Báo cáo tiến độ công việc

Người thực hiện : Trịnh Đình Nam

Chức vụ: Thức tập sinh

Người yêu cầu: Nguyễn An Hưng (Trưởng phòng kỹ thuật)

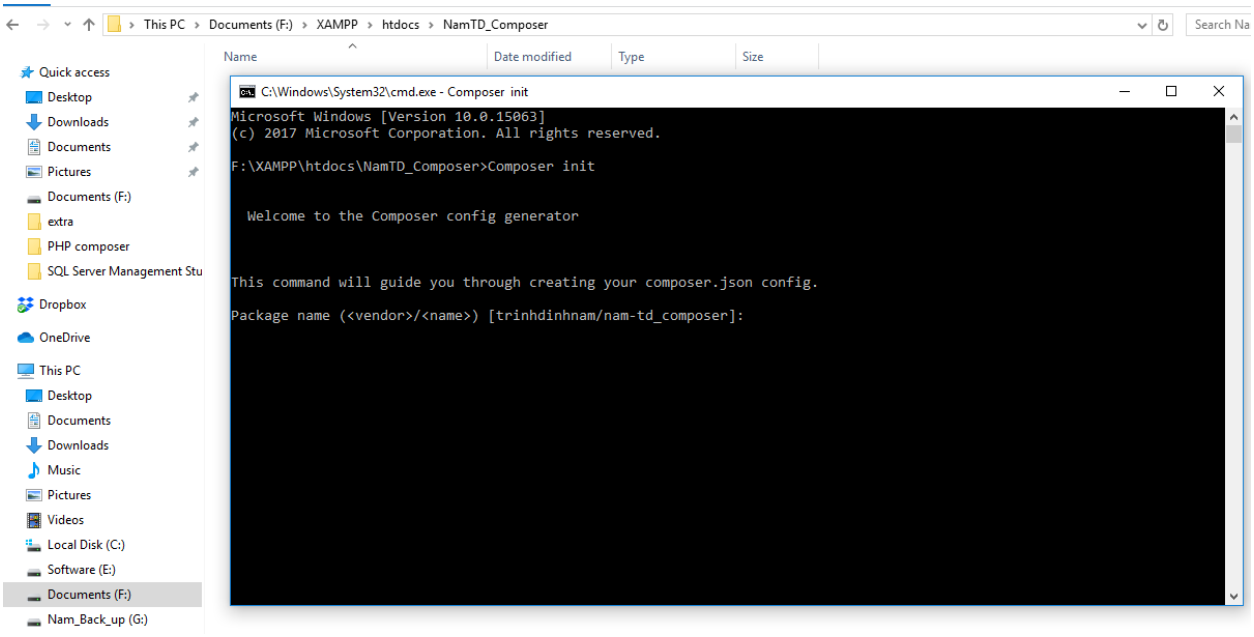
1. **Đầu việc 1: Tìm hiểu về PHP Composer, cách thức tạo Pakage Composer.**

- Tiến độ thực hiện: 100%.
- Thời gian dự kiến: 3 ngày.
- Thời gian thực hiện: 2 ngày.

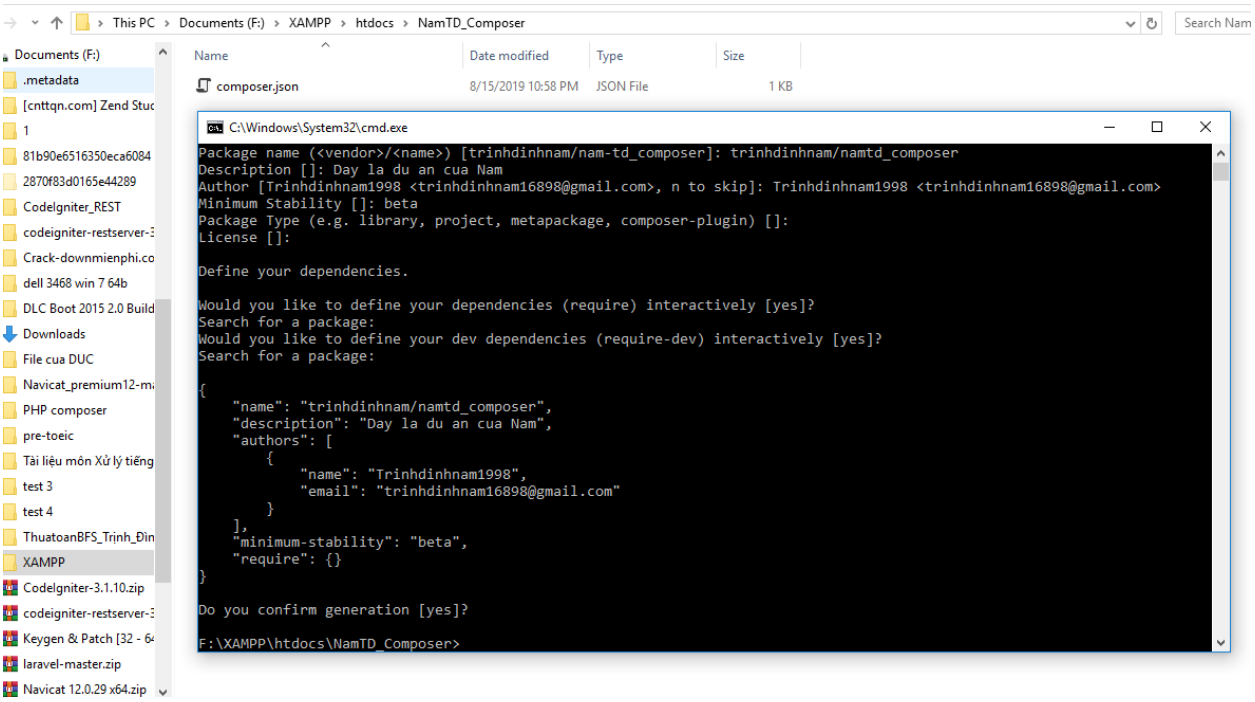
Mô tả:

Bước 1: Tạo một dự án có tên là NamTD_Composer trong F:\XAMPP\htdocs và thao tác cmd trong Project vừa tạo.

- Để khởi tạo Composer gõ lệnh *Composer init.*



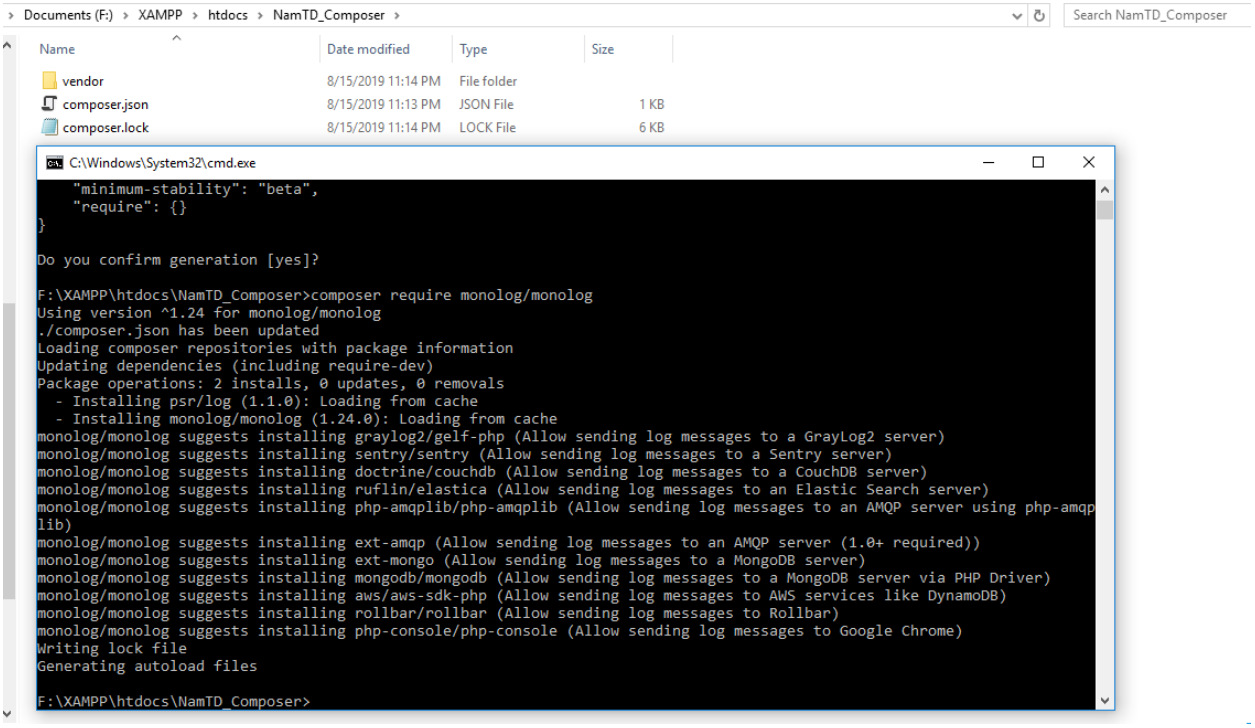
- Sau đó chúng ta sẽ cấu hình cho Composer bằng các lệnh dưới đấy và thự mục sẽ được tạo ra một file *composer.json.*



- Muốn cài đặt 1 Package ta có thể thao tác trên Cmd hoặc cài đặt trên file Composer vừa tạo.

  + Ở đây ví dụ ta muốn tạo ra Package Monolog trên dòng lệnh ta thực hiện chạy lệnh sau:

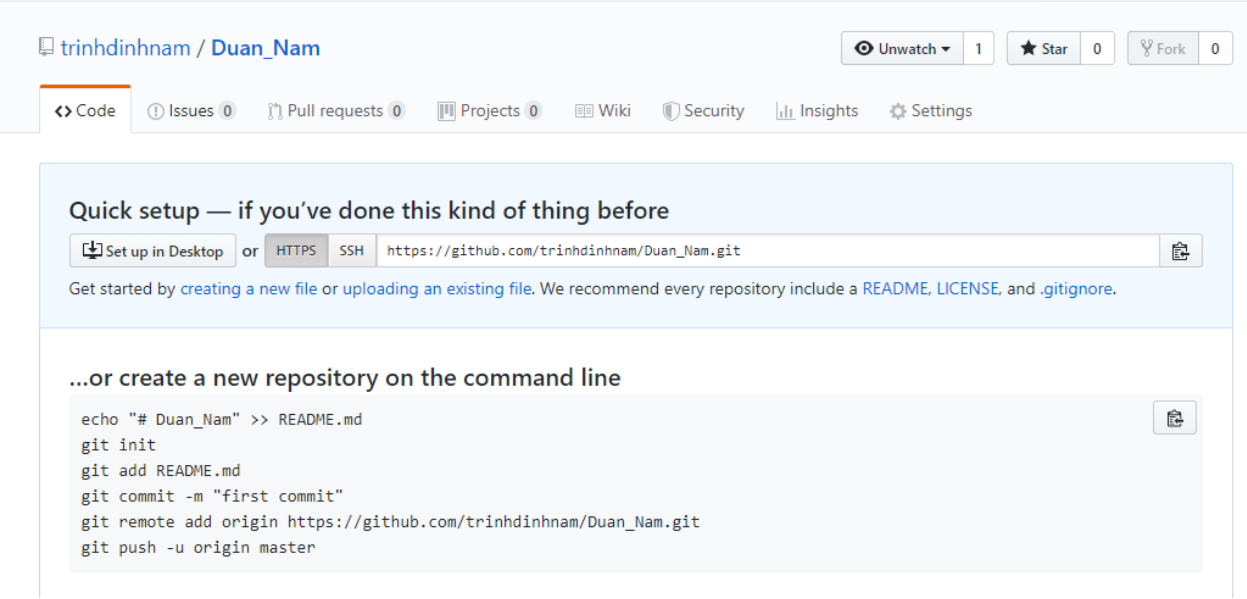
"Composer require monolog/monolog".



+ Sau đó máy sẽ tự cài đặt **monolog** vào tron dự án của mình, đồng thời tạo thêm 1 thư mục *vendor* và file *composer.lock*.

Bước 2: Up 2 file Composer.json và composer.lock lên Github.

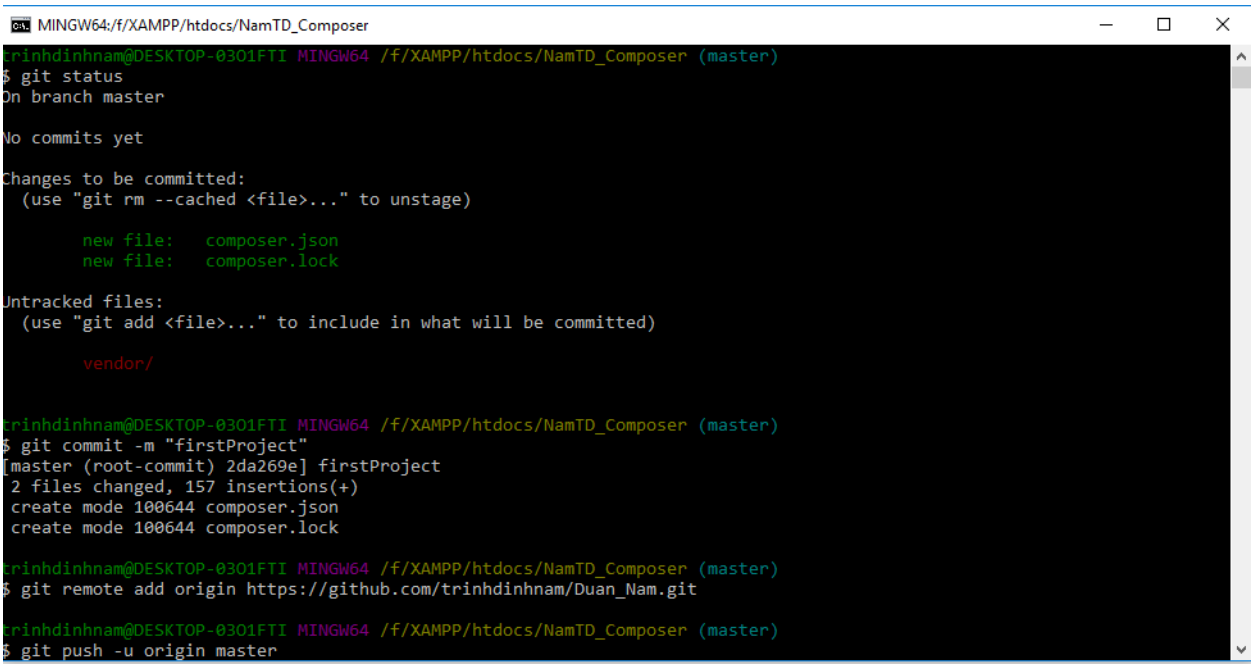
- Vào trang của Github tạo 1 Repo có tên là Duan_Nam:



- Vào thư mục chứa 2 file .json và .lock mở Git bash:

Thực hiện lần lượt các lệnh:

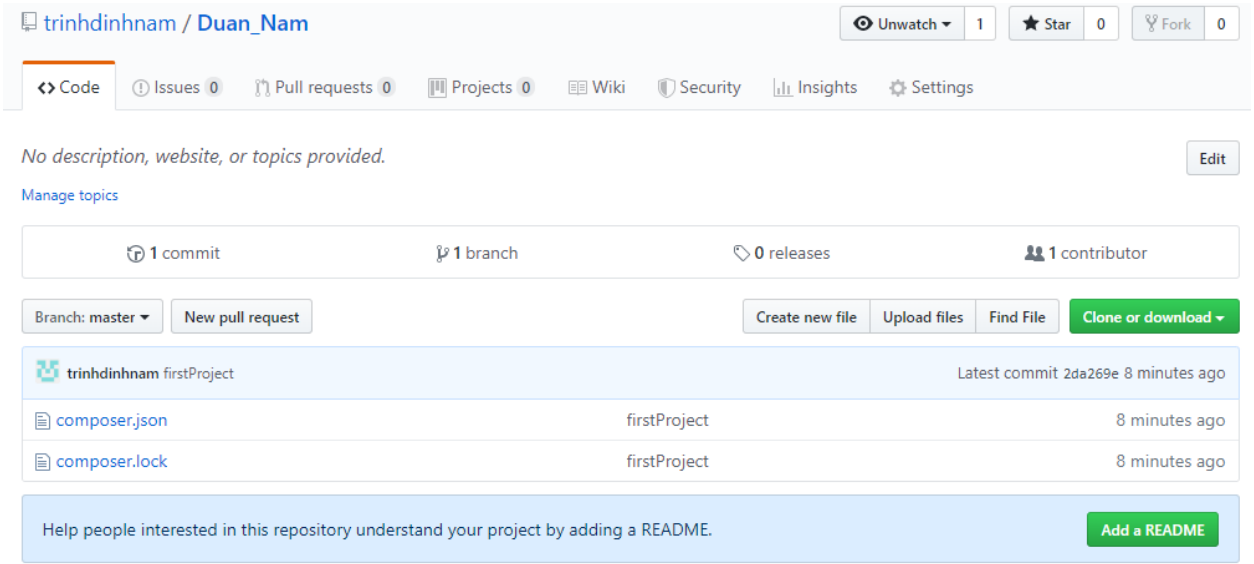
+ *git init* để khởi tạo
+ *git status* để kiểm tra trạng thái các file đã được add hay chưa.
+ Sau đó thực hiện add lần lượt: *git add composer.json* và *git add composer.lock*.



+ Sau đó kiểm tra trạng thái 2 file nếu màu xanh là đã add thành công.
+ Thực hiện commit bằng lệnh: *git commit –m "mô tả"*
+ Để push 2 file lên ta thực hiện 2 lệnh:

Git remote add origin https://github.com/trinhdinhnam/Duan_Nam.git.

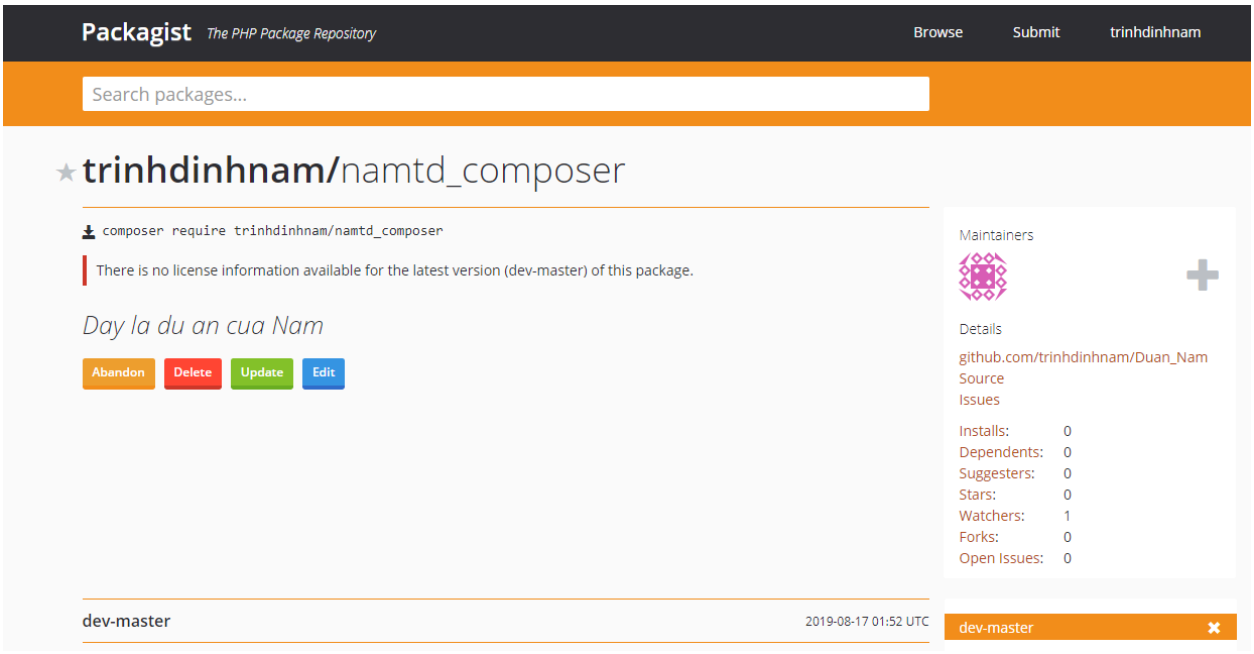
Git push –u origin master.



- Kiểm tra trên Github thì 2 file đã được push lên.

Bước 3: Lưu lên Packagist.

- Tạo một tài khoản Packagist liên kết với git hub.
- Chọn mục Submit và dán đường dẫn trên github chọn check như thế thư mục trên github đã được lưu lại trên Packagist.



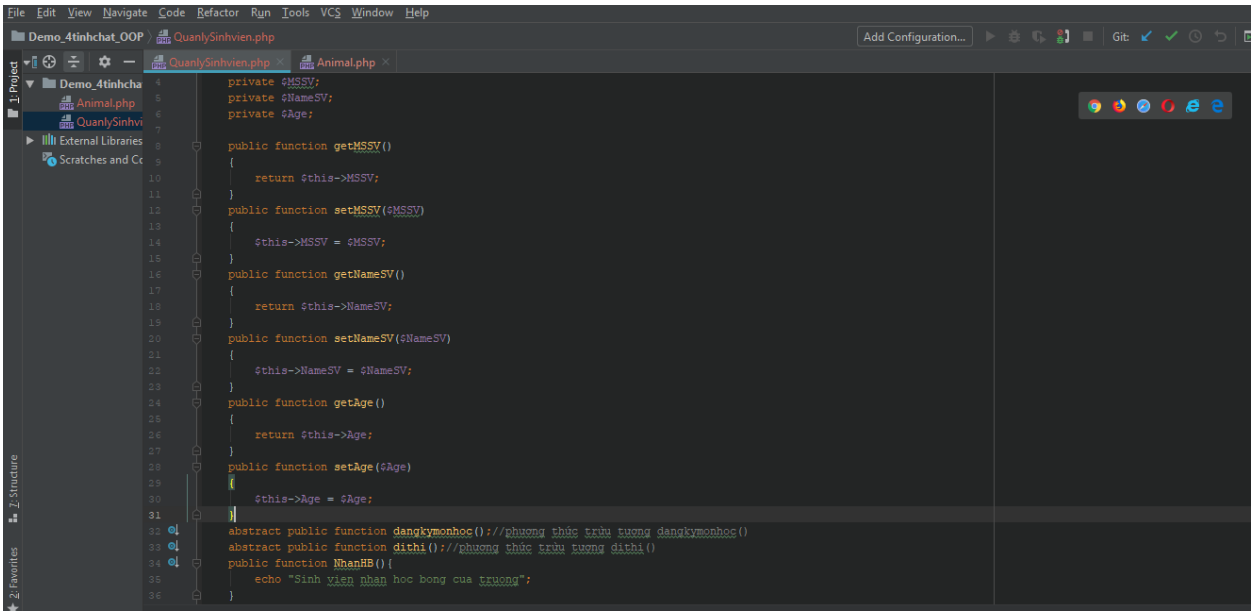
2. Đầu việc 3: Tìm hiểu về OOP PHP.

- Tiến độ thực hiện: 100%.
- Thời gian dự kiến: 2 ngày.
- Thời gian thực hiện: 1 ngày.

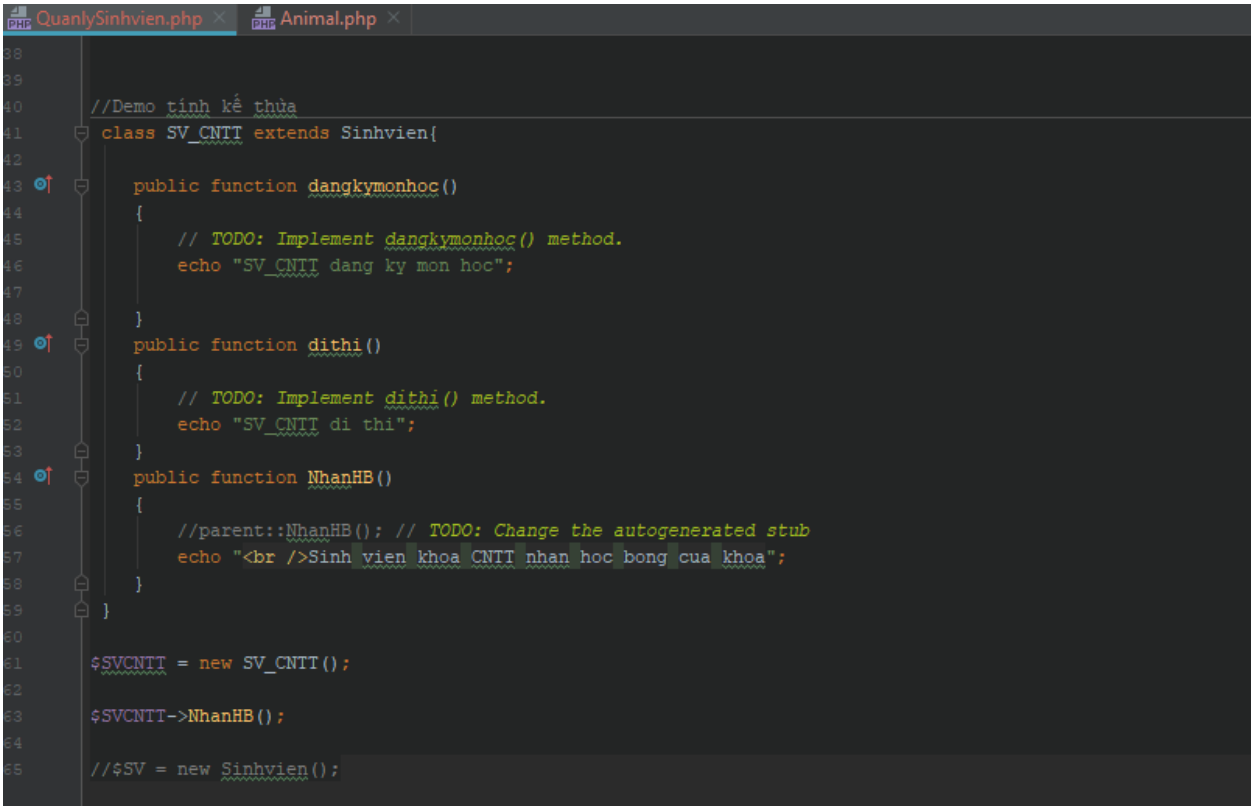
Mô tả: **Demo 4 tính chất trong lập trình hướng đối tượng.**

a) Tính trừu tượng:

- Tạo một lớp trừu tượng Sinhvien có từ khóa Abstract và có 2 phương thức trừu tượng là *dangkymonhoc()* và *dithi():*

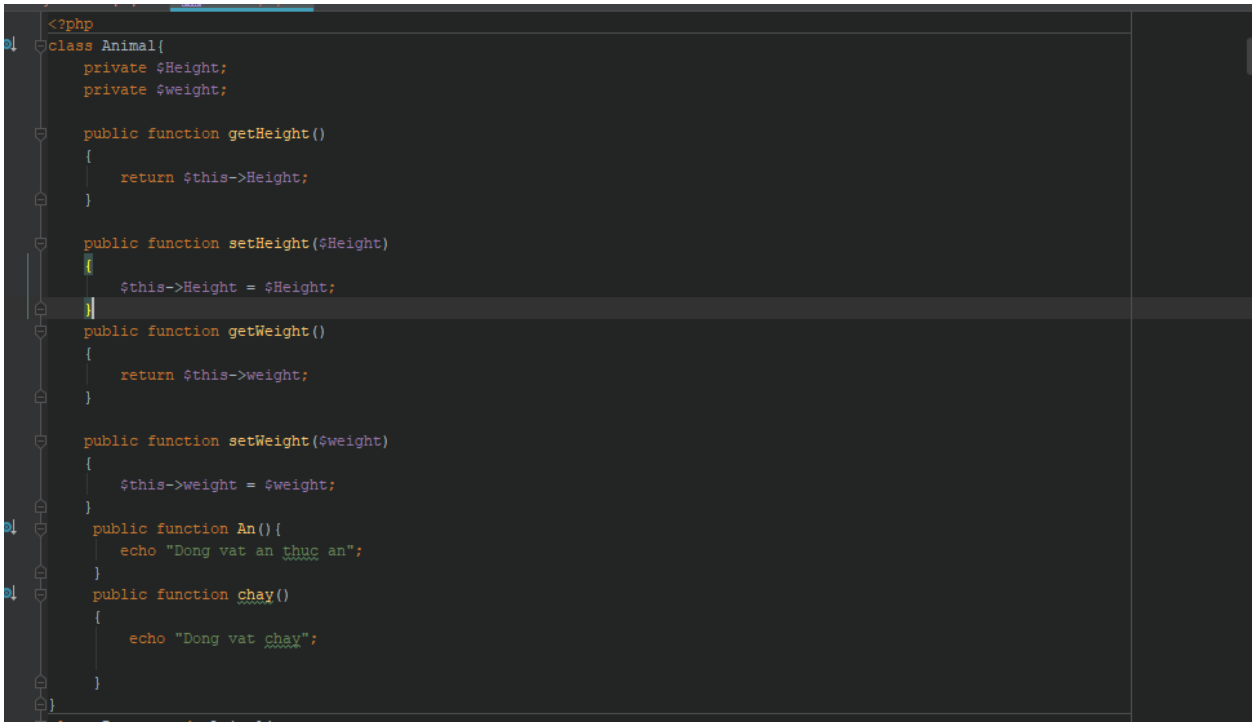


- 2 phương thức này không được định nghĩa mà ngầm định tới lớp kế thừa là bắt buộc phải định nghĩa 2 phương thức này:
  Ví dụ: lớp SV_CNTT kế thừa lại phải định nghĩa 2 phương thức trừu tượng ở lớp cha:



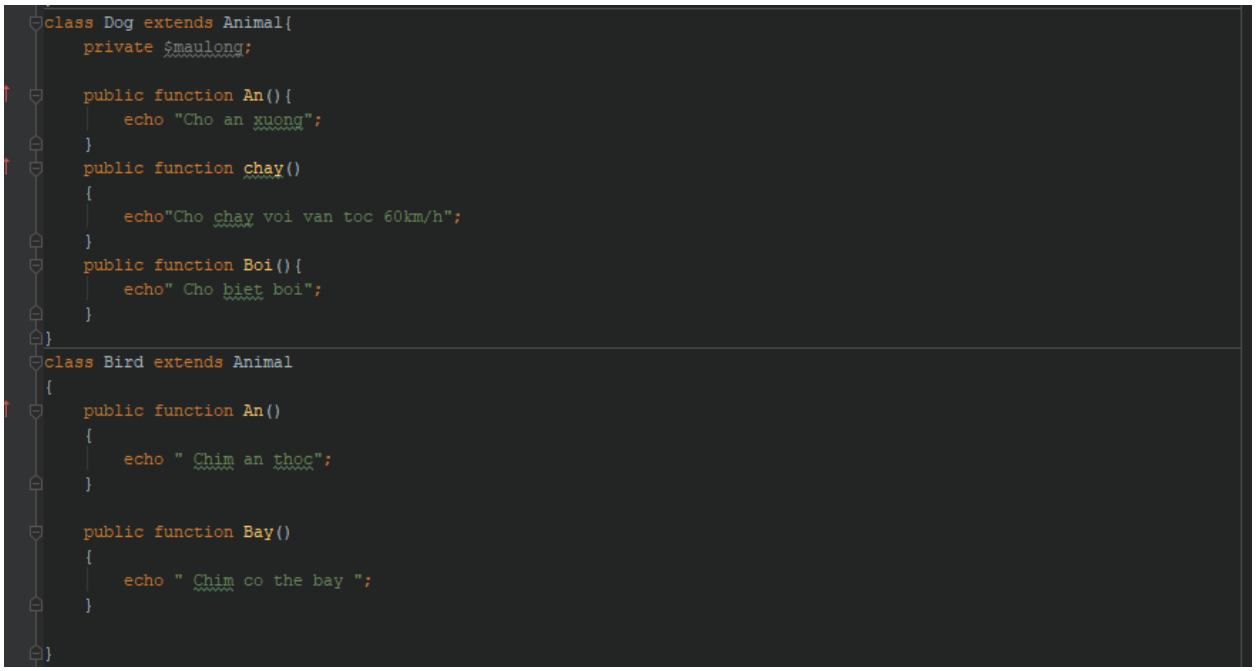
b) Tính kế thừa:

- Ở đây ta khởi tạo một lớp có tên là Animal có các thuộc tính *Height* và *weight* có 2 phương thức *An()* và *chay():*



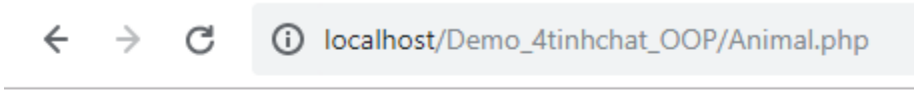
```php
<?php
class Animal{
    private $Height;
    private $weight;

    public function getHeight()
    {
        return $this->Height;
    }

    public function setHeight($Height)
    {
        $this->Height = $Height;
    }
    public function getWeight()
    {
        return $this->weight;
    }

    public function setWeight($weight)
    {
        $this->weight = $weight;
    }
     public function An(){
        echo "Dong vat an thuc an";
     }
     public function chay()
     {
         echo "Dong vat chay";

     }
}
```

- Khởi tạo lớp Dog kế thừa lớp Animal thì lớp Dog kế thừa tất cả các phương thức của lớp Animal và định nghĩa lại chúng và có thể khởi tạo thêm những phương thức mới khác.



```php
class Dog extends Animal{
    private $maulong;

    public function An(){
        echo "Cho an xuong";
    }
    public function chay()
    {
        echo"Cho chay voi van toc 60km/h";
    }
    public function Boi(){
        echo" Cho biet boi";
    }
}
class Bird extends Animal
{
    public function An()
    {
        echo " Chim an thoc";
    }

    public function Bay()
    {
        echo " Chim co the bay ";
    }

}
```

```php
$dl= new Dog();
$dl->setHeight( Height: 50);
$dl->setWeight( weight: 25);
echo "Loai cho co chieu cao: " . $dl->getHeight() . ", can nang la: " . $dl->getWeight() . "<br />";
echo $dl->An() . "<br />";
echo $dl->Chay() . "<br />";
echo $dl->Boi() . "<br />";
```

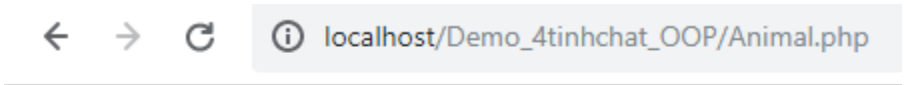


Loai cho co chieu cao: 50, can nang la: 25
Cho an xuong
Cho chay voi van toc 60km/h
Cho biet boi

c) Tính đa hình.

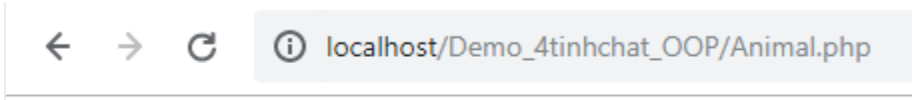
- Cũng trên ví dụ trên để thể hiện tính đa hình ta khởi tạo một đối tượng là $_a thuộc lớp Animal và cho nó trỏ đến phương thức An() kết quả in ra sẽ trả về giá trị là hàm An() của lớp Animal.

```php
//Tính đa hình

$_a= new Animal();
//$a=&$dl;
$_a->An();
```



Dong vat an thuc an

- Nhưng khi ta tham chiếu tới đối tượng $d1: *$_a= &$d1;* và cho đối tượng $_a trỏ lại phương thức An() kết quả trả về sẽ là phương thức An() của lớp Dog chứ không còn là phương thức An() của lớp Animal nữa.



Cho an xuong

d) Tính đóng gói.

- Thực chất tính đóng gói của lập trình hướng đối tượng là mô tả qua 3 điều kiện bảo mật private, protected và public.

+ **Private:** là tính chất bí mật thường dùng cho thuộc tính, nó không cho phép lớp ngoài hay kể cả lớp kế thừa truy cập đến, các lớp khác chỉ được truy cập đến phương thức get(), set() mà lớp này tạo ra**.**
+ **Protected:** là bảo vệ các thuộc tính hay phương thức được sử dụng chỉ cho phép lớp kế thừa truy cập đến, các lớp bên ngoài khồng được phép truy cập đến.
+ **Public:** là công cộng bất cứ lớp bên ngoài hay lớp kế thừa đều có thể truy cập đến.