

OOP上机实验5(e)

(Version: 0.74 Date: 2025/4/24 Author: wangxp@fudan.edu.cn)

1 实验目标

通过本实验，拟达到如下目标

- 编写完整的 C++ 类，熟练掌握类的构造函数、析构函数和拷贝构造函数。
- 掌握操作符重载，包括输入输出操作符。
- 理解成员函数与非成员函数的区别，以及友元函数的使用。
- 熟悉类的静态属性和静态成员函数。
- 理解完整类代码的组成，包括成员属性、成员函数、静态属性、静态成员函数，以及非成员函数（特别是输入输出重载操作符、不能作为成员函数的双目或多目操作符、关系操作符）。
- 掌握基于测试驱动的程序开发流程。

2 实验内容

2.0 背景

在 C++ 中，最大的整数类型为 unsigned long long，其表示范围因平台不同而异。然而，在许多场景中，需要支持任意长度整数或指定长度整数（例如 512 位或 2048 位整数）。

2.1 目标

设计并实现一个支持任意长度整数的类 Bignum，支持加法、减法、乘法和除法运算。实验目标：设计一个支持任意长度的整数表示，而且需要支持加法、减法、乘法和除法。

本次实验，提供给大家的文件包括

源文件：bignum.h [bignum.cc](#) bignum_test.cc minitest.h
cmake文件：CMakeLists.txt

2.1.1 bignum.h和bignum.cc文件

bignum.h文件定义类Bignum的接口，包括：

- 构造函数
- 正负号、数值和长度相关操作
- 输入输出操作符重载
- 双目运算：+、-、*、/

- 用户定义字面量[1]，用于创建大整数

bignum.cc文件：

- 实现类Bignum的接口；
- 辅助函数。

类Bignum的内部表示：

- sign_：char类型，记录符号，0表示正，1表示负；
- value_：std::string类型，记录数值部分。例如，数值+12345678表示为
 - sign_ = 0
 - value_ = "12345678"

2.1.2 minitest.h

minitest.h 提供了一个简化的测试框架（类似 Google Test[2]），包括：

- TestCase 结构体和全局变量 test_cases（类型为 static inline），用于记录测试函数。
- 测试函数统一为 void() 类型。
- 宏 TEST(PREFIX, name)：定义并自动注册测试函数。
- 宏 ASSERT(condition)：检查条件是否为真，失败则抛出异常。
- 宏 ASSERT_EQ(expected, actual)：检查 expected 和 actual 是否相等。
- 异常类 TestFailure：用于处理测试失败。
- 宏 TEST_MAIN：生成测试程序的 main 函数，自动运行所有注册的测试函数并报告结果。

minitest.h 无需修改。

2.1.3 bignum_test.cc

包含所有测试用例。每个测试用例使用 TEST 宏定义，例如：

```
TEST(Bignum, assign_normal)
{
    Bignum num;
    num.assign("123");

    ASSERT_EQ(num.sign(), 0);
    ASSERT_EQ(num.value(), "123");
}
```

测试函数通过 TEST 宏注册，并在函数体内设置测试条件后使用 ASSERT_EQ 检查结果。

2.1.4 要求

1. 检查和完善bignum.h

- 确保函数声明的正确性，例如参数传递方式（值传递、引用传递）、是否需要 `const` 修饰等。
- 在修改处添加简洁注释说明理由。

2. 完善bignum.cc

- 与 `bignum.h` 的修改保持一致。
- 补充代码实现（共 5 处，标有 `// TODO:`）。

3. 补充 bignum_test.cc

- 添加必要的测试用例，确保覆盖所有功能。
- 提示：可使用 Python 生成测试数据，并将结果写入测试用例。

4. 效率分析与优化建议

- 分析当前实现的效率，提出可能的优化方法（例如改进表示形式或算法）；
- 描述优化方法的实现难点；
- 说明如何通过测试比较当前方法与优化方法的性能；
- 该部分无需提交代码，可写在单独文档中并上传。

2.2 提交要求

提交下面的文件：**提交bignum.h、bignum.cc和bignum_test.cc文件**。提交前，需确保所有程序能够正确编译和运行。

3 参考资料

[1] 用户定义字面量。 https://en.cppreference.com/w/cpp/language/user_literal

[2] GTest。Google测试框架。 <https://github.com/google/googletest>

4 实验要求

4.1 实验提交要求

- 如果有源代码提交的要求，请不要改变源代码原来的名字；
- 如果有截屏的要求，把截屏的内容放在PDF文件中(文件名为：学号_实验编号.pdf)，并在前面写明实验内容的编号和源代码名称；
- 把所有要提交的源代码和PDF文档放在同一个文件夹内，把该文件夹压缩；
- 压缩文件支持zip或rar两种形式。

4.2 压缩文件命名

压缩文件名为：学号_姓名_实验编号.rar 或学号_姓名_实验编号.zip。例如第1次实验，学号为19307130022,姓名为张其的学生，提交的压缩文件名为 22307130022_张其_1.rar或22307130022_张其_1.zip)。请大家务必按照命名规则来正确命名文件的名称，否则可能导致评分无法进行，影响实验评分!

请注意：提交的文件不包括*.obj和*.exe，只需要包含*.cpp 和*.h 文件即可。如果有说明，请把说明包含在文本文件中，例如代码demo.cpp的说明文件名为demo.cpp.txt。

4.3 截至时间

截至时间通常为下周五晚上12点前，请大家安排时间准时完成作业。