**LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG TRONG SOLIDITY\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Biên soạn:** Nhóm Nghiên Cứu Blockchain Khoa HTTT

--------------------------------------------------------------------------------------------------------------

1. **Định nghĩa Smart Contract trong Solidity.**

* Một Contract trong Solidity được định nghĩa như sau:

|  |
| --- |
| pragma solidity <version>;  contract <ContractName> {  // 1. Định các state variables  // 1.1 Định nghĩa private state variables  // 1.2 Định nghĩa public state variables  // 2. Constructor, Getter, Setter  // 3. Định nghĩa các function  // 3.1 Private Function  // 3.2 Public Function  } |

* Tạo mới một đối tượng cơ bản

<ContractName> <nameVar> = new <ContractName>();

Ví dụ: SinhVien sv = new SinhVien();

1. **Tính kế thừa trong Solidity.**

* Cũng giống như những ngôn ngữ lập trình khác Solidity cũng có tính kế thừa (Inheritance).
* Contract con CÓ THỂ truy cập được tất hàm và biến không PRIVATE của Contract cha.
* Function có thể được override Function của Contract cha nhưng về mặt khai báo hàm phải giống với Contract cha. Nội dung thực thi bên trong thì có thể khác.
* Có thể dùng từ khoá **super** để gọi hàm của Contract cha.
* Khai báo kế thừa: contract <ContractChild> is <ContractParent> {}
* Xem ví dụ tham khảo: 08\_InheritanceContract.sol

1. **Tài liệu tham khảo**

[1] https://www.tutorialspoint.com/solidity/solidity\_contracts.htm, [Online] [Thời gian truy cập: 23/06/2022]

[2] <https://www.tutorialspoint.com/solidity/solidity_inheritance.htm>, [Online] [Thời gian truy cập: 23/06/2022]

[3] <https://www.tutorialspoint.com/solidity/solidity_constructors.htm>, [Online] [Thời gian truy cập: 23/06/2022]