目标：

编写C++程序来达到：接收输入的正整数n并且用至少两种方法计算不超过n的所有正偶数之和。在此过程中学习好的程序编写模式。

原理：

编写头文件和源文件，用面向对象的方法编程，通过对软件构件库代码、构件库测试代码、构件库应用代码的编写实现程序的目标并且进行测试。

**如何编程和运行：**

* 软件构件库部分：

在头文件中定义类和类的构造、析构函数，声明outStart函数，定义变量。

在源文件中include头文件并且对头文件中声明的outStart函数编写代码：首先读入数据（正整数n）同时进行判断是否满足正整数的条件，如果不满足或者读入数据失败则让用户再次输入。在其中有两种方法来计算不超过n的正偶数之和，第一种方法是累加，第二种方法是用数学方法计算出公式后利用公式计算。

在定义变量的时候，正整数n使用int型，而为了使答案正常显示，最后算出的结果用unsigned long long型。

在编写代码时，为了过滤掉错误输入进行清空缓冲区的操作，以免错误输入造成干扰。

* 构件库测试部分：

编写头文件，声明测试函数。

在源文件中对测试函数的代码进行编写。首先构造类的对象，之后调用软件构件库内的函数进行计算操作。

* 构件库应用部分：

在该部分的源程序中编写main函数，其中调用构件库测试部分中的函数。在这部分中为了实现循环测试编写了部分代码（多次调用测试代码达到循环测试的效果）。

结果：

程序运行结果：

|  |  |  |  |
| --- | --- | --- | --- |
| 输入 | 输出n | 第一种方法 | 第二种方法 |
| 1 | 1 | 0 | 0 |
| 2 | 2 | 2 | 2 |
| 8 | 8 | 20 | 20 |
| 15 | 15 | 56 | 56 |
| 925 | 925 | 213906 | 213906 |
| 1589 | 1589 | 631230 | 631230 |
| 898652 | 898652 | 201894303602 | 201894303602 |
| 2147483647 | 2147483647 | 1152921503533105152 | 1152921503533105152 |
| 2147483648 | 0 | 未显示，要求再输入 | 未显示，要求再输入 |
| 2147483895 | 0 | 未显示，要求再输入 | 未显示，要求再输入 |
| -1 | -1 | 未显示，要求再输入 | 未显示，要求再输入 |
| -596856 | -596856 | 未显示，要求再输入 | 未显示，要求再输入 |
| -2147483648 | -2147483648 | 未显示，要求再输入 | 未显示，要求再输入 |
| -2147483688 | 0 | 未显示，要求再输入 | 未显示，要求再输入 |
| -5896598545 | 0 | 未显示，要求再输入 | 未显示，要求再输入 |
| 123dsf | 123 | 3782 | 3782 |
| Joijefwo123 | 0 | 未显示，要求再输入 | 未显示，要求再输入 |

程序验证结果报告：

* 对于64位的计算机

当输入从1到2147483647的整数时，都可以正确输出相应的整数及运算结果。

当输入的整数超出此范围或者输入非数字开头的其他字符，则输出的n均为0，即n的值没有发生变化，同时要求用户再次输入。

当输入内容以此范围内的整数开头时，程序自动读取输入内容开始的整数并且进行计算。

* 当输入的整数超出范围或输入的不是整数时，上面程序会要求再次输入正确数据。

两种方法优缺点：

* 第一种计算方法（直接法，根据题目描述进行累加操作）：

优点：简单直接，便于编写，不易出错。

缺点：运行速度慢，运行时间长。

* 第二种计算方法（先用数学方法推出计算公式，再编写程序）：

优点：运行速度快。

缺点：出错可能性大。

结论：

该C++程序运行没有问题，这种编写模式结构清晰，便于重用，体现了面向对象编程的特点。