

C++ INI 解析器实现

介绍

涉及知识点

- 面向对象设计
- 构造函数重载
- 函数重载
- 运算符重载
- 标准模板库

常用配置文件有哪些类型？

1. INI
2. XML
3. JSON
4. YAML

INI

```
1 [server]
2 ip = 127.0.0.1
3 port = 80
4
5 [profile]
6 name = jack
7 gender = male
8 age = 30
```

优点：非常简单、书写方便

缺点：不能表达复杂的数据格式

应用场景：配置文件

XML

```
1 <?xml version="1.0"?>
2 <root>
3     <server>
4         <ip>127.0.0.1</ip>
5         <port>80</port>
6     </server>
```

```
7     <profile>
8         <name>jack</name>
9         <gender>male</gender>
10        <age>30</age>
11    </profile>
12 </root>
```

优点：可以表达复杂的数据格式

缺点：复杂、书写麻烦

应用场景：配置文件，数据传输

JSON

```
1 {
2     "server": {
3         "ip": "127.0.0.1",
4         "port": 80
5     },
6     "profile": {
7         "name": "jack",
8         "gender": "male",
9         "age": 30
10    }
11 }
```

优点：简单、可以表达复杂的数据格式

缺点：书写麻烦

应用场景：配置文件、数据传输

YAML

```
1 server:
2     ip: 127.0.0.1
3     port: 80
4
5 profile:
6     name: jack
7     gender: male
8     age: 30
```

优点：简单、直观、书写方便、可以表达复杂的数据格式

缺点：它使用空白符号缩进，使用场景受限，无法进行数据传输

运用场景：配置文件

为什么要用 INI 配置？

1. 非常简单、书写方便
2. 运用广泛：mysql、redis、php的配置等

INI 格式

INI文件的格式很简单，最基本的三个要素是：sections、parameters、comments

```
1 [section]
2 key = value
3 # 这是一条注释
```

实例：

main.ini

```
1 [server]
2 ip = 127.0.0.1
3 port = 80
4
5 [profile]
6 name = jack
7 gender = male
8 age = 30
```