جلسه ی 12 اموزش php

در این جلسه برای autoload کردن کلاس ها چند کلاس را در فولدری به نام inc ذخیره می کند. که عده ای از ان ها کلاس هایی هستند که در جلسات قبل نوشته شده بودند.

از کلاس students که در جلسه ی قبل استفاده کردیم استفاده کرد.

کلاس point را به کلاس point2d تغییر نام داد و کلاس point3d را هم ایجاد کرد که از ان ارث بری می کند.

ادامه ی بحث شی گرایی:

فیلد ها و متد های ایستا:

بدون نیاز به ایجاد شی و با نام کلاس قابل استفاده اند.

Self:: در staticها نقش همان this$ را در غیر استاتیک ها دارد

متدهای استاتیک باید public باشند و الا نمی توان ان هارا با نام کلاس صدا زد

در متد استاتیک یک کلاس نمی توان به فیلدهای ان کلاس دسترسی پیدا کرد زیرا نمی توانیم در متد استاتیک از $this استفاده کنیم چون صدا زدن ان متد از طریق خود کلاس بوده و شی وجود ندارد تا $this به ان اشاره کند

سازنده نمی شود استاتیک باشد

وقتی یک فیلد استاتیک داریم بهتر است دست یاب به ان را هم استاتیک بگزاریم.

echo dirname(\_\_File\_\_);

مسیر فایل را میدهد مثلا دستور فوق را در یکی از برنامه هایمان نوشتیم داد:

C:\xampp\htdocs\php

چند ریختی:یعنی اینکه کلاس فرزند متدی از کلاس والد را لغو کند یا به قول این مرد بازنویسی کند این مطلب موجب می شود یک متد را اگر با شی1 از کلاس والد صدا بزنیم یک خروجی داشته باشد و وقتی متد کلاس فرزند با همان نام را با شی2 ازکلاس فرزند صدا بزنیم خروجی دیگر. اقای شهرکی این مطلب را چند ریختی می داند که البته احتمالا اشتباه می کند .

چگونه جلوی بازنویسی یا overwrite شدن یک متد را می گیریم و اعلان می کنیم این نسخه اخرین نسخه از این متد است؟

با کلمه ی final قبل از نام متد

Final public function display(){

}

کلمه ی کلیدی final قبل از نام کلاس موجب میشود دیگر کلاسی نتواند از ان ارث بری کند یعنی دیگر ان متد بدون فرزند میشود نسلش منقرض می شود

Instanceof :

ایا شی ای از نوع کلاسی هست یا نه؟

ایا شی ای به هر نحو حتی به طور غیرمستقیم یعنی از طریق ارث بری امکانات کلاسی را دارد یا نه؟

دقیقه ی 1:24:00

کلاس چکیده یا abstract:

یک کلاس abstract می تواند یک کلاس abstract دیگر را پیاده سازی کند

<?php

Abstract class shape{

Public abstract function shape();

}

?>

حتی اگر یک متد abstract داشته باشیم باز هم باید کلاس را از نوع abstract اعلان بکنیم

واسط ها

کلمه کلیدیه is\_a هم مثل instaneof عمل می کند

هر کلاس فقط می تواند از یک کلاس مجازی ارث بری کند اما محدودیتی در پیاده کردن واسط ها ندارد

-کلاس مجازی و متدمجازی را نمی توان final کرد

-متدهای مجازی در کلاس مجازی را نمیتوان private کرد زیرا قراراست از ان ها ارث بری شود اما متدهای واسط میتوانند private باشند زیرا در واسط ها ارث بری مطرح نیست بلکه پیاده سازی مطرح است

تا دقیقه ی 70 خوادیم