POSD2019f_Midterm

Clone the project on .

https://ssl-gitlab.csie.ntut.edu.tw/course/posd2019f_midterm

You will find that Solid is given to you already.

We want to add four different solids:

1. **Cone**
2. **Square Cylinder**
3. **Triangular Pyramid**
4. **Complex Solids**

These four solids should inherit Solid class.

You should use Composite pattern.

Use Iterator to access your vector rather than index.

1. You should implement following methods

A. bottomArea

B. volume

Cone:

bottomArea = radius ^ 2 x π

(π -> You must use M_PI from math.h/cmath)

volume = bottomArea x height x ⅓

Note: Give you **Center of mind** and **a point on the circle**

that you can calculate radius

If radius length is equal to zero it should

throw "Bottom is not a Circle!"

Square Cylinder:

bottomArea = edge ^ 2

volume = bottomArea x height

Triangular Pyramid:

bottomArea = $\sqrt{s(s-a)(s-b)(s-c)}$

Note: Give you three points that you can get three edge a, b, c

$$s = \frac{a+b+c}{2}$$

If three edge can not **form a triangle** that it should

throw "Bottom is not a Triangle!"

volume = bottomArea x height x ⅓

2. A Complex Solids object is composite and can be made of multiple Solids, including both leaf and composite in the Composite Pattern.

Note:

Use Iterator to access your vector rather than index.

A. Member functions In **ComplexSolids** class:

i. **Constructor**:

ComplexSolids(std::vector<Solid*> * solids)

ii. **add()**: add the solid to composite

void add(Solid *s)

iii. **bottomArea()**: sum of the child bottomArea

double bottomArea() const

iv. **volume()**: sum of the child volume

double volume() const

v. **numberOfChild()**: return child amount

int numberOfChild()

3. Add an **find()** method to implement an operation to find solids– leaf nodes with

volume inside [volumeMin, volumeMax] and

bottomArea inside [bottomAreaMin, bottomAreaMax]

A. Add a **find ()** method to the " **Solid** " hierarchy.

find (): return result satisfies the condition below:

1. volumeMin <= volume <= volumeMax,

2. bottomAreaMin <= bottomArea <= bottomAreaMax

skeleton:

vector<Solid*> find(double volumeMin, double volumeMax, double bottomAreaMin, double bottomAreaMax)

In **leaf node**, you should check if leaf itself satisfies the condition.

If so, the result that **find()** returns will contain leaf itself

In **composite node**, you should traverse the composite structure to find all **leaf** that satisfies the condition.

Note:

“It only need to store

Cone or Square Cylinder or Triangular Pyramid

No need to store ComplexSolid “

Submit your code

Submit your code by committing to your repository

(https://ssl-gitlab.csie.ntut.edu.tw/users/sign_in)

before the deadline

(2019-10-19 17:30 pm). Submit frequently!

Don’t wait till the last minute!

Midterm Project Structure

Example:

- bin
  - ut_all
- src
  - complex_solids.h
  - cone.h
  - solid.h
  - square_cylinder.h
  - triangular_pyramid.h
- test
  - ut_main.cpp
  - ut_solid.h

makefile

Note

1. **Make sure your code are writen in the correct file.**
2. **The score of this exam will be divided into three parts:**
    1. **Code(40%)**

       You should pass the CI Test first!

    2. **Unit tests given to you byTA (20%),**
    3. **Unit tests on Jenkins but not given to you (40%).**

Resources allowed to use

1. **Design Patterns (GoF) textbook**

   (Only English Version)

2. **projects in class (https://ssl-gitlab.csie.ntut.edu.tw/yccheng/posd2019f)**
3. **Your own homework repository on Gitlab**
4. **cplusplus.com (http://www.cplusplus.com)**
5. **Prescribed Dictionary (https://dictionary.cambridge.org/zht)**

Attention!!

You cannot visit any other website and you must turn off your mobile phone during the midterm exam, or you will be considered as cheating.

Violation of the rules:

First time : Deduct points

Second time: Calculated by zero