**Bài tập về nhà (Deadline 13h00 ngày 30/10/2020)**

**Bài 1** (Lý thuyết): Tìm hiểu về WebComponent: khái niệm? Công dụng?

**Khái niệm:** Web components (thành phần web) là một tập các quy chuẩn công nghệ dùng cho việc xây dựng các thành phần trang web được đóng gói (tách biệt với phần code còn lại của ứng dụng) và có thể tái sử dụng. Đó là là các thành phần UI của 1 trang web được đóng gói lại để sử dụng cho nhiều project khác nhau, bất kể khung framework gì.

Là tập hợp các Web APIs cho phép chúng ta tạo ra một thẻ HTML riêng, mang các đặc tính riêng, đóng gói, có thể tái sử dụng. Web component được xây dựng trên chuẩn web hiện tại, vì thế đang (và sẽ) có thể hoạt động trên tất cả các trình duyệt, có thể tương thích với tất cả các library và framework Javascript có thể làm việc với HTML, phạm vi sử dụng là trình duyệt, chứ ko gói gọn ở phạm vi framework.

Để đạt được khả năng đóng gói kể trên thì Web components sử dụng 3 công nghệ là Custom Elements, Shadow DOM, và HTML Templates.

**Công dụng:** giúp các framework có tiếng nói chung trong việc tái sử dụng component xuyên suốt nền tảng và sử dụng component bên thứ 3.

**Bài 2 (OOP):** Hãy viết chương trình quản lý tài liệu cá nhân bằng cách cài đặt các class sau

- Document:

+ id

+ name

+ author

+ price

+ publishDate

+ get info(): return thông tin về tài liệu đó

+ set info(data): sửa thông tin của document với data (data là 1 object)

- EBook kế thừa Document

+ discount: giảm giá, tính theo phần trăm

+ get info(): return thông tin về sách điện tử đó, price tính theo discount

- HardCopy kế thừa Document

+ source: tên quyển sách được sao chép

+ get info(): return thông tin về bản hardcopy, có cả source

- DocumentCase:

+ id

+ name

+ documents: là 1 mảng chứa các quyển sách

+ owner: tên người sở hữu

+ dateModified

+ addDocument(document): thêm 1 document.

Tham số document truyền vào là thể hiện của Document.

Hỏi: liệu có truyền thể hiện của EBook hoặc HardCopy được không? Vì sao?

+ findDocuments(name): tìm tất cả document theo tên

+ deleteDocument(id): xóa 1 document theo id

+ showDocuments(): in ra console các documents hiện có

+ updateDocument(id, data): sửa thông tin của document với id và data

Lấy ví dụ với 1 DocumentCase, 2 EBook, 2 HardCopy

**Bài 3 (Nâng cao OOP + Thuật toán, không bắt buộc phải làm): Tìm** hiểu về cấu trúc dữ liệu LinkedList và cài đặt?

- Cấu trúc dữ liệu LinkedList: là cấu trúc dữ liệu dùng để lưu 1 tập dữ liệu dưới kiểu danh sách (gần giống như mảng)

- Nó là 1 dãy dữ liệu lên kết với nhau thông qua các liên kết

- Tham khảo: [https://www.geeksforgeeks.org/data-structures/linked-list/](https://l.facebook.com/l.php?u=https%3A%2F%2Fwww.geeksforgeeks.org%2Fdata-structures%2Flinked-list%2F%3Ffbclid%3DIwAR2KiA_xFdV8u30MTHIxdJdMa4jKul89dkKLj95rhqSag_fC40vPEcdMMQY&h=AT2dr1cyKPctD37jMPufnS58dSZ29WAydqipSCKehe3auR0vfCYVZ-wUBwYD2t8LCRkX5KGeSV-h4--PoqR1U5mqZ-cnTnu9X0APISJ6naGl1tf0-D4xxKI1hh5aY6_f6A&__tn__=-UK-R&c%5b0%5d=AT1-wvn0QlnIlS_psbgw3MU6Ufu1d4CVAvYlBvDo4FaS-wc7baJnqZYH7gE2ZXXBYw0-F1Y8l_Oi34J7ncgAwEURPBzqwKVAzMu_zRJM6J0kHlFsAd4fMzKo2fNgvzH6fImjdHVvGebeVaQSLCqtBAbKqb3Bhk8UFRYyZHAAauomCH5qdwAIl1YE_ViiqoGJ3CJ18uqmHd_wVuill35UdM0dltPPrXCvugz7aQKUgRg)

- Cách cài đặt:

- Node:

+ data: lưu dữ liệu của node

+ next: trỏ đến 1 node khác

- LinkedList

+ firstNode: chứa node đầu tiên của LinkedList

+ tranversal(): duyệt tất cả các node

+ pushNode(node): thêm 1 node vào cuối danh sách

+ unshiftNode(node): thêm 1 node vào đầu danh sách

+ ...

Lấy 1 ví dụ

LinkedList là một chuỗi các cấu trúc dữ liệu, với mỗi node bao gồm hai phần thông tin: dữ liệu của node và tham chiếu đến node kế tiếp trong chuỗi.