|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SKILL** | **ITEM** | **DESCRIPTION** | **USE** | | **LIMIT** |
| Programing Principles | SOLID  KISS  YAGNI  DRY | Kỹ năng viết code, roles trong lập trình (em từng có nghe “viết code sạch”) | Tránh trường hợp code dài, loằn ngoằn, rối. Áp dụng các quy tắc vào code giúp dễ phân biệt, dễ đọc code | | Chưa biết |
| General Development Skill | Git flow | Tool hỗ trợ để control và quản lý souce code cũng như những lịch sử sửa, viết của source code | Một chức năng nâng cao của git, dùng để thao tác, quản lý theo framework phân nhánh | | public |
| Azure Repos | Azure cũng tương tự như git tuy nhiên điểm khác biệt là azure có tính private hơn git, thường được sử dụng cho các dự án kín | | private |
| dotnet CLI | Chưa biết | | Chưa biết |
| C# | Types | Kiểu dữ liệu là tập hợp các giá trị mà biến thuộc kiểu dó có thể nhận được. Kiểu dữ liệu bao gồm: Kiểu tham chiếu và Kiểu giá trị. | Kiểu giá trị:   * Kiểu số nguyên (byte[0,255]; sbyte[-128,127]; short; ushort; int; uint; long(8byte); ulong) * Kiểu ký tự (char) * Kiểu logic (bool) true or false * Kieur số thực (float, double, decimal) * Kiểu cấu trúc Struct | |  |
| Kiểu tham chiếu:   * Object: cung cấp một số method giúp overload để sử dụng như ToString() chuyển đổi kiểu dữ liệu về kiểu chuỗi; GetHashCode() trả về mã băm; Equals() so sánh hai đối tượng True khi bằng nhau và ngược lại là False; GetType() trả về kiểu dữ liệu của đối tượng * Kiểu String * Class * Delegate * interface | |
| Nullable reference types | Cho phép gán null cho biến kiểu giá trị | Int a = null // chương trình bị lỗi biên dịch  Sử dụng nullable như sau: Nullable<int> a = null or int? a = null | |  |
| Statements, Expression, Operators | **Statement**: Câu lệnh thực thi chương trình và kết thúc bằng dấu chấm phẩy. Các câu lệnh biên dịch từ trên xuống.  **Expression**: Biểu thức là sự kết hợp của giá trị, hằng số, biến, toán tử.  **Operators**: Toán tử dùng để thực hiện các phép toán hạng. Có 6 loại toán tử:   * số học: +, -, \*, /, %, ++, -- * quan hệ: ==, !=, >, <, >=, <= * logic: &&, ||, ! * so sánh bit: băm ra nhị phân và so sánh * gán: =, +=, -=, \*=, /=, %=,… * hỗn hợp | int a = 1;  int b = 2;  int sum = a + b;  Console.WriteLine("{0} + {1} = {2}", a, b, sum);  Console.ReadLine(); | |  |
| Classes, Structs | **Class:** là tập hợp các đối tượng có chung đặc điểm, phương thức  **Struct:** là tập hợp dữ liệu thuộc nhiều kiểu khác nhau | **Classes:**   * Có tính kế thừa * Kiểu dữ liệu tham chiếu * Có thể null | **Structs:**   * Không có tính kế thừa * Kiểu dữ liệu giá trị * Không thể null |  |
| Interfaces | Khai báo phương thức và không có nội dung trong phương thức đó, không định nghĩa đối tượng như class, | Sử dụng trong đa kế thừa | | Chậm hơn so với Abstract Class, không có phạm vi truy cập vì default là public |
| Delegate & Event | Event là các hành động của người dùng như click chuột, nhấm phím,…  Delegate là trỏ đến các hàm nhằm thực hiện event mà người dùng mới hành động. | Delegate thường được sử dụng phối hợp với event trong winform | |  |
| Method | Là nhiều câu lệnh thực thi một nhiệm vụ | Sử dụng để tạo các phương thức như create, update, delete, sort,… | |  |
| Properties | (Đặc tính) là một thành viên của class | Trong trường hợp bảo mật thông tin fields riêng của class, không thể truy cập các fields này từ bên ngoài phạm vi class, khi sử dụng properties thì ta có thể truy cập được các fields riêng tư này. | |  |
| Generics | Generic sử dụng khi muốn tạo một danh sách, được định ghĩa bằng cách sử dụng dấu ngoặc nhọn <> | Dùng Generic với classes, method, interface | |  |
| Namespaces | Được tạo ra để tổ chức quản lý code | Namespaces có thể chứa các đơn vị khác như:   * Namespaces khác * Struct * Class * Interface * delegate | |  |
| Collection | collection là một thư viện, giúp việc code trở nên ngắn gọn và dễ dàng hơn, trong collection có các phwng thức để them, bớt, lấy tổng phần tử | Collection kiểu Generic sử dụng System.Collections.Generic   * List<T> là phiên bản generic của ArrayList * Dictionary<Tkey, Tvalue> là thay thế cho Hashtable   SortedDictionary<Tkey, Tvalue> là thay thế cho SortedList  Một số phương thức để tìm kiếm trong List nư Find, FindLast, FindAll,… | |  |
| Linq | Ngôn ngữ truy vấn tích hợp, câu lệnh giống như SQL vào bên trong c# để truy cập nguồn CSDL | Sử dụng 2 thư viện để dùng linq:   * using System.Collections.Generic: * using System.Linq; | |  |
| Asynchronous programming | Lập trình bất đồng bộ là lập trình mà chương trình hoạt động không theo tuần tự từ trên xuống, đoạn code ở dưới có thể chạy tước đoạn code ở trên |  | |  |
| Attribute |  |  | |  |
| Reflection |  |  | |  |
| ASP.NET core | MVC | Model – View – Controller  Tạo lập giao diện người dùng trên máy tính, được sử dụng để lập trình Web.  Model có chức năng là lưu trữ dữ liệu của ứng dụng  View là phần giao diện cho người dùng, dùng để hiể thị các đối tượng trong môt ứng dụng  Controller là phần xử lý yêu cầu của người dùng   * request của client gửi đến server thì controller nhận để xử lý rồi vào model, model trả data lại cho controller, cuối cùng thì controller gửi dữ liệu lại view và hiển thị cho người dùng trên trình duyệt web | -Hỗ trợ quá trình phát triển nhanh chóng giúp tiết kiệm thời gian (sử dụng mô hình lập trình bất đồng bộ) | |  |
| Middleaware |  |  | |  |
| Filter & Attributes |  |  | |  |
| Application settings & configuaration |  |  | |  |
| Authentication & authorization | Authentication Là quá trình xác định vấc nhận danh tính của user khi vào một ứng dụng  Authorization là phân quyền cho user để truy cập và xử lý tính năng của website |  | |  |
| Dependency injection |  |  | |  |
| Environments | Công cụ cần thiết để lập trình   * .Net framework (ứng dụng windows, ứng dụng web, djch vụ web), có nhiều ngôn ngữ như C#, C++, VB * IDE (Visual Studio) |  | |  |
| Routing |  |  | |  |
| HTTP request | http là giao thức thiết kế theo kiểu client và server, client đưa ra request và server sẽ respone lại cho client  Có hai phương thức là GET và POST |  | |  |
| Error Handling | Xử lý ngoại lệ |  | |  |
| Logging |  |  | |  |
| Entity Framework | The basic (model, query, schema, save data, …) | Là một ánh xạ CSDL, hỗ trợ truy vấn Linq, hoạt động ới SQL server | Tạo model, truy xuất dữ liệu, Dbset, SaveChanges | |  |
| Code First & Migrations | Code First: thiết ké CSDL trước rồi tạo các lớp khớp với thiết kế CSDL bằng C#  Migration: là tương tác với CSDL, code cập nhật cậu trúc CSDL sẽ được cập nhật lên CSDL giúp data không bi mất | Lệnh add-migration, remove-migration | |  |
| Change Tracker |  |  | |  |
| Lazy, Eager, Explicit Loading |  | Sử dụng FirstOrDefault | |  |
| TPT, TPH |  |  | |  |
| SQL | Database, table design | Tạo cơ sở dữ liệu và tạo bảng | Sử dụng SQL server với câu lệnh query hoặc tạo bằng giao diện UI | |  |
| SQL syntax | Câu lệnh khởi tạo database, table, procedure, view,… | Viết lệnh trong query | |  |
| Query data | Viết lệnh query để xem dữ liệu, insert data, update data, delete data,… | Sử dụng các lệnh INSERT, DELETE, SELECT FROM WHERE,… | |  |
| Index | Bảng tra cứu dữ liệu như truy vấn SELECT chứa WHERE hoặc ORDER nhanh hơn | CREATE INDEX tên\_index ON tên\_table (tên\_colum); | | Không sử dụng cho các dữ liệu nhỏ, không sử dụng cho bảng update hoặc insert thường xuyên |
| VueJs |  |  |  | |  |
| Unit Test |  |  |  | |  |
| Agile Scrum |  |  |  | |  |