23= 5</code>	<code>sum = 30.242</code>
<code>index2=3</code>	<code>error-lines = 9</code>
<code>qqindex7=3</code>	
<code>value73=-10</code>	

[illegible]

Define the `TestRunner` class in the **default** package with unit tests for the static method (see previous task).

## Замечания и ограничения к задаче 2

- Не использовать коллекции.
- Информационных классов не создавать. Реализовать в статическом методе (сигнатура другая!) тестового раннера, т.е. в форме скрипта. Структура класса `TestRunner` должна быть та же, как в задаче `strings1`.
- Анализировать (проверять на соответствие) ключи и значения, выделять подстроки с помощью регулярного выражения, т.е. через функционал класса `Matcher` (см. И. Блинов, стр. 177–181). Из строковых операций разрешены только присваивание `=` и сцепление `+`.

– Properties–файл читать с помощью класса ResourceBundle (см. И. Блинов, стр. 181–185). Метод `getKeys()` позволяет прочитать все ключи в объект `Enumeration<String>`. Далее каждый ключ анализируется. Схема такая.

```
ResourceBundle rb = ResourceBundle.getBundle("in");
```

```
Enumeration<String> keys = rb.getKeys();
```

```
String key;
```

```
while(keys.hasMoreElements()) {
```

```
    key = keys.nextElement();
```

```
    ...
```

```
}
```

– Обратите внимание, что порядок ключей в экземпляре `Enumeration<String>` не совпадает с порядком ключей в исходном файле.

– Прямой доступ к значению по ключу `key` обеспечивается вызовом метода `getString(key)` класса `ResourceBundle`.

– При чтении ключа как лидирующие, так и завершающие пробельные символы автоматически отсекаются. Такой ключ также должен быть обработан.

– При чтении значения `valueij` по ключу `indexi` началом значения `valueij` является первый непробельный символ. Следовательно, в начале значения `valueij` не могут оказаться пробелы. С другой стороны, завершающие пробельные символы являются частью значения `valueij`. Они должны отсекаются с помощью метода `trim()`.

– Правильное значение по ключу `valueij` должно быть вещественным числом, записанным в любой форме. Его не надо прогонять через регулярное выражение. Если строка преобразуется через оболочечный класс `Double` к числу, то суммируем, иначе в блоке `catch` увеличиваем счетчик ошибок.