**Work Shop 5**

1. **Viết báo cáo tìm hiểu UML** 
   1. **Giới Thiệu.**

UML (Unified Modeling Language) là ngôn ngữ dành cho việc đặc tả, hình dung, xây dựng và làm tài liệu của các hệ thống phần mềm. UML tạo cơ hội để viết thiết kế hệ thống, bao gồm những khái niệm như tiến trình nghiệp vụ và các chức năng của hệ thống. Cụ thể, nó hữu dụng cho những ngôn ngữ khai báo, giản đồ cơ sở dữ liệu, thành phần phần mềm có khả năng tái sử dụng. UML được phát triển bởi Rational Rose và một số nhóm cộng tác, nó nhanh chóng trở thành một trong những ngôn ngữ chuẩn để xây dựng hệ thống phần mềm hướng đối tượng (Object-Oriented). Đây là ngôn ngữ kế vị xứng đáng cho những ngôn ngữ mô hình hoá như Booch, OOSE/Jacobson, OMT và một số các phương thức khác.

* 1. **Mục Tiêu Của Uml.**

a. UML cung cấp cho người dùng một ngôn ngữ mô hình hoá trực quan sẵn sàng để dùng và có ý nghĩa; cho phép phát triển và trao đổi những mô hình mang nhiều ý nghĩa.

b. Cung cấp khả năng mở rộng và chuyên môn hoá để mở rộng những khái niệm cốt lõi.

c. Độc lập với ngôn ngữ lập trình chuyên biệt và các tiến trình phát triển.

d. Cung cấp nền tảng về sự hiểu biết ngôn ngữ mô hình hoá.

e. Khuyến khích và hỗ trợ sự phát triển của các công cụ hướng đối tượng.

f.Hỗ trợ những khái niệm phát triển cấp độ cao như collaboration, framework, pattern and component.

g. Tích hợp một cách tốt nhất với thực tiễn.

Một điểm quan trọng cần chú ý là UML là một “ngôn ngữ” dành cho việc chỉ định, đặc tả chứ không phải là các phương thức hay thủ tục. UML được sử dụng để định nghĩa hệ thống phần mềm, chi tiết về hệ thống, làm tài liệu và xây dựng hệ thống, nó là một ngôn ngữ để lên kế hoạch cho việc viết chi tiết. UML có thể được sử dụng theo nhiều cách để hỗ trợ cho các phương pháp luận trong việc phát triển phần mềm. Để tiếp cận với UML bạn phải tiếp xúc với nhiều khái niệm. Trong khuôn khổ bài viết này tôi không thể cung cấp cho bạn tất cả, tuy nhiên tôi sẽ cố gắng cung cấp những biểu đồ cơ bản của UML để bạn có thể hiểu được tổng quan về cách phân tích và thiết kế một hệ thống UML cung cấp 9 loại biểu đồ:

* Biểu Đồ Lớp (Class Diagram)
* Biểu Đồ Gói (Package Diagram)
* Biểu Đồ Chức Năng
* Biểu Đồ Tương Tác
* Biểu Đồ Tiến Trình
* Biểu Đồ Trạng Thái (Statechart Diagram)
* Biểu Đồ Hoạt Động (Activity Diagram)
* Biểu Đồ Thành Phần (Component Diagram)
* Biểu Đồ Triển Khai (Deployment Diagram)

**Ví Dụ :**

1. **Thuật toán là gì ?**

**Thuật toán** , còn gọi là **giải thuật**, là một tập hợp hữu hạn của các chỉ thị hay phương cách được định nghĩa rõ ràng cho việc hoàn tất một số sự việc từ một trạng thái ban đầu cho trước; khi các chỉ thị này được áp dụng triệt để thì sẽ dẫn đến kết quả sau cùng như đã dự đoán.

Nói cách khác, thuật toán là một bộ các quy tắc hay quy trình cụ thể nhằm giải quyết một vấn đề trong một số bước hữu hạn, hoặc nhằm cung cấp một kết quả từ một tập hợp của các dữ kiện đưa vào.

**Ví dụ:**

thuật toán để giải phương trình bậc nhất P(x): *a*x + *b* = *c*, (*a*, *b*, *c* là các số thực), trong tập hợp các số thực có thể là một bộ các bước sau đây:

1. Nếu *a* = 0
   * *b* = *c* thì P(x) có nghiệm bất kì
   * *b* ≠ *c* thì P(*c*) vô nghiệm
2. Nếu *a* ≠ 0
   * P(x) có duy nhất một nghiệm x = (*c* - *b*)/*a*
3. **Mã giải là gì ?**

**Mã giả** (tiếng Anh: *Pseudocode*, xuất phát từ chữ pseudo và code) là một bản mô tả giải thuật lập trình máy tính ngắn gọn và không chính thức cấp cao, trong đó sử dụng những quy ước có cấu trúc của một số ngôn ngữ lập trình, nhưng thường bỏ đi những chi tiết không cần thiết để giúp hiểu rõ giải thuật hơn, như bỏ đi chương trình con, khai báo biến và những đoạn mã đặc biệt của hệ thống. Ngôn ngữ lập trình được bổ sung bằng những mô tả chi tiết bằng ngôn ngữ tự nhiên ở nơi thích hợp, hoặc bằng ký hiệu toán học đơn giản. Mục đích của việc sử dụng mã giả là nó có thể con người đọc hiểu dễ dàng hơn những ngôn ngữ lập trình thông thường, và mã giả có thể là một bản mô tả cô đọng và độc lập với môi trường phát triển ghi lại những nguyên lý chủ chốt của một giải thuật. Không có bất cứ tiêu chuẩn nào cho cú pháp mã giả, vì một chương trình viết bằng mã giả không phải là một chương trình có thể thực thi được.

**Ví dụ về mã giải.**

Một ví dụ về mã giả khác nhau với mã thường.

|  |  |
| --- | --- |
| Mã thông thường (viết bằng [PHP](https://vi.wikipedia.org/wiki/PHP)):  <?php  **if** (is\_valid($cc\_number)) {  execute\_transaction($cc\_number, $order);  } **else** {  show\_failure();  }  ?> | **Mã giả:**  **if** số thẻ tín dụng là đúng **then**  thực hiện giao dịch dựa trên số và thứ tự  **else**  hiển thị thông báo thất bại chung chung  **end if** |