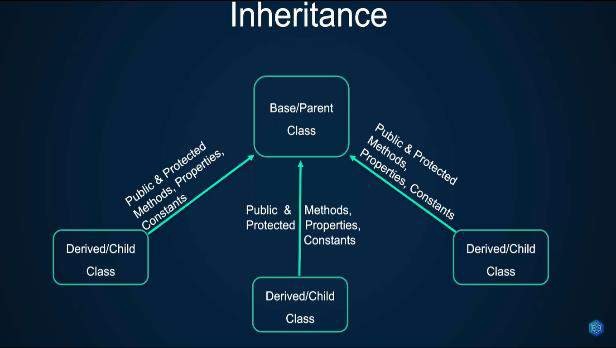
# Kế thừa trong PHP - Giải thích - Kế thừa có tốt không? - Hướng dẫn

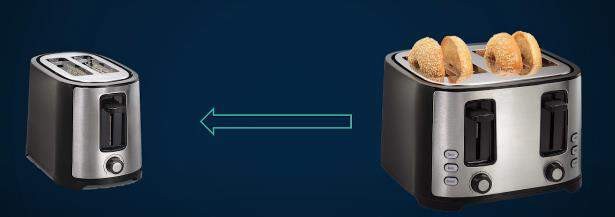
**PHP 8 đầy đủ**

* Tính kế thừa cho phép có 1 lớp dẫn xuất từ 1 lớp khác cũng được gọi là lớp cha mẹ con trong đó lớp cha mẹ là lớp cơ sở mà bạn lấy được từ lớp con sẽ kế thừa các hằng số và thuộc tính của phương thức công khai và được bảo vệ mà sau đó có thể được ghi đè ngoài việc kế thừa những thứ này trong lớp con, tất nhiên bạn cũng có thể có các hằng thuộc tính và phương thức bổ sung mà chúng tôi sẽ đề cập đến 1 số ví dụ ở đây.



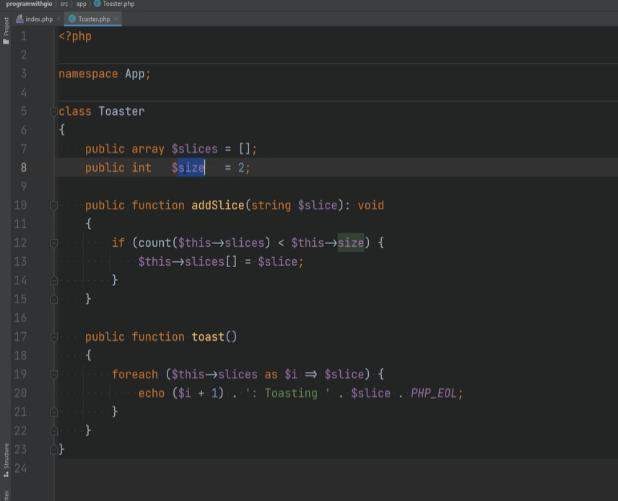
**Ví dụ đầu tiên tôi** sẽ cho bạn xem trong mã thực tế và các ví dụ khác mà tôi sẽ cho bạn xem 1 số sơ đồ và điểm qua 1 số trường hợp sử dụng tính kế thừa

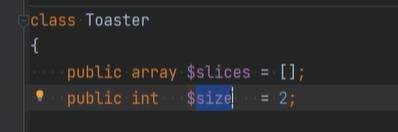
* + Giả sử chúng ta đang triển khai 1 trình mô phỏng máy nướng bánh mì (**Toaster**) có thể nướng bánh mì



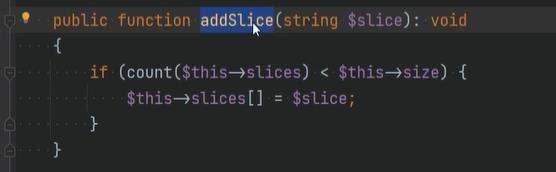
* + Có 1 lớp **Toaster** có 2 thuộc tính chung, **slices** lưu trữ các lát sẽ được

nướng và kích thước **(size)** cho biết số lát tối đa mà máy nướng bánh mì này có thể nướng cùng lúc



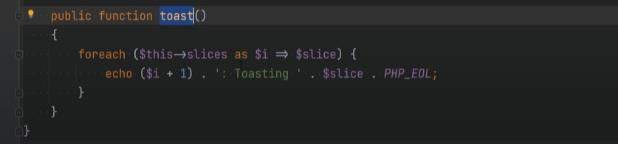


* + Có phương thức **addSlice**, chỉ cần thêm lát vào mảng lát và nó chỉ thực hiện điều đó nếu chúng ta chưa đạt đến số lát tối đa

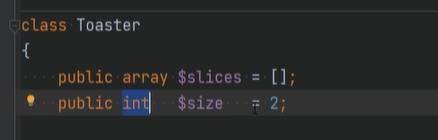


* + Có phương thức nướng bánh mì **(toast)** chỉ đơn giản lặp qua mảng lát và in

lát bánh mì nướng ra màn hình



* + Tại sao tôi không sử dụng hằng số ở đây



Vì **size** không thực sự thay đổi trong thời gian chạy, máy nướng bánh mì này chỉ có thể nướng 2 miếng cắt lát cùng 1 lúc nên sẽ hợp lý nếu coi cái này là hằng số. Lý do là vì tôi muốn bao gồm các hằng số và số liệu thống kê được đánh giá quá cao trong 1 video riêng biệt sau khi chúng ta chuyển sang chủ đề liên kết tĩnh muộn nên bây giờ điều này vẫn ổn, vì vậy hãy bắt đầu kiểm tra điều này trên index.php.

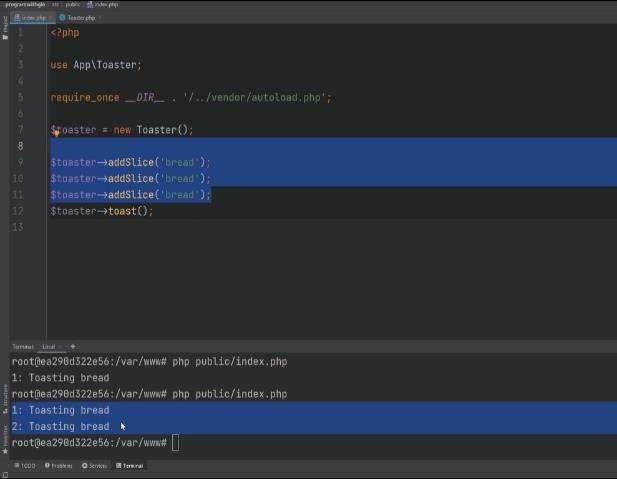
* + Tôi đang tạo đối tượng của lớp **Toaster** và tôi đang thêm 1 lát bánh mì (

**bread**), sau đó tôi đang gọi phương thức **toast**

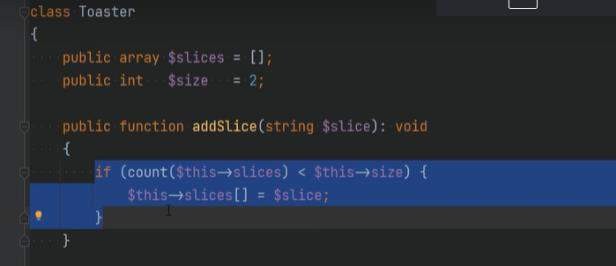


* + Mở terminal và chạy php public index.php và sẽ thấy bây giờ nó đang

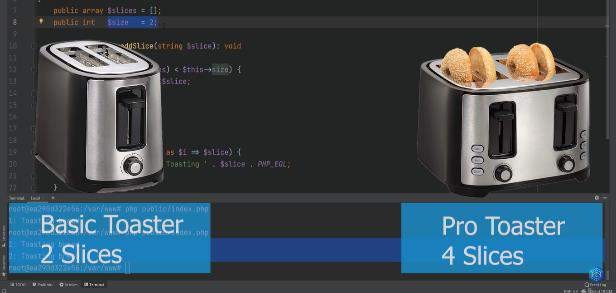
nướng 1 lát bánh mì, nó chỉ nướng 2 lát mặc dù đã thêm 3 lát ở đây vì số lát tối đa mà máy nướng bánh mì này có thể nướng là 2 lát



* + Cách lý tưởng là ném 1 ngoại lệ vào đây nếu đạt đến số lượng lát cắt tối đa nhưng vì chúng ta chưa đề cập đến các ngoại lệ nên tôi quyết định giữ điều này đơn giản và thực hiện theo cách này

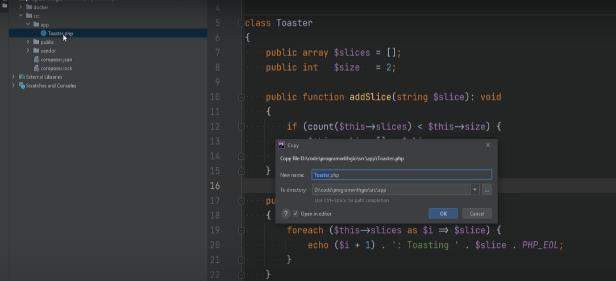


* + Muốn triển khai trình mô phỏng cho 1 máy nướng bánh mì cao cấp hơn có nhiều chức năng hơn 1 chút và nhiều tính năng hơn và g/s rằng đó là phiên bản chuyên nghiệp của máy nướng bánh mì ban đầu

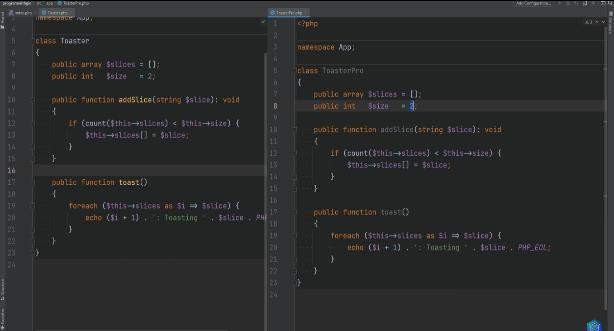


Thay vì chỉ nướng 2 lát bánh mì cùng lúc, **Toaster Pro** có thể nướng 4 lát cùng lúc. Và ngoài việc có các khả năng nướng cơ bản, nó còn có cài đặt sẵn để nướng nó như bánh mì tròn, vì vậy hãy tạo lớp đó.

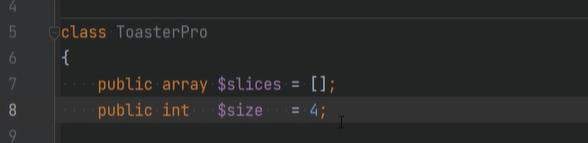
* + Tôi sẽ x2 lớp **Toaster**



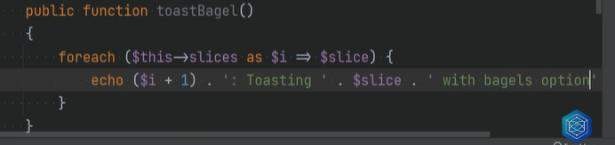
* + Và tạo máy nướng bánh mì chuyên nghiệp **Toaster Pro** và hãy chia chế độ xem ở đây và hãy thay đổi chế độ xem này thành máy nướng bánh mì chuyên nghiệp **Toaster Pro**



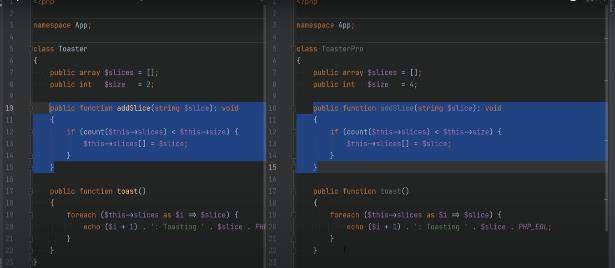
* + Thay đổi kích thước **size** thành 4 vì điều này có thể nướng 4 lát cùng 1 lúc



* + Thêm 1 phương thức ở đây để nướng với tùy chọn **toastBagel** để có thể làm điều tương tự như phương pháp nướng bánh mì
  + Có thể thêm từ ngữ ở đây tức là nướng bánh mì với tùy chọn bánh mì tròn



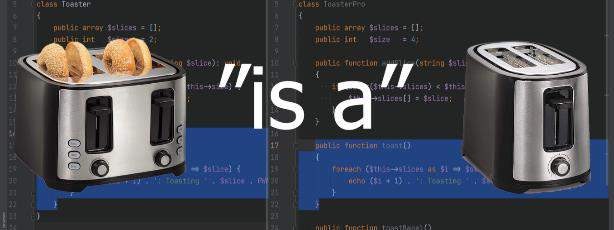
* + Phương thức thêm lát **( addSlice)** giống hệt như phương thức nướng bánh mì thêm lát **( addSlice)**



* + Phương thức nướng **( toast)** hoàn toàn giống như phương pháp nướng bánh mì ở đây **( toast)**



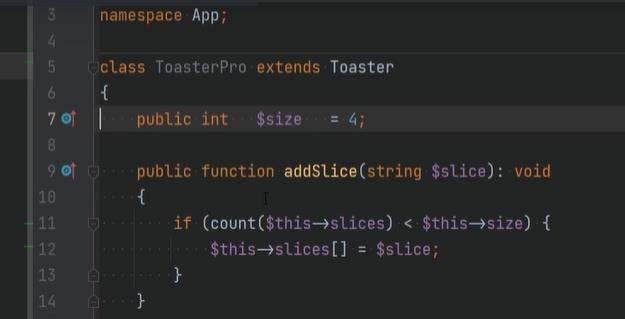
* + Và đó là bởi vì máy nướng bánh mì chuyên nghiệp **Toaster Pro** cũng là máy nướng bánh mì **Toaster** đúng không



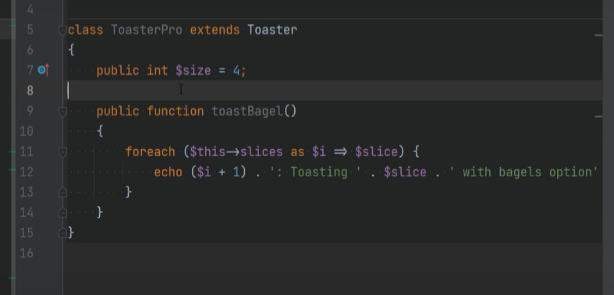
* + Và hãy chú ý mối quan hệ ở đây máy nướng bánh mì chuyên nghiệp là máy nướng bánh mì, nghĩa là nó có thể nướng bánh mì với các cài đặt cơ bản thông thường nhưng nó cũng có thể nướng bánh mì như cài đặt bánh mì tròn và nó có thể nướng 4 lát cùng lúc thay vì 2 lát vì vậy
  + Vì điều này, chúng ta có thể tận dụng sức mạnh của sự kế thừa và kế thừa lớp máy nướng bánh mì **Toaster** trong máy nướng bánh mì chuyên nghiệp **Toaster Pro** của mình và chúng ta có thể làm điều đó bằng cách sử dụng từ khóa mở rộng **extends** để chúng ta có thể mở rộng máy nướng bánh mì **Toaster**



* + Điều đó có nghĩa là chúng ta có thể loại bỏ các lát cắt 1 cách chính xác



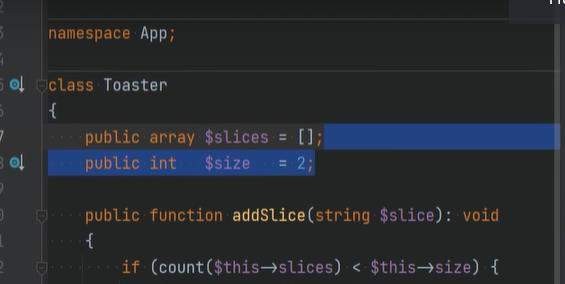
* + Chúng ta cũng có thể loại bỏ phương thức **addSlice** và phương thức toast và chúng ta chỉ còn lại phương thức bánh mì tròn nướng



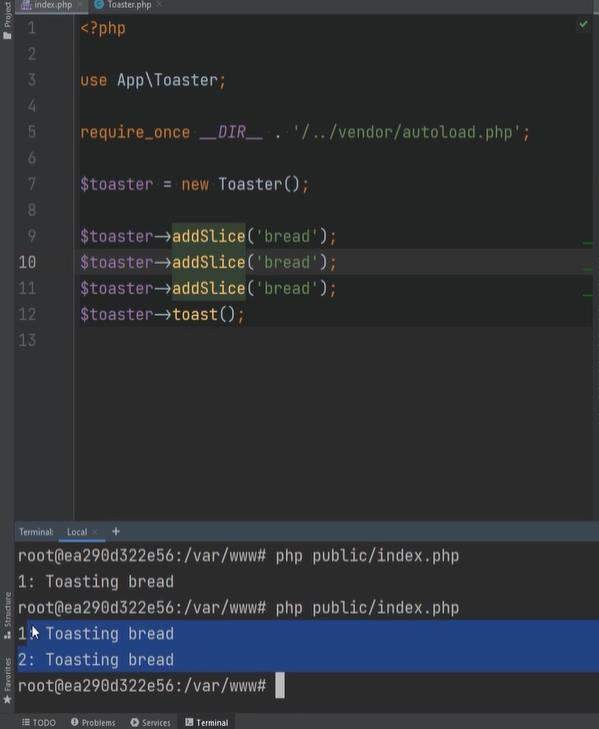
* + Như đã đề cập trước khi bạn có thể ghi đè các hằng số và phương thức thuộc tính công khai và được bảo vệ và đó chính xác là những gì chúng tôi đang làm ở đây, chúng tôi sẽ ghi đè thuộc tính **size** với giá trị là 4 vì **toaster pro** có thể nướng 4 lát cùng 1 lúc



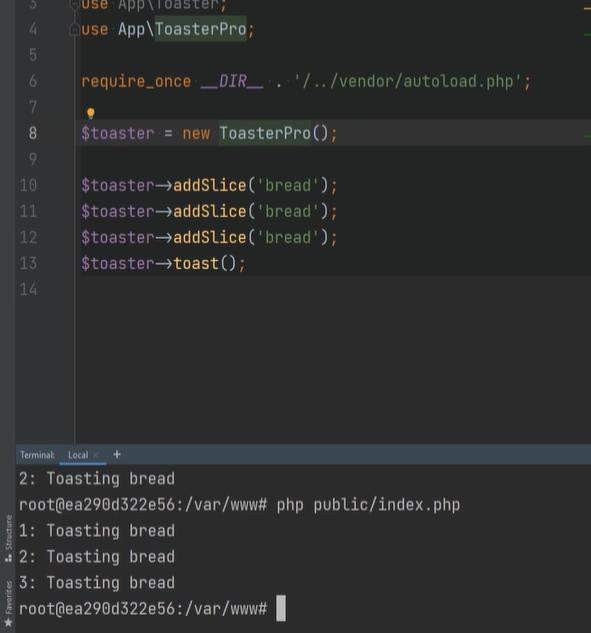
* + Trong khi Máy nướng bánh mì **toaster** chỉ có thể thực hiện 2 việc



* + Nên nếu tôi truy cập vào file index.php và mở terminal thì nó chỉ có thể nướng đúng 2 lát

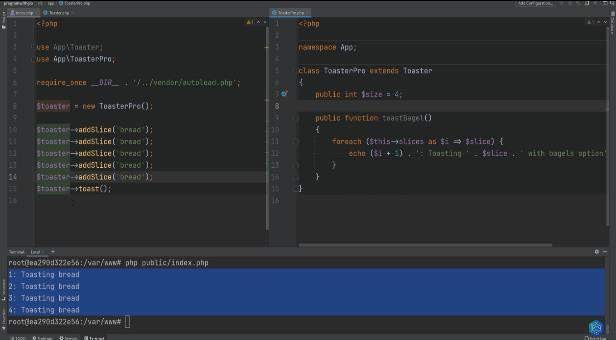


* + Nhưng nếu tôi đổi cái này thành **toaster pro** và hãy nhập nó và chạy lại thì chúng ta thấy rằng bây giờ nó đang nướng 3 lát

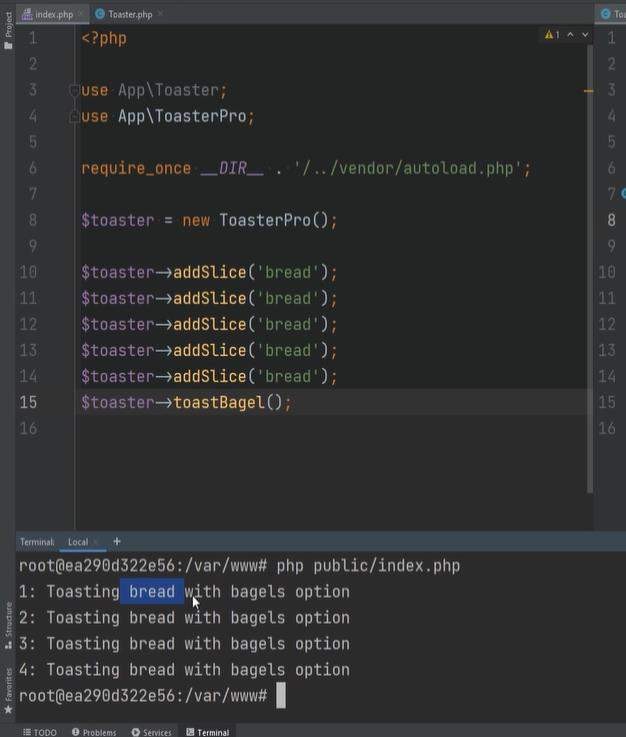


* + Tôi có thể thêm vài lát nữa ở đây và chúng tôi thấy rằng số lát tối đa là 4 vì

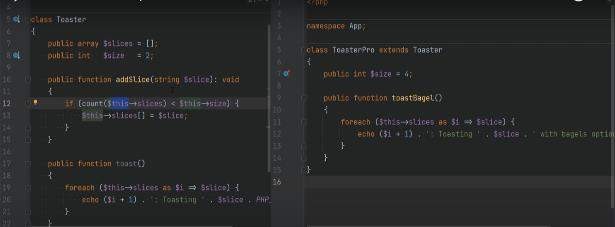
đó là những gì chúng tôi đặt ở đây



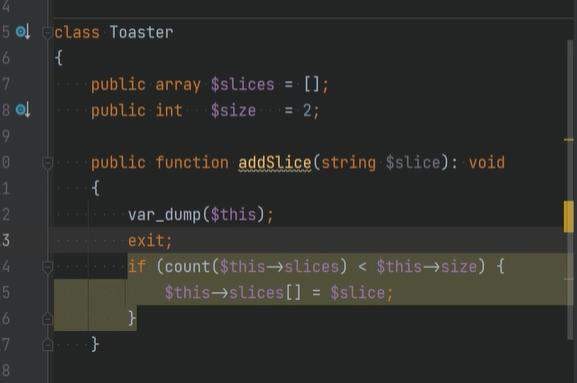
* + Bây giờ ngoài việc nướng nó như bánh mì thông thường chúng tôi cũng có thể nướng nó như bánh mì tròn đúng không, vì vậy chúng tôi có thể gọi bánh mì nướng **toastBagel** và chúng tôi chạy nó và chúng ta thấy rằng nó đang nướng bánh mì với tùy chọn của bagel



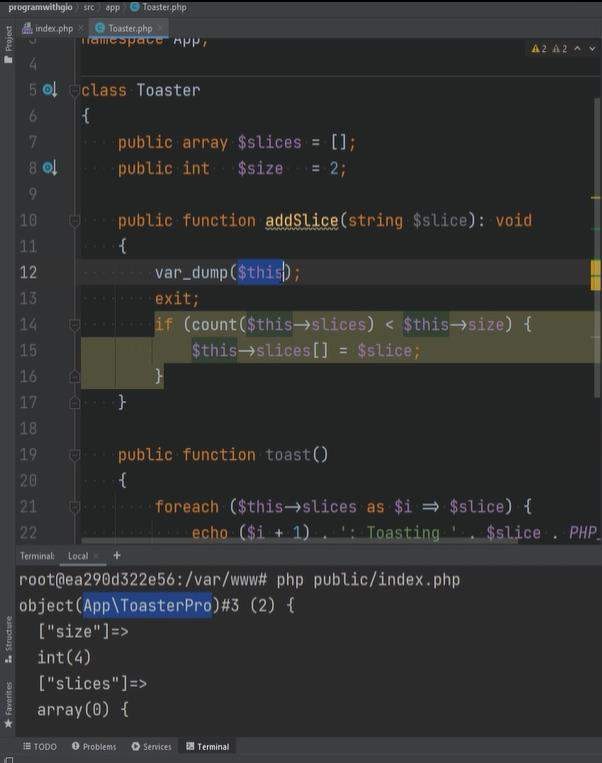
* + Cũng có 1 điều đáng chú ý là nếu bạn nhớ định nghĩa của biến **this** từ 1 trong những bài học trước thì biến này đề cập đến đối tượng gọi trong trường hợp này khi chúng ta truy cập các thuộc tính hoặc các phương thức sử dụng biến này điều đó không nhất thiết có nghĩa là nó đang truy cập các thuộc tính đó trên lớp cha mà nó có thể truy cập chúng trên lớp con.



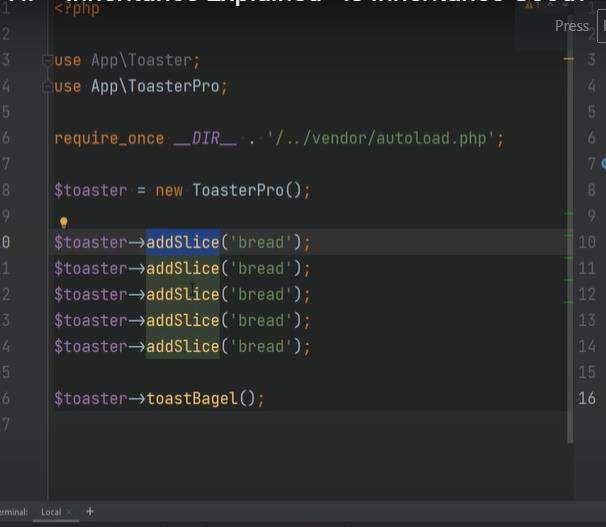
* + Hãy thay đổi biến này trong lớp cha và xem những gì chúng ta nhận được



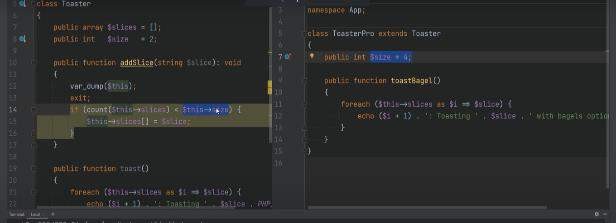
* + Khi bạn có thể thấy biến này đề cập đến lớp toaster pro vì đó là đối tượng gọi



* + Nếu chúng ta quay lại index.php, chúng ta đang khởi tạo toaster pro
* đây và chúng ta đang gọi phương thức từ lớp toaster



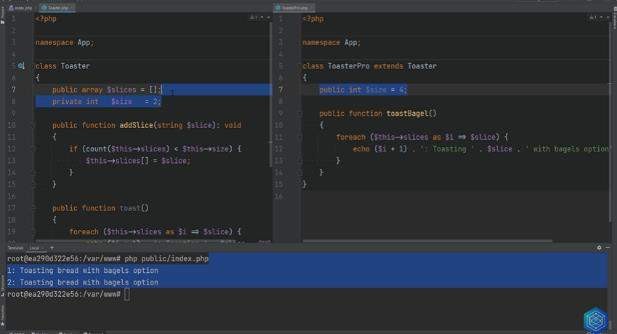
* + Vì toaster pro kế thừa hoặc mở rộng lớp toaster, đó là lý do tại sao biến này trong lớp cha hoặc lớp nướng bánh mì



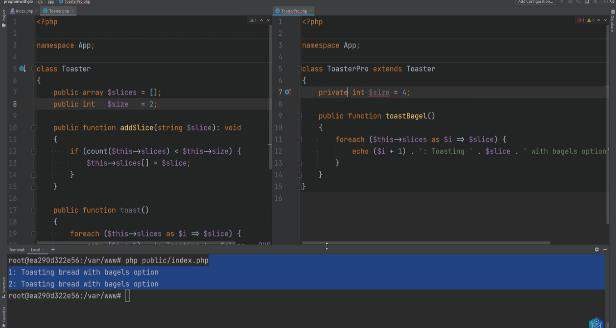
* + Đang đề cập đến phiên bản toaster pro bây giờ chúng ta đã biết những điều cơ bản về kế thừa, hãy nói về 1 số chi tiết quan trọng khác nếu chúng ta đặt thuộc tính size thành riêng tư và chạy mã mà chúng ta thấy rằng bây giờ nó đang nướng chỉ có hai lát mặc dù chúng tôi đã chỉ định bốn lát ở đây vì bạn không thể ghi đè các thuộc tính riêng tư vì các thuộc tính riêng tư chỉ thuộc về và tồn tại trong lớp đó ngay cả lớp mở rộng lớp nướng bánh mỳ

cũng không thể truy cập các thuộc tính riêng tư mà bạn chỉ có thể truy cập

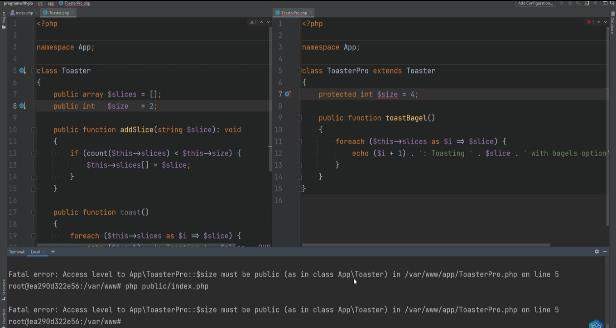
và ghi đè các phương thức thuộc tính công khai và được bảo vệ và hằng số



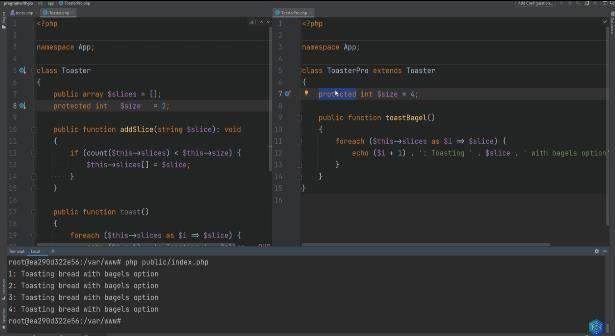
* + 1 quy tắc khác là bạn không thể giảm mức độ hiển thị của các thuộc tính, vì vậy nếu bạn đặt kích thước thành công khai, bạn không thể giảm mức độ hiển thị của kích thước nên bạn không thể đặt kích thước này thành riêng tư, nó sẽ báo lỗi cho bạn và nó sẽ gạch chân nó nếu tôi chạy



* + Tôi sẽ gặp lỗi đó, bạn cũng không thể đặt nó để bảo vệ nó vì được bảo vệ thấp hơn công khai và về cơ bản bạn đang giảm khả năng hiển thị và mặt khác nó sẽ không hoạt động



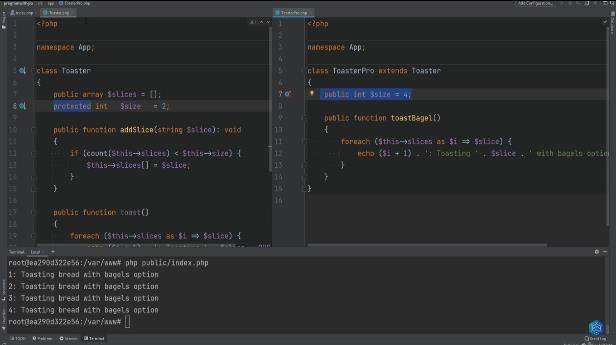
* + Nếu bạn đặt tùy chọn này thành bảo vệ, bạn có thể ghi đè nó được bảo vệ, trong trường hợp này sẽ hoạt động vì vậy nếu tôi chạy nó thì mọi thứ đều hoạt động



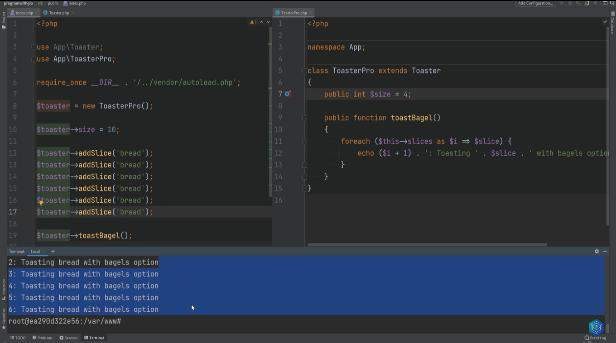
* + Hoặc tôi có thể công khai và điều này vẫn sẽ hoạt động theo các quy tắc chính xác tương tự áp dụng cho các phương thức, thuộc tính tĩnh, phương thức và hằng số cũng như bạn biết các thuộc tính công cộng thì không lý tưởng vì chúng phá vỡ sự đóng gói ngay trong trường hợp này, thuộc tính size được đặt để bảo vệ nó trong lớp cha nhưng chúng tôi đang ghi đè nó

bằng khả năng hiển thị công khai, điều đó có nghĩa là chúng tôi đang tăng

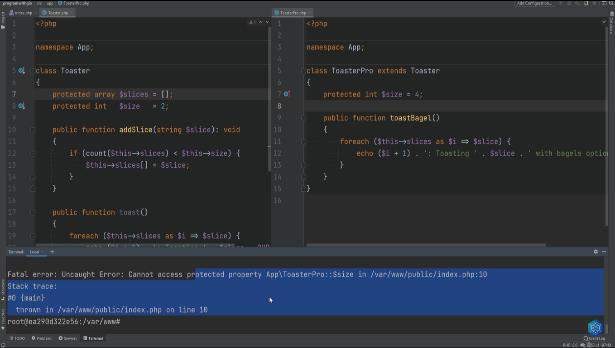
khả năng hiển thị trong toaster pro



* + Và về cơ bản chúng tôi có thể làm được đại loại như thế này, chúng ta có thể làm kích thước máy nướng bánh mì bằng 10 và bây giờ chúng tôi đã đặt kích thước thành 10 1 cách rõ ràng và mặc dù máy nướng bánh mì này chỉ có thể nướng 4 lát cùng lúc nhưng chúng tôi đã thay đổi điều đó và nếu tôi thêm nhiều lát hơn ở đây và tôi sẽ chạy nó chúng tôi có thể nướng nhiều hơn 4 lát, 1 cách khắc phục như bạn biết là đặt phần này thành riêng tư nhưng sau đó chúng tôi không thể ghi đè các thuộc tính ngay.



* + 1 cách khắc phục khác là sử dụng công cụ sửa đổi quyền truy cập được bảo vệ, về cơ bản giống như riêng tư nhưng nó mở rộng cho trẻ em các lớp vì vậy hãy thiết lập mọi thứ để bảo vệ nó và bây giờ nếu tôi chạy nó, chúng ta sẽ gặp lỗi này là chúng ta không thể truy cập thuộc tính size nữa

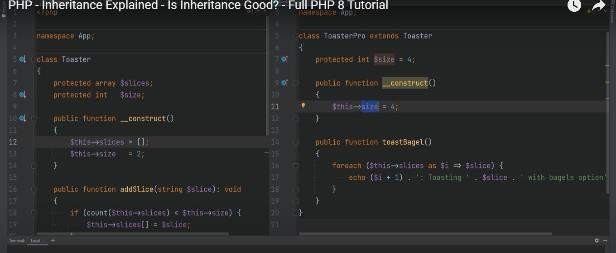


* + Vì vậy hãy nói về ghi đè phương thức. Tôi sẽ cho bạn xem ví dụ về ghi đè các hàm tạo. Đầu tiên, sau đó tôi sẽ chỉ cho bạn ví dụ về ghi đè các phương thức khác vì bạn nhận thấy rằng chúng ta không có bất kỳ hàm tạo nào trong lớp toaster hoặc toaster pro và đó là vì các hàm tạo không bắt buộc

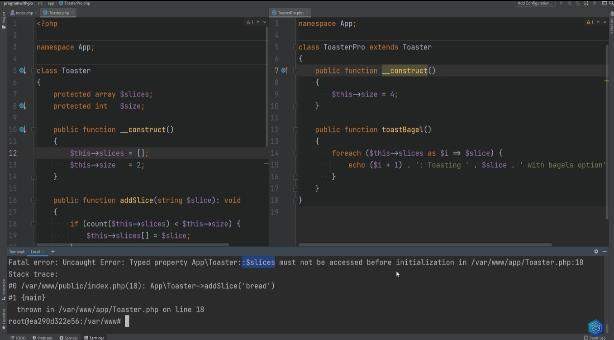


* + Hãy thêm phương thức hàm tạo vào lớp toaster ở đây và thay vì khởi tạo thuộc tính size thành 2 ở đây, chúng ta có thể chỉ cần thực hiện điều đó trong hàm tạo theo cách này và loại bỏ việc khởi tạo từ đây và hãy làm điều tương tự cho các lát cắt và chúng ta có thể làm điều tương tự ở đây hãy xây dựng hàm công khai và chúng ta có thể chỉ cần ghi đè thuộc tính kích

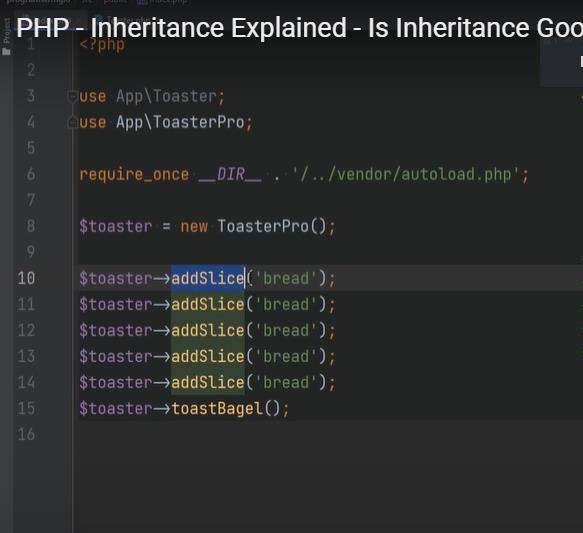
thước thành 4 và vì kích thước được đặt để bảo vệ nó nên chúng tôi có thể truy cập đúng cách



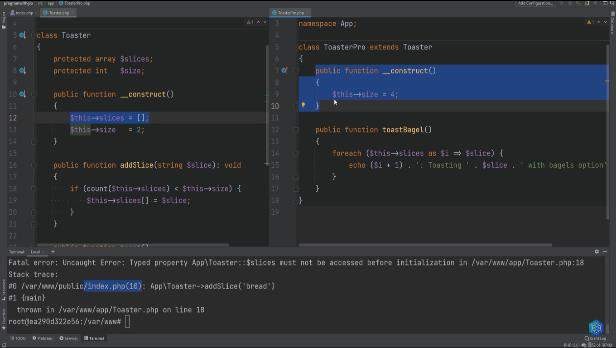
* + Và chúng tôi có thể xóa thuộc tính này khi ghi ngay bây giờ nếu tôi chạy lại, chúng tôi sẽ gặp phải lỗi lạ nói rằng mảng lát cắt phải không được truy cập trước khi khởi tạo



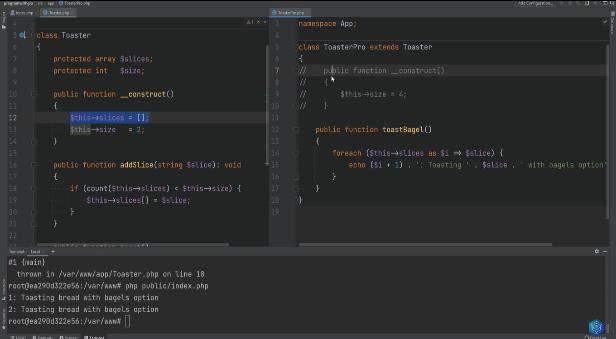
* + Vì vậy những gì đang xảy ra ở đây, hãy đi đến dòng 10 của index.php và gỡ lỗi này để trên dòng 10, chúng ta đang gọi thêm lát bánh mì



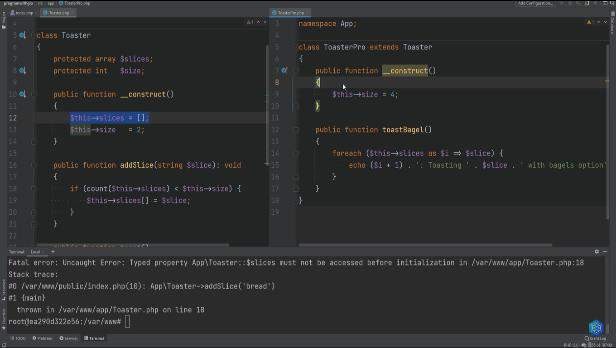
* + Vì vậy hãy đi đến thêm lát và lát đó nằm trong lớp máy nướng bánh mì và những gì điều này đang làm là nó đang truy cập vào thuộc tính slice ở ngay đây nhưng nó nói rằng thuộc tính này không được khởi tạo nhưng chúng ta đã khởi tạo nó trong hàm tạo này, vậy tại sao nó không hoạt động, lý do nó không hoạt động là khi bạn ghi đè Hàm tạo ở đây theo mặc định, nó sẽ không gọi phương thức hàm tạo của cha mẹ



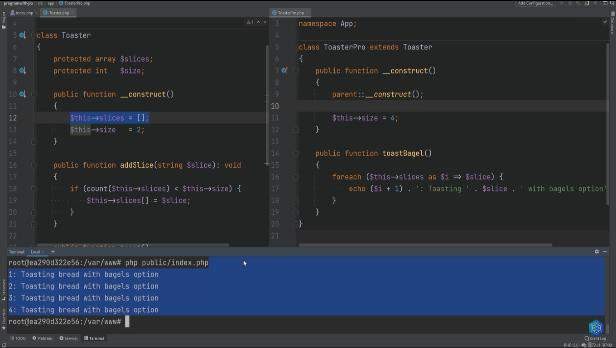
* + Vì vậy nếu tôi nhận xét phần này là phần ghi đè và tôi chạy lại thì mọi thứ vẫn hoạt động không còn lỗi nữa



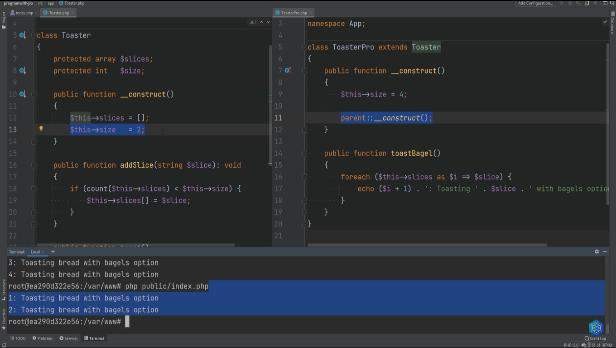
* + Nhưng khi tôi ghi đè phương thức hàm tạo hoặc bất kỳ phương thức nào theo nghĩa đó thì nó sẽ không hoạt động để gọi phương thức cha theo mặc định, vì vậy nếu tôi chạy lại nó, chúng tôi sẽ gặp lỗi đó



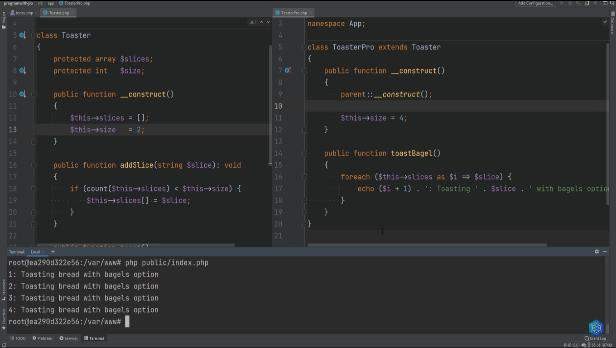
* + Bạn cần gọi phương thức cha 1 cách rõ ràng bằng từ khóa cha, sau đó dùng dấu hai chấm kép là toán tử phân giải phạm vi và sau đó gọi cấu trúc theo cách này ngay bây giờ nếu tôi chạy nó lần nữa, mọi thứ vẫn đang hoạt động.



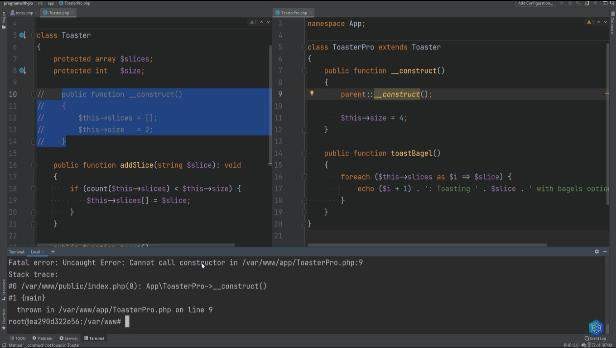
* + Điều cần lưu ý ở đây là bạn cần cẩn thận khi gọi phương thức xây dựng gốc nếu tôi gọi phương thức này sau khi ghi đè thuộc tính size, vì vậy nếu tôi gọi nó sau ngay tại đây thì điều này sẽ gây ra sự cố bởi vì chúng tôi đang đặt thuộc tính kích thước thành bốn và sau đó chúng tôi đang gọi cấu trúc gốc mà ở đây nó đặt thuộc tính kích thước thành hai nên về cơ bản nó sẽ đặt lại thuộc tính kích thước thành hai nếu tôi chạy mã mà chúng tôi thấy rằng bây giờ nó chỉ đang nướng hai vì kích thước hiện được đặt thành hai nên hãy cẩn thận khi bạn gọi phương thức xây dựng cha mẹ.



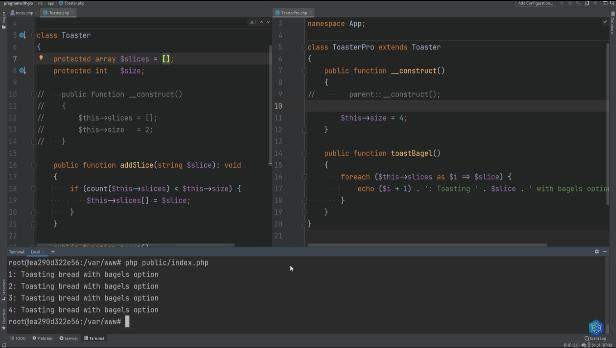
* + Cá nhân tôi luôn đặt nó lên trên trừ khi tôi có 1 số logic đặc biệt khi tôi cần thực hiện 1 số nội dung trước khi gọi cấu trúc cha mẹ như có thể chuẩn bị các đối số rằng cần phải được chuyển vào hàm tạo gốc, v.v. vì vậy nếu tôi chạy lại cái này thì mọi thứ sẽ trở lại bình thường.



* + Điều cần lưu ý là nếu bạn không có hàm tạo trong lớp cha thì bạn không thể gọi cấu trúc gốc nếu tôi thử để làm điều đó chúng ta sẽ gặp lỗi là không có hàm tạo nào trong lớp nướng bánh mì



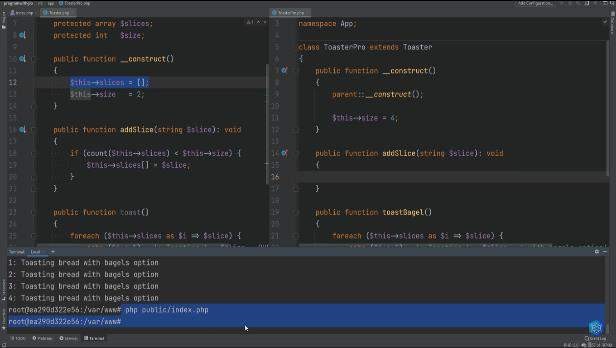
* + Nên tôi cũng cần phải nhận xét điều đó và điều này hoàn toàn hợp lệ, bạn không cần phải có hàm tạo trong lớp cha để để có 1hàm tạo trong lớp con và vì chúng ta không khởi tạo mảng lát cắt nên chúng ta cần khởi tạo nó để tránh gặp lỗi vì vậy hãy tạm thời đặt nó nếu tôi chạy nó, mọi thứ sẽ trở lại bình thường.



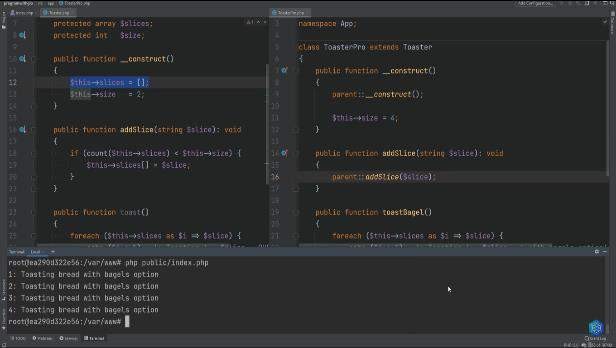
* + Tôi biết có rất nhiều thứ cần phải xử lý đây nhưng đừng lo lắng về điều đó, hãy xem video này bao nhiêu lần tùy thích và nó cũng sẽ đi kèm với việc thực hành bạn không cần phải ghi nhớ bất kỳ điều nào trong số này khi bạn đang viết mã, bạn sẽ gặp những lỗi đó và bạn sẽ có thể sửa chúng bất cứ khi nào chúng xuất hiện nên đừng lo lắng về điều này quá nhiều
  + Vì vậy hãy thay đổi lại điều này và tiếp tục áp dụng tương tự cho các

phương thức khác ngoài phương thức khởi tạo

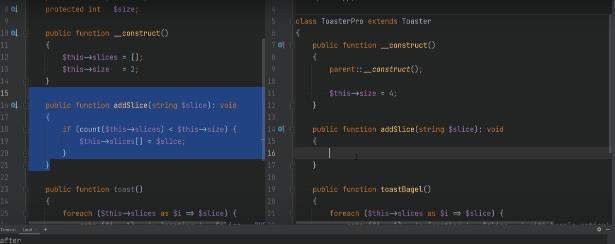
* + Vì vậy giả sử rằng chúng ta muốn ghi đè phương thức add slice chấp nhận lát cắt làm đối số nếu chúng ta không gọi phương thức lát cắt cha mẹ 1 cách rõ ràng thì nó sẽ không tự động thực hiện điều đó cho chúng ta vì vậy nếu tôi chạy lại cái này, chúng ta sẽ thấy rằng không có gì được in và đó là 1 lần nữa vì nó không tự động gọi phương thức cha mẹ



* + Chúng ta cần gọi nó một cách rõ ràng bằng cách sử dụng cha mẹ thêm lát cắt và chuyển bất kỳ đối số cần thiết nào là lát cắt trong trường hợp này và nếu chúng ta chạy lại nó thì mọi thứ vẫn hoạt động



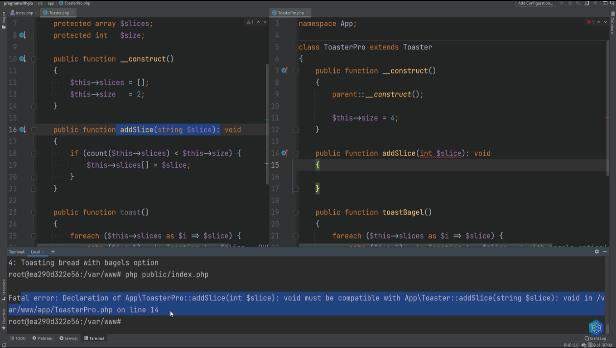
* + Thêm logic tùy chỉnh vào trước khi gọi phương thức cha mẹ hoặc sau đó nhưng đây là không bắt buộc, không cần gọi phương thức gốc nếu muốn ghi đè phương thức và có logic tùy chỉnh trong lớp con của mình, có thể làm điều đó và có thể thực hiện logic tùy chỉnh ở đây



* + Chẳng hạn như trong lát cắt bổ sung mà chúng tôi mong đợi chuỗi được cắt sang bên phải nếu chúng tôi thay đổi chuỗi này thành lát cắt số nguyên,

điều này sẽ không hoạt động nếu tôi chạy mã, bạn thấy rằng chúng tôi đang

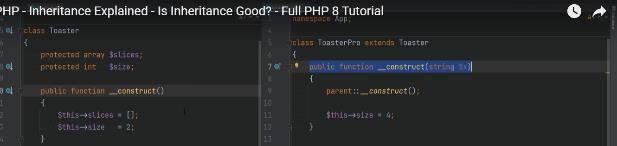
gặp phải lỗi nghiêm trọng tương tự đối với kết quả trả về



* + Vì vậy nếu tôi thay đổi kiểu này trở lại thành chuỗi và giả sử chúng ta thay đổi kiểu trả về thành số nguyên thì điều này cũng sẽ dẫn đến một lỗi nghiêm trọng vì chữ ký phương thức không tương thích với chữ ký phương thức cha, có những thứ như hiệp phương sai và nội dung các biến thể làm cho kiểu tham số và kiểu trả về trở nên linh hoạt hơn một chút khi nói đến kế thừa



* + Nhưng đừng lo lắng về điều đó bây giờ chúng ta sẽ trình bày những biến thể đó trong một video riêng. các quy tắc tương thích không áp dụng cho hàm tạo nên hãy để tôi xóa phần này khỏi đây và hãy cùng bắt đầu chấp nhận một số loại tham số trong hàm tạo con để chúng ta có thể làm một cái gì đó như chuỗi x để bạn có thể thấy chữ ký phương thức của lớp con không giống với chữ ký phương thức của lớp cha nhưng vì đây là hàm tạo nên điều này không sao các quy tắc chữ ký không áp dụng cho các phương thức của hàm tạo



* + Bạn cũng có thể chấp nhận các loại đối số khác nhau trong hàm tạo gốc để chúng tôi có thể chấp nhận một cái gì đó như int x và sau đó chúng tôi chấp nhận chuỗi y ở đây chẳng hạn và có thể chúng tôi chấp nhận int



* + Và sau đó là điều duy nhất điều bạn cần làm là bạn cần đảm bảo rằng bạn đang chuyển các đối số thích hợp cho lệnh gọi hàm tạo cha mẹ nếu bạn đang gọi cấu trúc cha mẹ



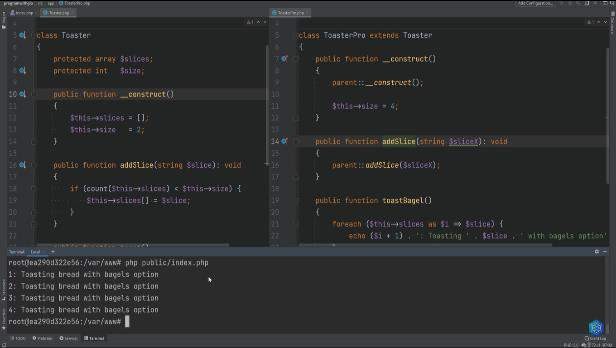
* + Nếu bạn không gọi cái này thì bạn không cần phải lo lắng về điều đó
  + Bạn vẫn có thể chấp nhận chuỗi x làm đối số đầu tiên ở đây và chuyển nó xuống và chỉ chấp nhận hai đối số bổ sung trong lớp con hoặc bạn có thể chỉ cần mã hóa cứng thứ này thành thứ gì đó và không chấp nhận nó làm đối số



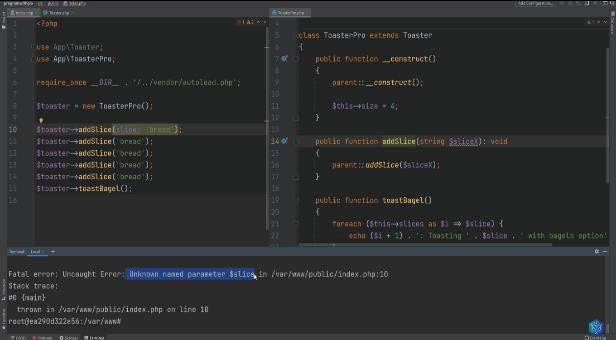
* + Nhưng về cơ bản, việc bạn muốn xử lý nó như thế nào tùy thuộc vào logic

ứng dụng và yêu cầu của bạn, được rồi, hãy xóa điều này khỏi đây

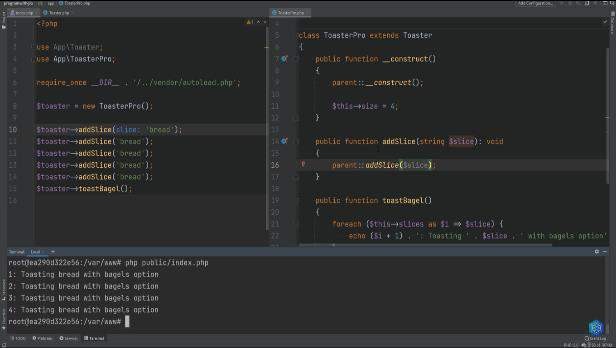
* + Và tiếp tục bạn cũng có thể có các tên khác nhau cho các tham số phương thức, điều này sẽ không gây ra vấn đề tương thích nhưng chúng có thể gây ra lỗi nếu tính năng đối số được đặt tên được sử dụng lại nếu chúng ta ghi đè phương thức thêm lát cắt ở đây và chúng ta gọi nó là lát x và chuyển nó trong điều này hoàn toàn hợp lệ, nó vẫn hoạt động nếu tôi chạy index.php, chúng tôi thấy rằng mọi thứ đều hoạt động



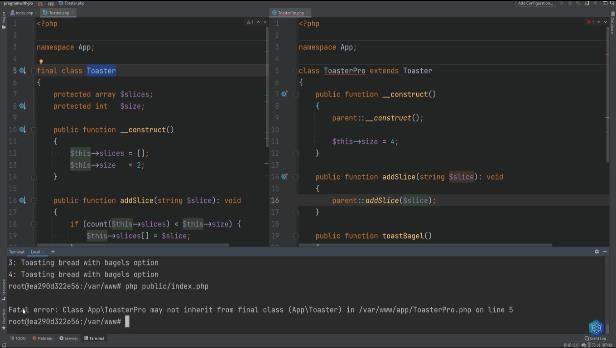
* + Nhưng nếu chúng tôi đang sử dụng các đối số được đặt tên ở đây, chúng tôi thấy rằng nó sẽ không hoạt động nếu tôi chạy lại nó, chúng tôi gặp lỗi đó là một lát tham số có tên không xác định, vì vậy vì tính năng đối số được đặt tên đã được giới thiệu nên tốt hơn là nên khớp với tên tham số



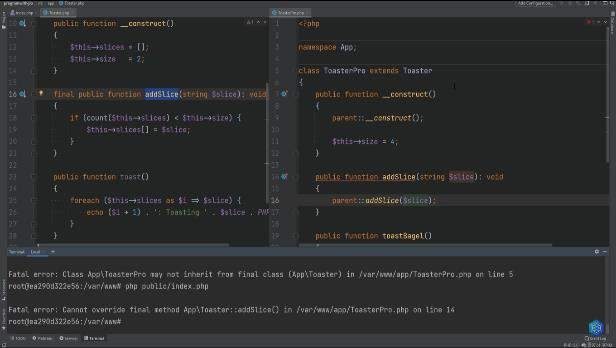
* + Vì vậy chúng tôi sẽ thay đổi tên này thành lát và mọi thứ sẽ hoạt động



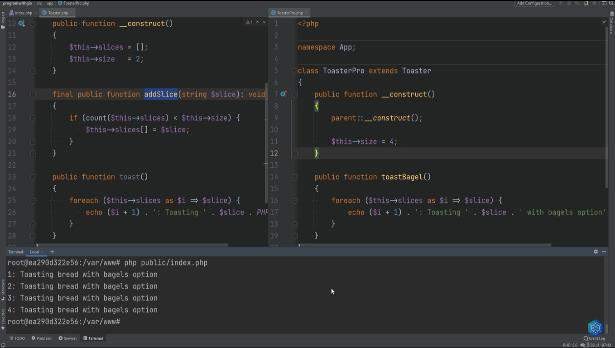
* + Nếu bạn muốn đảm bảo rằng một lớp hoặc một phương thức cụ thể có thể được mở rộng từ trong trường hợp đó, bạn có thể sử dụng từ khóa cuối cùng
  + Nếu bạn xác định một lớp là cuối cùng để bạn có từ khóa cuối cùng trước từ khóa lớp, điều này khiến bạn không thể mở rộng từ lớp này vì bạn có thể thấy id của tôi được gạch chân ở đây và cho biết với tôi rằng tôi không thể kế thừa hoặc mở rộng từ lớp cuối cùng vì vậy nếu tôi chạy mã này, chúng ta sẽ gặp phải lỗi nghiêm trọng



* + Tương tự như các lớp là cuối cùng, bạn cũng có thể có các phương thức là cuối cùng nên bạn có thể nói rằng phương thức thêm lát là cuối cùng, điều này tạo nên do đó, phương thức add slice không thể bị ghi đè nên nếu tôi chạy nó bây giờ thì chúng ta sẽ gặp một lỗi nghiêm trọng vì chúng ta đang cố ghi đè phương thức cuối cùng



* + Nếu tôi loại bỏ phương thức này và chạy nó ngay bây giờ thì mọi thứ đều hoạt động nên về cơ bản từ khóa cuối cùng có thể được sử dụng để ngăn chặn sự kế thừa lớp và ghi đè phương thức trước khi chúng ta chuyển sang các ví dụ khác.

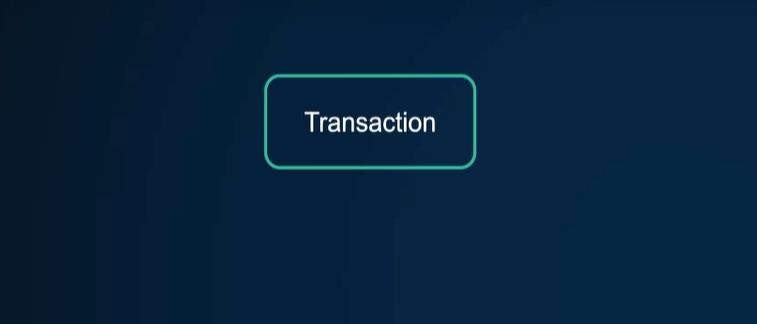


* + Php không hỗ trợ đa kế thừa, không thể mở rộng nhiều hơn một lớp tuy nhiên nó hỗ trợ kế thừa đa cấp có thể được sử dụng để đạt được kế thừa phân cấp
  + Có thể mở rộng **Toaster Pro** và lớp đó sẽ kế thừa tất cả các thuộc tính và hằng số phương thức công khai và được bảo vệ của cả lớp toaster pro và toaster



* + G/s rằng chúng ta có một lớp giao dịch nơi chúng ta xử lý các giao dịch,

v.v. và nó có tất cả các thuộc tính và phương thức cần thiết để xử lý giao dịch đó



* + Bây giờ hãy nói rằng chúng ta cũng có một loại giao dịch khác là giao dịch hoàn tiền và có thể được biểu diễn dưới dạng một lớp khác, vâng, bạn có thể có phương thức hoàn tiền trong lớp giao dịch nhưng nếu bạn có logic cụ thể về cách bạn hoàn tiền cho các giao dịch của mình trong trường hợp đó, giao dịch hoàn tiền có thể là một lớp khác có chức năng tùy chỉnh riêng

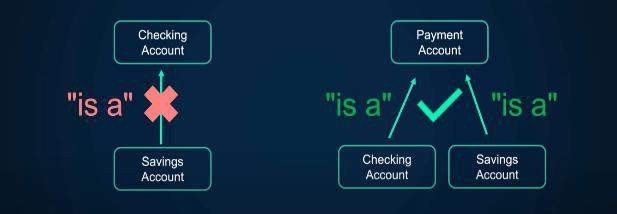
nhưng nó cũng kế thừa tất cả các thuộc tính và phương thức cần thiết từ lớp giao dịch.



* + Một ví dụ khác là với tài khoản thanh toán, bạn có thể có hai loại tài khoản, ví dụ: bạn có thể có tài khoản séc và bạn có thể có tài khoản tiết kiệm ngay bây giờ vì tài khoản séc và tiết kiệm chia sẻ một số chức năng phổ biến mà bạn có thể có tài khoản thanh toán làm lớp cơ sở và mở rộng tài khoản séc và tiết kiệm từ chúng



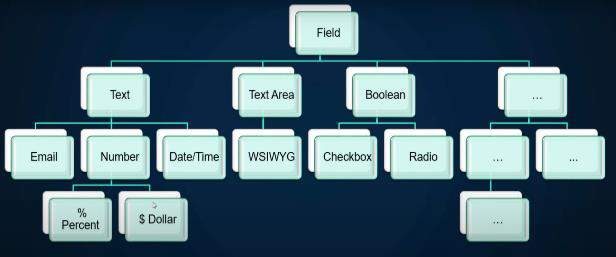
* + Hoặc bạn chỉ cần có một tài khoản séc và sau đó mở rộng tài khoản đó từ tài khoản tiết kiệm nhưng điều đó không thực sự có ý nghĩa bởi vì tài khoản tiết kiệm không phải là tài khoản séc và hãy nhớ rằng đó là một mối quan hệ và chúng ta sẽ đề cập đến vấn đề đó sau một phút nữa và tôi sẽ giải thích điều đó có nghĩa là gì.



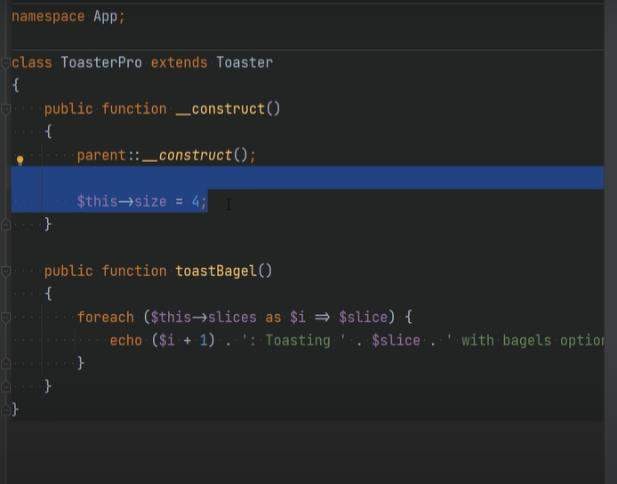
* + Ví dụ tiếp theo là với các trường biểu mẫu nên hãy coi nó như cấu trúc cây mà chúng ta có một trường cơ sở có thể có một số thuộc tính và một số phương thức áp dụng cho trường cơ bản , sau đó chúng tôi mở rộng trường cơ sở đó thành các trường văn bản thành các trường vùng văn bản thành

các trường loại boolean, v.v., sau đó mỗi trường loại đó có thể mở rộng hơn

nữa và chúng tôi có thể có trường văn bản email

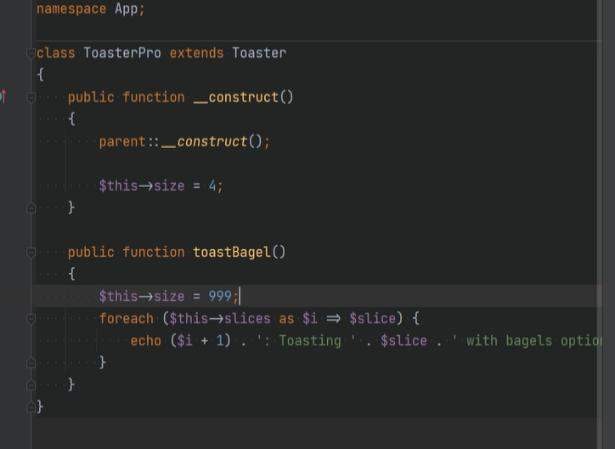


* + Ví dụ như trong trường hợp này tôi Tôi có thể trực tiếp thay đổi các thuộc tính của lớp cha và điều này có thể ổn vì đây là do thiết kế nhưng nó cũng có thể phá vỡ sự đóng gói khi bạn được phép làm điều này



* + Bạn có thể làm điều này theo bất kỳ phương pháp nào khác, bạn có thể thay

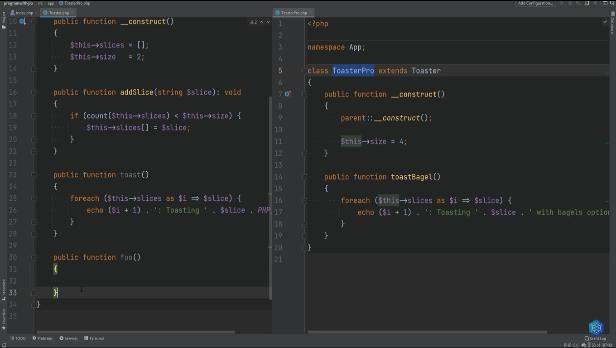
đổi thuộc tính size thành 999, v.v.,



* + Một nhược điểm khác của kế thừa là khi bạn kế thừa từ lớp cha, bạn thực sự đang kế thừa tất cả các phương thức công khai và được bảo vệ mà bạn thậm chí có thể không cần trong

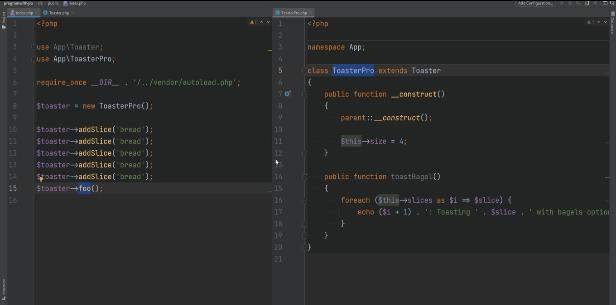
lớp con đó

* + Ví dụ: giả sử lớp máy nướng bánh mì ở đây có một số loại phương thức bổ sung không áp dụng cho máy nướng bánh mì chuyên nghiệp
  + Giả sử chúng ta đang làm một cái gì đó cụ thể cho lớp máy nướng bánh mì cơ bản bây giờ khi chúng tôi mở rộng máy nướng bánh mì sang máy nướng bánh mì chuyên nghiệp, máy nướng bánh mì chuyên nghiệp này có quyền truy cập vào phương thức đầy đủ này



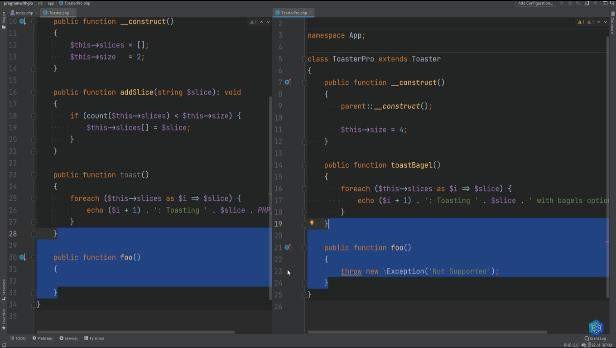
* + Và khi chúng tôi khởi tạo máy nướng bánh mì chuyên nghiệp ở đây, chúng tôi có thể gọi phương thức đầy đủ đó và điều đó có thể không lý tưởng vì

điều này Phương thức này có thể hoàn toàn không áp dụng cho toaster pro và bạn có thể không muốn cho phép gọi hàm đó trên toaster pro.

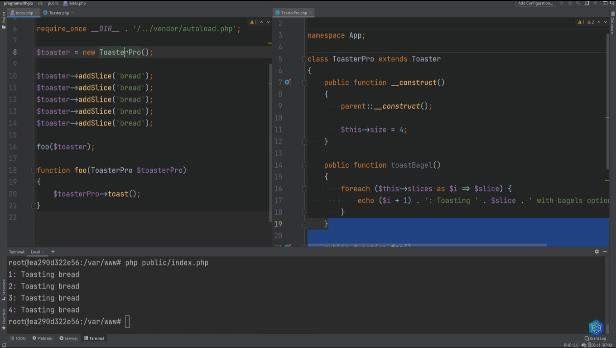


* + Giải pháp điển hình mà bạn có thể thấy trong một số cơ sở mã là ghi đè phương thức đó và chỉ cần ném một ngoại lệ và điều này vào Ý kiến của

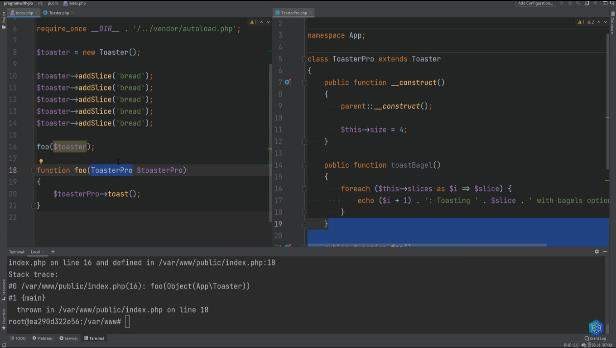
tôi không phải là một thiết kế tốt vì vậy nếu bạn thấy mình đang thực hiện kế thừa theo cách mà bạn buộc phải ghi đè một phương thức và đưa ra một ngoại lệ theo cách này



* + Vì vậy nếu tôi truy cập index.php, tôi sẽ chứng minh điều này bằng cách có một hàm tùy chỉnh ở đây chứ không phải một phương thức nên tôi sẽ thực hiện hàm và giả sử rằng chúng tôi đang chấp nhận toaster pro và gọi toaster pro đó và chúng ta hãy gọi toaster pro toast bây giờ chúng ta có thể gọi phương thức đó ở đây và chuyển toaster làm đối số rồi hãy loại bỏ nó và nó sẽ hoạt động

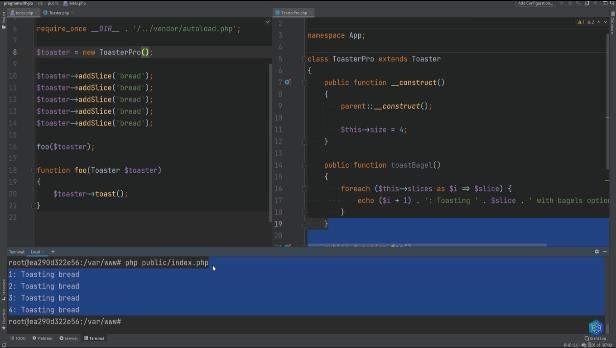


* + Ngay bây giờ nếu chúng ta chuyển vào toaster pro thay vì toaster pro thì điều này sẽ xảy ra dẫn đến lỗi vì máy nướng bánh mì không phải là máy nướng bánh mì chuyên nghiệp, máy nướng bánh mì không mở rộng máy nướng bánh mì chuyên nghiệp thay vào đó máy nướng bánh mì chuyên nghiệp mở rộng lớp máy nướng bánh mì nên máy nướng bánh mì là máy nướng bánh mì nhưng bản thân máy nướng bánh mì không phải là máy nướng bánh mì chuyên nghiệp.

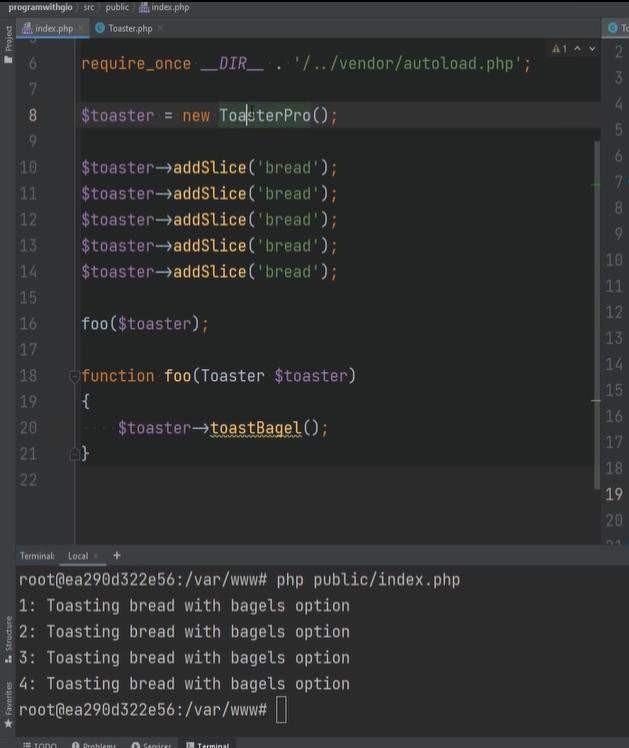


* + Cách bạn có thể khắc phục điều này là bạn có thể chỉ cần gõ gợi ý lớp cơ sở là máy nướng bánh mì và hãy thay đổi nó thành máy nướng bánh mì và phương thức nướng bánh mì có sẵn trên cả máy nướng bánh mì và máy

nướng bánh mì chuyên nghiệp và điều này sẽ hoạt động và nó cũng sẽ hoạt động nếu chúng ta chuyển vào máy nướng bánh mì chuyên nghiệp vì vậy nếu tôi chạy lại bây giờ nó đang nướng bốn lát

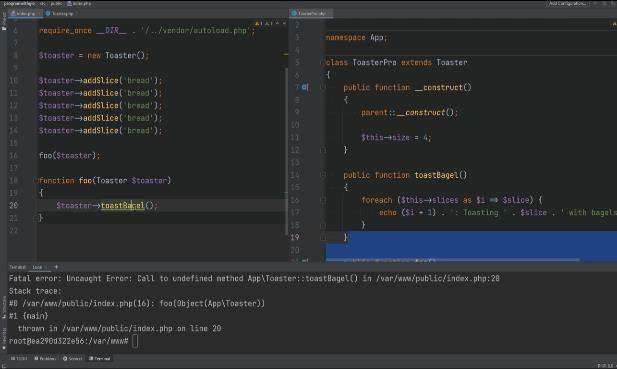


* + Có một vấn đề, tuy nhiên nếu bạn cố gắng gọi bánh mì nướng bagel thì điều này sẽ hiệu quả nếu bạn chuyển vào đối tượng chuyên nghiệp máy nướng bánh mì

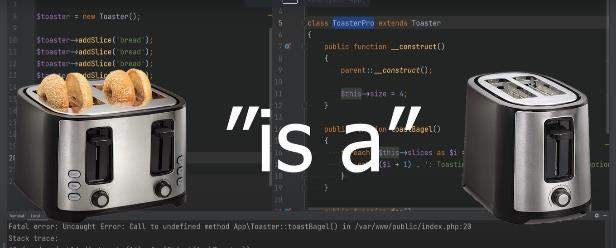


* + Nhưng nó sẽ không hoạt động nếu bạn chuyển vào máy nướng bánh mì

thông thường vì bánh mì nướng bagel không tồn tại



* + Và bạn có thể sử dụng mối quan hệ đó để quyết định xem bạn có thực sự cần quyền kế thừa hay không nếu bạn có một lớp mà bạn có thể áp dụng là một mối quan hệ, sau đó bạn có thể tận dụng sức mạnh của sự kế thừa trong trường hợp của chúng tôi là máy nướng bánh mì chuyên nghiệp là một máy nướng bánh mì



* + Giả sử rằng sau này bạn phải triển khai một trình mô phỏng cho một trong những lò nướng ưa thích cũng có chức năng nướng bánh mì và bánh mì tròn

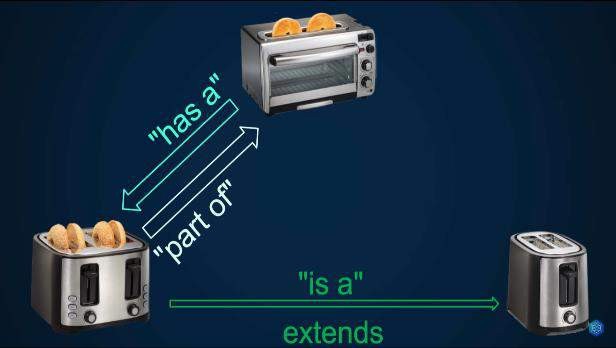


* + Bạn có thể muốn triển khai lò nướng đó bằng cách đơn giản mở rộng từ lớp máy nướng bánh mì hoặc đơn giản là mở rộng từ máy nướng bánh mì pro để kế thừa tất cả chức năng của máy nướng bánh mì và máy nướng bánh mì chuyên nghiệp , sau đó thêm chức năng bổ sung mà lò nướng đó hỗ trợ nhưng chúng sẽ gây ra nhiều vấn đề hơn trong khi duy trì và bổ sung thêm nhiều chức năng hơn cho ứng dụng của bạn

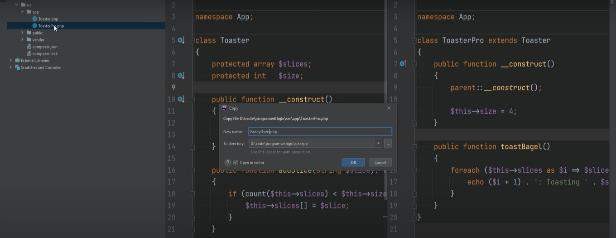


* + Vì vậy nếu bạn nghĩ theo cách khác nếu lò nướng không phải là máy

nướng bánh mì và không đủ điều kiện để trở thành một mối quan hệ thì làm thế nào chúng ta có thể kết hợp máy nướng bánh mì vào lớp lò nướng của mình mối quan hệ với tôi có vẻ như lò nướng có một máy nướng bánh mì hoặc máy nướng bánh mì là một phần của lớp lò nướng phải không

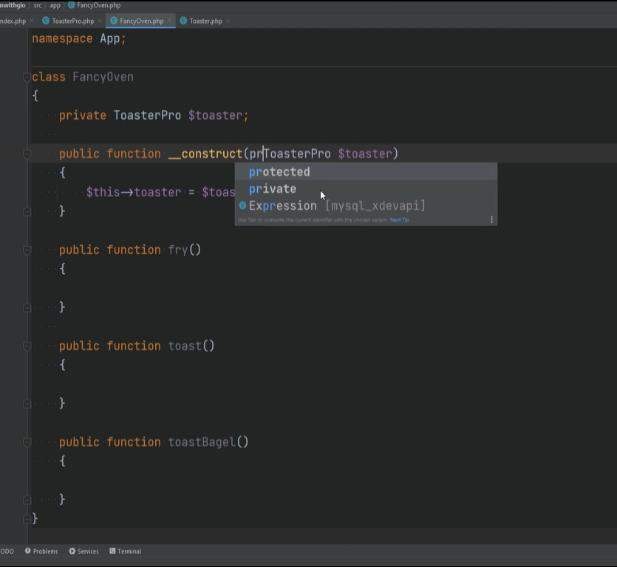


* + Chúng ta chỉ cần nhân đôi máy nướng bánh mì chuyên nghiệp ở đây và hãy cùng thực hiện hãy đặt tên nó là lò nướng ưa thích và hãy đóng cái này lại

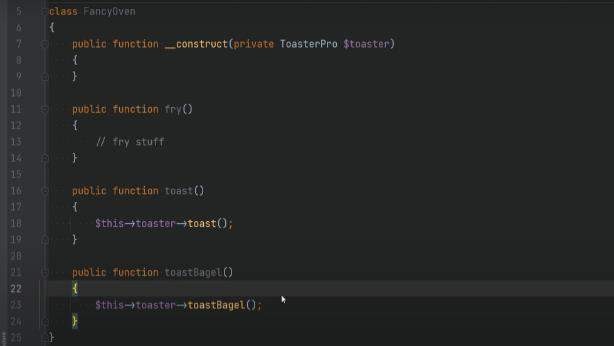


* + Có nó ngay tại đây và bánh mì nướng tồn tại trên lớp máy nướng bánh mì thay vì mở rộng máy nướng bánh mì chuyên nghiệp ở đây chúng ta có thể chỉ cần sử dụng một thứ gọi là thành phần trong đó lò nướng ưa thích có chức năng nướng bánh mỳ chuyên nghiệp và chúng ta có thể làm điều đó bằng cách sử dụng các thuộc tính đơn giản để chúng ta có thể làm máy

nướng bánh mỳ chuyên nghiệp riêng

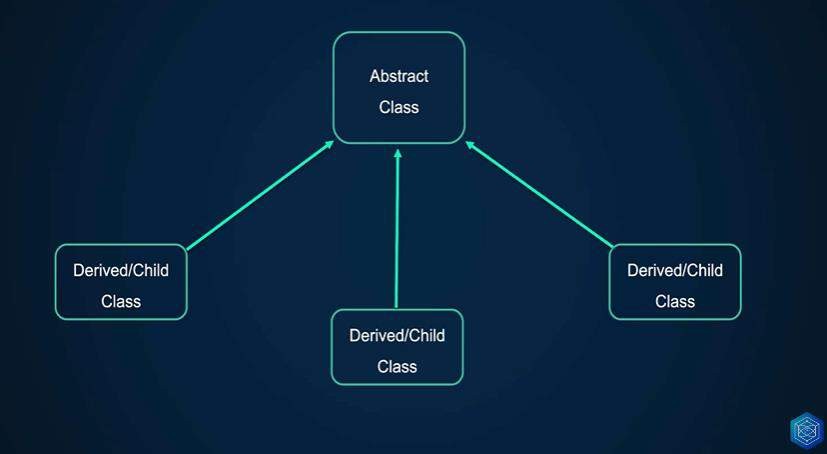


* + Gọi đơn giản là bánh mì nướng và bánh mì nướng bánh mì tròn này để bạn có thể thấy không sd tính kế thừa nhưng chúng ta vẫn có thể sd chức năng của Toaster Pro bằng cách sd thứ gọi là bố cục



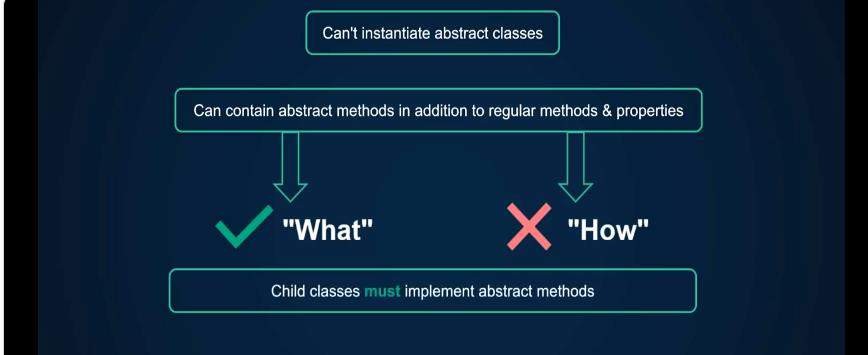
# Lớp và Phương thức Trừu tượng trong PHP - Hướng dẫn đầy đủ về PHP 8

* + - Một lớp trừu tượng giống như một khuôn mẫu hoặc lớp cơ sở mà đứa trẻ hoặc các lớp cụ thể có thể mở rộng.

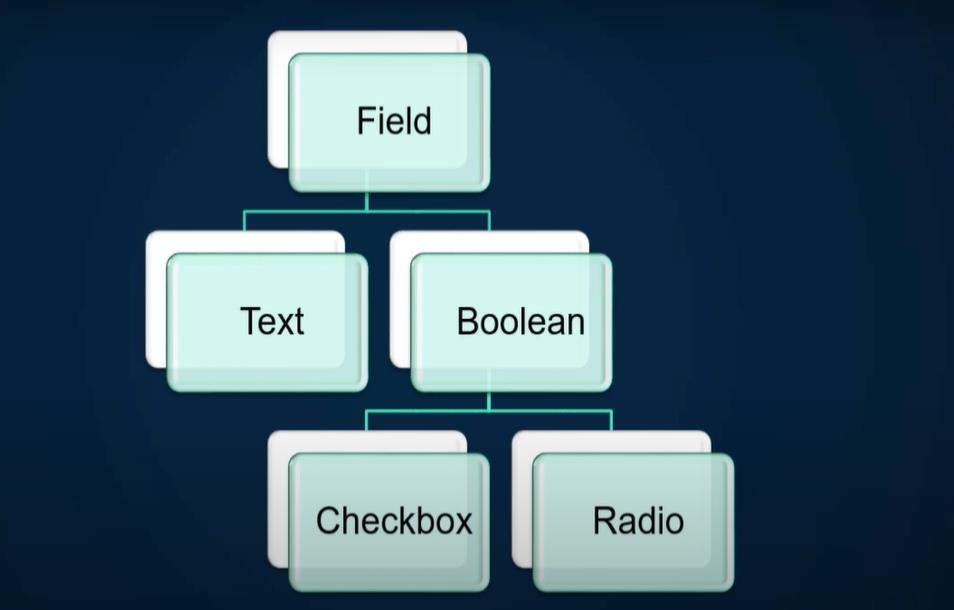


* + - Điều này được thực hiện bằng cách tận dụng sức mạnh của thừa kế, đã phục hồi trong bài học cuối cùng. Có rất ít quy tắc khi nói đến các lớp trừu tượng và các phương pháp trừu tượng.
    - Ví dụ, một lớp trừu tượng không thể được khởi tạo.

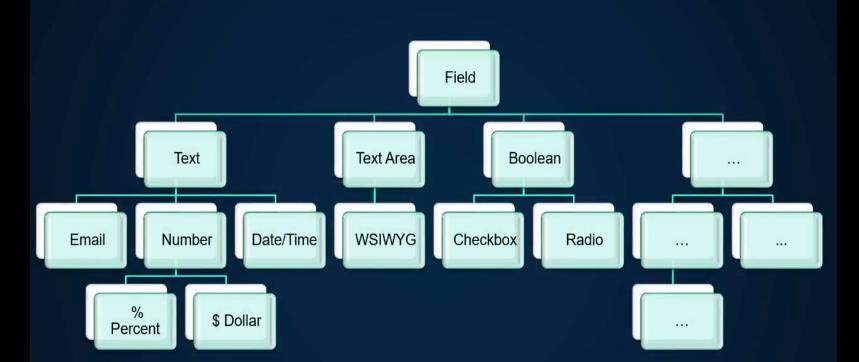
Bạn chỉ có thể mở rộng nó, có nghĩa là bạn không thể tạo các đối tượng trực tiếp từ lớp trừu tượng. Một lớp trừu tượng có thể có các phương thức trừu tượng chỉ chứa phương thức chữ ký hoặc định nghĩa mà không thực hiện thực tế. Về cơ bản, lớp trừu tượng biết cái gì, nhưng nó không biết làm thế nào. Phần làm thế nào được thực hiện trong các lớp con. Trách nhiệm của lớp con là thực hiện các phương pháp trừu tượng này.



* + - Hãy xem ví dụ này từ bài học trước. Chúng tôi có trường cơ sở ở đây và sau đó chúng tôi Có trường văn bản, trường vùng văn bản và trường boolean mở rộng từ trường cơ sở.



* + - Và sau đó chúng tôi đi sâu hơn.



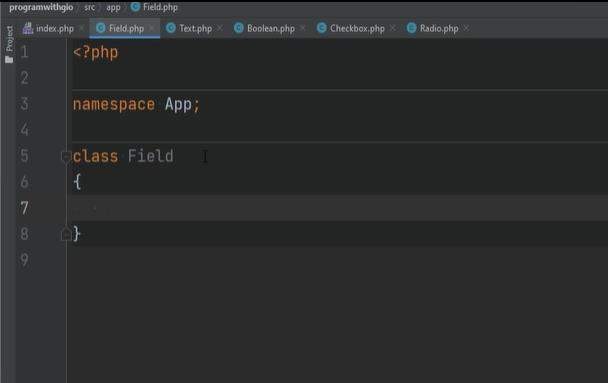
* + - Hãy triển khai phiên bản đơn giản hơn của điều này, nơi chúng ta sẽ có

trường cơ sở và sau đó chúng ta sẽ có trường văn bản và boolean trường

mở rộng từ trường cơ sở, và sau đó chúng ta sẽ có hộp kiểm và các trường nút radio mở rộng từ trường Boolean.

* + - Tôi có các lớp này được tạo ở đây. Tôi có lớp trường cơ sở, Tôi có lớp văn

bản, lớp boolean, hộp kiểm và tôi có nút radio.



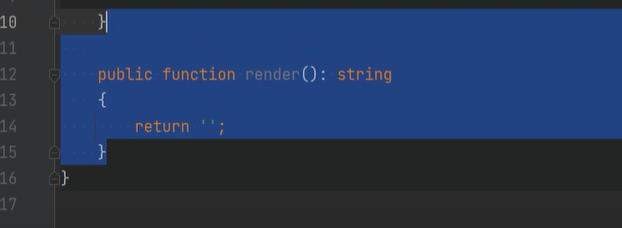
* + - Chúng chỉ trống rỗng ngay bây giờ, vì vậy hãy điền chúng vào. Trước khi chúng ta nhảy vào các lớp trừu tượng, chúng ta hãy chỉ sử dụng kế thừa thông thường.
    - Trường cơ sở có thể có một số thuộc tính và các phương thức được chia sẻ bởi tất cả các trường. Tất cả các trường có thể có tên, Ít nhất là tất cả các trường mà chúng ta có ngay tại đây, trường văn bản và hộp kiểm và các nút radio, tất cả đều có các thuộc tính tên. Chúng ta có thể định nghĩa name là một trong những thuộc tính.
    - Tôi sẽ tạo construct() và tôi sẽ sử dụng tính năng khuyến mãi bất động sản

ở đây.

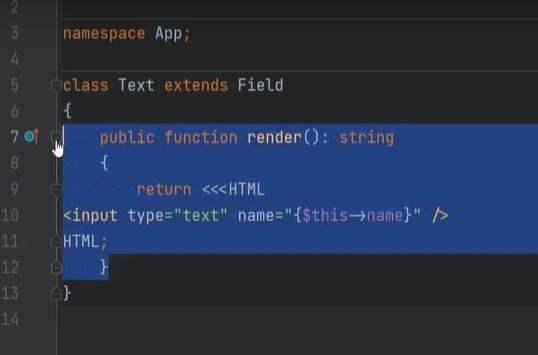
* + - Chúng ta sẽ có contruct(protected string $name)



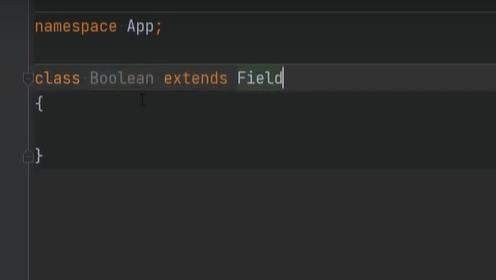
* + - Và sau đó chúng ta sẽ có một phương thức để render trường. Chúng ta có thể có kết xuất hàm công cộng trả về chuỗi và trường cơ sở này chỉ đơn giản trả về một chuỗi trống vì chúng ta chưa biết phải kết xuất cái gì.



* + - Hãy sao chép cái này và đưa nó vào các trường khác. Ngoài ra, chúng ta có thể mở rộng trường cơ sở ở đây Và hãy đặt nó vào để ghi đè lên phương thức kết xuất.
    - Tôi sẽ sử dụng ở đây. Cú pháp Dox để trả về HTML. Chúng ta sẽ có đầu vào, loại, văn bản, và sau đó đặt tên, và chúng ta có thể đặt tên này và sau đó đóng nó lại.



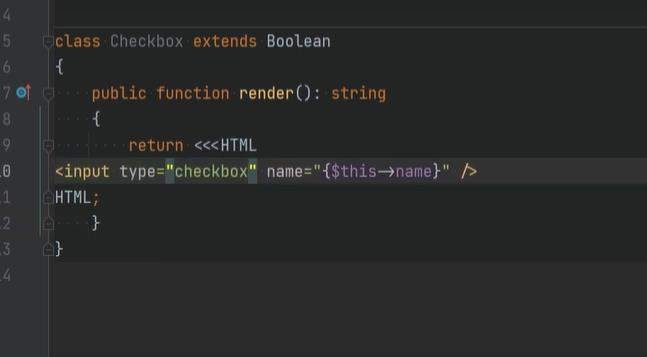
* + - Bây giờ, tất nhiên, bạn không phải sử dụng ở đây.(<<<HTML). Cú pháp Dox trong ứng dụng thực. Bạn có thể sẽ sử dụng một tập tin xem và hiển thị nội dung thành một chuỗi và trả về nội dung đó thay thế. Nhưng hãy giữ mọi thứ đơn giản cho ví dụ này.
    - Tôi sẽ sao chép cái này và đưa nó vào các lĩnh vực khác.Trong lớp boolean, chúng ta có thể chỉ cần mở rộng trường cơ sở.



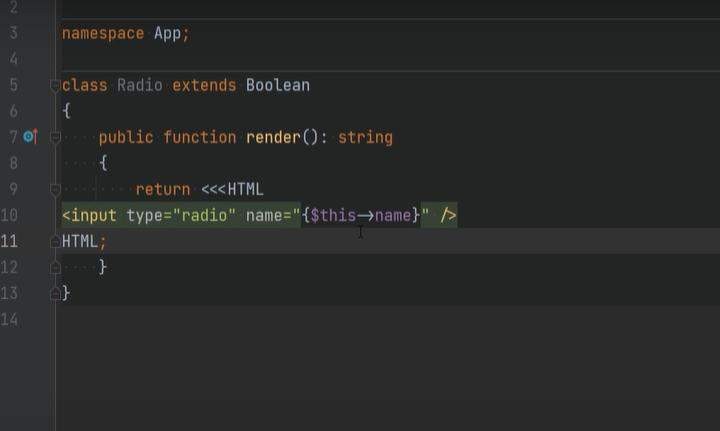
* + - Bởi vì chúng tôi không biết làm thế nào để hiển thị một trường Boolean.

Thay vào đó, chúng ta sẽ ghi đè lên. Phương thức kết xuất trong các trường hộp kiểm và nút radio.

* + - Chúng ta đi đến hộp kiểm và hãy mở rộng lớp boolean và chúng ta ghi đè lên phương thức kết xuất và chúng tôi chỉ cần đặt hộp kiểm loại và chúng tôi đi đến radio



* + - Mở rộng boolean, và chúng tôi thay đổi loại thành radio.



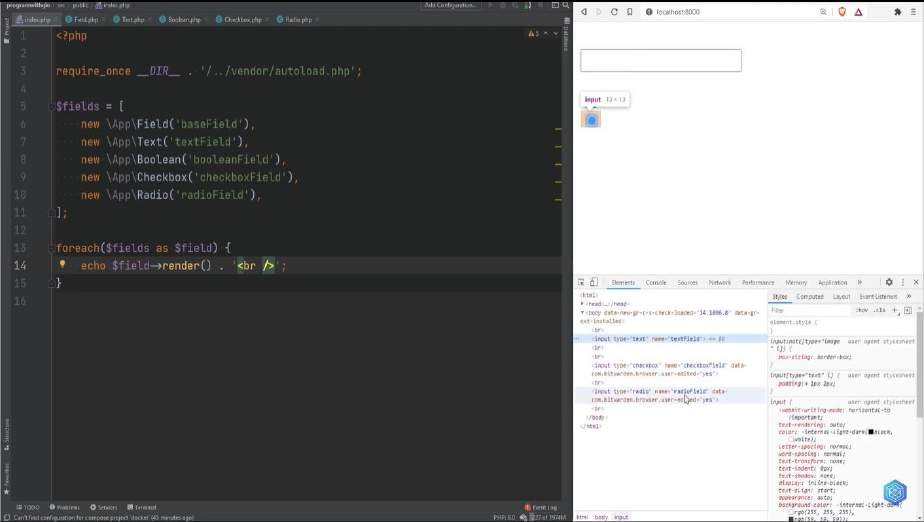
* + - Bây giờ, tất nhiên, có nhiều cách khác nhau để thực hiện những điều như

vậy.

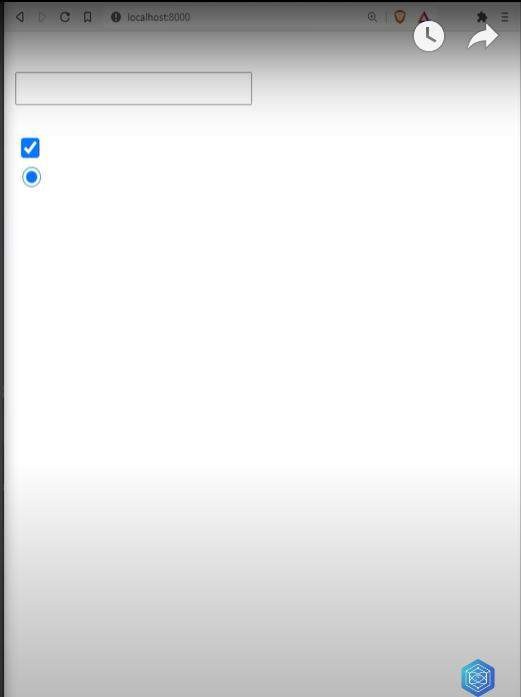
* + - Ví dụ: thay vì có nhiều lớp, bạn chỉ cần lưu trữ Nhập vào một thuộc tính và hiển thị nó theo cách đó. Nhưng bởi vì mỗi lĩnh vực riêng lẻ có thể có logic tùy chỉnh riêng, nó tốt hơn để tách các mối quan tâm và đặt chúng trong các lớp riêng của nó, ngay cả khi bạn đang lặp lại một chút mã.
    - Bằng cách đó bạn có thể linh hoạt hơn và nhiều chỗ hơn để thêm vào các

tính năng cụ thể của trường cho lớp cụ thể đó.

* + - Ngay bây giờ, chúng tôi chưa sử dụng bản tóm tắt.
    - Hãy đi đến index.php và tạo một mảng các trường sẽ chứa một thể hiện của từng trường đó. Hãy tạo một đối tượng của trường cơ sở và truyền vào một số tên. Chúng ta sẽ gọi trường cơ sở này. Hãy sao chép trường này bốn lần Và hãy thay đổi chúng thành các trường tương ứng.Chúng tôi có một mảng gồm năm lĩnh vực.

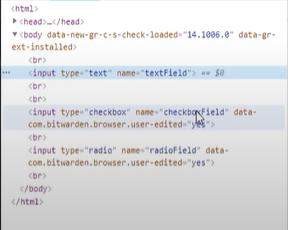


* + - Bây giờ chúng ta hãy xem qua từng trường này và hãy hiển thị và in ra HTML. Hãy thêm một đường ngắt ở đây và hãy làm mới trang và chúng ta thấy một hộp nhập, Chúng tôi thấy một hộp kiểm và chúng tôi thấy một nút radio. Nó dường như đang hoạt động.



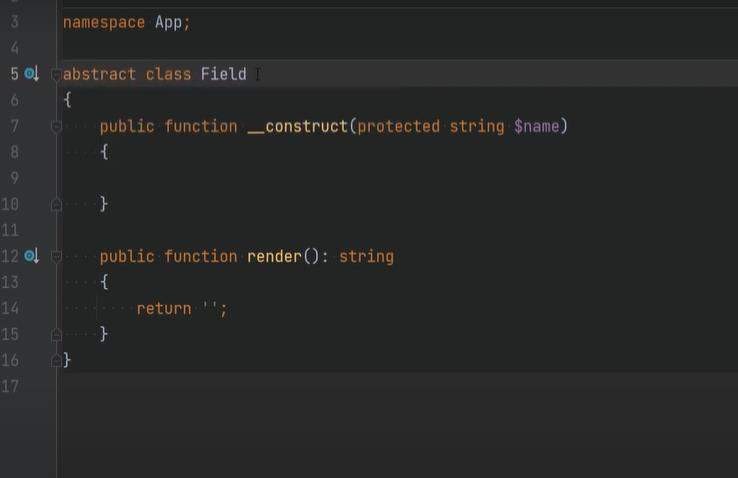
* + - Sau đó, chúng ta có khoảng trống cho trường cơ sở Và chúng ta có khoảng trống cho trường Boolean.
    - Nếu chúng ta mở DevTools, chúng ta thấy rằng tên được đặt đúng. Nó

được đặt thành trường văn bản, trường hộp kiểm và trường radio.

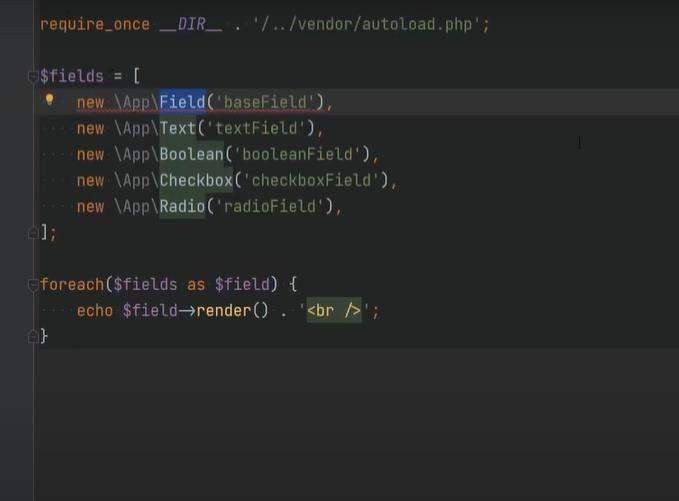


* + - Nhưng có một vấn đề ở đây. Chúng tôi có thể tạo các phiên bản của trường cơ sở và trường boolean cơ sở trực tiếp. Đó là điều mà chúng tôi không muốn cho phép vì chúng tôi không biết làm thế nào để hiển thị trường cơ sở và chúng tôi không biết cách hiển thị trường Boolean.
    - Chúng tôi biết làm thế nào để kết xuất các lớp cụ thể. Chúng tôi biết cách hiển thị trường văn bản. Chúng tôi biết làm thế nào để hiển thị hộp kiểm và radio.
    - Nhưng hai điều này là các trường cơ sở hoặc mẫu cung cấp một số cơ sở chức năng cho tất cả các trường này mở rộng từ. Đó là nơi bạn có thể sử dụng các lớp trừu tượng.
    - Nếu chúng ta đi đến lớp trường, chúng ta có thể thay đổi lớp này thành một lớp trừu tượng. Chúng ta có thể làm điều đó bằng cách sử dụng từ khóa trừu tượng ngay trước lớp. Bây giờ chúng ta không còn có thể tạo các đối

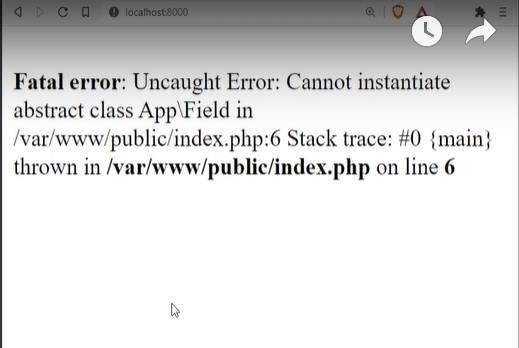
tượng của lớp này nữa.



* + - Nếu chúng ta đi đến index.php chúng ta thấy rằng nó đã được gạch chân ngay tại đây.



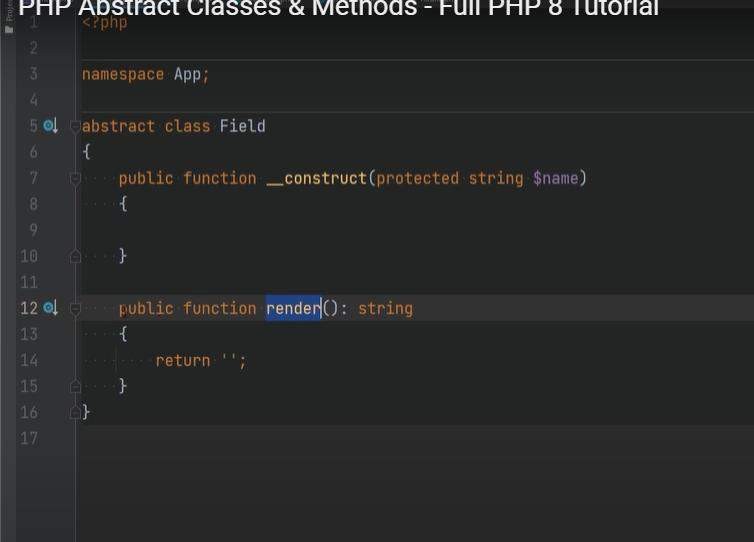
* + - Nếu tôi làm mới, chúng ta sẽ nhận được lỗi nghiêm trọng.



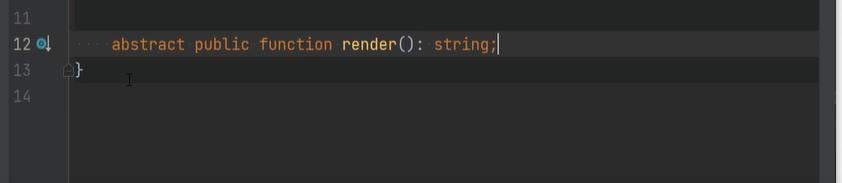
Hãy loại bỏ điều đó khỏi đây và hãy tiếp tục.

* + - Một điều khác chúng ta có thể làm ở đây là rằng phương thức render chỉ trả về một chuỗi trống.
    - Bây giờ, như tôi đã đề cập ở phần đầu Trong bài học, bạn có thể có các phương pháp trừu tượng. Một phương pháp trừu tượng chỉ có định nghĩa mà không thực hiện.Đây là một ví dụ hoàn hảo cho điều đó.Chúng tôi biết rằng tất cả các lĩnh vực này cần phải có
    - Phương thức kết xuất, nhưng chúng tôi không biết chúng thực sự được triển khai như thế nào. Điều đó nên tùy thuộc vào lớp con để xác định cách nó thực sự được thực hiện. Đó là nơi chúng ta có thể sử dụng các phương pháp trừu tượng bởi vì ngay bây giờ, chúng ta có thể đơn giản Để trống trường này và trường văn bản sẽ không được hiển thị và điều đó không lý tưởng.
    - Chúng ta muốn buộc các lớp con thực hiện phương thức render. Để làm

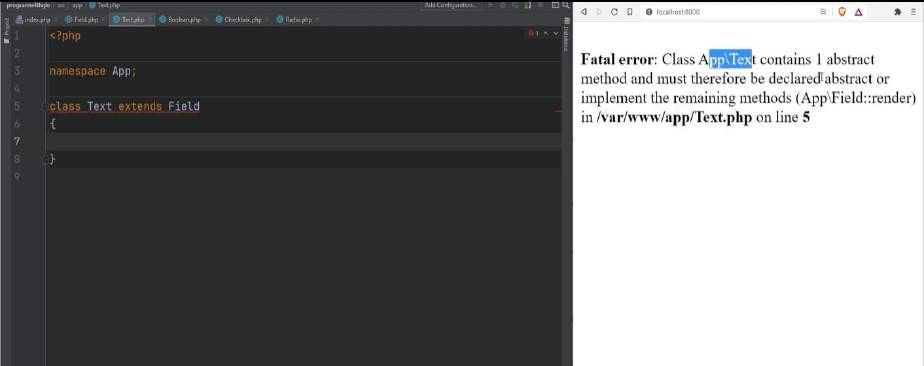
điều đó, chúng ta có thể sử dụng các phương pháp trừu tượng. Chúng ta có thể đến đây và viết trước public Chúng ta có thể làm abstract public function Nó được gạch chân ở đây bởi vì bạn chỉ có thể có chữ ký phương thức, bạn không thể thực hiện.



Chúng ta cần loại bỏ việc triển khai và đặt dấu chấm phẩy ngay tại đây.

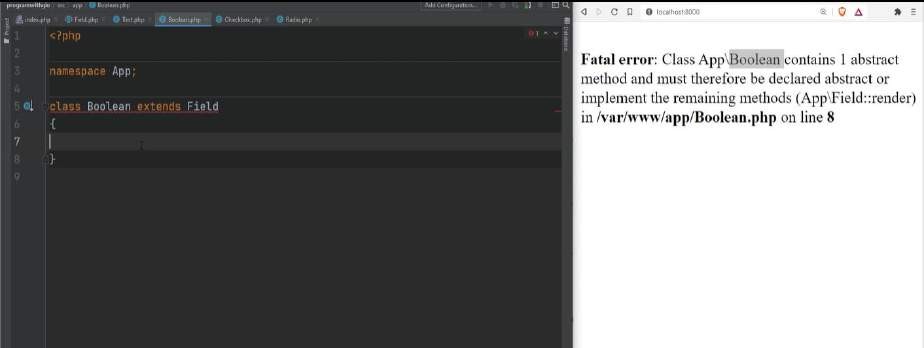


* + - Bây giờ, điều này có nghĩa là tất cả các lớp Lớp trường mở rộng đó cần triển khai phương thức kết xuất. Nếu bạn đi đến text class bây giờ, Nó gây ra lỗi cho chúng tôi vì chúng tôi đã xóa phương thức kết xuất. Nếu tôi làm mới trang, chúng tôi sẽ gặp lỗi nghiêm trọng



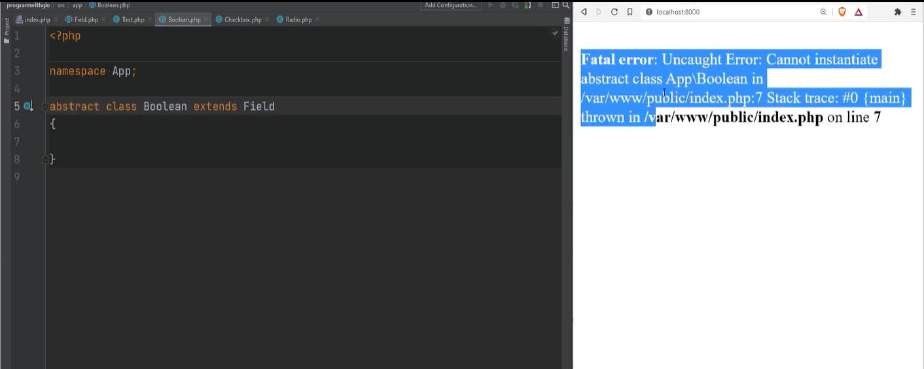
đó là văn bản lớp chứa một phương thức trừu tượng và do đó nó cần được khai báo tóm tắt hoặc thực hiện các phương pháp còn lại.

* + - Điều đó có nghĩa là nếu bạn không. Thực hiện phương pháp trừu tượng, sau đó bạn cần đánh dấu lớp là trừu tượng. Nếu bạn có ít nhất một phương pháp trừu tượng, Bản thân lớp cần được khai báo là trừu tượng. Bạn có một phương pháp trừu tượng ở đây bởi vì bạn thừa hưởng tất cả công chúng và các phương thức được bảo vệ từ lớp trường cơ sở.
    - Do đó, vì lớp trường có phương pháp trừu tượng, Chúng ta cần khai báo class này abstract hoặc chúng ta cần thực hiện phương thức render.
    - Trong trường hợp này, chúng tôi sẽ chỉ mang theo Quay lại phương thức render và chúng tôi làm mới trang và chúng tôi vẫn gặp lỗi này. Nhưng bây giờ lỗi đến từ lớp boolean.

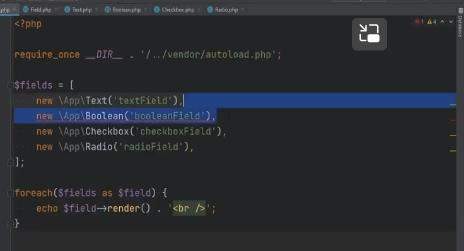


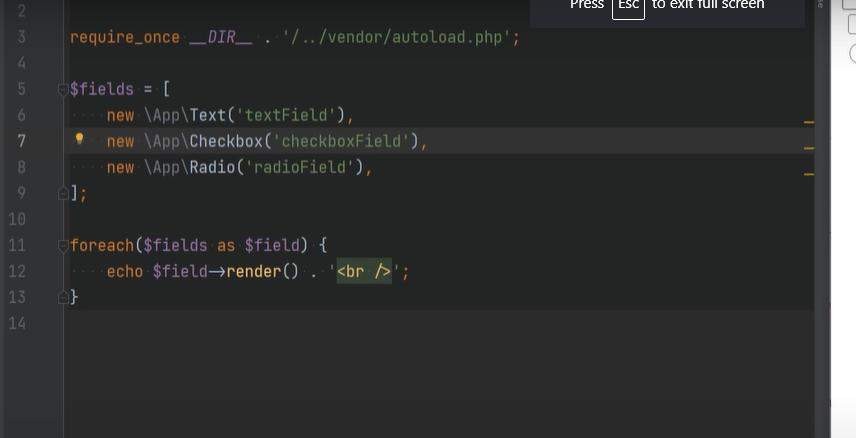
* + - Nếu chúng ta chuyển sang Boolean, Chúng tôi thấy rằng chúng tôi chưa bao giờ triển khai phương thức kết xuất ở đây. Trong trường hợp này, chúng ta không cần phải thực hiện phương thức render bởi vì, một lần nữa, chúng ta không biết cách hiển thị các trường Boolean. Tùy thuộc vào các lớp cụ thể để tìm ra cách hiển thị các trường của riêng họ. Hộp kiểm phải tìm ra cách hiển thị hộp kiểm, v.v.
    - Trong trường hợp này, chúng ta cũng có thể đánh dấu lớp này là trừu

tượng. Nếu tôi làm mới trang, chúng tôi sẽ gặp lỗi nghiêm trọng.

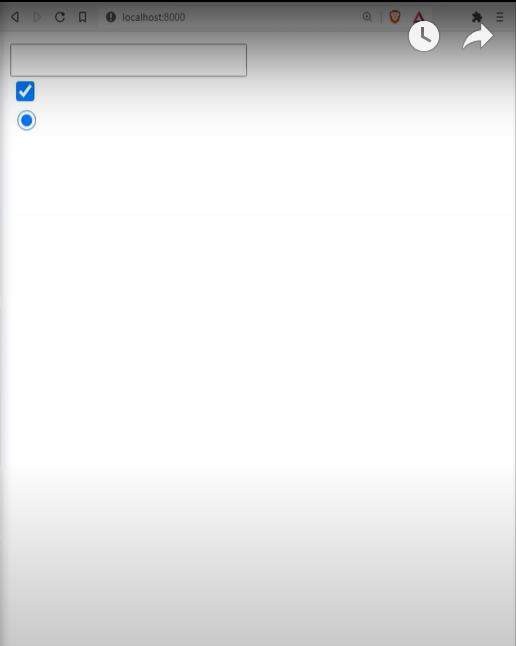


* + - Đó là bởi vì chúng ta đang cố gắng tạo một đối tượng của lớp boolean trong index.php
    - Chúng ta cũng cần xóa cái này.





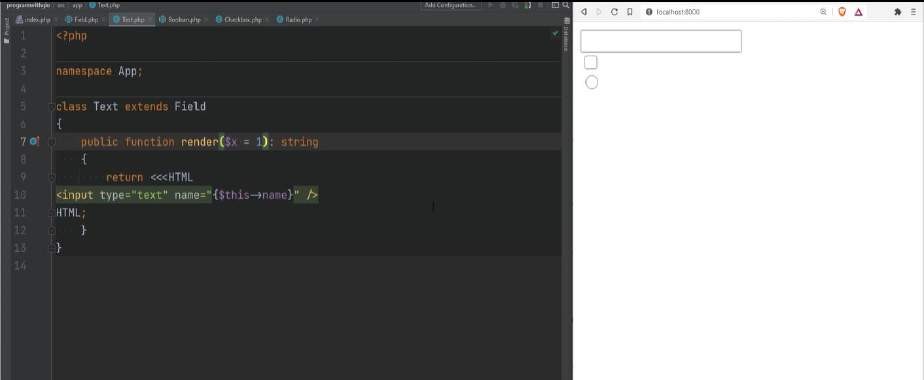
* + - Mọi thứ đã trở lại bình thường. Bây giờ chúng ta có trường văn bản, trường hộp kiểm và nút radio.



* + - Bởi vì chúng ta đang sử dụng kế thừa ở đây, tất cả các lớp con phải tuân theo Các quy tắc kế thừa thông thường và tương thích chữ ký.
    - Chúng tôi đã thảo luận điều này trong bài học cuối cùng như tốt, nhưng tôi quên đề cập đến một điều mà các phương pháp mà bạn đang ghi đè
    - Trong lớp con thực sự có thể có các đối số bổ sung với các giá trị mặc định.

Điều này cũng áp dụng cho việc thực hiện các phương pháp trừu tượng.

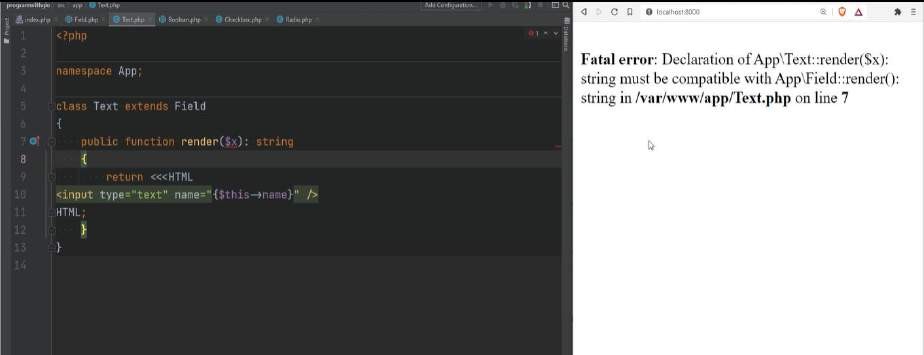
* + - Ví dụ: phương thức kết xuất, nó không chấp nhận bất kỳ đối số nào, Nhưng bạn có thể ghi đè lên nó và chấp nhận một số đối số với các giá trị mặc định.
    - Bạn có thể nói X bằng một, và điều này là hợp lệ. Nếu tôi làm mới trang, mọi thứ vẫn hoạt động.



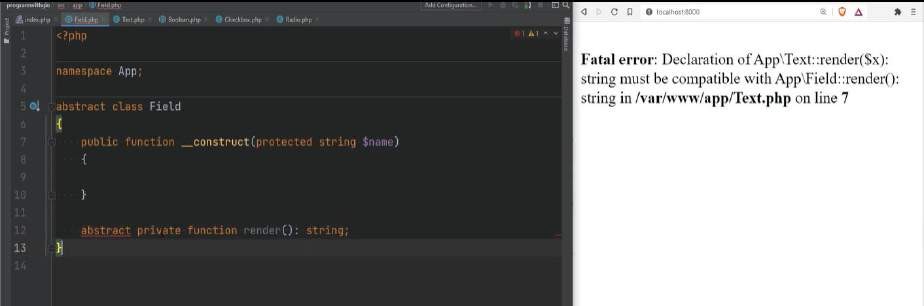
* + - Tuy nhiên, nếu bạn không xác định các giá trị mặc định, thì điều này không còn tương thích nữa và nó sẽ thất bại vì phương thức cha mẹ không có bất kỳ đối số nào và đứa trẻ .Phương pháp, bạn đang yêu cầu một đối số và

điều đó không hoạt động.

* + - Nếu tôi làm mới, chúng ta sẽ nhận được lỗi nghiêm trọng.



* + - Một điều khác bạn nên biết là Khả năng hiển thị của phương pháp trừu tượng đó có thể được công khai hoặc được bảo vệ.
    - Bạn không thể đặt tùy chọn này thành riêng tư. Điều đó có ý nghĩa bởi vì bạn không thể ghi đè lên các phương thức riêng tư. Khi bạn khai báo các phương thức là trừu tượng,

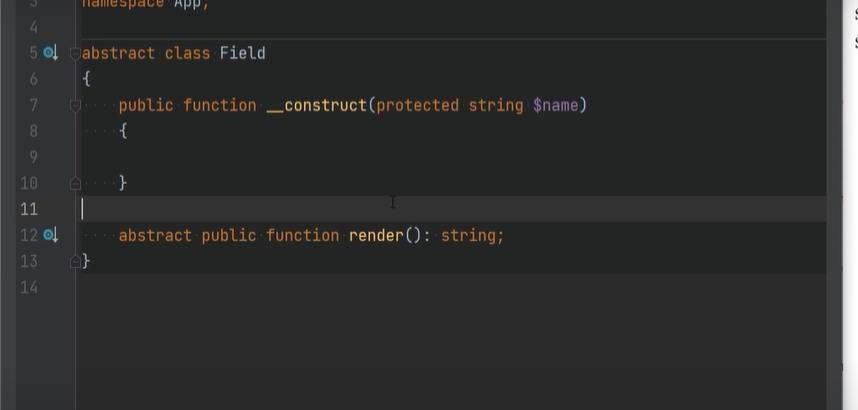


* + - Bạn cần triển khai phương thức đó trong lớp con và chúng ta không thể ghi

đè lên

* + - Phương pháp riêng trong lớp con và do đó bạn sẽ không làm việc. Đó là lý

do tại sao bạn cần phải có điều này là được bảo vệ hoặc công khai.



Khi nào bạn sẽ sử dụng một lớp trừu tượng?

* + - Bạn có thể sử dụng một lớp trừu tượng khi bạn muốn buộc các lớp con

để thực hiện các phương pháp trong khi bạn cung cấp một số chức năng cơ bản.

Theo chức năng cơ bản, ý tôi là ngay bây giờ chúng tôi không có nhiều

Chức năng cơ bản trong lớp này, nhưng bạn có thể có một loạt các chức năng

khác

Các phương pháp không trừu tượng có thể được sử dụng trên tất cả các lớp con.

Sau đó, bạn có thể có một số thuộc tính chung ở đây áp dụng cho tất cả các lớp học trẻ em và như vậy.

Bây giờ, nếu bạn có một lớp trừu tượng trong đó Bạn có một loạt các phương pháp trừu tượng, thì có lẽ bạn nên xem xét không sử dụng trừu tượng và thay vào đó tiếp cận với một thứ gọi là giao diện.

Chúng ta sẽ nói về giao diện trong bài học tiếp theo.