

مغازه سی دی فروشی (Factory method)

این هفته هیپولی توی کلاس AP با کلاس انتزاعی و توابع مجازی آشنا شده ولی دقیق نمی دونه که چقد استفاده از این می تونه کمکش کنه.

بعد از کلاس میره مغازه سی دی فروشی ای که عموش تازه باز کرده. عموش ازش می خواد اطلاعات سی دی های مغازشو مرتب یه جا داشته باشه و به راحتی بتونه بهشون دسترسی داشته باشه.

حالا هیپولی مونده و کلی سی دی مرتب نشده!!



محتوای سی دی ها میتونن فیلم سینمایی (Movie) و یا سریال (Series) باشن و فیلم ها قابلیت اجاره گرفتن دارن. پس یه کلاس CD داریم که عضو های داده ای نام اثر (name)، کارگردان اثر (director) و مدت زمان به دقیقه (duration) داره. دو کلاس Movie و Series از کلاس CD ارث میبرن.

در کلاس Movie عضوهای داده ای release_year و language و rented رو داریم. همچنین چهار تابع عضو زیر هم وجود دارن:

- تابع *setdata* : وظیفه ی گرفتن تمام اطلاعات فیلم سینمایی شامل نام اثر، کارگردان، مدت زمان، سال انتشار، زبان فیلم و قابلیت اجاره گرفتن رو داره.
- تابع *showdata* : وظیفه ی نمایش تمام اطلاعات فیلم سینمایی رو داره.
- تابع *rentMovie* : با صدا زدن این تابع نشان می دهیم فیلم اجاره داده شده.
- تابع *returnMovie* : با صدا زدن این تابع نشان می دهیم فیلم به مغازه بازگردانده شده.

در کلاس Series دو عضو داده ای season و episode رو داریم. همچنین دو تابع عضو زیر هم وجود دارن:

- تابع *setdata* : وظیفه ی گرفتن تمام اطلاعات سریال شامل نام اثر، کارگردان، مدت زمان، فصل و قسمت رو داره.
- تابع *showdata* : وظیفه ی نمایش تمام اطلاعات سریال رو داره.

حالا به هپولی کمک کنید که این کلاس ها رو بهینه پیاده سازی کنه جوری که از کلاس انتزاعی، توابع مجازی، فکتوری متد و داینامیک کست استفاده بشه. همچنین شما می توانید عضو های داده ای بیشتری در کلاس ها بنابر نیاز ایجاد کنید.

ورودی :

خط اول ورودی تعداد شی هایی است که می سازیم سپس در صورتی که کاربر در خط بعدی عدد 1 را وارد کرد به معنای ساخت شی فیلم سینمایی، در صورتی که 2 را وارد کرد به معنای ساخت شی سریال است.

بعد از وارد کردن نوع شی در خط بعدی موارد مورد نیاز هر شی از کاربر گرفته می شود . برای مثال برای فیلم سینمایی به ترتیب نام اثر، کارگردان، مدت زمان، سال انتشار و زبان فیلم گرفته میشه و یا در صورتی که سریال بود به ترتیب نام اثر، کارگردان، مدت زمان، فصل و قسمت گرفته میشه.

سپس در q خط بعدی به ترتیب شماره سی دی و اجاره گرفتن (1) و یا پس دادن را (0) آن را ورودی می گیریم.

خروجی :

ابتدا به ترتیب اطلاعات تمامی CD ها را بستگی به نوع شئی ها خروجی می دهیم.(اطلاعات با یک اسپیس فاصله) سپس به تعداد query ها به شکل زیر خروجی می دهید:

- فیلم قبلا اجاره گرفته شده بود و در دسترس نیست عبارت "Not Returned Yet"
- با موفقیت اجاره داده شد عبارت "Rented Successfully"
- با موفقیت پس داده شد "Returned Successfully"
- در غیر این صورت "ERROR"

مثال :

ورودی :

```
4
1
Interstellar Christopher_Nolan 169 2014 Eng 0
2
Friends David_Crane/Marta_Kauffman 25 4 7
2
Skam Julie_Andem 26 2 1
1
The_Book_Thief Brian_Percival 130 2013 Eng/Ger 0
5
1 0
4 1
4 1
4 0
2 1
```

خروجی :

Interstellar Christopher_Nolan 169 2014 Eng 0
 Friends David_Crane/Marta_Kauffman 25 4 7
 Skam Julie_Andem 26 2 1
 The_Book_Thief Brian_Percival 130 2013 Eng/Ger 0
 ERROR
 Rented Successfully
 Not Returned Yet
 Returned Successfully
 ERROR

کد main :

```

1  int main(){
2
3      int n, val, q;
4      cin>>n; //The number of objects that is going to be created.
5      CD** cd_object = new CD* [n];
6
7      for(int i = 0; i < n; i++){
8          cin>>val;
9          cd_object[i] = CreateObject(val); //Call Factory Method here
10         cd_object[i]->getdata(); // Get the data from the user.
11     }
12
13     for(int i=0; i<n; i++)
14         cd_object[i]->putdata(); // Print the required output for each object.
15
16     cin>>q;
17     //your code
18
19     return 0;
20 }
```

شیپولی حسود (OOP)

پیش‌نیاز مسئله: خانواده شیپولی (مفاهیم پیشرفته شیئی گرایی)

یکی از دلایل خوشحالی همیشگی هیپولی اینه که وقتی میره برای خودش قهوه بریزه بیاد لپتاپش میره رو حالت اسکرین سیور و کلی حباب قشنگ میاد که دارن میخورن بهم :))) (سیستم عاملش ویندوز سونه!).

مورد بالا باعث شده که شیپولی به هیپولی حسودی کنه و دلش بخواد که اونم همچین چیزی داشته باشه، حتی یه چیز بهتر!

در این تمرین قراره شما به کمک خانواده شیپولی یک اسکرین سیور در صفحه ترمینال طراحی کنید که علاوه بر دایره، مربع و مستطیل هم داشته باشه که در صفحه با سرعت های مختلف حرکت می‌کنند و در صورت برخورد بهم مسیرشون عوض میشه.

از این لینک می‌توانید نحوه کار این اسکرین سیور را ببینین.

در این برنامه در هر مرحله صفحه ترمینال پاک می‌شود و به کمک توابع پایین شکل های مختلف کشیده می‌شوند و در هر مرحله در یک جهت با سرعتی معین جابه جا می‌شوند.

• تنظیمات اولیه:

```
1 | #include <windows.h>
2 | #include <time.h>
3 | #include <math.h>
4 | double pi = 2*acos(0.0);
5 | HWND myconsole = GetConsoleWindow();
6 | HDC hdc = GetDC(myconsole);
7 | COLORREF color;
8 | color = RGB(rand() % 256, rand() % 256, rand() % 256);
```

• تابع کشیدن دایره:

```

1  for (double theta = 0; theta < 2 * PI; theta += .001) {
2      SetPixel(hdc, x - radius * cos(theta), y - radius * sin(theta), color);
3  }
4  // Filled circle
5  int r2 = radius * radius;
6  for (int i = -radius; i < radius; i++) {
7      for (int j = -radius; j < radius; j++) {
8          if (i * i + j * j < r2)
9              SetPixel(hdc, x - i, y - j, color);
10     }
11 }
12 Sleep(1000);

```

• تابع کشیدن مستطیل:

```

1  for (int i = -width / 2; i < width / 2; i++)
2  {
3      SetPixel(hdc, x + i, y - height / 2, color);
4      SetPixel(hdc, x + i, y + height / 2, color);
5  }
6  for (int j = -height / 2; j < height / 2; j++)
7  {
8      SetPixel(hdc, x - width / 2, y + j, color);
9      SetPixel(hdc, x + width / 2, y + j, color);
10 }
11 // Filled rectangle
12 for (int i = -width / 2; i < width / 2; i++)
13 {
14     for (int j = -height / 2; j < height / 2; j++)
15     {
16         SetPixel(hdc, x + i, y + j, color);
17     }
18 }

```

راهنمایی‌ها:

۱. با تغییر جزئی کد خانواده شیپولی و اضافه کردن مواردی مثل یک تابع `move` و ... استفاده کنید.
۲. اشیاء را می‌توانید بصورت رندوم در کلاس مربوط به خودشان تولید کنید و بر حسب ابعاد صفحه نمایش و تعداد شیئی‌های در صفحه بسازید.
۳. برای سرعت هر شکل می‌توانید یک بردار که جهت x و y را دارد در نظر بگیرید.
۴. برای تلاقی دایره با مستطیل، می‌توانید دایره محیطی مستطیل را در نظر بگیرید.

موارد امتیازی: (تا ۵۰ نمره)

- تلاقی دو مستطیل باهم بدون فرض دایره بودن و بر حسب اضلاع هر کدام
- تعریف طول عمر برای هر یک از اشیاء
- هر ایده موردی که مسئله یا طراحی شیئی گرایشی شما را بهبود بخشد!

نکات مهم:

۱. تمامیه متغیرها باید از نوع مناسب باشند.
۲. کلاس‌های شما باید به بهترین نوع ممکن ارث برده باشند. (مانند سوال خانواده شیپولی)
۳. به تفاوت تلاقی اشکال با یکدیگر توجه کنید، برای مثال فاصله دو دایره حداقل جمع دو شعاع آنهاست.
۴. طراحی کلاس‌های شما مهم‌ترین قسمت این مسئله است، پس به تمامی مباحث درس در طراحی توجه کنید.
۵. برای مواردی که به طور شفاف در صورت سؤال بیان نشده اند (مثلاً برخورد دایره با مستطیل) با یک فرض پیش روید.
۶. هنگام تلاقی دو شیئی می‌توانید صرفاً جهت مسیر هر شکل را برعکس کنید. (۱۸۰ درجه چرخش)
۷. به فایل بندی کلاس‌هایتان دقت کنید.

سپهر خسته (Exception handling)

پیش‌نیاز مسئله: کلاس Date-Time (مباحث پیشرفته شیء گرای)

همون طور که میدونیم سپهر یکمی دست و پا چلفتیه!!

واسه همین ممکنه ورودی های غلط به برنامه ای که برای تاریخ و زمان نوشته بده! از طرفی دیگه از دیباگ کردن کدش و assert نوشتن خسته شده! حالا سپهر به کمک شما احتیاج داره که اگه ورودی هاشو غلط داد بتونه هندلشون کنه.

حالا مشکلاتی که میتونن پیش بیان به شکل زیر هستن:

کلاس Time

- در صورت بروز خطا در متغیر دقیقه یا ساعت، یک شیء از کلاس `TimeException` را `throw` کنید که شامل یک متغیر برای نمایش مقدار خطا و یک متن برای توضیح بیشتر خطاست.

کلاس Date

- در صورت بروز خطا برای روز، مقدار `int` را `throw` کنید.
- اگر برای مقدار ماه خطا پیش اومد، `out_of_range` را `throw` کنید. (برای اطلاعات بیشتر درمورد این کلاس، به این لینک مراجعه کنید.)
- در صورت بروز خطا برای سال یک شیء از کلاس `DateException` را `throw` کنید که مشابه کلاس `TimeException` است.

کلاس DateTime

- خطاهای مربوط به متغیر دقیقه کلاس `Time` را رسیدگی کنید و نزدیک ترین عدد را کنید.
- خطای `int` مربوط به روز در کلاس `Date` را رسیدگی کنید و نزدیک ترین عدد را جایگزین کنید.
- خطاهای مربوط به ساعت و یا سال را `catch` کنید و خطای `DateTimeException` را `throw` کنید. (چون خطای این دو مورد مهم هستند و امکان رسیدگی به آن در این کلاس نیست.)

موارد مهم:

- در برنامه خود تا حد امکان از تکرار کد خودداری کنید و تا حد امکان خطاها را در یک قسمت رسیدگی کنید.
- کلاس‌های خطای شما باید از کلاس `Exception` ارث ببرند. (توضیحات این کلاس در این لینک است.)
- در `main` برنامه خطای مربوط به `DateTimeException` را `catch` کنید و در صفحه کنسول متن مربوط به خطا را چاپ کنید.
- در `main` برنامه مثال‌هایی برای چک کردن سناریوهای بالا بزنید.

دست گرمی پروژه (: (OOP)

صورت سوال در فایل PDF تمرین است.

در این سوال شما باید با توجه به فایل پروژه ای که مربوط به پروژه های قبلی این درس بوده است فایل PDF شامل یکی از پروژه های درس برنامه نویسی پیشرفته است. در این سوال شما باید با توجه به توضیحات پروژه UML این پروژه را طراحی کنید.

در نهایت یک فایل PDF شامل طراحی خود به همراه لینک یک ویدیو حداکثر ۳ دقیقه ای برای توضیح و دفاع از طراحی خود آپلود کنید.

در هنگام طراحی به موارد زیر توجه کنید:

- به خوبی از روابط شیئی گرایی استفاده کنید و تا حد امکان از اصول SOLID که در کارگاه بحث شد استفاده کنید. (برای اطلاعات بیشتر می‌توانید به [این لینک](#) مراجعه کنید).
- نیازی به جزئیات کامل نیست و توابع مهم در طراحی کافی است.
- به دسترسی های هر کلاس با توجه به توابعی که دارد دقت کنید