[Git-Version Control](#Git)

[SSH](#SSH)

[HTTP – HTTPs and APIs](#HTTP)

[Basic Terminal Usage](#Terminal)

[Learn to Research](#learntoResearch)

[Data Structures & Algorithmx](#DataAltho)

[Character Encodings](#Character)

[Design Patterns](#designPatterns)

[GitHub](#GitHub)

|  |  |
| --- | --- |
| Đề nghị | Tùy khả năng |

# Git-Version Control

# SSH

SSH (Secure Shell) là một giao thức điều khiển từ xa cho phép người dùng kiểm soát và chỉnh sửa server từ xa qua Internet. Dịch vụ được tạo ra nhằm thay thế cho trình Telnet vốn không có mã hóa và sử dụng kỹ thuật cryptographic (mã hóa) để đảm bảo tất cả giao tiếp gửi tới và gửi từ server từ xa diễn ra trong tình trạng mã hóa. Nó cung cấp thuật toán để chứng thực người dùng từ xa, chuyển input từ client tới host, và relay kết quả trả về tới khách hàng.

Windows users có thể sử dụng những SSH clients như là Putty. Bạn có thể thực thi lệnh shell cũng như việc bạn đang thực sự vận hành máy tính vật lý.

Ref

[1]. <https://www.hostinger.vn/huong-dan/ssh-la-gi-va-cach-su-dung-ssh-cho-nguoi-moi-bat-dau/>

# HTTP – HTTPs and APIs

# Basic Terminal Usage

# Learn to Research

# Data Structures & Algorithmx

# Character Encodings

# Design Patterns

Các loại design patterns. Về cơ bản thì design pattern sẽ được chia làm 3 dạng chính và mỗi dạng chính và có tổng cộng 32 mẫu design:

## Creational Pattern ( nhóm khởi tạo):

Nhóm này sẽ giúp bạn rất nhiều trong việc khởi tạo đối tượng, mà bạn khó có thể nhận ra (nó sẽ không dùng từ khóa new như bình thường). Nhóm này gồm 9 mẫu design là:

* Abstract Factory.
* Builder.
* Factory Method.
* Multiton.
* Pool.
* Prototype.
* Simple Factory.
* Singleton.
* Static Factory.

## Structural (nhóm cấu trúc):

Nhóm này sẽ giúp chúng ta thiết lập, định nghĩa quan hệ giữa các đối tượng. Nhóm này gồm có 11 mẫu design là:

* Adapter/ Wrapper.
* Bridge.
* Composite.
* Data Mapper.
* Decorator.
* Dependency Injection.
* Facade.
* Fluent Interface.
* Flyweight.
* Registry.
* Proxy

## Behavioral patterns (nhóm ứng xử):

Nhóm này sẽ tập trung thực hiện các hành vi của đối tượng. Gồm 12 mẫu design là:

* Chain Of Responsibilities.
* Command.
* Iterator.
* Mediator.
* Memento.
* Null Object.
* Observer.
* Specification.
* State.
* Strategy.
* Template Method.
* Visitor. Ngoài ra thì trong thời gian gần đây đã xuất hiện thêm 4 mẫu design nữa đó là:
* Delegation.
* Service Locator.
* Repository.
* Entity-Attribute-Value (EAV).

Ref

[1]. <https://en.wikipedia.org/wiki/Software_design_pattern>

[2]. <https://www.oreilly.com/ideas/5-reasons-to-finally-learn-design-patterns>

[3]. <https://toidicodedao.com/2016/03/01/nhap-mon-design-pattern-phong-cach-kiem-hiep/>

[4]. <https://toidicode.com/designe-patterns-la-gi-130.html>

# GitHub