

面向对象程序设计基础作业三 设计文档

1. 模型部分

a. 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，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。

b. 算法

本程序使用的算法为模拟法。

周长计算遵循以下公式：

$$Circumference = na$$

其中，a 表示边长，n 表示边数；在圆形的情况下， $n=2\pi$ 。

面积计算遵循以下公式：

$$Area = \lambda a^2$$

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	$\frac{\sqrt{3}}{4}$	0.4330127019
4	1	1
5	$\frac{5\tan(54^\circ)}{4}$	1.720477401
6	$\frac{3\sqrt{3}}{2}$	2.598076211
$+\infty$	π	3.141592654

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。

c. 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。

```
1  #ifndef HW3_SHAPE_DEALER_H
2  #define HW3_SHAPE_DEALER_H
3
4  #include <vector>
5  #include "shape.h"
6  class shape_dealer
7  {
8  public:
9      shape_dealer();           // constructor, automatically triggering input function
10     ~shape_dealer() = default; // trivial destructor
11
12 private:
13     std::vector<shape> shapes; // stores all shapes
14     bool push_shape(int command, double data); // add a shape to (vector)shapes
15     void report();           // displays total circumference and total area
16 };
17
18
19 #endif //HW3_SHAPE_DEALER_H
```

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  #ifndef HW3_SHAPE_H
2  #define HW3_SHAPE_H
3
4  const double pi = 3.141592654;
5  class shape
6  {
7  public:
8      shape(double nn, double ll);
9      ~shape() = default;
10     double getarea() const; // get the area of the shape
11     double getcir() const;  // get the circumference of the shape
12
13 private:
14     double side_length;
15     int n;                // number of sides, 5 for circle
16 };
17
18
19 #endif //HW3_SHAPE_H

```

2. 验证部分

a. 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	$\frac{\sqrt{3}}{4}$
2	4	8	8	4	4
3	5	15	15	15.4843	$\frac{9\sqrt{25 + 10\sqrt{5}}}{4}$
4	6	24	24	41.5692	$24\sqrt{3}$

5	圆	31.4159	10π	78.5398	25π
---	---	---------	---------	---------	---------

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。

b. 整体测试

手工测试结果如下：

```

/Users/casorazitora/CLionProjects/HW3/cmake-build-debug/HW3
Input command and corresponding data: -1 5
Error in input!!
Input command and corresponding data: 0.5 5
Error in input!!
Input command and corresponding data: 10 5
Error in input!!
Input command and corresponding data: 1 1
Input command and corresponding data: 2 2
Input command and corresponding data: 3 3
Input command and corresponding data: 4 4
Input command and corresponding data: 5 5
Input command and corresponding data: 0
5 shapes read with a total circumference of 81.4159 and a total area of 140.026
Process finished with exit code 0

```

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。