Class Abstraction, Interface and Type Hints

Daryl K Wood

CTO DATASHUTTLE.NET

@datashuttle | www.datashuttle.net | www.linkedin.com/in/datashuttle

Module Overview

Class and method abstraction

Interfaces

Type hints

Type hinting with interfaces

Abstraction

Class and method abstraction

Add new code efficiently

Minimize refactoring

```
abstract class SomeClass{
   //class content ...
}
```

Abstract Class

Add the keyword “abstract” to the class signature.

Abstract Class

Frequently used as a superclass

Abstract classes can inherit from other abstract classes

Cannot be instantiated as an object

Why Mark as Abstract

Provides a measure of control and prevents unintended class instantiation

Used as a class template or a provider of other objects

Abstract Class Best Practice

Mark a class as “abstract” when only used as a base class, or

Otherwise not intended for constructing instance of itself

Where concrete subclasses are extending

Think in terms of a foundation or starting point of other classes

Demo

Implement abstraction in the Form Generator

Up Next

PHP Traits and the abstract method.

The Abstract Method

Declared with the “abstract” keyword

No special implementation as declared

Must be implemented in extending subclasses

Use Me, But...

If you do, you must provide implementation

But Why?

Subclasses provides unique implementation

Method flexability

Classes marked abstract that contain abstract methods

But Wait!

Could subclasses just override?

But what if the code base expects to call the method?

Abstract method ensures availability to the code base

Demo

Abstract method in VehicleBase class

Up Next

Abstract Methods and PHP Interfaces.

PHP Interface

Conceptually difficult to understand

Let's keep it simple

Look at use cases

```
interface SomeInterface{
   //interface declarations ...
}
```

Interface Construct

A construct, like a class or trait, that contains only method declarations and no implementation.

Think of a Contract Stipulation

CONTRACT

FAILURE

t of employment is entered into between (hereinafter referred to

/ Ms

der the terms and conditions of employment below

Commencement of Employment * Effective from

* until either party terminates the contract

* for a fixed term contract for a period of

* days / month(s) / year(s), ending on

2. Position and Section Employed

3. Place of Work

4. Working Hours From hours to hours (days per week)

5. Wages (a) wage rate Basic wages of $ per * day / month

plus the following allowance(s) -

* Meal allowance of $ per * day / month

* Travelling allowance of $ per * day / month

* Attendance bonus of $ per * day / month

(amount) (details of criteria and calculation of payment, if any)

g. commission, tips) (amount) (details of criteria and calculation of payment, if any)

ages / per hour

day and times of normal w

day of th

day

(d

But Why?

Required by a code base dependency

Without the dependency—the code breaks

Another Reason Comes Later

Flexibility with type hinting

Interfaces

Vs.

Abstract Classes

Abstract Classes

Commonly serve as a base class

Can declare and provide method implementation

Only for classes extending the abstract

Interface

Declares methods only

Can be used by any number of classes by implementation

Decision Defined at Application Design

Up Next

Implementation in the form generator application.

Type Hinting

Specifies a data type

Type declaration in a method or function signature

Why is it Important?

Avoids type coercion, or

Type juggling

Ensures the correct data type received and returned

Demo

Type hinting

Type Hinting Availability

Scalar data types
(int, string, float, boolean)

Arrays, callables and interfaces

Up Next

Type hinting with interfaces.

Type Hinting on Interfaces

Ensures correct data type

Passing objects as parameters

Passed objects provide dependent functionality

Dependency injection

Dependency Injection

Type hinting on an object class name

Demo

Type hinting with class names

Up Next

Type hinting with interfaces rather than classes.

Type Hinting on Interfaces (continued)

Continue with the View class type hint for a Plane class instance

Take Aways on Interface Type Hints

Ensure correct objects are passed

Allow for swapping implementation

Much easier to write test code

Up Next

Type hinting with interfaces in the form generator application.

Interface Type Hinting In the Form Generator

Update the Form Generator appliction

Demo

Evaluate for interfaces

Module Summary

Class and method abstraction

Interface construct

Interface use cases

Type hinting to ensure correct types

Interfaces as type hints