|  |  |
| --- | --- |
| 1. 正则表达式 2. 使用正则表达式库 3. 指定regex对象的选项 4. 指定或使用正则表达式时的错误 5. 正则表达式类和输入序列类型 | Regex类，函数regex\_match,regex\_search  Smatch容器类，附加参数保存信息  String pattern(“[^c]ei”);  Pattern = ….  Regex r(pattern);  Smatch results;  String test\_str = “sdfasdf”;  If(regex\_search(test\_str,results,r))  3.regex r(“…”,regex::icase)  Smatch…  4.try{  Regex r(“sdfasdf”);  }catch(regex\_error e){  Cout<<e.what(),e.code()  }   1. cmatch results;   regex\_search(“sdfsad.cc”,results,r); |
| 1. 正则表达式 2. 是一种描述字符序列的方法。正则表达式库（RE库） 3. Regex类表示一个正则表达式，表17.6 4. 函数regex\_match和regex\_search确定一个给定字符序列与一个给定regex是否匹配   Regex\_match判断整个输入序列，返回bool，regex\_search判断输入序列的子串，返回一个bool，还有函数regex\_replace   1. 上述的函数都重载了，其中一个版本接受一个类型为smatch的附加参数 2. 使用正则表达式库 3. pattern = “[[:alpha:]]\*[^c]ei[[:alpha:]]\*” regex r(pattern)，regex使用的正则表达式语言是ECMAScript 4. regex\_search 找到一个匹配子串就停止查找 5. 指定regex对象的选项 6. 指定一些标志来影响regex如何操作,表17.6   Regex r(”sdfa”,regex::icase);   1. 正则表达式有特殊字符。字符点（.）通常匹配任意字符，放置一个反斜线去除特殊含义.但是反斜线是c++的特殊字符，连续使用两个反斜线告诉c+我们想要一个普通的反斜线字符。\\. 2. 指定或使用正则表达式时的错误 3. 可以把正则表达式看成简单程序设计语言编写的“程序”。不是由c+编译器解释，正则表达式运行时，当一个regex对象被初始化或被赋予一个新模式时，才被“编译的”。也有可能出错。 4. 用try cath 包围regex   Note：正则表达式的语法是否正确是在运行时解析的   1. 会抛出regex\_error，有what(),code()对应数值编码。表17.7第一个对应下标0 2. Note：正则表达式是在运行时编译的。非常慢的的操作，构造一个regex对象操作一个regex对象都非常耗时的。避免创建很多不必要的regex。在循环外使用 3. 正则表达式类和输入序列类型 4. Regex r(pattern);   Smatch results;  If(regex\_search(“sfsdf”\_str,results,r))  其中regex\_search的第一个参数是字符数组输入序列，则不能使用smatch对象，得使用cmatch。   1. 表17.8对应关系。   简要：smatch表示string类型输入序列  Cmatch表示字符数组序列  Wsmatch表示宽字符串  Wcmatch表示宽字符数组  并且不同的序列类型，对迭代器类型也有所差异。 | |









