به نام خدا

قسمت دوم

آموزش Object Oriented

زمان تخمینی: ۱۵ ساعت

مفهوم شی گرایی (Object Oriented) یکی از مهم‌ترین و پایه‌ای ترین مفاهیم دانش برنامه نویسی می‌باشد. به همین منظور یادگیری برنامه نویسی شی‌گرا برای هر فردی که در این زمینه مشغول به فعالیت است امری ضروری است.

در لینک‌های زیر این مفاهیم توضیح داده شده اند:

به دلخواه خود یکی از زبان‌های زیر را انتخاب نمایید و لینک‌های مربوط به آن را مطالعه نمایید.

For java:

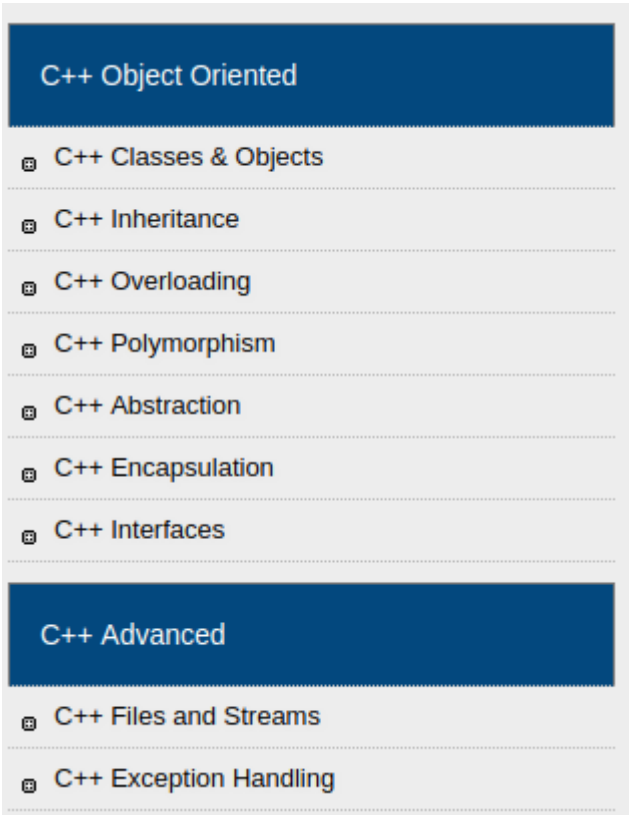
https://www.tutorialspoint.com/java/index.htm

برای زبان جاوا از لینک بالا استفاده نمایید.

For cpp:

https://www.tutorialspoint.com/cplusplus/index.htm

لینک بالا را باز کنید و از سمت چپ تمامی موارد قسمت C++ Object Oriented و همچنین دو قسمت اول C++ Advance را مطالعه نمایید



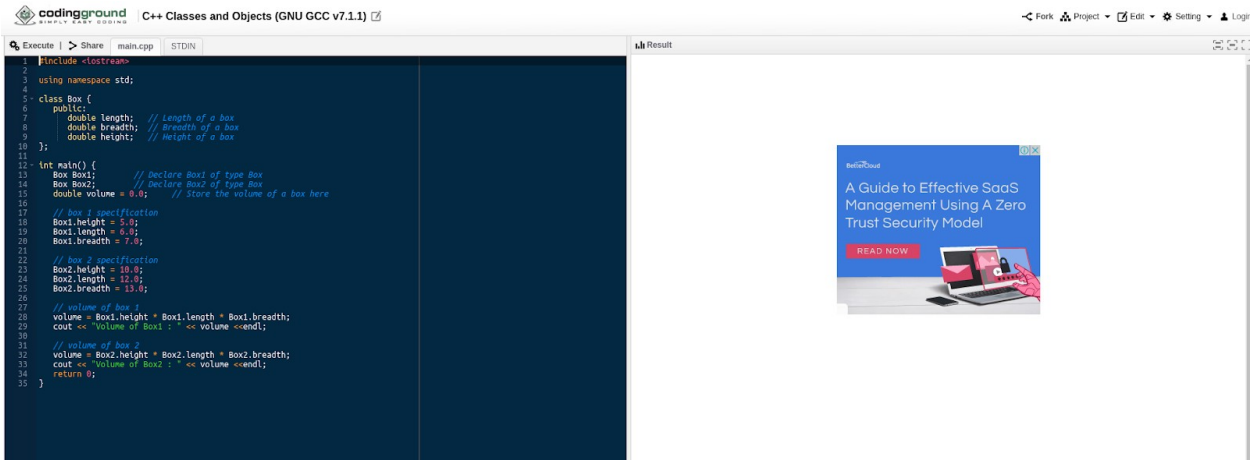
همان طور که در شکل زیر می‌بینید با کلیک روی Live Demo می‌توانید به صورت آنلاین کد بزنید:

Live Demo

```cpp
#include <iostream>

using namespace std;

class Box {
   public:
      double length;   // Length of a box
      double breadth;  // Breadth of a box
      double height;   // Height of a box
};
```



برای تفهیم بهتر مفاهیم شی گرایی از لینک زیر می‌توانید استفاده کنید.

https://dev.to/charanrajgolla/beginners-guide---object-oriented-programming

شما باید پس از خواندن این اسناد مباحث Inheritance, static fields and methods , Object, Class, Encapsulation, Abstraction را فرا گرفته باشید. لازم به ذکر است که یادگیری همین مباحث کافی است و نیازی به یادگیری مباحث بیشتری از جمله Polymorphism نیست.

تمرین: (این تمرین را با زبان برنامه نویسی دلخواه خود بنویسید)

در این تمرین می‌خواهیم با توجه به چیزهایی که در این قسمت یاد گرفتیم موجودیت(class)ها و فرآیند(method)ها یک سیستم تبلیغات آنلاین ساده را طراحی و پیاده سازی کنیم. در سیستم‌های تبلیغاتی کلیکی موجودیت‌های تبلیغ کننده (کسی که تبلیغ می‌کند) و تبلیغ (آگهی تبلیغاتی) همواره مطرح هستند. این سیستم یک سیستم تبلیغات کلیکی است لذا هر بار تبلیغات نمایش داده می‌شود باید این تعداد شمرده شود همچنین با کلیک کاربران روی تبلیغ هر بار یک عدد به تعداد کلیک‌های آن باید اضافه شود.

در این سیستم یک کلاس به نام Advertiser داریم که حاوی اطلاعات و method های تبلیغ کننده‌های سیستم است. کلاسی دیگر به نام Ad داریم که این کلاس حاوی اطلاعات و method های مربوط به تبلیغ است.

Advertiser Class Fields:

- id
- name
- clicks
- views

توضیحات:

فیلد id شناسه خاص هر تبلیغ کننده است که معمولا از نوع عدد طبیعی است و باید به صورت یکتا برای هر تبلیغ کننده باشد. فیلد name همان نام تبلیغ کننده است که به صورت string ذخیره می‌شود. فیلدهای clicks و views نیز به ترتیب برابر با تعداد کلیک‌ها و تعداد نمایش‌های یک تبلیغ کننده است.

لازم به ذکر است که با توجه به قواعد encapsulation که فرا گرفته اید این فیلدها بایستی private باشند.

Advertiser Class Methods:

- getName()
- setName(string name)
- help()
- getTotalClicks()
- getClicks()
- getViews()
- incClicks()
- incViews()
- describeMe()

توضیحات:

متد getName نام تبلیغ کننده را به عنوان خروجی برمی‌گرداند. متد setName نیز یک نام به عنوان ورودی می‌گیرد و با آن نام جدیدی برای تبلیغ‌کننده تنظیم می‌کند.

توابع getClicks و getViews به ترتیب تعداد کلیک‌ها و نمایش‌های یک تبلیغ‌کننده را بر می‌گردانند.

توابع incClicks و incViews به ترتیب تعداد کلیک‌ها و تعداد نمایش‌ها را یک واحد اضافه می‌کنند.

تابع help باید به عنوان خروجی یک توضیح مختصر در قالب یک string در مورد فیلدهای این کلاس به کاربر برگرداند.

تابع describeMe باید وظیفه این کلاس را به عنوان خروجی برگرداند.

تابع getTotalClicks یک متد static است که وظیفه آن این است که به عنوان خروجی مجموع تعداد کلیک‌های تمام تبلیغ کنندگان سیستم را برگرداند.

Ad Class Fields:

- id
- title
- imgUrl
- link
- clicks
- views

توضیحات:

فیلد id شناسه خاص هر تبلیغ است که معمولا از نوع عدد طبیعی است و باید به صورت یکتا برای هر تبلیغ باشد. فیلد title عنوانی است که برای تبلیغ انتخاب می‌شود و از نوع string است. فیلد imgUrl همان آدرس عکس تبلیغ است. فیلد link آدرسی است که کاربر پس از کلیک روی تبلیغ به آن صفحه هدایت می‌شود. فیلدهای clicks و views نیز به ترتیب برابر با تعداد کلیک‌ها و تعداد نمایش‌های یک تبلیغ است.

لازم به ذکر است که با توجه به قواعد encapsulation که فرا گرفته اید این فیلدها بایستی private باشند.

Ad Class Methods:

- getTitle()

- setTitle(string title)
- getImgUrl()
- setImgUrl(string url)
- getLink()
- setLink(string link)
- setAdvertiser(Advertiser advertiser)
- describeMe()
- getClicks()
- getViews()
- incClicks()
- incViews()

توضیحات:

تابع getTitle عنوان تبلیغ را به عنوان خروجی برمی‌گرداند. متد setTitle نیز یک عنوان به عنوان ورودی می‌گیرد و با آن عنوان جدیدی برای تبلیغ تنظیم می‌کند.

توابع getImgUrl و setImgUrl برای دریافت و ست کردن آدرس url عکس محتوا استفاده می‌شود.

توابع getLink و setLink برای دریافت و ست کردن آدرس کلیک محتوا استفاده می‌شود.

توابع getClicks و getViews به ترتیب تعداد کلیک‌ها و نمایش‌های یک تبلیغ را بر می‌گردانند.

توابع incClicks و incViews به ترتیب تعداد کلیک‌ها و تعداد نمایش‌ها را یک واحد اضافه می‌کنند. لازم به ذکر است که با افزایش تعداد کلیک یا نمایش یک تبلیغ نیاز است که تعداد کلیک‌ها یا نمایش‌های تبلیغ کننده مربوط به آن تبلیغ نیز افزایش یابد.

تابع describeMe باید وظیفه این کلاس را به عنوان خروجی برگرداند.

با توجه به این که این دو کلاس که در بالا معرفی شدند، فیلدها و توابع مشترک زیادی دارند بهتر است که هر دو فرزند یک کلاس دیگر باشند تا قواعد وراثت که در شی گرایی به آن برخوردیم رعایت شوند. لذا یک یک کلاس پدر با نام BaseAdvertising تعریف کنید که کلاس‌های Advertiser, Ad از آن ارث ببرند. و توابع مشترک را در کلاس پدر بنویسید و توابعی که نیاز است که در هر یک از فرزندها متفاوت باشد را در فرزندان override کنید. همچنین ممکن است فرزندان توابعی داشته باشند که در پدر نیست مثل تابع help در کلاس Advertiser.

لازم به ذکر است جهت تمیز بودن کد بهتر است که هر یک کلاس‌ها در یک فایل جداگانه تعریف شود و در مواردی که نیاز به استفاده از کلاس‌ها یا شی های یک فایل دیگر دارید از import استفاده کنید. به همین منظور چهار فایل ad.cpp, advertiser.cpp, baes_model.cpp, main.cpp را تعریف کنید. در فایل main.cpp تابع main خود را بنویسید و هر یک از ۳ کلاس را در فایل مربوط به خود بنویسید.

در نهایت کدهای زیر را در تابع main نوشته و خروجی‌های خط‌های مختلف آن را چاپ کنید.

```
baseAdvertising = new BaseAdvertising();
advertiser1 = new Advertiser(1, “name1”);
advertiser2 = new Advertiser(2, “name2”);
```

```
ad1 = new Ad(1, “title1”, “img-url1”, “link1”, advertiser1);

ad2 = new Ad(2, “title2”, “img-url2”, “link2”, advertiser2);

baseAdvertising.describeMe();

ad2.describeMe();

advertiser1.describeMe();

ad1.incViews();

ad1.incViews();

ad1.incViews();

ad1.incViews();

ad2.incViews();

ad1.incClicks();

ad1.incClicks();

ad2.incClicks();

advertiser2.getName();

advertiser2.setName(“new name”);

advertiser2.getName();

ad1.getClicks();

advertiser2.getClicks();

Advertiser.getTotalClicks();

Advertiser.help();
```

کدهای خود را در اکانت گیت هاب‌ای که در قسمت اول ساخته بودید push کنید.