

تمرین های سری چهارم درس برنامه نویسی پیشرفته کامپیوتر

امیر جهانشاهی

۹ فروردین ۱۳۹۷

۱. این سوال ادامه سوال دوم تمرین سری پیش می باشد.

در این بخش قصد داریم با تعریف کلاس `polygon` برای چند ضلعی ها کلاس های نوشته شده `point` و `line` را تکمیل نماییم. برای این کار کلاس نوشته شده باید شامل موارد زیر باشد:

```
۱ class polygon
۲ {
۳ ...
۴     polygon(point* arr,int size);
۵     polygon(line* arr,int size);
۶     bool isTriangle();
۷     bool isSquare();
۸     bool isEqual(polygon p);
۹     ...
۱۰ };
```

- `polygon(point* arr, int size)` این تابع یکی از constructor های این کلاس است و در آن باید شی مربوط به چند ضلعی بر اساس نقاط داده شده ساخته شود. (نقاط- راس های چند ضلعی- به ترتیب در آرایه قرار خواهند گرفت)
- `polygon(line* arr, int size)` این تابع یکی از constructor های این کلاس است و در آن باید شی مربوط به چند ضلعی بر اساس خطوط داده شده ساخته شود. (خطوط- ضلع های چند ضلعی- به ترتیب در آرایه قرار خواهند گرفت)
- `bool isTriangle()` در صورت مثلث بودن این چند ضلعی `true` و در غیر این صورت `false` باز می گرداند.
- `bool isSquare()` در صورت مربع بودن این چند ضلعی `true` و در غیر این صورت `false` باز می گرداند.

- پیاده سازی این بخش اختیاری بوده و دارای نمره امتیازی می باشد:

`bool isEqual(polygon p)` در صورت برابر بودن دو چند ضلعی `true` و در غیر این صورت `false` باز می گرداند.

توجه کنید که: توابع نوشته شده دقیقاً مشابه توابع ذکر شده باشند و از تغییر نام توابع خودداری کنید. در صورت نیاز می توانید توابعی جدیدی نیز تعریف نمایید.

۲. در این سوال قصد داریم کلاسی با نام `Shape` را پیاده سازی نماییم. این کلاس را به صورت خلاصه به عنوان کلاس پایه و به صورتی که توانایی پیاده سازی سایر کلاس ها را داشته باشید پیاده سازی کنید. در مرحله بعدی دو کلاس `TwoDimensionalShape` و `ThreeDimensionalShape` برای اشکال دو بعدی و سه بعدی از کلاس `Shape` ارث می برند. در مرحله بعدی کلاس های `sphere`، `square`، `cube` و `circle` را که از دو کلاس `TwoDimensionalShape` و `ThreeDimensionalShape` ارث می برند را پیاده سازی نمایید. در نهایت کلاس های پیاده سازی شده باید `main` نوشته شده زیر را اجرا کرده و خروجی های مورد نظر را ایجاد نماید.

توجه کنید که: خروجی می بایست دقیقاً برابر با خروجی ذکر شده باشد.

```

1  #include <iostream>
2
3  #include "circle.h"
4  #include "square.h"
5  #include "sphere.h"
6  #include "cube.h"
7
8  using std::cout;
9
10 int main()
11 {
12     Square sqr( 12, 2, 2 );
13     Sphere sph( 5, 1.5, 4.5 );
14     Cube cub( 2.2 );
15     Circle cir( 3.5, 6, 9 );
16     Shape *ptr[ 4 ] = { &cir, &sqr, &sph, &cub };
17
18     for ( int x = 0; x < 4; ++x )
19         cout << *( ptr[ x ] ) << '\n';
20
21     return 0;
22 }
```

خروجی `main` به صورت زیر می باشد.

```

1  Circle radius = 3.5
2  center --> (6, 9)
```

```

۳ area of 38.4845
۴
۵ Square side length = 12
۶ center --> (2, 2)
۷ area of 144
۸
۹ Sphere radius = 5
۱۰ center --> (1.5, 4.5, 0)
۱۱ area of 314.159 & volume of 523.598
۱۲
۱۳ Cube side length = 2.2
۱۴ center --> (0, 0, 0)
۱۵ area of 29.04 & volume of 10.648

```

در پیاده سازی خود نکات زیر را رعایت کنید:

به منظور به دست آوردن خروجی مطلوب برای کلاس ها یک تابع به نام `print` تعریف نمایید و در مواقع مورد نیاز این تابع را به صورت `virtual` تعریف کنید.

همچنین در کلاس هایی که از کلاس مبنا ارث میبرند توابع `area` و `volume` را به نحوی تعریف کنید که محاسبات مساحت و حجم برای اشکال انجام گیرد. توجه کنید که برای اشکال دو بعدی حجم بی معنی خواهد بود.

آنچه که در چاپ خروجی به فرمت داده شده بسیار مهم است بازنویسی یکی از `operator` یعنی << به نحو صحیح می باشد.

در تمرین قبلی کلاس `point` را تعریف کرده و پیاده سازی نمودید. حال سعی کنید `operator` جمع را برای کلاس هایی که از کلاس `Shape` ارث می برند به نحوی بازنویسی نمایید که با جمع عنصری از کلاس `point` و کلاس های مشتق شده بتواند مرکز اشکال را به اندازه مختصات نقطه جا به جا نماید. این بخش را به انتهای فایل `main` اضافه نمایید.

۳. در سوال قبل آیا نیازی به تعریف توابع `print`، `area` و `volume` به صورت `virtual` داریم یا خیر؟ توضیح دهید که به صورت کلی چه زمانی از `virtual` استفاده می کنیم.

۴. در این سوال می خواهیم کلاس `Rational` را برای کار با اعداد گویا (کسری) پیاده سازی کنیم. قسمتی از برنامه که در زیر نوشته شده است را درون فایل `main.cpp` خود قرار دهید و کلاس مربوطه را به صورتی پیاده سازی نمایید که برنامه به درستی اجرا شود. در واقع کلاس مربوطه باید خواص زیر را دارا باشد:

- ایجاد `constructor` که از پذیرفتن صفر و اعداد منفی در مخرج پرهیز کند و هم چنین کسر ها به فرم کسر ساده شده تبدیل کند .
- اپراتور های جمع ، تفریق ، ضرب و تقسیم برای این کلاس تعریف شوند.

• اپراتورهای مقایسه ای (< , > , <= , >= , == , !=) برای این کلاس تعریف شوند.

• تابع `show` که وظیفه نمایش کسر ها را بر عهده دارد.

```
1 using std::cout;
2 using std::endl;
3
4 int main()
5 {
6     RationalNumber a{7, 3 }, b{3, 9}, x;
7     RationalNumber c = a;
8
9     c.show();
10    cout << '\n';
11
12    a.show();
13    cout << " + " ;
14    b.show();
15    cout << " = ";
16    //x = a + b;
17    //x = a - b;
18    //x = a * b;
19    x = a / b;
20    x.show();
21
22    cout << '\n';
23    a.show();
24    cout << " is:\n";
25
26    cout << ( ( a > b ) ? " > " : " <= " );
27    b.show();
28    cout << " according to the overloaded > operator\n";
29
30    cout << ( ( a != b ) ? " != " : " == " );
31    b.show();
32    cout << " according to the overloaded != operator" << endl;
33
34    return 0;
35 }
```

۵. این کلمات و عبارات برای شما حاوی چه معنایی هستند؟

(a) explicit

(b) inline

(c) override

(d) pure abstract

(e) CMessage() = default;

جهت تحویل تمارین، هر تمرین را داخل یک فولدر بریزید که با شماره تمرین نام گذاری شده است. Q1, Q2, ... گزارش کار را به صورت PDF در فولدر اصلی تمرین ها قرار دهید. در نتیجه در فولدر اصلی فقط یک فایل گزارش موجود می باشد و تعدادی فولدر که با شماره تمرین ها نام گذاری شده است. اسم فولدر اصلی را به صورت زیر نام گذاری و سپس فشرده سازی و در قالب یک فایل ارسال کنید. توجه نمایید که از قالب فشرده سازی rar استفاده نکنید.

شماره دانشجویی-AP-HW4.zip

مهلت تحویل: تا ساعت ۲۳ جمعه ۲۴ فروردین ماه ۱۳۹۷