

面向对象程序设计基础 第一次作业

徐浩博 2020010108

模型部分

为了解决此问题，我共建立了两种模型以计算出结果。

□ 计算方法 1

利用等差数列求和公式。设读入的正整数为 x 。先考虑 $x \geq 2$ 的情况，则不大于 x 的最大偶数为 $(x/2)*2$ ，从 2 至 $(x/2)*2$ 共有 $(x/2)$ 个偶数，采用等差数列的求和公式 $sum = \text{项数} * (\text{首项} + \text{末项}) / 2 = (x/2) * (2 + (x/2)*2) / 2$ 。特别地，若 $x=1$ ，则 $(x/2)=0$ ，恰也满足此公式。
综上： $sum = \text{项数} * (\text{首项} + \text{末项}) / 2 = (x/2) * (2 + (x/2)*2) / 2$ 。

□ 计算方法 2

设读入的正整数为 x 。利用循环，从 2 开始，枚举每一个小于等于 x 的正偶数。将如上的正偶数加和，即得答案。

验证部分

等价类划分

- 输入的不是纯数字: a, 1+2
- 输入的不是正整数: 1.2, 0, -1
- 输入的正整数超过了 int 范围($[2147483648, +\infty)$)
- 输入的正整数虽未超 int 范围, 但计算出的结果超过了 int 范围($[92682, 2147483648]$)
输入的是计算结果未超过 int 范围的正整数 ($[1, 92681]$)

案例选取

- 输入的不是纯数字: a, 1+2
- 输入的不是正整数: 1.2, 0, -1
- 输入的正整数超过了 int 范围: 2147483648
- 输入的正整数虽然未超 int 范围, 但计算出的结果超过了 int 范围: 10000000
- 输入的是计算结果未超过 int 范围的正整数: 1, 10, 19, 100000

测试结果

输入	第一种计算方法结果	第二种计算方法结果
a	输入非法	输入非法
1+2	输入非法	输入非法
1.2	输入非法	输入非法
0	输入非法	输入非法
-1	输入非法	输入非法
2147483648	输入超过计算范围	输入超过计算范围
10000000	结果超过计算范围	结果超过计算范围
1	0	0
10	30	30
19	90	90
10000	25005000	25005000

分析部分

- ❑ 当输入的不是正整数时，程序能够给出错误提示。当输入的数字因超过 `int` 数据范围而无法计算出结果时，程序也能够给出错误提示。当输入合适的数字时，程序用两种方法运算出的答案均一致且正确。
- ❑ 第一种计算方法时间复杂度为 $O(1)$ ，第二种计算方法时间复杂度为 $O(n)$ ，相较而言第一种计算方法时间开销较小，但由于能够计算出结果的数据范围较小，因此直观上，两种计算方法没有产生显著的速度差异。