**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**

|  |
| --- |
| Properties [server] ip = 127.0.0.1 port = 80  [profile] name = jack gender = male 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**

|  |
| --- |
| XML <?xml version="1.0"?> <root>  <server>  <ip>127.0.0.1</ip>  <port>80</port>  </server>  <profile>  <name>jack</name>  <gender>male</gender>  <age>30</age>  </profile> </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**

|  |
| --- |
| YAML {  "server": {  "ip": "127.0.0.1",  "port": 80  },  "profile": {  "name": "jack",  "gender": "male",  "age": 30  } } |

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**

|  |
| --- |
| YAML server:  ip: 127.0.0.1  port: 80  profile:  name: jack  gender: male  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

|  |
| --- |
| Properties [section] key = value # 这是一条注释 |

实例：

main.ini

|  |
| --- |
| Properties [server] ip = 127.0.0.1 port = 80  [profile] name = jack gender = male age = 30 |