实现思路：

作业要求对 yaml 的一个子集有良好的解析效果，并且能生成对应的 json 文件，还支持查询操作。

词法分析：

对于 yamlite，整体可分为：键值对与数组，而键值对与数组的值既可是基本类型，也可是键值对、数组（以递归实现）。对应于 json 类型，即要将其对应成 json 对象或 json 数组。对于读进的文档，按行读取并获得行号后，对每一行进行解析，并生成对应的 token 流，并保存在一个 ArrayList 中。token 类型包含了题中要求的基本类型（用正则表达式并运用部分字符串解析方法可判断基本类型），并定义出 identifier、array 等类型。（当 token 为 identifier 时，被认为将读进一个键，为 Array 时读进一个数组），token 流中包含了类型、名称、缩进层次以及是否是数组元素等信息。这些信息将用于转 json 时的文法分析。

文法分析：

对于转 json 类型，首先定义一个关于 json 值类型的统一接口，所有 json 对象及数组、数组元素、键值对都实现该接口，接口中定义的类型与 token 定义的类型相关，即可实现由 yaml token 向 json 的转换。实现两个子类 JsonObject 与 JsonArray，分别对应于 json 对象与数组，并给出了 json 的返回格式。其中 JsonArray 实现了一个 Arraylist，用于保存该数组元素的 json 类型与值，JsonObject 实现了一个 LinkedHashMap，用于保存键值对与数组对象。JsonElement 对应于 Json 值。规定 JsonFormat 为 Json 格式的对象，分析过程即由 token 的 arraylist 向 JsonFormat 转换的过程。Parse 类中定义了两个方法，parseObject 与

parseArray，分别对应于扫描到两种 json 类型的 token 时的处理过程，并且通过 token 的 layer 属性实现缩进层级与递归调用。

有关查询：

在查询中，使用已完成文法分析的 JsonFormat 作为查询对象，并使用递归的思想实现多层次查询。查询的返回对象同样分为 Array 和 Object 两种类型，最终均转换为 JsonValue，并以 Json 格式字符串输出。对传进的 path 参数解析后，即可获得每次查询的键与数组名，并递进查询直到 path 分解完毕。

运行截图：

检查合法：

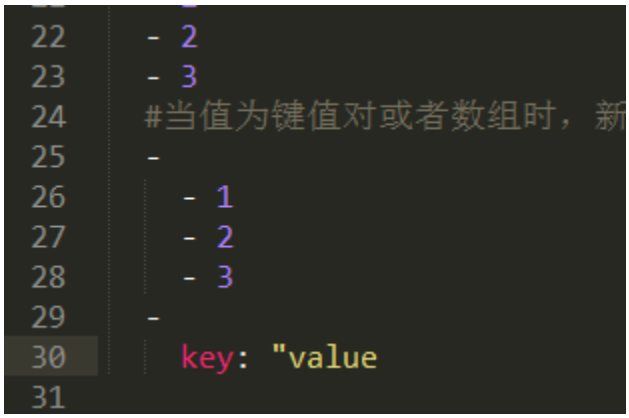
```
#字符串
string: "value2"
#整数
int: 2333
#浮点数
float: 2333.33
#科学计数法
scientific_notation: 1.234e-10
#布尔值
bool: true

#当值为键值对或者数组时，新增一行，缩进增加两个空格
#数组中的每个元素值跟随在“-”和一个空格后面
array:
  - 1
  - 2
  - 3
  #当值为键值对或者数组时，新增一行，缩进增加两个空格
  -
    - 1
    - 2
    - 3
  -
    key: "value"
```

运行命令并输出：

```
E:\Woo\Homework\yamlTestTwo\out\artifacts\yamlTestTwo_jar>java -jar yamlTestTwo.jar E:\Woo\Homework\sample.yml
valid
```

检查不合法：



运行命令并输出：

```
E:\Woo\Homework\yamlTestTwo\out\artifacts\yamlTestTwo_jar>java -jar yamlTestTwo.jar E:\Woo\Homework\sample.yml
line:30, position15: expect <">
```

转成 Json：

```
E:\Woo\Homework\yamlTestTwo\out\artifacts\yamlTestTwo_jar>java -jar yamlTestTwo.jar -json E:\Woo\Homework\sample.yml
The file: E:\Woo\Homework\sample.json had been created.
```

生成的 Json 文件：

Find 查询(包含：数组，键值对，数组嵌套数组，数组嵌套键值对)：

```
E:\Woo\Homework\yamlTestTwo\out\artifacts\yamlTestTwo_jar>java -jar yamlTestTwo.jar -find array[4].key E:\Woo\Homework\sample.yml
"value"

E:\Woo\Homework\yamlTestTwo\out\artifacts\yamlTestTwo_jar>java -jar yamlTestTwo.jar -find array[3] E:\Woo\Homework\sample.yml
[
  1,
  2,
  3
]

E:\Woo\Homework\yamlTestTwo\out\artifacts\yamlTestTwo_jar>java -jar yamlTestTwo.jar -find array[3][0] E:\Woo\Homework\sample.yml
1

E:\Woo\Homework\yamlTestTwo\out\artifacts\yamlTestTwo_jar>java -jar yamlTestTwo.jar -find int E:\Woo\Homework\sample.yml
2333

E:\Woo\Homework\yamlTestTwo\out\artifacts\yamlTestTwo_jar>
```