

整数表示格式分析

整数字符串的有效格式必须满足以下条件：

1. 不能为空字符串
2. 可以有一个可选的符号（+或-），但只能在字符串开头
3. 除了开头的符号外，其余字符必须全部是数字（0-9）
4. 表示的数值必须在 int 类型范围内（-2147483648 到 2147483647）

不符合整数格式的原因与测试案例

1. 空字符串

- 案例: ""
- 抛出: EmptyStringException

2. 包含非法字符

- 案例: "abc", "123a"
- 抛出: InvalidCharacterException

3. 只有符号

- 案例: "+", "-"
- 抛出: OnlySignException

4. 多个符号或符号位置不正确

- 案例: "++123", "-+123", "123+"
- 抛出: MultipleSignsException 或 InvalidSignPositionException

5. 数值溢出

- 案例: "1234567890123456789", "-1234567890123456789"
- 抛出: OverflowException

测试报告

程序包含两种测试方式：

1. 预定义测试用例的自动测试
2. 交互式测试（用户输入）

运行测试后，每个测试用例会显示以下信息：

- 测试的字符串
- 测试用例的描述
- 测试结果（有效整数值或异常信息）

这种设计能够覆盖所有可能的错误情况，并根据不同的错误原因抛出相应的异常，满足了面向对象的设计需求。

程序实际运行结果

```

PS C:\Files\OOP\HW12\build> .\HW12.exe
=== 测试用例 ===
测试: "123" (有效整数) --> 有效整数: 123
测试: "-456" (有效负整数) --> 有效整数: -456
测试: "+789" (有效带正号整数) --> 有效整数: 789
测试: "" (空字符串) --> 异常: 空字符串不能转换为整数
测试: "abc" (包含非法字符) --> 异常: 字符串中包含非法字符: a
测试: "123a" (数字后有非法字符) --> 异常: 字符串中包含非法字符: a
测试: "+" (只有符号) --> 异常: 字符串只包含符号, 没有数字
测试: "-" (只有符号) --> 异常: 字符串只包含符号, 没有数字
测试: "abc" (包含非法字符) --> 异常: 字符串中包含非法字符: a
测试: "123a" (数字后有非法字符) --> 异常: 字符串中包含非法字符: a
测试: "+" (只有符号) --> 异常: 字符串只包含符号, 没有数字
测试: "-" (只有符号) --> 异常: 字符串只包含符号, 没有数字
测试: "+" (只有符号) --> 异常: 字符串只包含符号, 没有数字
测试: "-" (只有符号) --> 异常: 字符串只包含符号, 没有数字
测试: "-" (只有符号) --> 异常: 字符串只包含符号, 没有数字
测试: "++123" (多个符号) --> 异常: 符号位置不正确
测试: "++123" (多个符号) --> 异常: 符号位置不正确
测试: "--123" (多个符号) --> 异常: 符号位置不正确
测试: "--123" (多个符号) --> 异常: 符号位置不正确
测试: "123+" (符号位置不正确) --> 异常: 符号位置不正确
测试: "1234567890123456789" (整数溢出) --> 异常: 整数值超出表示范围

=== 交互式测试 ===
=== 交互式测试 ===
请输入要解析的字符串 (输入'exit'退出): exit
  
```