OOP 第三周作业文档

2019010175 孔瑞阳 土木 92

一、功能说明

不断进行以下操作：输入一个 0-5 的自然数表示操作类型。

1) 输入一个三角形的边长。

2) 输入一个正方形的边长。

3) 输入一个正五边形的边长。

4) 输入一个正六边形的边长。

5) 输入一个圆形的半径。

0) 结束程序，输出结果。

结束后输出输入的图形总个数、所有图形的总周长、所有图形的总面积。

二、模型

1、输入与输出

首输入和输出采用 C++标准库中的输入输出流 cin 和 cout 进行。

在实现过程中，发现课上所讲的输入一个正整数中的如下语句：

```
while (m_data <= 0)
{
    cin.clear();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');
    cin >> m_data;
}
```

并不能通过直接把<=0 改成<0 来完成输入一个自然数的程序，因为在输入一个不是数字的数据后，m_data 的值依然是 0，这在正整数中算是非法输入，但是在自然数中是合法的。

经过查阅资料后，采用以下方式实现：

```
while (m_data < 0 || cin.rdstate() == ios_base::failbit)
```

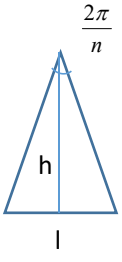
2、正多边形的周长和面积计算

记正多边形的边数为 n，边长为 l。

则周长：$C = l * n$ 。

对于面积，我们考虑把正 n 边形分成 n 个相邻顶点为底角、中心点为顶角的三角形。

则顶角的角度是 $\frac{2\pi}{n}$，则可以得到高 $h = \frac{l}{2} * \cot\left(\frac{\pi}{n}\right)$。



一个三角形的面积是 $\frac{hl}{2}$，则总面积 $S = \frac{nhl}{2} = \frac{l^2 n}{4}\cot\left(\frac{\pi}{n}\right)$。

运用 cmath 库中的三角函数实现，其中 cot 使用 1/tan 实现。

3、圆形的周长和面积计算

记圆形的半径为 r。

则周长：$C = 2\pi r$。

面积：$S = \pi r^2$。

4、图形总个数、总周长和总面积的统计

记使用三个计数器 polygonNumber，totalPerimetre，totalSquare 记录如上信息。
每次输入一个图形之后即进行相应的修改。

三、软件构件介绍

文件	功能介绍
CP_NaturalInteger.h/cpp	实现自然数的输入和操作
CP_PositiveReal.h/cpp	实现正实数的输入和操作
CP_Circle.h/cpp	有关于圆的信息、操作和计算
CP_RegularPolygon.h/cpp	有关于正多边形的信息、操作和计算
regularPolygonTest.h/cpp	对于圆、正多边形的计算的测试
regularPolygonMain.cpp	主程序

四、单元测试

1、输入单元的测试

对于输入一个自然数操作符 opt:（ CP_NaturalInteger.h/cpp ）

(1) 等价类划分

① 输入的不是 0-5 的某个自然数。

② 输入的是 0-5 的某个自然数。

(2) 测试：

等价类	选取案例	输出结果（下一步操作）
等价类①	HelloWorld!	您输入的不是自然数，请输入一个自然数：
	1919.810	您输入的不是自然数，请输入一个自然数：
	-10	您输入的不是自然数，请输入一个自然数：
	12	您输入的自然数超过了 5，请输入一个自然数：
等价类②	1	请输入正三角形的边长，请输入一个正实数：
	2	请输入正方形的边长，请输入一个正实数：
	3	请输入正五边形的边长，请输入一个正实数：
	4	请输入正六边形的边长，请输入一个正实数：
	5	请输入圆形的半径，请输入一个正实数：
	0	输入的图形总个数是……（程序结束）

对于输入一个正实数边长/半径：（ CP_PositiveReal.h/cpp ）

(1) 等价类划分

① 输入的不是正实数。

② 输入的是正实数。

(2) 测试：

等价类	选取案例	输出结果（下一步操作）
等价类①	HelloWorld!	您输入的不是正实数，请输入一个正实数：
	-114.514	您输入的不是正实数，请输入一个正实数：
等价类②	1	请输入一个 0-5……（进入下一步操作）
	1919.810	请输入一个 0-5……（进入下一步操作）

2、周长与面积的验证（ CP_RegularPolygon.h/cpp , CP_Circle.h/cpp ）

测试单元：regularPolygonTest.h/cpp

按照图形的种类来划分等价类。

第一步：选取边长/半径为 1 的图形进行计算，并与理论值进行比较。

正三角形：$C=3, S=\frac{\sqrt{3}}{4}\approx 0.433$

正方形：$C=4, S=1$

正五边形：$C=5, S=\frac{\sqrt{25+10\sqrt{5}}}{4}\approx 1.720$

正六边形：$C=6, S=\frac{3\sqrt{3}}{2}\approx 2.598$

圆形：$C=2\pi\approx 6.2832, S=\pi\approx 3.1416$

测试：

采用如下语句进行测试：

```
CP_RegularPolygon rp(i, 1);
cout << rp.perimeter() << ' ' << rp.square() << endl;
CP_Circle cc(1);
cout << cc.perimeter() << ' ' << cc.square() << endl;
```

测试结果：

图形	边长/半径	周长 面积
正三角形	1	3 0.433013
正方形	1	4 1
正五边形	1	5 1.72048
正六边形	1	6 2.59808
圆形	1	6.28319 3.14159

与预测值相同，初步测试成功。

第二步：利用第一步的结果进行对拍。

由于边长/半径为 1 的情况我们已经验证正确，并且根据理论，边长/半径为 a 的图形的周长应该是其的 a 倍，面积是其的 a^2 倍，那么可以每次随机出一个图形和相应的边长/半径，计算其的周长和面积是否是相应边长/半径为 1 的图形的相应倍数，当两个值的相对误差小于 10^{-9} 时我们即认为是相等的。

运行 regularPolygonTest2()，在一分钟之内程序没有发现错误暂停。事实上，在一分钟内计算机可以进行上亿次验证，基本可以验证程序的正确性。

3、计数器的测试（ regularPolygonMain.cpp ）

为了考虑测试正确性以及非法输入对于程序的可能影响，设计一组输入数据如下：

其中，红色表示非法输入，运用测试 2.1 中的数据进行测试，理论输出应该为：

个数：5　　总周长：24.2832　　总面积：8.89316

输入	输出
HelloWorld! 3 1 2 1 4 HelloWorld! -10 1 203 1 0 1 5 1 12 0	输入的图形总个数是：5 所有图形的总周长是：24.2832 所有图形的总面积是：8.89316

与理论输出相同，由于计算的过程已经在 2 中得到验证，那么可以认为计数器实现没有问题，并且不会受到非法输入的影响。

五、文档其他部分

1、如何充分进行单元测试

（1）保证测试的全面性，不能出现某个单元没有测试到的情况。确保所有的基本功能都可以正确实现。

（2）保证测试数据覆盖了全部等价类，考虑特殊情况，保证程序没有考虑某些情况。甚至在可以的情况下，测试数据覆盖所有情况。

（3）对于边界值进行测试，防止边界情况出现溢出。

（4）对于某些输入的非法情况进行测试，防止程序因此而崩溃。

2、C 语言与 C++在本例中的程序扩展性差异

首先，如果要直接扩展，由于在 C 语言中数据和处理的过程是独立的，所以在扩展的过程中对于原来的过程的依赖性很高，从而使得程序的耦合性变得很高，难以维护和扩展。而 C++进行扩展时，对于对象本身的扩展是非常简便的，并且如果仅是要扩展出一个对于本对象进行数据处理的程序，不需要再去调用其他的过程，直接对于对象进行数据处理即可，耦合性较低。

其次，在测试的过程中，使用 C++面向对象的方法进行测试，可以直接利用过程来检测对象的特性，但如果用 C 语言来实现，则会出现：测试过程（Test）->测试对象（多边形）->调用测试对象的过程（计算）->调用测试对象的另一个数据（多边形的数据）->……这样的情况，使得扩展之后的单元测试更加麻烦，也从另一方面降低了 C 语言程序的可扩展性。