

# 面向对象程序设计基础作业一 设计文档

## 1. 模型部分

### a. 功能简述

本程序实现了如下功能：从控制台输入任一字符串，程序将进行合法性判断，若为一整数则输出小于等于该整数的所有正偶数之和。该和采用两种不同方法计算（见 1.b）。

```
[(base) casorazitara@Chens-MacBook-Pro cmake-build-debug % ./HW1 ]
Input a number: 400000
Using mathematical algorithm, the answer is: 40000200000
Using iterative algorithm, the answer is: 40000200000

(base) casorazitara@Chens-MacBook-Pro cmake-build-debug % ./HW1
Input a number: -50
Using mathematical algorithm, the answer is: 0
Using iterative algorithm, the answer is: 0

(base) casorazitara@Chens-MacBook-Pro cmake-build-debug % ./HW1 ]
Input a number: foobar
Input a number: 3.1415926
Input a number: -
Input a number: 100
Using mathematical algorithm, the answer is: 2550
Using iterative algorithm, the answer is: 2550
```

图 1 使用示例（从上至下分别展示正整数输入、负整数输入、非法数据输入）

同时，本程序自带交叉验证功能。从命令行执行本程序时，可以通过在指令后添加：-t [正整数]，程序将验证小于等于该正整数的整数是否计算正确（见 2.）。

### b. 算法

本程序采用两种方法计算所期望的和，称为“公式法”和“循环法”。本部分的讨论过程假设输入的数是合法的正数  $n$ 。设： $\text{temp} = \left\lfloor \frac{n}{2} \right\rfloor$ （在程序中这一步由  $\text{temp} = n \gg 2$  实现）。

公式法，即利用数学公式：

$$\text{Sum} = \frac{(a_1 + a_n) \times n}{2}$$

在本题中， $a_i = 2 \times (i - 1)$ ，故  $Sum = \frac{temp \times (temp - 1)}{2}$ 。计算此公式即得结果。本法的时间复杂度为  $O(1)$ 。

循环法，即利用定义式：

$$Sum = \sum_{k=1}^n a_k = \sum_{k=1}^{temp} 2 \times k$$

利用循环累加模拟此过程即得结果。本法的时间复杂度为  $O(n)$ 。

从时间复杂度上考虑，公式法相比循环法的优势是明显的，尤其在  $n$  较大 ( $>1e8$ ) 时，公式法的计算时间几乎可以忽略不计，而循环法的计算时间变得肉眼可见。但由于循环法的本质是模拟法，模拟的过程相比公式在一些层次上更加可靠。

### c. 数据结构

本程序包括主函数源文件 (main.cpp) 和 1 个额外源文件及其头文件 (CPP\_getint.cpp/h)。在 CPP\_getint.h 中定义了类 customdata::cpp\_int，暂时包括一个存放整数值的私有成员变量和部分函数接口 (见图 2)，并在对应源文件中进行了实现。

```
#ifndef HW1_CPP_GETINT_H
#define HW1_CPP_GETINT_H

namespace customdata
{
    class cpp_int
    {
    private:
        int value = 0;
    public:
        short set_num(long long to_change);
        cpp_int() = default;
        ~cpp_int() = default;
        int get_value() const;
        bool read_num();
        long long calculate_positive_smaller_even_num(int method_type = 0) const;
    };
}

void test(long long upper_limit);
#endif //HW1_CPP_GETINT_H
```

图 2 CPP\_getint.h

## 2. 验证部分

本程序从命令行执行时可使用自动验证程序。

从控制台调用此程序时，若读入了`-t [正整数]`指令，程序将进行交叉比对。比对的过程是：从 0 开始，计算两种方法得到的结果并比较是否一致，若一致则以 1073741824 为初始步长检查下一数字；若不一致，或者此时将要验证的数字已经大于读入正整数，则回退一个步长并减半步长，直到步长为 0，就输出验证到的最大数字。这一验证过程将调用约 $\log_2 n$ 数量级的计算过程，每个计算过程的时间复杂度均为  $O(n)$ ，因此整体的验证时间复杂度大约为  $O(n \log n)$ 。

```
[(base) casorazitara@Chens-MacBook-Pro cmake-build-debug % ./HW1 -t 10000 ]
<-----This part is for verification only----->
All answers are correct up to 10000
<-----Verification ended----->
Input a number: 10000
Using mathematical algorithm, the answer is: 25005000
Using iterative algorithm, the answer is: 25005000
```

图 3 验证过程示例

这一验证方法并非逐一验证：我们假设，越大的数字越容易出现错误（最常见的是由于数据规模溢出），且如果验证了 A 与 B 均正确，则假设 A、B 间的数都正确。同时，由于我们只验证了两种算法是否一致，无法完全排除算法本身的瑕疵。