**Magic methods**

* \_\_construct()
  + Gọi \_\_construct của lớp cha: parent::\_construct
* \_\_destruct(): thực thi khi obj bị xóa?
* \_\_get($pro): dùng chung với property\_exists, dùng để truy xuất 1 pro private, protected, nếu có truy xuất trực tiếp, PHP sẽ tìm coi có hàm này ko
* \_\_set($pro, $val)
* \_\_isset(), \_\_unset()
* \_\_toString(): khi echo 1 object, nó sẽ gọi tới hàm này trc
* ­­\_\_call($method, $params): dùng khi client gọi method ko public
* Clone: copy object, chứ ko chỉ lấy tham chiếu, nhưng nếu có obj trong obj thì sẽ vẫn còn tham chiếu => dùng \_\_clone
* \_\_invoke: dùng obj như 1 hàm

**Exception**

* Throwable interface
* <https://trowski.com/2015/06/24/throwable-exceptions-and-errors-in-php7/>
* <https://www.quora.com/What-are-the-major-difference-between-PHP-5-and-PHP-7>
* <http://blog.teamtreehouse.com/5-new-features-php-7>
* <http://ralphschindler.com/2010/09/15/exception-best-practices-in-php-5-3>
* <https://www.theodo.fr/blog/2014/04/manage-php-errors-and-exceptions-in-your-project/>
* <https://daveyshafik.com/archives/69237-an-exceptional-change-in-php-7-0.html>
* <http://www.guru99.com/error-handling-and-exceptions.html>
* <https://www.alainschlesser.com/structuring-php-exceptions/>
* <https://www.sitepoint.com/exceptional-exceptions/>
* <https://wiki.haskell.org/Error_vs._Exception>
* <https://blog.pascal-martin.fr/post/php71-en-flashback-php70.html>
* <https://www.tutorialspoint.com/php/php_error_handling.htm>
* PHP 5 ko xư lý các lỗi nghiêm trọng (fatal error) mà sẽ dừng thực thi khi gặp chúng. Khi ấy, user sẽ thấy 1 trang trắng (blank page) vì ko có thông báo lỗi nào. Nó cũng tạo ra vấn đề đối với những nguồn tài nguyên đag đc sử dụng mà chưa đc đóng
* PHP 7: exception sẽ đc throw khi 1 lỗi nghiêm trọng xảy ra, hơn là ngừng đoạn script
* Có những lỗi vẫn sẽ dừng đoạn script
* Uncaught exception tiếp tục là 1 fatal error trong php 7
* Dùng throwable trong try/catch để bắt exception và error, nhưng sẽ tốt hơn nếu chỉ rõ ra loại exception, error
* Lỗi fatal error: catchable và non-catchable?
* PHP 7 sẽ quăng exception cho những lỗi PHP tiêu chuẩn, nghĩa là ta có thể xử chúng bằng try/catch
* Trc PHP 5: ko có xử lý exception
* Error đc thiết kế để hiển thị message và shut down chương trình
* Nếu ko dùng loại Error này thì các fatal error sẽ crash, shut down chương trình
* Ta có thể hiển thị message cho user, ghi vào log, hoặc bỏ qua
* Finally: close file, databases
* Nesting: getPrevious(): lấy cái exception trc đó
* New core exception types: trc đó, các dev thường đưa thêm thông tin vào exception để làm rõ nó
  + Có 13 loại exception mới của SPL, gồm 2 loại lớn là LogicException, RuntimeException
* Error reporting level: phụ thuộc vào mức độ của error, chương trình có thể chạy tiếp hoặc dừng lại, dùng set\_error\_handler để catch error
* Exception: nếu ko có catch thì trở thành fatal error
* Trc php 7, để convert error sang exception, dùng throw exception trong hàm set\_error\_handler
* Tạo ra class exception của chính mình để xử lý phù hợp với loại exception đó. VD: FileNotFoundException, xử lý: tạo 1 file
* Error: là kết quả ko mong muốn mà chương trình ko thể tự xử lý, phải sửa lại code VD: vòng lặp vô hạn. exception: kết quả ko mong muốn mà chương trình có thể tự giải quyết. VD: mở 1 file ko tồn tại
* Xử lý exception để:
  + Tránh những kết quả ko mong muốn
  + Thay đổi flow của chương trình khi có lỗi dự đoán đc xảy ra
* Set\_error\_handler: cho phép chỉnh sửa message, log ra file, email cho dev @@