

١٥

أكتوبر

٩ ربيع أول ١٤٤٣

الجمعة

باب ١٧٢٨

Friday
15
October

OOP Session ⑤

8

- ٩١- class , object , this , construct , destruct
- ٩٢- inheritance and polymorphism.
- ٩٣- access modifiers .
- ٩٤- Encapsulation .
- ٩٥- Abstraction and interface
- ٩٦- static method and property
- ٩٧- method chaining
- ٩٨- Namespaces .
- ٩٩- SOLID principles .

2

3

4

5

6

7

Fri.	Sat.	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Su.	
١	٢	٣	٤	٥	٦	٧	٨	٩	١٠	١١	١٢	١٣	١٤	١٥	١٦	١٧	١٨	١٩	٢٠	٢١	٢٢	٢٣	٢٤	٢٥

Thursday
14
October

الخميس

باب ١٧٨ ق

① class , object :-

(@) اية الفرق بين الـ oop و procedural

الاثنتين عبارة عن طرق لكتابة الـ code / program

الفرق :-

الـ procedural : هي طريقة لكتابة الكود يتبعها التقسيم

functions

الـ oop : هي طريقة لكتابة الكود بـ كل الحالات الواقع

يستخدم التقسيم objects من data و كل

functions

الـ oop اسهل و انيط من procedural لأن

طريقة كتابة الكود كـ oop احترافية لـ oop

الـ oop يـ simluate reality من الكود.

١- الـ oop يـ بناء على classes حاميات class والـ object

٢- الـ object من التـ language ما هو data type يـ خزن فيه

values / تـ values data

٣- الـ oop يـ من الـ properties variables بـ

methods بـ functions

٤- الـ oop يـ reality simulate

Wednesday
13 October

١٣

أكتوبر

٧ ربیع اول ١٤٤٣ھ

الأربعاء

٣ باب ١٧٣٨

٧) ابيه الفرق بين الـ class والـ object

٨) الـ class هو مجردة أو مطلقة template مفخن بتحتها
عما صفات وتفاصيل الـ object مطلقة.

٩) الـ class هو مجرد مطلقة أو وصف
عما للـ object بالتأكيد ميادين حيز عن
الـ memory.

١٠) الـ object هو السمة المادية الـ class طبعناها من

الـ class، سمة مادية يعني الـ
غير data يعني يتأخذ غير من الـ memory.

١١) الـ class هو مجرد وصف عما.

١٢) الـ object هو السمة المادية من الـ class.

١٣) أمثلة: انسان هو مجرد وصف عما انسان هو

الـ person يعني الشخص من الـ person.

١٤) methods هي الـ functions التي تتأخذ من data.

١٥) عربية هو مجرد وصف عما، إنما person من

الـ person الذي منه.

١٦) كل object سيكون مستقل عن الـ class.

١٧) methods و properties يحتوى على object.

١٨) كل object يحتوى على methods و properties.

Fri.	Sat.	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.							
١	٢	٣	٤	٥	٦	٧	٨	٩	١٠	١١	١٢	١٣	١٤	١٥	١٦	١٧	١٨	١٩	٢٠	٢١	٢٢	٢٣	٢٤	٢٥	٢٦	٢٧	٢٨	٢٩	٣٠

Tuesday
12
October

١٢

الثلاثاء

أكتوبر

باب ٤٧٣٨

١٤٤٣ هـ ربیع اول

٦) ایہ ہو $\$this$ ؟

یہ keyword عبارت عن placeholder بیتال ویختط مکانہ object کا بھسوٹہ۔

ہمارے وظیفتہ ہے پیشوار کے ال current object property کی کیا نظر رکھو تو اسے اور method کی کیا نظر رکھو اسے اسے اسے اسے اسے object کا بھسوٹہ تصور کریں۔

منقرض نہیں کیا method دا انہی property کیا $\$this$ کیا خواہ ال

٧) ایہ ہیں ال magic methods

کوہ لونگھم سے كل تسلیقائی calling methods کیا جائیں اور جیسا کہ میرے مارے کیا جائیں۔

امنگھے ال constructor و ال destructor و ال builder کیا جائیں۔

وہاں بھیجے ال data و ال data بھیجے ال data و مانہہ بین ال object و ال object بینہوں۔

Monday
١١
October

أكتوبر
١٤٤٣ هـ ربيع أول

الإثنين

١٢٣٨٦

٩) الفرق بين constructor وال destructor

- ٨) باستخدام `__construct()` وال `__destruct()` يمكن
- نها `call()` object أو `built-in method` ويتطلب
- ٩) بكل تلاقى لوحدهم عن مرحلة معينة من ال `object`

١٠) `data` أو `constructor` وال `destructor` واحد بخط `data`
والثانى بـ `destructor`.

١١

`--construct()`, `--destruct()`

١٢

- ١) `object` أو `call` أو `script` يتعلّق `call` عن ٣ صفات
- ٢) `unset` `object` يمس `exit` / `unset` `script` يخلص.
- ٣) `exit` `script` يتعلّق `exit` أو `script`

٣

٤) `initial value` من `object`

٤

٥) `data` من `object` يمكن `data` من `object` أو `script`

٥

٦) `oop` كنائة الكود

- ٧) `procedural` وال `object-oriented` من `oop`
- ٨) `encapsulation` / `polymorphism` / `inheritance` من `oop`
- ٩) `oop` `concepts` إلى `oop` `concepts`

Fri.	Sat.	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Sunday
10
October

الاحد

اكتسوبر

دوينج اول ١٩٩٣

٢٠١٧٣٨

• object چوں خیجے class جیسے نہیں (@)

```
class User {  
    // properties  
    var $name;  
    public $age;  
    // methods
```

function __construct(\$name, \$age){

\$this->name = \$name;
 \$this->age = \$age;

}

function sayHi(){

echo "Hello \$this->name";

}

}

\$Ahmed = new User();

\$Ahmed->name = 'Ahmed';

\$Ahmed->age = 15;

\$Ahmed->sayHi();

\$bl7 = new User('bl7', 17);

\$bl7->age = 25;

\$bl7->sayHi();

member \$'s access :-

• من object ہے ()

→ arrow ہے (>)

• dot ہے (.)

الحمد لله رب العالمين والصلوة والسلام على سيد العالمين أجمعين سلطان العرش العظيم الإمام محمد بن عبد الله بن عبد الرحمن الباقر عليهما السلام

Saturday
9

October

٩

اكتوبر

٣ ربیع اول ١٤٤٣ھ

السبت

٢٩ سوتو ١٧٣٨ق

٧) اهم احقال لغة برمجة ايل داپ (OOP)

لو اتوفر خمس اى principles لـ OOP
 وهم :
 ٨ inheritance
 ٩ polymorphism
 ١٠ encapsulation
 ١١ abstraction

من خرط ان اللغة يمكن فنر .
 تكون OOP ، بالمثل ان ذكر مفهوم
 ١٢ object-based classes لذرة
 مثل class-based

١) هل الکرسی الک انته قاسی
 ٢) ولا class و class

٣) object

لأنه ~~شيء~~ ماض لـ حيز ، إنما ال class هو
~~شيء~~ معنوى ، مجرد وصف عالم الکس ، المادي (objects)

٤) اهم امر في OOP انه يجعل simulation
 ٥) الواقع reality في الکور الرجس ، يبع ما نراه في الواقع
 ٦) کائنات ، يمكن نهائ کور بوجس ، وای ، هی بیگلوون
 ٧) من ، خصائص او صفات الک هی ال کور variables
 و افعال او action الک هی ال functions

Fri.	Sat.	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Friday
8 October



اكتوبر
٢ ربیع أول ١٤٤٣ هـ

الجمعة

٢٨ قوتو ١٤٤٣

inheritance & polymorphism

① ايجاد مفهوم inheritance

هو احد مبادىء OOP، فكرته قائمة على ان properties احنا نعبر عنها class يورثه الى class ثان وال methods يتبع class ثان.

الميزة من inheritance هي reusability، ان احنا نستطيع تكرار الکود.

② ايجاد مفهوم inheritance

(1) من يخرج الحجات المتركة بين classes ومتطرفة من class واحد يكون الاب.

(2) يستخدم أمر extends بين class يورث / جيد / يتجزئ من parent class extends من

final Keyword

هو keyword ينطوي قبل class معين او method معين.

ينطبق قبل class مكتوب في المقدمة اي انه يورث منه override

Thursday
V

اكتوبر

١ ربیع أول ١٤٤٣ھ

الخميس

٢٧ قتوت ١٤٣٨ق

October

٧ اية هـ در الـ polymorphism (Q)

٨ polymorphism یعنی ساخت الا - کال .
دو نفوس ایماجر لکن بارہ کال مختلفه .
دو نفوس ال method لکن ب method مختلفه .

٩ ١٠ ایام مبارس ال oop ذکریتے قائمۃ
لیں ال classes method ، لکن
کل واحد برقی بارہ کال مختلف .

١١ ١٢ ایة ال oop چو منہ الا - کال

یعنی بیکروں میں اپنے class لکن اولادی
بینندگی بارہ کال مختلفہ .

١٣ ١٤ ایة انواع ال polymorphism و ایة الفرق یعنی

١٥ overriding و overloading

١٦ ١٧ ایة overriding : چو نہیں ال method لکن ورثتہ
implementation لکن ب مختلفیں ؟

١٨ inheritance یعنی ال ایسا نہیں
نسل ۴ method ایسا وارثتہ .

١٩ ٢٠ ایة overloading : چو نہیں ال method لکن ب مختلفیں
parameters لکن ب مختلفیں .

٢١ ٢٢ overloading میں ال php

Fri.	Sat.	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

الأربعاء

٢٩ سبتمبر ١٤٢٨

③ access modifiers :

ج) ايجاد صيغة

هي كلمات أو keywords ينطوي قبل الـ property أو visibility قبل الـ method يمكن تغييرها بحسب اقتراح او مقدار خصوصية فائدة

ج) ايجاد علوا على access modifiers

يمكن نقل الـ changes في الـ properties

ج) وظيفة الـ access modifier

تحت الـ الماكن الـ ذكر نتوات في الـ property أو الـ method

ج) ايجاد الـ الماكن الـ ذكر نحصل على الـ property والـ method

قابلية الوصول من شخص class	child class	خارج الـ class بيان من او object بتاجه
private	✓	✗
protected	✓	✗
public	✓	✓

Tuesday

5

October

أكتوبر

٢٨ صفر ١٤٤٣

الثلاثاء

٢٥ سبتمبر ١٤٤٣

Q) اية الفرق بين الـ private , protected , public ؟

8) public يعني نقدر نستعمله في كل صفات

9) protected يعني نقدر نستعمله في كل صفات class

والـ child class الى الابنوية بس .

10) private يعني نقدر نستعمله في الـ class بتاتحة فقط .

11) ازاً خرد انهم في modifier

12) القاعدة بتقول ابداً بالاضيق بعد كل دالة تابعه

توسيع ايجي و سعر

13) يعني صفاتي أى private property في البداية

بعد كدة لما يتغير توسيع اى visibility بتاتحة

ابعد و سعر

14) ~~ابعد و سعر~~

15) ~~ابعد و سعر~~

16) ~~ابعد و سعر~~

17) ~~ابعد و سعر~~

18) ~~ابعد و سعر~~

Fri.	Sat.	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

٢٩

سبتمبر
٢٢ صفر ١٤٤٣

الأربعاء

١٩ سبتمبر ١٤٢٨

Wednesday
29

September

④ Encapsulation:

٨

ما هي حوار ال encapsulation ؟

٩

ال encapsulation هو ابراص مهارى ال oop فكرته قائمة على حماية ال data عن طريق تغليفها.

١٠

ما هي نتائج ال encapsulation ؟

١١

أولاً : التزمن من ال encapsulation هو حماية ال data.

١٢

ثانياً : بنقل ال encapsulation عن طريق :-

١

١) يجمع properties وال methods .

٢

class مرتبطين بعضهم ببعض وامثلة ال class

٣

٣) بيني ال modifier لـ properties

٤

٤) هناك visibility ستكون

٥

٥) صدور ال class الى ملائتها فقط .

٦

٦) بنقل getter method

٧

٧) private property

٨

٨) بنقل setter method

٩

٩) private property

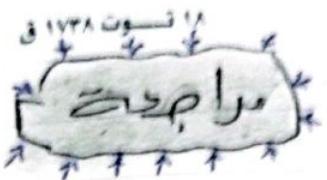
Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu
١	٢	٣	٤	٥	٦	٧	٨	٩	١٠	١١	١٢	١٣	١٤	١٥	١٦	١٧	١٨	١٩	٢٠	٢١	٢٢	٢٣

Tuesday
28 September

٢٨

سunday

الثلاثاء



إيه الذكره من الـ oop
الذكره من الـ oop ان احنا بنتعامل مع البرججت
(من اشياء ذي الاستياء الـ oop بنتعامل معاها يومياً من الحياة الواقعية (الانسان ، الحيوان ، السيارة ...))
إيه الفرق بين الـ class والـ object

الـ class ما صو الـ template بتحمل الاوصاف والـ actions
الـ shir مصبع بي وكل ما او مخرج بيون data .
الـ object هو الـ class في data الخليله.

إيه هـ الصفات الـ classy للـ oop
إيه هـ المبادئ الـ classy للـ oop

① الـ oop هو encapsulation يعني بجمع الخصائص والـ actions al لوحـم علاقـة بـعـضه بـحـمة وـلـدة الاوصـف class وبيقدر حـجم الـ visibility بتـاعة الـ data الـ class فـيـه

② الـ oop هو inheritance يعني الـ class يـورـثـ الخـصـائـصـ وـالـاـفـعـالـ بـتـاعـةـ لـ class ثـانـيـ ، بتـاعـهـ نـهـنـجـلـ تـكـلـارـ الرـكـودـ.

③ الـ oop هو polymorphism او متعدد الـ action كل بـوـينـ الـ classes مختلف يـورـثـواـ من نفس الـ class ، لكن كل class بيـعـنـ بيـفـدـ الـ action الـ class وـرـثـهـ بـكـلـ مـخـتلفـ.

٢٧

سبتuber

٤ شهر سبتمبر ٢٠٢٣

الإثنين

١٧ سبتمبر ٢٠٢٣

Monday
21
September

⑤ Abstraction :

abstraction يعني التجريد، يعني احنا نفترض
الشيء وينفصل عن الشخص التفاصيل الكثيرة
عن مهنة بالبساطة.

الاندماج هو كييفية الاستخدام فقط
بروى الدليل من تفاصيل.

الoop هو object أو class يمكن مكتوئاً عبارة عن اسم
functions واصناً من مارفين implementation عامل ازاي.

يعني احنا عارفين اسم الـ function لكن مني عارفين
الـ implementation بيتانية، وهو نوع التجريد أو الشيء
يعني سهل التفاصيل المبرحة وركذ على الاستخدام.

(١) اما الـ oop يخدم مبدأ الـ abstraction

من الـ oop يختلف نوعين من الـ classes ينبعوا
من مبدأ الـ abstraction وهما: ① الـ abstract class ② الـ interface

ذلك التجريد، ان احنا ندخل التفاصيل المبرحة والغامضة
لك احنا لست من عارفين مستشفى ازاي !!!

Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
١	٢	٣	٤	٥	٦	٧	٨	٩	١٠	١١	١٢	١٣	١٤	١٥	١٦	١٧	١٨	١٩	٢٠	٢١	٢٢

RV

سبتمبر
٢٠ صفر ١٤٤٣ هـ

الإثنين

١٧ Tuesday ١٩٨١

Monday
27
September

⑤ Abstraction !

abstraction يعنى التجريد، يعنى اصحابنا يعرضون^٧ خدمة^٨، ويستبدلون^٩ عن الشخص^{١٠} التخاطل^{١١} الذي^{١٢} تدور^{١٣} سهرة^{١٤} بالساعة^{١٥} بالآخر.

ياصحاحاً^{١٦} abstraction هو كميته^{١٧} الاستخدام فقط^{١٨} بدون الدخول^{١٩} من^{٢٠} تفاصيل^{٢١}.

oop هو^{١١} abstraction^{١٢} object^{١٣} أو^{١٤} class^{١٥} يمكن^{١٦} سكوئوا^{١٧} مشاركة^{١٨} عن^{١٩} اصحاب^{٢٠} functions^{٢١} واصحاحاً^{٢٢} من^{٢٣} عارفين^{٢٤} implementation^{٢٥} عامل^{٢٦} ازاي^{٢٧}.

يعنى^١ اصحابنا^٢ عارفين^٣ اسم^٤ الـ function^٥ لكن^٦ من^٧ عارفين^٨ او^٩ implementation^{١٠} او^{١١} التجريد^{١٢} او^{١٣} التبديد^{١٤} يعنى^{١٥} استبدل^{١٦} التفاصيل^{١٧} المبرهنة^{١٨} ورثث عالاستخدام^{١٩}.

٣) abstraction في oop يخدم مبدأ^١ الـ^٢ ازاي^٣ الـ^٤ oop^٥

٦) في oop يمثل^٧ متعدد^٨ من^٩ classes^{١٠} ينفذ^{١١} مبدأ^{١٢} الـ^{١٣} interface^{١٤} وهو^{١٥}: ① الـ^{١٦} مبدأ^{١٧} الـ^{١٨} abstraction^{١٩} ② الـ^{٢٠} abstract class^{٢١} ③ الـ^{٢٣} class^{٢٤}

٧) ذكرة التجريد^١ لاصحاحاً^٢ من^٣ التفاصيل^٤ المبرهنة^٥ والخامفاته^٦
لذلك^٧ اصحابنا^٨ من^٩ عارفين^{١٠} يستبعد^{١١} ازاي^{١٢} !!!

Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

Sunday
26
September

٢٦

سبتمبر
١٩ صفر ١٤٤٣

الأحد

١٧٣٨ - ١٩

① abstract class

إيه هو الـ abstract class

هو class منفرد ليس نعمل منه object ! . لكن نعمل
موريث منه وبتحتو على الأقل ١٠ abstract method

إيه هو الـ abstract method

implementation / body (معرفة المنهج) named method
signature (معرفة المنهج) هو abstract method
خطط بدون body .
الـ abstract class تكون سبب لـ abstract method
أيـ abstract class يرث معرفة من class .
كل abstract methods لكل class فـ implementation

إيه الغرض من الـ abstract

الغرض الـ abstract هو تنظيم للأمور معاً أكثر

ـ لا نعمل object من abstract class
ـ لا نعمل implementation method لـ abstract class
ـ لا نعمل datatype يدخل إلـ abstract class
ـ من خـ abstract class (interface)
ـ لا نعمل نـ developer لـ implement abstract method
ـ معين من خـ abstract

٥٠

سبتمبر
١٨ صفر ١٤٤٣

السبت

١٥ سפטمبر ١٤٤٣

Saturday
25
September

7. $\left\{ \begin{array}{l} \text{abstract class} \\ \text{abstract class Bank} \end{array} \right.$ نظر ١٥/١ (Q)
8. $\left\{ \begin{array}{l} \text{protected } \$x; \\ \text{protected } \$y; \end{array} \right.$
9. $\left\{ \begin{array}{l} \text{abstract function } m1(); \\ \text{abstract function } m2(int \$a); \end{array} \right.$
10. $\}$ نظر ١٥/٢ (Q)

\hookrightarrow type hinting

11. $\left\{ \begin{array}{l} \text{دو عبارت نوع الـ argument} \\ \text{الـ argument المـ required} \\ \text{او الـ argument المـ optional} \\ \text{او الـ argument المـ nullable} \\ \text{او الـ argument المـ mixed} \end{array} \right.$ نظر ١٥/٣ (Q)
12. $\left\{ \begin{array}{l} \text{ex) function test(string \$txt)} \\ \text{---} \\ \text{---} \end{array} \right.$ نظر ١٥/٤ (Q)
13. $\left\{ \begin{array}{l} \text{---} \\ \text{---} \end{array} \right.$ نظر ١٥/٥ (Q)
14. $\left\{ \begin{array}{l} \text{---} \\ \text{---} \end{array} \right.$ نظر ١٥/٦ (Q)
15. $\left\{ \begin{array}{l} \text{---} \\ \text{---} \end{array} \right.$ نظر ١٥/٧ (Q)
16. $\left\{ \begin{array}{l} \text{---} \\ \text{---} \end{array} \right.$ نظر ١٥/٨ (Q)
17. $\left\{ \begin{array}{l} \text{---} \\ \text{---} \end{array} \right.$ نظر ١٥/٩ (Q)
18. $\left\{ \begin{array}{l} \text{---} \\ \text{---} \end{array} \right.$ نظر ١٥/١٠ (Q)

Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

Friday
24
September

٢٤

الجمعة

١١ سبتمبر ١٤٤٣

سبتمبر
١٧ صفر ١٤٤٣

② interface / العقد

• هو بتحول إلى class blueprint هو وال class

• class blueprint هو interface

• object في php هو عامل أو العقد أو الاتصال الذي

• يحول object إلى datatype

• class يحول object إلى datatype

• كل class يحول object إلى datatype

• كل class يحول object إلى datatype

• كل class يحول object إلى datatype

الأربعاء الخميس الجمعة السبت الأحد الاثنين الثلاثاء الأربعاء الخميس الجمعة السبت الأحد الاثنين الثلاثاء الأربعاء الخميس

١ ٢ ٣ ٤ ٥ ٦ ٧ ٨ ٩ ١٠ ١١ ١٢ ١٣ ١٤ ١٥ ١٦ ١٧ ١٨ ١٩ ٢٠ ٢١ ٢٢ ٢٣ ٢٤ ٢٥ ٢٦ ٢٧ ٢٨ ٢٩ ٣٠ ٣١

٢٣

سبتمبر
١٦ صفر ١٤٤٣

الخميس

١٣ أكتوبر ١٧٣٨

Thursday
23
September

7 دو عقد أو اتفاق بينه وبين ال interface
ال اتفاق ينص على أن ال interface تمثل أي
فألا تزال class implements كل ال implementation
of name method class في target interface.
الا في

8 يساير ال interface تكون شاملة في فحص صيغة
أي ال class تبعها، يعني ال class يتحقق كل ال requirements
الا في جرأة.

12 what are the similarities between them?

(ex) ~~both are objects with their own methods~~

1 interface MyDatabase{

2 function connectToDB();

3 function queryDB();
}

4 class MySQL implements MyDatabase{
MySQL is the Database class.

5 function connectToDB(){
6 {
7 // needs the information, etc.

8 function queryDB(){
9 {
10 }

11 }
12 }

13 }
14 }

15 class SQLServer implements MyDatabase{
SQLServer is the Database class.

Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

Wednesday
22
September

٢٢

الأربعاء

١٤ سبتمبر ١٩٧٨

الستين

```
function runBankDb ( MyDatabase $db ) {
    $db->connectToDB();
    $db->queryDB();
}
```

\$obj1 = new MySQL();

\$obj2 = new SQLserver();

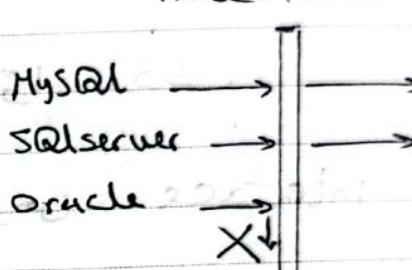
\$obj3 = new Oracle();

MyDatabase
interface

runBankDb (\$obj1); ✓

runBankDb (\$obj2); ✓

runBankDb (\$obj3); X



ما نجح في تضمينه أو نتمكن من إلزام class بالـ interface

methods بين class أو بين class والـ interface ممكن.

لا يتحقق ممكناً في class الذي ي Implements

interface كل الـ methods implement

interface (In case of Inheritance)

ويمكن إثبات ذلك

في سطح المكتب

أولاً نكتب class

ثانياً نكتب class

ثالثاً نكتب class

الرابع الخطيب الجمعة السبت الأحد الاثنين الثلاثاء الأربعاء الخميس الجمعة السبت الأحد الاثنين الثلاثاء الأربعاء الخميس الجمعة السبت الأحد الاثنين الثلاثاء الأربعاء الخميس الجمعة السبت الأحد الاثنين الثلاثاء الأربعاء الخميس

٣٠ ٢٩ ٢٨ ٢٧ ٢٦ ٢٥ ٢٤ ٢٣ ٢٢ ٢١ ٢٠ ١٩ ١٨ ١٧ ١٦ ١٥ ١٤ ١٣ ١٢ ١١ ١٠ ٩ ٨ ٧ ٦ ٥ ٤ ٣ ٢ ١

٢١

سبتuber
١٤ صفر ١٤٤٣ هـ

الثلاثاء

١١ سبتمبر ١٤٤٣ هـ

Tuesday
21
September

٧ الفرق بين الـ interface و abstract class .

٨ methods و properties يحتوي abstract class على و و (abstract) named methods .

٩ . فقط named methods يحتوي interface على .

١٠ . class يورث class فقط .

١١ . interface يقتصر في implement على class .

١٢ . interface من غيره و abstract class على object .

١٣ . class الذي يورث من abstract class لازم ينفذ كل اـ named methods .

١٤ . class الذي implement على interface (يعنى class يحقق العقد / الاتفاقي مع اـ interface) لازم ينفذ كل اـ named methods .

١٥ . class ينبع من abstract class على ميزة اـ object لكن يورث فقط .

١٦ . class ينبع من interface على ميزة ان class ينفذ اـ methods .

Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu							
١	٢	٣	٤	٥	٦	٧	٨	٩	١٠	١١	١٢	١٣	١٤	١٥	١٦	١٧	١٨	١٩	٢٠	٢١	٢٢	٢٣	٢٤	٢٥	٢٦	٢٧	٢٨	٢٩	٣٠

Monday
20 September

٢٠

الاثنين

١٩٩٨ م

سبتمبر
١٤٣٢ هـ

١) اسئلة من خصم الـ interface

- ١- لو عايزين ندخلن اى الـ class معين
- ٢- لو عدنا datatype + argument في تقبل function
- ٣- مختلفة كل مرتبة او عايز احص اية الـ datatypes امتحنة
- ٤- interface كل parameters من نوع interface
- ٥- classes معين، حبيهول كل اى implement
- ٦- interface implement

٢) اية الفرق بين الـ interface و الـ class

- ١- object يتحقق من interface ومن غيره properties في

- ٢- object يتحقق من كل الـ class
- ٣- كل اى interface يتحقق من كل الـ class
- ٤- هو يعني اية الـ methods كل لازم اى class يستخدمها.

٣) اية صور الـ interface

- ١- class يتحقق من interface

- ٢- abstract class يحتوي على abstract method

- ٣- abstract class يتحقق من abstract data type

- ٤- implementation يحتوي على methods

١٩

سبتمبر

١٢ صفر ١٤٤٣ هـ

الأحد

١٧٣٨ ق

Sunday
19
September

٦ static method and property

٨ static method الـ static method

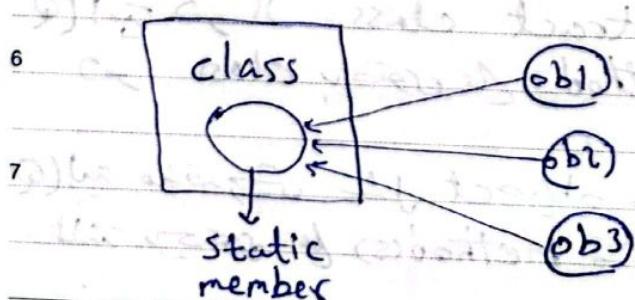
- ٩ نقدر نعمل على class call static method
- ١٠ الـ static من غير ما نعمل على object
- ١١ الـ static الـ static الميزة الـ static تكون متركة ما بين objects
- ١٢ الـ static خاص بـ object

٩ static property الـ static property

- ١ ينحصر في object معين ، بالتالي
- ٢ نقدر نعمل على access static على class
- ٣ من غير ما نعمل على object

٣ static Keyword الـ static keyword

- ٤ نصور static property ، static method على كل object
- ٥ ينحصر في كل object من objects



Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	
١	٢	٣	٤	٥	٦	٧	٨	٩	١٠	١١	١٢	١٣	١٤	١٥	١٦	١٧	١٨	١٩	٢٠	٢١	٢٢	٢٣	٢٤

Saturday
18
September

IA

سبتمبر
١٤٤٣ هـ
١١ صفر

السبت

١٧٢٨ ق

- (ما هو ؟)
- ١- اصحاب الـ static يتكونون من تكملة معايير كل او call object معيين ، بالتالي يمكننا الدخول الى بناء class ثم الدخول الى static member.
 - ٢- بناء class مع الـ static member يبدأ بـ #this وليس self:: keyword.

(ex)

```
١- اقام اد شفرة تخدم خارج الـ class كالتالي . public
class A{
    public static $x = 'hi';
    public static function m1() {
        public function m2() {
            self:: m1();
            return self::$x;
        }
    }
}

class B{
    public function m1() {
        A::m1();
    }
}
```

echo A::\$x;

echo A::m1();

الارجاء المطبي الجملة السبت الاحد الاثنين الثلاثاء الارجاء المطبي الجمعة السبت الاحد الاثنين الثلاثاء الارجاء المطبي

الارجاء المطبي الجمعة السبت الاحد الاثنين الثلاثاء الارجاء المطبي الجمعة السبت الاحد الاثنين الثلاثاء الارجاء المطبي

١ ٢ ٣ ٤ ٥ ٦ ٧ ٨ ٩ ١٠ ١١ ١٢ ١٣ ١٤ ١٥ ١٦ ١٧ ١٨ ١٩ ٢٠ ٢١ ٢٢ ٢٣ ٢٤ ٢٥ ٢٦ ٢٧ ٢٨ ٢٩ ٢٩ ٣٠ ٣١

IV

سبت

١٩ سبتمبر ٢٠٢٤

الجمعة

١٧ سبتمبر ٢٠٢٤

Friday

١٧

September

{ static method

نعلم

أي

- ٦) دايرن نعمل static method بعماش لو دايرن دايرن properties و methods اور class اور static method اور non-static ميقياش اور static method اور this لارنه ميغيره اس اور static method بعماش
- ٧)
- ٨)
- ٩)
- ١٠)

{ static property

نعلم

أي

- ١١) دايرن نعمل static property معينة ما بين كل ال objects ، وال data ديره ميتحميش معين ، بعن تمامه (كلاس)
- ١٢)

{ static member

نعلم

أي

- ١٣) دايرن خارج ال class اسم class
- ١٤) دايرن خارج ال class self
- ١٥) دايرن داخل ال class parent
- ١٦) دايرن داخل child class parent

5) className:: — ;

6) self:: — ;

parent:: — ;

- ١٧) دايرن operator :: عاريف :: operator دايرن قبليه (::) scope resolution operator دايرن static member اور static class دايرن

Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

رتو

ستمبر
٩ صفر ١٤٤٣

Thursday
16
September

الخميس

٦ ذي القعده ١٤٢٨

(Q) ما هو الفرق بين الـ static العادي والـ method

① الـ method متصل بـ objects مرتدي ما بين كل الـ static method

الـ object العادي يحصل منه سبحة لكل method

② الـ static member مبني على static method غير non-static

" الـ method العادي يقبل أي صاحب non-static أو static

③ الـ method العادي ينادي static method بالـ this كـ member

الـ method العادي للـ object يتعل method العادي object

لأن كل object له سبحة بـ static

④ أين هي الـ static member

في class member static member class member static member

في class member static member class member static member

الـ static member في class member static member class member static member

١٥

سبتمبر
٢٠٢٣ صفر ٨

الأربعاء

٩ سبتمبر ٢٠٢٣

Wednesday
15

September

- ٧ ⑦ method chaining

٨ ٨) method chaining

٩ ٩) programming technique

١٠ ١٠) invoke other methods

١١ ١١) object

١٢ ١٢) technique

١٣ ١٣) object

١٤ ١٤) return

١٥ ١٥) another

١٦ ١٦) readability

١٧ ١٧) object

١٨ ١٨) operations

١٩ ١٩) Martin Fowler said ("

٢٠ ٢٠) أي واحد غير قادر على كتابة كود الكائنات بغيره لكن المبرمج الشهير يعذر يمكن كتابة كود اى جزء

٢١ ٢١) معاذ الله

Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	
١	٢	٣	٤	٥	٦	٧	٨	٩	١٠	١١	١٢	١٣	١٤	١٥	١٦	١٧	١٨	١٩	٢٠	٢١	٢٢	٢٣	٢٤

Tuesday
14 September

١٤

سبت

٢٠١٣/٩/١٤

الثلاثاء

٢٠١٣/٩/١٤

```
(ex) class FakeString {
    private $str;
    function __construct() {
        $this->str = '';
    }
    function addA() {
        $this->str .= $this->str . 'A';
        return $this;
    }
    function addB() {
        $this->str .= $this->str . 'B';
        return $this;
    }
    function getStr() {
        echo $this->str;
    }
}
```

```
$fs = new FakeString();
$fs->addB()->addA()->addB()->addA()->getStr();
```

O/P: 'BABBA'

in constructor is created object & then after
then add B, then A, then B again then A.

Object creation & then calling object.

١٣

سبتمبر

٦ صفر ١٤٤٣ھ

الإثنين

٢٠ سبتمبر ١٩٣٨ق

Monday

13

September

⑧ namespaces :

دورة طريقة ظهرت على بحث كلية التعارض بين الأسماء في الأكواد.

٩

لية الكلمة يعني مشروع بسيكير، الملفات التي يكتب
 ١٠ كلة انت لها libraries ومتغيرات يمكنها في
 ١١ آخر من developers خلال المشروع، وبالتالي يمكن للناس تعيين الـ function أو class الجديدة أو file جديدة بنفس
 ١٢ الأسماء لكن ثانية تأتي في المشروع، وبالتالي يمكن أن يكون هناك衝突 conflict.

لية أكل؟ نستخدم طريقة الـ
 ٢ . namespaces
 ٣ . naming conflicts

٤

لية حول namespaces؟ وازاي يستعمل؟

هو يعني زى صناديق وده يعني بجمع فيه classes
 ٥ . functions والـ functions يمكنه اولو جزء خارج
 ٦ او class او function او موجودة من الصندوق للزم
 يركبته تم الصندوق الـ كما فيه "أولاً".

٧ . naming conflicts دورة يطبع كل دورة
 ٨ . طريقة دورة يطبع كل دورة

Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

Sunday
2
ember

جامعة

جامعة

الأحد

٢٠١٣/١٢/٥

{ namespaces أو namespaces

أولاً: كمثال المصروف الودع بـ namespace Keyword

(ex)

UserA.php

```
namespace A;  
class User{  
function getUser(){  
}
```

UserB.php

```
namespace B;  
class User{  
function getUser(){  
}
```

index.php

```
require-once "UserA.php";
```

```
require-once "UserB.php";
```

ثانياً: مسحات تستخدم لـ code موجود في namespace

namespace B\

backslash

```
$user1 = new A\User();
```

```
$user2 = new B\User();
```

أكملوا بـ \ المصروف

backslash

Saturday

سبتمبر

السبت

١٤٤٣ هـ
٢٠١٨

September

١٤٤٣ هـ

(ملاحظات)

٨- يمكن نصيحة قوله أي عدداً من classes.

٩- دار دار functions .

١٠- لا namespace عامل، نرى الصندوق (و) مجتمع فيه

١١- اصحابه . المترتبة ببعضها .

١٢- كل الاك صناعة خط / هنر و ت من namespace .

١٣- ملف .php .

F1.php
 ١ <?php
 ٢ namespace A;
 ٣ >?
 F2.php
 ١ <?php
 ٢ >?
 ٣ >?

٤ namespace A;
 ٥ >?
 ٦ >?

٧ >?

٨- مثلاً نستخرج اي صادر جداً ال namespace ، لذا نذكر بـ \tum ال namespace أولاده بعد كدة .

٩- namespace \tum ال الصندوق A .

١٠- نرى اقتصر بنطلة متن نسترك كل مرة نذكر بـ +

١١- الى جه .

١٢- use A\User ;
 ١٣- new = لون

Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	
١	٢	٣	٤	٥	٦	٧	٨	٩	١٠	١١	١٢	١٣	١٤	١٥	١٦	١٧	١٨	١٩	٢٠	٢١	٢٢	٢٣	٢٤

Friday
10
September

الجمعة

فیض المکتب

١٧٣٧ ق

1883 Jan. 19

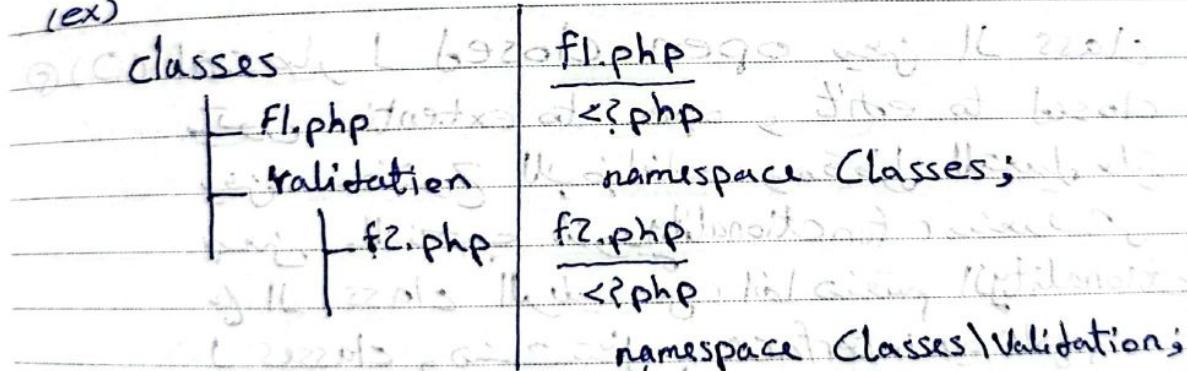
لارم ليس له ايات اخري تدعى بالابيات،
ويكتب classes والـ USL مثل هذا:

(ex) use A\User; X
use B\User; X

use A\user as b1f;
use B\user as ttf;

فهي namespace تعزى على الـ standard بـ php namespace standard الـ
الـ folder يكون تغير تم مسار الـ namespace (الـ folder) الذي يحتوى على الـ php

(ex)



٩

سبتمبر

٢ صفر ١٤٤٣ھ

الخميس

٤ سبتمبر ٢٠٢٤

Thursday

٩

September

⑨ SOLID principles :

٨ اية هو ال SOLID principle (Q)
 اية هو ال SOLID principle (Q)

٩ هنا ملخص عن خمس مبادئ بناء البرمجيات
 ذات نتائج كبيرة تجربة oop و مبادل التعديل
 ملخص

١١ اية هو اضياعار SOLID وشرائط (Q)

١٢ اضياعار لـ single responsibility (S) (Q)
 اى class الواحد يمثل حماقة واحدة بسيطة

١ فلو لغتة او class الواحد يمثل المهام
 قسم المهام في class

٦ اضياعار لـ open-closed (O) (Q)
 يتحول closed to edit و open to extenstions
 يعني مفتوح للإضافات و مقفل للتعديل عليه
 يعني لا يجب شمول functionality بمفهوم class
 مع اى class الرئيسي إنما ينقسم الى functionality
 لـ regular interface و من متصل به classes

٧

Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri
١	٢	٣	٤	٥	٦	٧	٨	٩	١٠	١١	١٢	١٣	١٤	١٥	١٦	١٧	١٨	١٩	٢٠	٢١	٢٢	٢٣	٢٤	٢٥	٢٦	٢٧	٢٨	٢٩	٣٠	

Wednesday
8
September



الأرباع

سبتمبر

١٧٩٧ في

١٤٤٦ هـ

① Liskov Substitution لـ اضطرار لـ

ورقة المبدأ، الـ Liskov Substitution إن أمكن نعمل على inheritance

والمبدأ يقول إن أي object من الـ parent ينبع من الـ child.

يتحقق مكانة object من الـ child.

المبدأ دعوه بـ Inheritance بـ Inheritance اثناء الـ inheritance:

لو الـ parent فيه صفات مثل كل الـ child

ويحتاجها يمكن نفصلها عن parent class

والـ parent الثاني له الـ inheritance يورث الـ parent الرئيس.

لو نعمل override فـ change في عدد parameters

أو نوع parameters أو نوع الـ return

② Interface Segregation لـ اضطرار لـ

ورقة المبدأ، الـ Interface Segregation إن أمكن نعمل على

المبدأ دعوه بـ Interface Segregation لـ implement

المبدأ يقول: لو فيه method من الـ interface

من كل الـ classes يـ implement

تحتاج ، يـ implement interface من مختلف

③ Dependency Inversion لـ اضطرار لـ

الـ maven يقول أوصى خاص class يـ implement

object من class ثانى يمكن معاشر

انما يـ implement على interface

يـ implement لـ objects من classes مختلفـ كـ functions مختلفـ

أوصى تـ functions يـ implement أو يـ implement function من class

من كلـ معاشر، انما function يـ implement وـ interface

Tuesday

7

September

سبتمبر

٢٠ محرم ١٤٤٣ هـ

الثلاثاء

٢ نسيان ١٩٧٧ ق

النحو صيغة

- ٨ ٥: ال class الوارد يجعل حاجة واحدة بس .
- ٩ ٦: ال class يقبل التوسيع وليس التعديل عليه .
- ١٠ ٧: ال class لوصورته ، لازم كل ولاية تكونوا جنبة بالتطبيق ، والا افضلاته لاكثر من parent .
- ١١ ٨: الا interface لازم كل ال classes الاتى يتبعه تكون سكون .
١٢ ٩: يحتاج كل الات قيمة ، والا افضلاته لاكثر من interface .
- ١٣ ١٠: ال class ادعى تخلقه يعتمد على object من class .
١٤ ١١: ثانى بـ كل مبارتر ، انتا ضلالة يعتمد على interface .
- ١٥ ١٢:
- ١٦ ١٣:
- ١٧ ١٤:
- ١٨ ١٥:
- ١٩ ١٦:
- ٢٠ ١٧:
- ٢١ ١٨:
- ٢٢ ١٩:
- ٢٣ ٢٠:
- ٢٤ ٢١:
- ٢٥ ٢٢:
- ٢٦ ٢٣:
- ٢٧ ٢٤:
- ٢٨ ٢٥:
- ٢٩ ٢٦:
- ٣٠ ٢٧:

١) اى متاح خدم interface ومتى

٢) اى بـ خدم ما يكون فيه actions وبيانات موجودة classes متاحة .

٣) لو كان متاح actions الات عاينهم يستخروا ال actions

٤) بعد ملاحظة بعض او ليه نفس الاصف يستخدم abstract class .

٥) لكن لو ال classes دول ماحظوا ملاحظة بعض غير مفهوم

٦) حامض تربطهم ومواهش اصل وادى بيقس يستخدم interface .

d	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu
	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

٦

سبتمبر

٢٩ محرم ١٤٤٣هـ

الاثنين

١١٢٧ق

Monday
6 September

- ① الـ interface عبارة عن عقد فنية مجبرة مترددة على functions التي لها جملة ذات حد مقدر متكامل مع العقد تو سترت أنه ينفذ كل الشرط الذي فيه.
- ② لكن abstract classes لازم الـ classes الذي هنورث منه يكونوا مخربين من أصل وحده ولهم علاقة بعض.

١	٢	٣	٤	٥	٦	٧	٨	٩	١٠	١١	١٢	١٣	١٤	١٥	١٦	١٧	١٨	١٩	٢٠	٢١	٢٢	٢٣	٢٤	٢٥	٢٦	٢٧	٢٨	٢٩	٣٠	٣١
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----