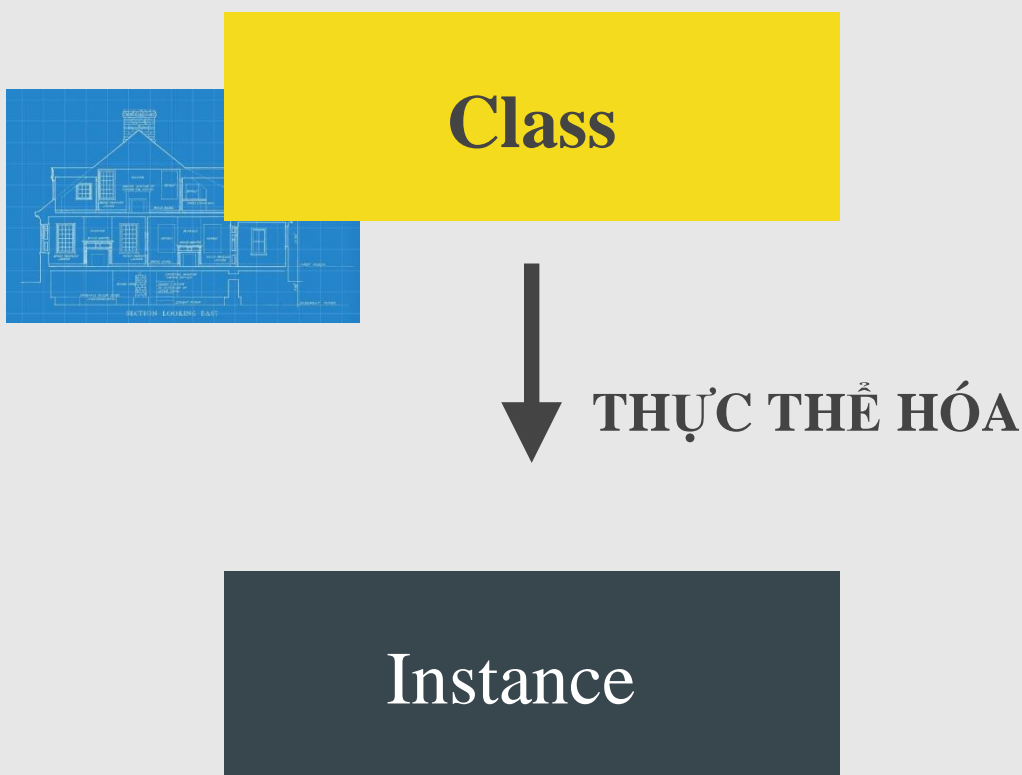


OOP TRONG JAVASCRIPT: NGUYÊN MẪU

“OOP CỔ ĐIỂN”: CLASS

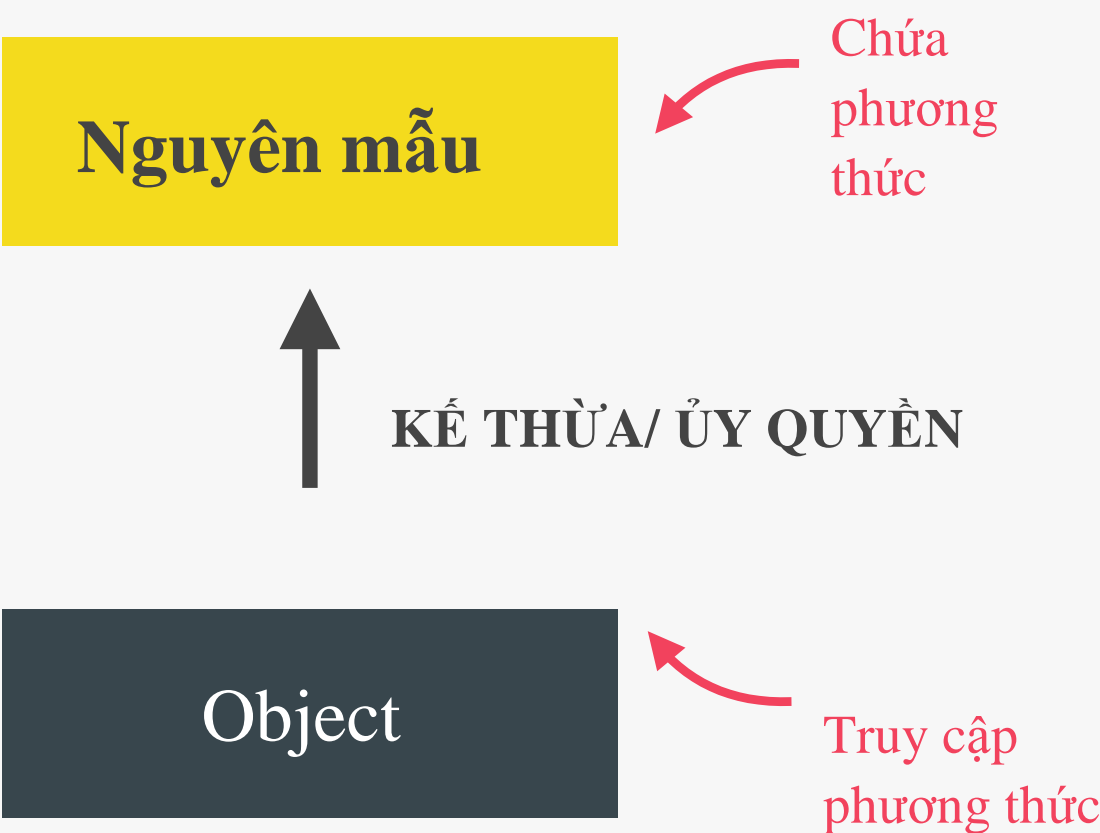


Object (instance) được **thực thể hóa** từ một class, có chức năng như một bản thiết kế chi tiết;



Hành vi (phương thức) được **sao chép** từ class đến tất cả các instance.

OOP TRONG JS: NGUYÊN MẪU



Các đối tượng **liên kết với** đối tượng nguyên mẫu



Kế thừa Prototypal: Nguyên mẫu chứa các phương thức (hành vi) có thể **truy cập** cho tất cả các đối tượng được liên kết với nguyên mẫu đó;



Hành vi được **ủy quyền** cho đối tượng nguyên mẫu được liên kết.



Ví dụ: Array

```
const num = [1, 2, 3];  
num.map(v => v * 2);
```



MDN web docs
moz://a

```
Array.prototype.keys()  
Array.prototype.lastIndexOf()  
Array.prototype.map()
```

Array.prototype là nguyên mẫu của tất cả các đối tượng array mà chúng ta tạo trong JavaScript.

Do đó, **tất cả các array** đều có quyền truy cập vào phương thức **map**!

```
▼ f Array() ⓘ  
  arguments: (...)  
  caller: (...)  
  length: 1  
  name: "Array"  
  prototype: Array(0)  
    ▶ unique: f ()  
      length: 0  
    ▶ constructor: f Array()  
    ▶ concat: f concat()  
    ▶ map: f map()
```

3 CÁCH THỰC HIỆN KẾ THỪA PROTOTYPAL TRONG JAVASCRIPT

🤔 “Làm thế nào để thực sự tạo ra các nguyên mẫu? Và làm thế nào để liên kết các đối tượng với các nguyên mẫu? Làm thế nào để tạo ra các đối tượng mới, mà không cần phải có các class?”

1 Các hàm khởi tạo

- 👉 Kỹ thuật tạo đối tượng từ một hàm;
- 👉 Đây là cách các đối tượng tích hợp như Arrays, Maps hay Sets được thực sự triển khai.

2 ES6 class

- 👉 Cú pháp thay thế cho hàm khởi tạo;
- 👉 "Cú pháp đặc biệt": các ES6 class hoạt động **chính xác** như các hàm tạo; Các
- 👉 ES6 class **KHÔNG** hoạt động như class trong "OOP cổ điển" (bài giảng gần nhất).

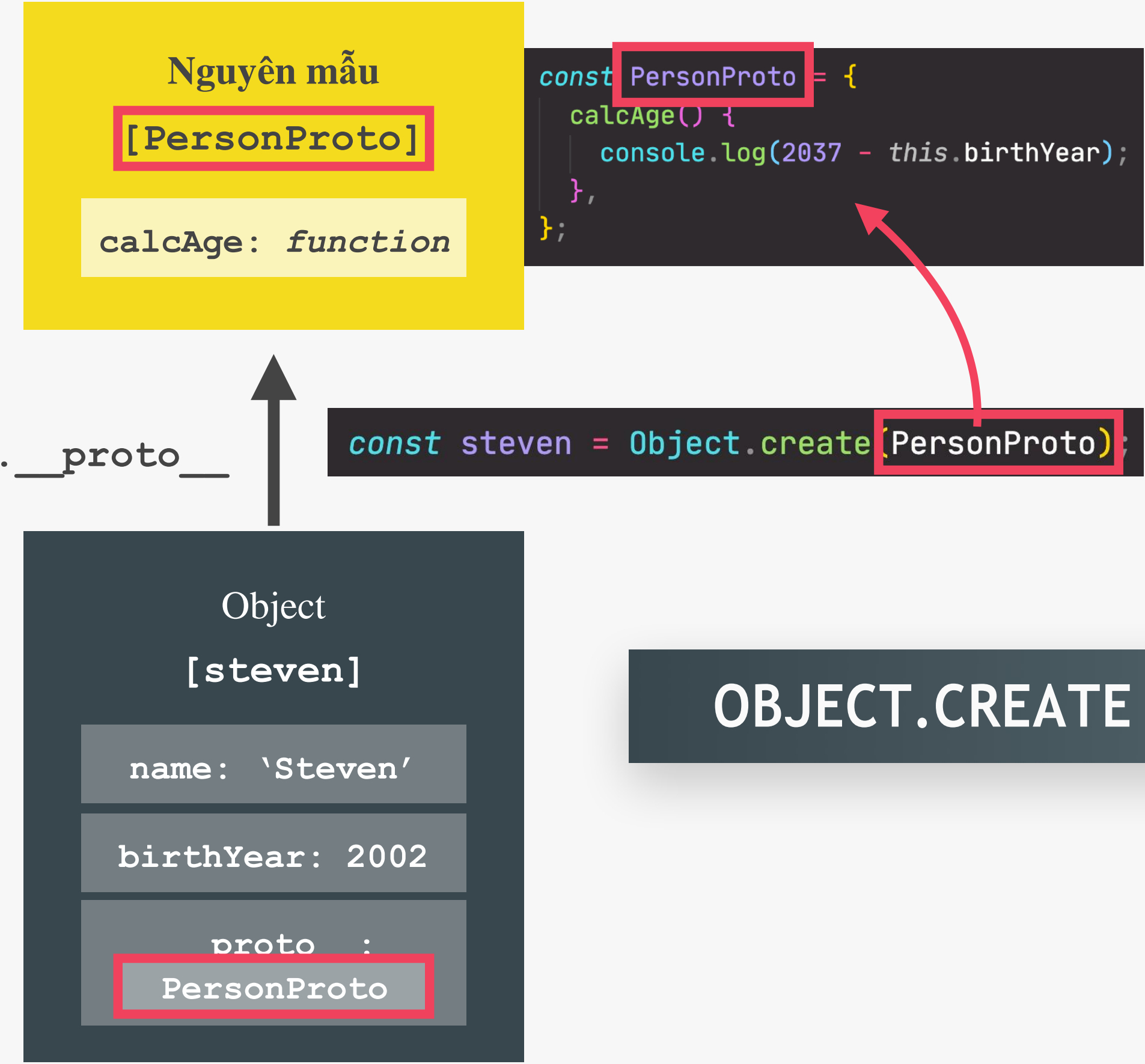
3 Object.create

- 👉 Là cách dễ nhất và đơn giản nhất để liên kết một đối tượng với một đối tượng nguyên mẫu.

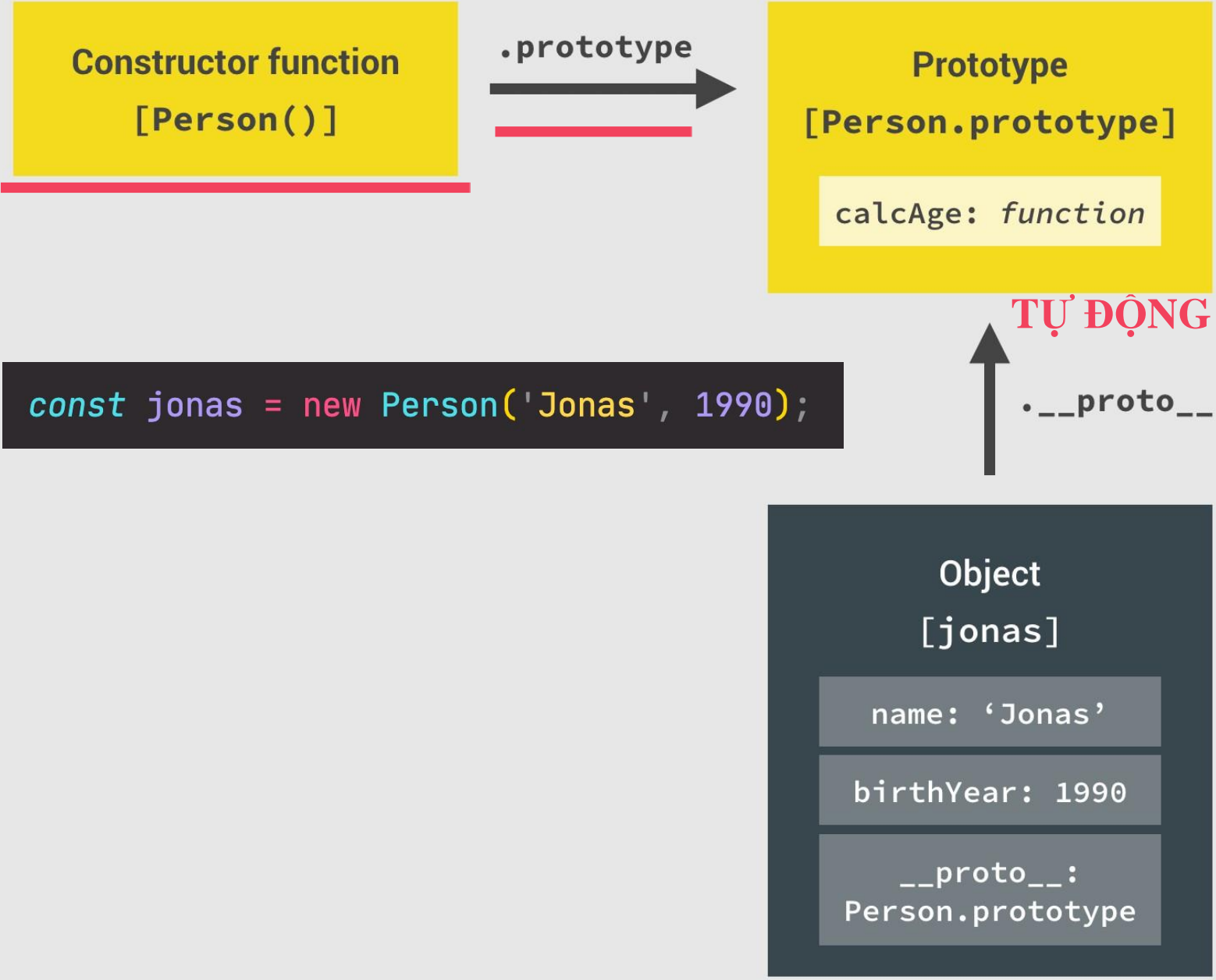
👉 **4 trụ cột của OOP vẫn còn giá trị!**

- 👉 Trừu Tượng
- 👉 Đóng gói
- 👉 Kế thừa
- 👉 Đa hình

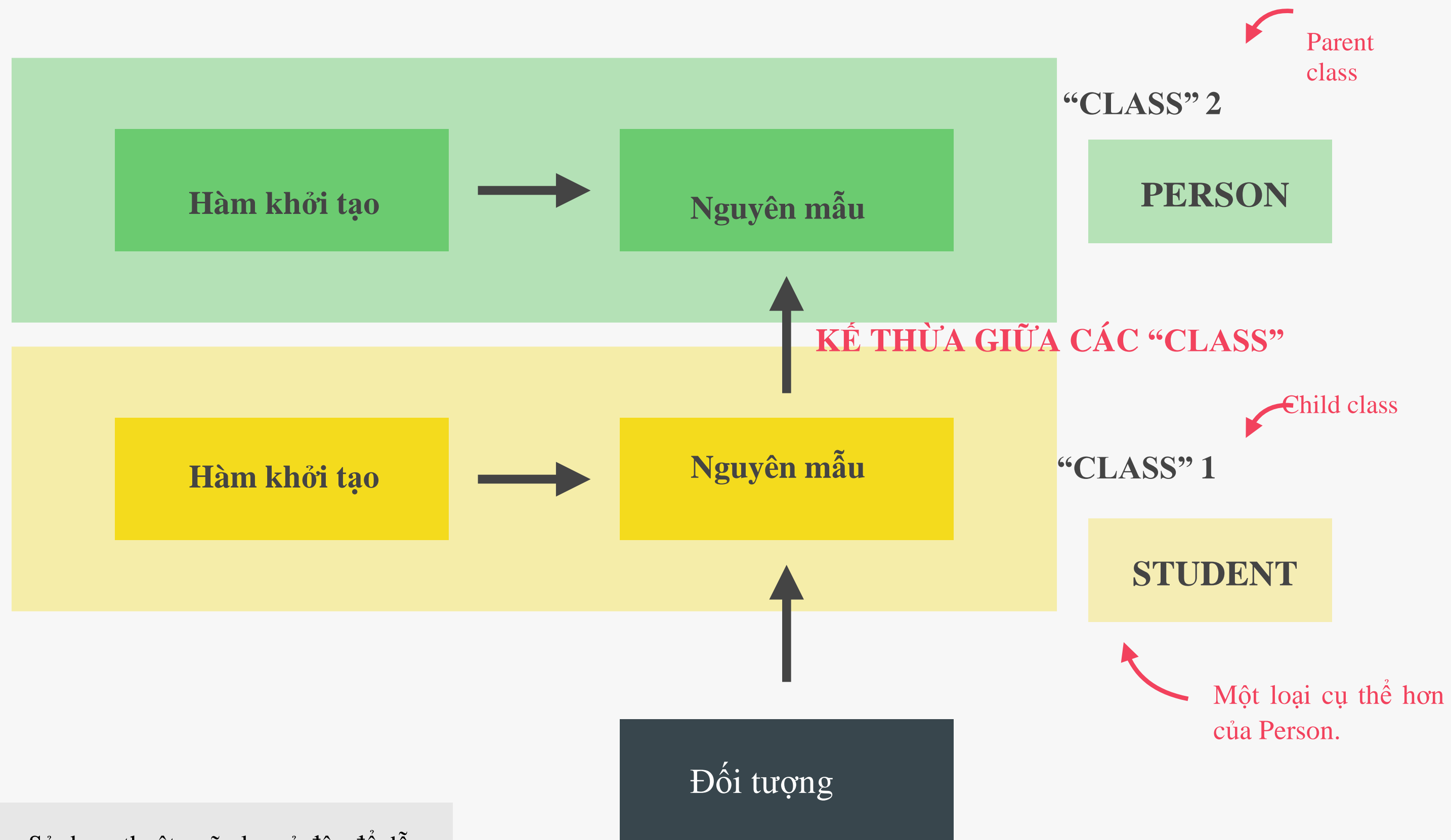
CÁCH OBJECT.CREATE HOẠT ĐỘNG



HÀM KHỞI TẠO



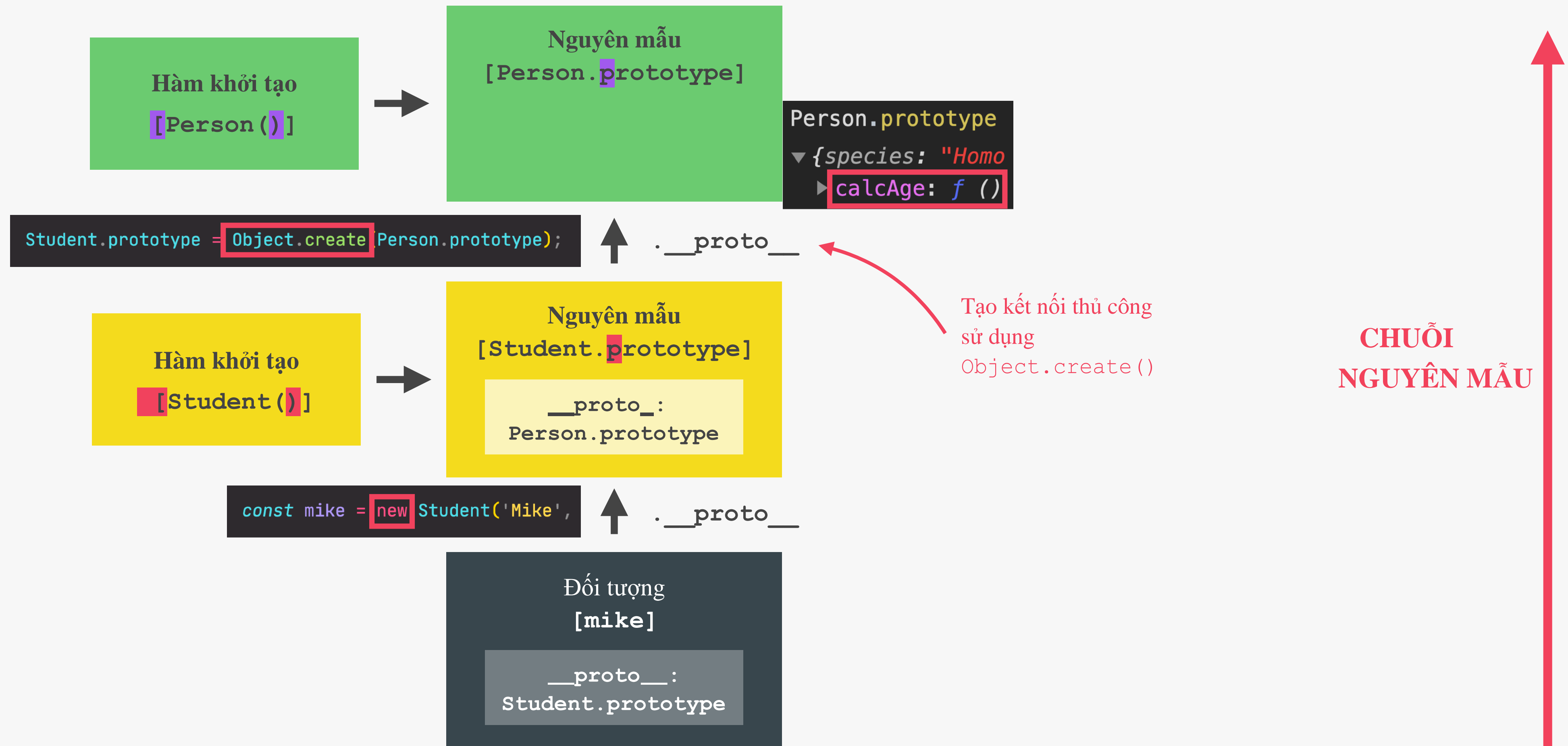
KẾ THỪA GIỮA CÁC “CLASS”



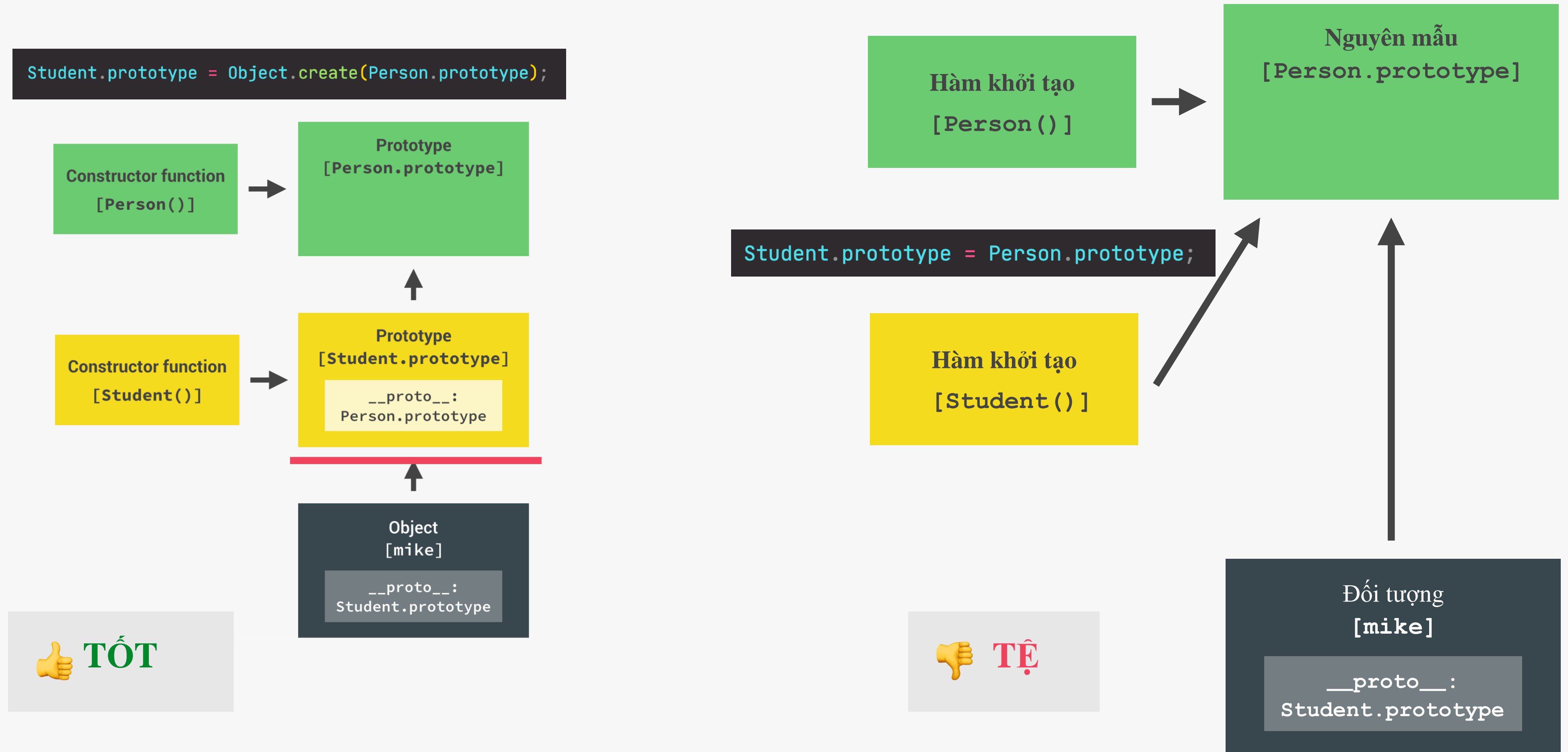
Sử dụng thuật ngữ class ở đây để dễ hiểu hơn.

- 1 Hàm khởi tạo
- 2 Class ES6
- 3 `Object.create()`

KẾ THỪA GIỮA CÁC "CLASS"



KẾ THỪA GIỮA CÁC "CLASS"



KẾ THỪA GIỮA CÁC "CLASS"

