面向对象程序设计基础作业三 设计文档

1. 模型部分
   1. 功能简述

本程序实现了如下功能：在程序提示后，用户可以输入一行数据，数据可以是：一个整数和一个浮点数，其中整数是1-5之间的正整数；或单独的一个0。如果用户输入的数据非法，则会提示：“Error in input!!”并要求用户重新输入。若输入的第一个整数非零，则依照下列对应关系（1:正三角形，2:正方形，3:正五边形，4:正六边形，5:圆），以第二个浮点数为边长构造对应的正多边形[[1]](#footnote-1)，并进行存储。若输入为零，则输出所有图形的总周长和总面积。

* 1. 算法

本程序使用的算法为模拟法。

周长计算遵循以下公式：

其中，a表示边长，n表示边数；在圆形的情况下，n=2π。

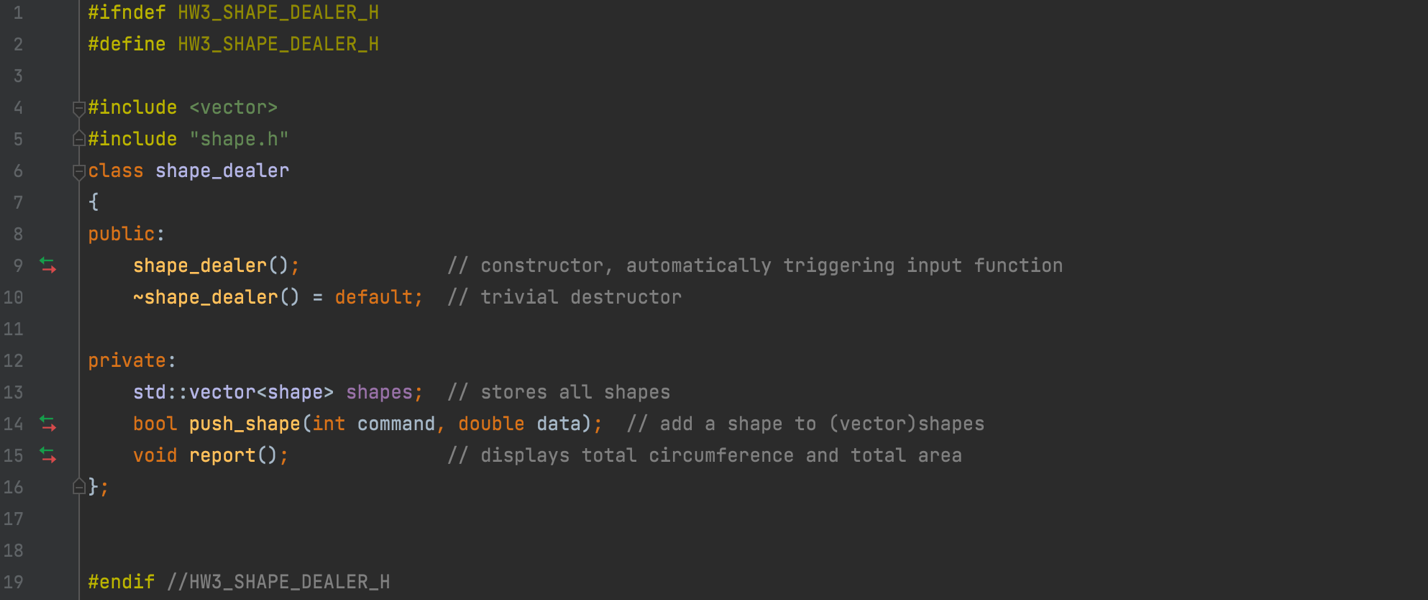
面积计算遵循以下公式：

其中，a表示边长，λ与边数有关，见下表：

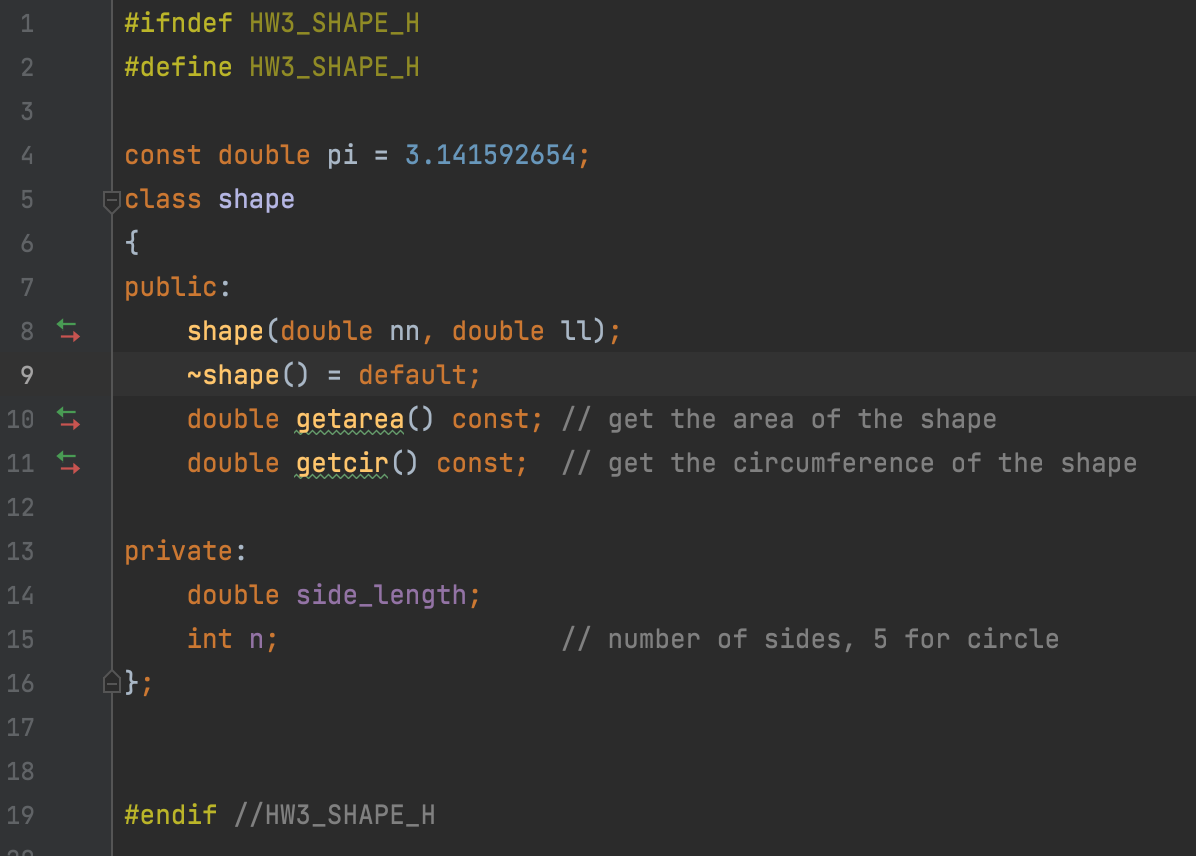
|  |  |  |
| --- | --- | --- |
| 边数 | λ的精确值 | λ的近似值（本程序使用） |
| 3 |  | 0.4330127019 |
| 4 |  | 1 |
| 5 |  | 1.720477401 |
| 6 |  | 2.598076211 |
|  |  | 3.141592654 |

* 1. 数据结构

本程序包括主函数源文件（main.cpp）和2个额外源文件及其头文件（shape\_dealer.cpp/h, shape.cpp/h）。

在shape\_dealer.h中定义了类shape\_dealer。这是一个负责处理所有本程序所需的计算及存储的类。其具体功能见图中注释，主要包括一个shape类的vector用于记录已有形状（及其对应的输入读取函数），及一个输出函数。

在shape.h中定义了类shape。此类用于描述一个正多边形，主要存储多边形的边长和边数，并提供公有接口调取其面积和周长。



1. 验证部分
   1. 单元测试

考虑到本程序最容易出现问题的地方在于shape::getarea()和shape::getcir()两函数，故对其进行单元测试。

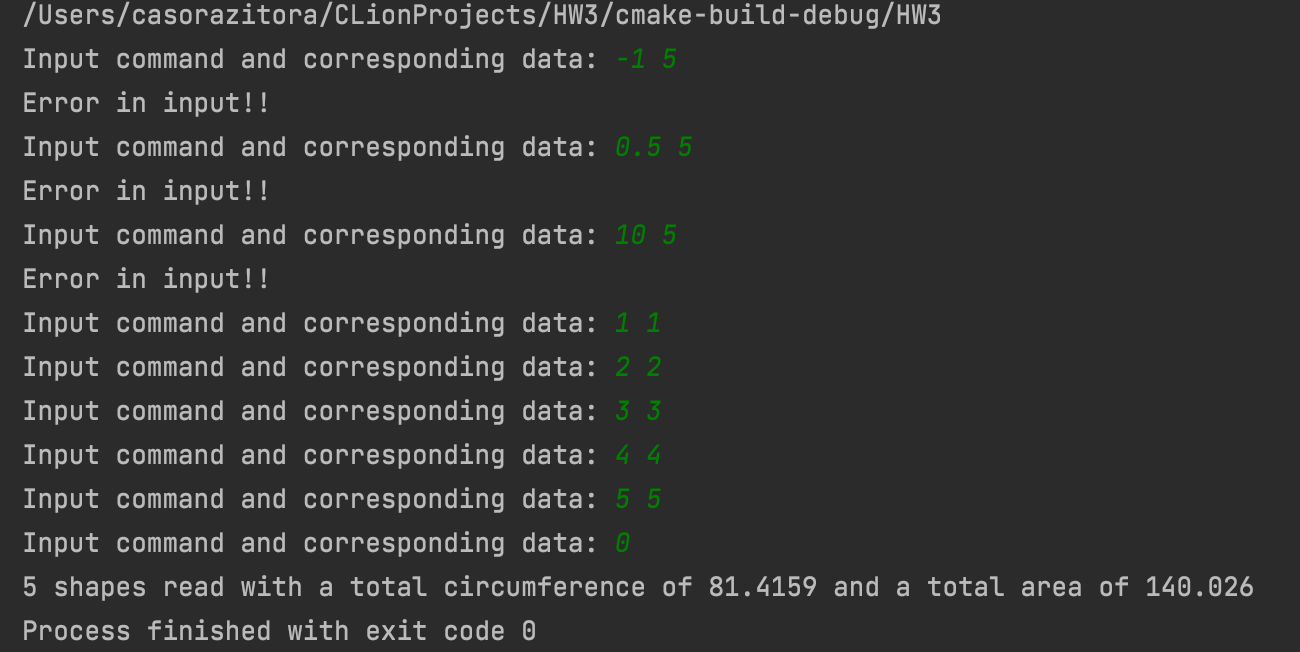
新增一个头文件test.h，内含一个测试函数test()。利用debug功能，读入一组数据，然后test()函数输出对应几何形状的getarea()及getcir()。这等同于对这两个函数进行了单元测试。测试结果如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 边长 | 边数 | getcir() | 理论周长 | getarea() | 理论面积 |
| 1 | 3 | 3 | 3 | 0.433013 |  |
| 2 | 4 | 8 | 8 | 4 | 4 |
| 3 | 5 | 15 | 15 | 15.4843 |  |
| 4 | 6 | 24 | 24 | 41.5692 |  |
| 5 | 圆 | 31.4159 |  | 78.5398 |  |

在10-6的精度下，测试结果和真实值一致。

在测试完毕后，将main函数中的test()删去。

* 1. 整体测试

手工测试结果如下：

首先测试了输入数据非法的状况，程序按设计拒绝了这些输入。之后，用a.中的数据逐一输入，并以0结束。将上一部分表内数据相加得总周长的真实值为81.4159、总面积的真实值为140.026345，和程序输出一致。

1. 为表述方便，本文档将统一把“圆”视作一种“正多边形”，在这种语境下，“边长”指的是圆的半径，“边数”定义为正无穷。 [↑](#footnote-ref-1)