**Các công nghệ nâng cao trong phát triển Phần mềm**

1. ***Lập trình hướng thành phần (Component Oriented Programming)***

Phương pháp lập trình Hướng thành phần (Component-base Development - CBD) ra đời, khắc phục các vấn đề mà OOP đơn thuần không thể làm được, đồng thời đem lại nhiều ưu điểm trong lập trình, bảo trì và nâng cấp sản phẩm

Khi bắt đầu lập trình, phương pháp lập trình Hướng đối tượng (Object Oriented Programming – OOP) là phương pháp rất quen thuộc. Các đối tượng sẽ kế thừa từ 1 lớp gọi là Object, các đối tượng có các đặc điểm tương tự nhau sẽ được gom lại và kế thừa từ 1 đối tượng cơ sở. Phương pháp này khá hiệu quả đối với những sản phẩm cỡ nhỏ, có ít loại đối tượng. Tuy nhiên, khi cần nâng cấp hoặc mở rộng thêm các tính năng khác, lập trình viên sẽ phải tuỳ biến chương trình rất nhiều để phù hợp với định hướng mới..

Phương pháp lập trình Hướng thành phần (Component-base Development - CBD) ra đời, khắc phục các vấn đề mà OOP đơn thuần không thể làm được, đồng thời đem lại nhiều ưu điểm trong lập trình, bảo trì và nâng cấp

Phương pháp CBD sẽ xem tất cả các đối tượng là giống nhau, không phân biệt Tank hay Bullet, Person hay Animal, ... Các đối tượng ban đầu sẽ ở trạng thái không có gì NULL. Khi đối tượng cần có 1 chức năng, thì lập trình và thêm chức năng (component) đó vào cho đối tượng cụ thể. Nhờ các chức năng đặc trưng đó mà có thể phân biệt được các đối tượng với nhau.

Phương pháp CBD đã khắc phục được các nhược điểm của OOP, đồng thời mang lại nhiều ưu điểm vượt trội so với OOP

Tuy nhiên, đổi lại việc thiết kế 1 chương trình theo CBD là phức tạp hơn, đòi hỏi lập trình viên tư duy theo hướng mở hơn.

1. ***Thiết kế mẫu (Design Pattern)***

Design Pattern là một kỹ thuật trong lập trình hướng đối tượng, nó khá quan trọng và mọi lập trình viên muốn giỏi đều phải biết. Được sử dụng thường xuyên trong các ngôn ngữ OOP. Nó sẽ cung cấp cho bạn các "mẫu thiết kế", giải pháp để giải quyết các vấn đề chung, thường gặp trong lập trình. Các vấn đề mà bạn gặp phải có thể bạn sẽ tự nghĩ ra cách giải quyết nhưng có thể nó chưa phải là tối ưu. Design Pattern giúp bạn giải quyết vấn đề một cách tối ưu nhất, cung cấp cho bạn các giải pháp trong lập trình OOP.

Design Patterns không phải là ngôn ngữ cụ thể nào cả. Nó có thể thực hiện được ở phần lớn các ngôn ngữ lập trình, chẳng hạn như Java, C#, thậm chí là Javascript hay bất kỳ ngôn ngữ lập trình nào khác.Design pattern chủ yếu được dựa theo những quy tắc sau đây về thiết kế hướng đối tượng.

+ Lập trình cho interface chứ không phải để implement interface đó.

+ Ưu tiên object composition hơn là thừa kế.

Pattern được hiểu theo nghĩa tái sử dụng ý tưởng hơn là mã lệnh.

Pattern đa tương thích

Pattern được phân loại ra làm 3 nhóm chính sau đây:

* Nhóm cấu thành (Creational Pattern)
* Nhóm cấu trúc tĩnh (Structural Pattern)
* Nhóm tương tác động (Behavioral Pattern)

1. ***Lập trình đa tầng***

3 tầng(3-tiers) là một kiến trúc kiểu client/server mà trong đó giao diện người dùng (UI-user interface), các quy tắc xử lý(BR-business rule hay BL-business logic), và việc lưu trữ dữ liệu được phát triển như những module độc lập, và hầu hết là được duy trì trên các nền tảng độc lập, và mô hình 3 tầng (3-tiers) được coi là một kiến trúc phần mềm và là một mẫu thiết kế

Như vậy, ta có thể mô hình này phân tách ứng dụng ra làm 3 module riêng biệt, bao gồm:

– Tầng Presentation: được dùng để giao tiếp với người dùng, nhiệm vụ chính là hiển thị dữ liệu và nhận dữ liệu từ người dùng.

– Tầng Business Logic: nhiệm vụ chính là cung cấp các chức năng của phần mềm.

– Tầng Data: lưu trữ dữ liệu, cho phép lớp Business Logic có thể tìm kiếm, trích xuất, cập nhật… dữ liệu.

1. ***Lập trình mạng***

Là một trong những nhiệm vụ căn bản để phát triển các ứng dụng doanh nghiệp. Một chương trình mạng được viết ra để các chương trình trên các máy tính khác nhau có thể truyền tin với nhau một cách hiệu quả và an toàn cho dù chúng được cài đặt trên mạng LAN, WAN hay mạng toàn cầu Internet, đây là điều căn bản đối với sự thành công của nhiều hệ thống.

Ngôn ngữ lập trình mạng: Java, .NET, C/C++, Delphi, Javascript.

Ứng dụng: Lập trình Socket với .NET Compact Framework

1. ***Kiến trúc hướng dịch vụ (Webservice)***

Kiến trúc hướng dịch vụ là một mẫu thiết kế. Nó được thiết kế để cung cấp dịch vụ cho các ứng dụng khác thông qua giao thức. Nó là một khái niệm duy nhất và không gắn liền với bất kỳ ngôn ngữ lập trình hoặc nền tảng nào.

Hầu hết web service là một công nghệ của SOA.

Dịch vụ (Service): Một dịch vụ được xác định rõ, chức năng khép kín đại diện cho đơn vị chức năng. Một dịch vụ có thể trao đổi thông tin từ một dịch vụ khác. Nó không phụ thuộc vào trạng thái của một dịch vụ khác.

Các kết nối dịch vụ (Service connection): Hình dưới đây minh họa kiến trúc hướng dịch vụ. Người dùng dịch vụ gửi yêu cầu tới nhà cung cấp dịch vụ và nhà cung cấp dịch vụ gửi phản hồi tới người dùng dịch vụ. Kết nối dịch vụ là dễ hiểu đối với cả người dùng và nhà cung cấp dịch vụ.

Ứng dụng: trong việc xây dựng kiến trúc bảo mật hướng dịch vụ, tích hợp hệ thống.

1. ***Công nghệ Hibernate***

Hibernate là một trong những ORM Framework. Hibernate framework là một framework cho persistence layer. Như vậy, nhờ có Hibernate framework mà giờ đây khi bạn phát triển ứng dụng bạn chỉ còn chú tâm vào những layer khác mà không phải bận tâm nhiều về persistence layer nữa.

Hibernate giúp lập trình viên viết ứng dụng Java có thể map các object (POJO) với hệ quản trị cơ sở dữ liệu quan hệ (database), và hỗ trợ thực hiện các khái niệm lập trình hướng đối tượng với cơ dữ liệu quan hệ.

Hibernate giúp lưu trữ và truy vấn dữ liệu quan hệ mạnh mẽ và nhanh. Hibernate cho phép bạn truy vẫn dữ liệu thông qua Java Persistence API (JPA) hoặc bằng ngôn ngữ SQL mở rộng của Hibernate (HQL) hoặc bằng SQL thuần (Native SQL).

1. ***Công nghệ Ajax***

AJAX (Asynchronous Javascript and XML) là phương thức trao đổi dữ liệu với máy chủ và cập nhật một hay nhiều phần của trang web, hoàn toàn không reload lại toàn bộ trang.

Ajax được viết bằng Javascript chạy trên client, tức là mỗi browser sẽ chạy độc lập hoàn toàn không ảnh hưởng lẫn nhau. Về mặt kỹ thuật, nó đề cập đến việc sử dụng các đối tượng XmlHttpRequest để tương tác với một máy chủ web thông qua Javascript.

AJAX được sử dụng để thực hiện một callback. Được dùng để thực hiện việc truy xuất dữ liệu hoặc lưu trữ dữ liệu mà không cần phải reload lại toàn bộ trang web. Với những server nhỏ thì việc này cũng tiết kiệm được băng thông cho chúng ta hơn.

Cần gì thì chỉ gửi dữ liệu phần đó, load lại 1 phần nhỏ để cập nhật thông tin chứ không load cả trang. Bằng cách này thì có thể giảm thiểu được tốc độ tải trang giúp người dùng có trải nghiệm tốt hơn.

Trang web bạn tạo ra cũng sẽ đa dạng và động hơn.

1. ***Lập trình API***

API là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của Application Programming Interface – giao diện lập trình ứng dụng. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng.

Web API: là hệ thống API được sử dụng trong các hệ thống website. Hầu hết các website đều ứng dụng đến Web API cho phép bạn kết nối, lấy dữ liệu hoặc cập nhật cơ sở dữ liệu. Ví dụ: Bạn thiết kế chức nằng login thông Google, Facebook, Twitter, Github… Điều này có nghĩa là bạn đang gọi đến API của. Hoặc như các ứng dụng di động đều lấy dữ liệu thông qua API.

API trên hệ điều hành: Windows hay Linux có rất nhiều API, họ cung cấp các tài liệu API là đặc tả các hàm, phương thức cũng như các giao thức kết nối. Nó giúp lập trình viên có thể tạo ra các phần mềm ứng dụng có thể tương tác trực tiếp với hệ điều hành.

API của thư viện phần mềm hay framework: API mô tả và quy định các hành động mong muốn mà các thư viện cung cấp. Một API có thể có nhiều cách triển khai khác nhau và nó cũng giúp cho một chương trình viết bằng ngôn ngữ này có thể sử dụng thư viện được viết bằng ngôn ngữ khác. Ví dụ bạn có thể dùng Php để yêu cầu một thư viện tạo file PDF được viết bằng C++.

API hiện nay đều tuân thủ theo tiêu chuẩn REST và HTTP, tạo sự thân thiện dễ sử dụng với nhà phát triển. Giúp người dùng dễ dàng truy cập, dễ hiểu hơn. Web API hiện đại dùng cho các đối tượng cụ thể, chẳng hạn như mobile developer với document, version khác nhau.

API key: Đây là loại code (string) được truyền tải bởi các chương trình máy tính gọi là API để xác định chương trình, nhà phát triển hoặc người dùng nó tới trang web. Các API key được sử dụng với mục đích nhằm giới hạn, kiểm soát sử dụng API. Chẳng hạn như ngăn chặn sự việc lạm dụng API.

1. ***Mã nguồn mở(Open Source)***

Open Source là phần mềm có bộ source code mà người dùng có thể dễ dàng tải về, tiến hành sửa đổi, hay thực hiện nâng cấp thêm các tính năng cần thiết nhằm đáp ứng cho nhu cầu sử dụng thực tế của chính mình. Thông thường đây đều là những mã nguồn được phát hành hoàn toàn miễn phí, thuộc quyền sở hữu của những đơn vị, những doanh nghiệp lớn trong lĩnh vực công nghệ. Đôi khi, mã nguồn mở được các lập trình viên phát triển, cung cấp nó với những khác biệt, độc đáo nhất định khi so sánh với phiên bản gốc.

Mã nguồn là phần mềm mà ở đó người dùng máy tính hầu hết đều không thể nhìn thấy. Tuy nhiên, đối với các lập trình viên thì việc truy cập vào mã nguồn, thực hiện những cải thiện bằn cách thêm, bớt, hay chỉnh sửa tính năng, các phần không hoạt động,… theo nhu cầu được thực hiện dễ dàng, chuẩn xác.

Một số ngôn ngữ lập trình mở như: PHP, Java,… Bên cạnh đó cũng có những ngôn ngữ “mở một phần” như .NET. Ngôn ngữ này thực ra là nguồn mở nhưng người sử dụng phải trả tiền một cách gián tiếp thông qua cách mua hệ điều hành Windows, chứ không dùng được trên hệ điều hành MAC OS.

Bên cạnh một số ngôn ngữ lập trình Engine game mở thì cũng có một số ngôn ngữ không thuộc phần mềm mã nguồn mở như: Unity. Nếu bạn tạo ra 1 game bằng mã nguồn này, bạn được dùng thoải mái nhưng không được bán chúng. Vì khi bạn bán chúng. ngay lập tức bản quyền sẽ bị store link report, khiếu nại khiến cho link bị kéo xuống và bị khai tử game đó. Muốn thương mại hóa bạn phải mua bản quyền sử dụng từ bên sở hữu ngôn ngữ lập trình.

Phần lớn mọi người ngoài lập trình viên đều cho rằng, mã nguồn mở và đóng liên quan đến bảo mật, chức năng của mã nguồn đó. Tuy nhiên, đó là một trong những hiểu lầm lớn nhất mà bạn từng có.

mã nguồn mở có tính bảo mật rất cao. Bằng chứng thực tế đó là hàng loạt các mã nguồn mở được dùng để điều hành phần lớn mạng Internet. Các công ty lớn như Google, Microsoft, IBM… là những đại diện lớn đang sử dụng mã nguồn mở để kinh doanh.

Do đó, sự an toàn bảo mật là ưu tiên hàng đầu đối với mã nguồn mở. Điều này đặc biệt quan trọng khi có nhiều người dùng cùng sử dụng, chia sẻ quyền truy cập máy chủ. Nếu không đảm bảo an toàn, các dữ liệu máy chủ sẽ bị đánh cắp nhanh chóng gây ra hậu quả cực kỳ nghiêm trọng.

Mã nguồn mở không liên quan đến cấu trúc hay kỹ thuật lập trình. Lấy ví dụ về kỹ thuật lập trình như sau:

Mã nguồn .NET hoạt động trên nguyên tắc mã hóa tất cả những file code của người dùng thành một tập lệnh DLL. Tập lệnh DLL này không cho phép bất kỳ phần mềm nào có thể dịch ngược lại được tập lệnh DLL, không ai có thể biết được bạn viết tệp lệnh đó như thế nào.

Tuy nhiên, việc có mã hóa được hay không không liên quan đến mã nguồn. Không phải mã nguồn đóng có thể mã hóa code để bảo mật còn mã nguồn mở sẽ không mã hóa được. Việc mã hóa tập lệnh để bảo mật liên quan đến cấu trúc và nguyên tắc hoạt động của chính phần mềm đó, không liên quan đến mã nguồn.

Mã nguồn mở không phải là một phần mềm được dùng một số chức năng miễn phí có sẵn. Chúng không giới hạn quyền truy cập, cho phép người dùng download thoải mái, dùng hết tất cả các chức năng mà không phải trả chi phí bản quyền.