

BỘ MÔN DUYỆT
Chủ nhiệm Bộ môn

ĐỀ CƯƠNG CHI TIẾT BÀI GIẢNG
(Dùng cho CNPM 8)

Thay mặt nhóm
môn học

Học phần: Thiết kế và xây dựng
phần mềm

Nhóm môn học:

Phan Nguyên Hải

Bộ môn: Công nghệ phần mềm

Phạm Văn Việt

Khoa: Công nghệ thông tin

Thông tin về nhóm môn học: Chuyên ngành chuyên sâu CNPM (Lý thuyết CNPM)

TT	Họ tên giáo viên	Học hàm	Học vị
1	Phạm Văn Việt	GV	TS
2	Phan Việt Anh	GV	ThS
3	Phạm Thị Bích Vân	TG	KS

Địa điểm làm việc: Bộ môn Công nghệ phần mềm, P1208, Nhà A1 (Gần đường HQ Việt)

Điện thoại, email: 01279858755, v.v.pham2012@gmail.com

Bài giảng 1: Giới thiệu về công nghệ phần mềm

Chương 1 Giới thiệu về công nghệ phần mềm

Tiết thứ: 1 – 4 Tuần thứ: 1

- **Mục đích, yêu cầu:** Giúp sinh viên ôn lại những kiến thức về công nghệ phần mềm:

- + Phần mềm là gì?
- + Tại sao cần có công nghệ phần mềm?
- + Các quy trình phát triển phần mềm.
- + Các mô hình phát triển phần mềm.
- + Quy trình quản lý dự án phần mềm và lập kế hoạch cho một dự án.

- **Hình thức tổ chức dạy học:** Lý thuyết, thảo luận, tự học, tự nghiên cứu

- **Thời gian:** Lý thuyết, thảo luận: 4t; Tự học, tự nghiên cứu: 8t

- **Địa điểm:** Giảng đường do P2 phân công.

- **Nội dung chính:**

I. Giới thiệu về phần mềm

1.1 Các loại phần mềm và quy mô phần mềm

1.2 Những khó khăn khi phát triển một dự án lớn

- Đòi hỏi nhiều công sức
- Chi phí cao
- Thời gian dài
- Yêu cầu thay đổi
- Rủi ro cao

1.3 Định nghĩa một dự án phần mềm thành công

- Dự án hoàn thành
- Sản phẩm có ý nghĩa, có thể dùng, và được dùng
- Đầu tư hiệu quả và có khả năng duy trì

1.4 Các lý do thất bại của một phần mềm

- Không giữ đúng lịch trình
- Chi phí quá cao
- Không giải quyết được yêu cầu của người dùng
- Chất lượng thấp
- Khả năng duy trì phần mềm không cao
- Không lập kế hoạch
- Các sản phẩm bàn giao không được định rõ

- Không hiểu rõ yêu cầu của người dùng
- Không kiểm soát và xem xét lại trong quá trình phát triển
- Trình độ kỹ thuật của lực lượng phát triển phần mềm không đáp ứng được
- Không hiểu rõ chi phí và công sức bỏ qua

1.5 Các quy tắc phát triển dự án kỹ thuật khác và ứng dụng trong công nghệ phần mềm

- Yêu cầu cách tiếp cận rõ ràng và có thể áp dụng cho nhiều dự án
- Yêu cầu quản lý dự án
- Yêu cầu chất lượng dự án

1.6 Những điểm giống và khác của một dự án phần mềm và một dự án kỹ thuật

1.7 Những khó khăn trong phát triển phần mềm

- Làm việc với người sử dụng
- Với đội ngũ kỹ thuật
- Với người quản lý

II. Các quy trình phần mềm

2.1 Định nghĩa quy trình

2.2 Quy trình phát triển phần mềm

2.3 Quy trình quản lý dự án

2.4 Quy trình quản lý thay đổi và cấu hình phần mềm

2.5 Quy trình quản lý chung

2.6 Các đặc tính của một quy trình tốt

III. Các mô hình phần mềm

3.1 Mô hình thác nước

3.2 Mô hình lập dự án mẫu

3.3 Mô hình tương tác

3.4 Mô hình xoắn ốc

IV. Quy trình quản lý dự án

V. Lập kế hoạch cho một dự án phần mềm

- ***Yêu cầu SV chuẩn bị:***

- + Đọc trước: Mục 6.1 và 6.2, Tài liệu 1.
- + Bài tập: Lập kế hoạch cho một dự án phần mềm.
- + Câu hỏi ôn tập:
 - a) Liệt kê các loại phần mềm
 - b) Liệt kê các lý do thất bại của một dự án phần mềm
 - c) Liệt kê các quy trình phần mềm

d) Liệt kê các mô hình phần mềm, điểm mạnh và điểm yếu của từng mô hình phần mềm

Bài giảng 2: Thiết kế phần mềm

Chương 2 Thiết kế phần mềm

Tiết thứ: 5 – 10 Tuần thứ: 2, 3

- **Mục đích, yêu cầu:** Giúp sinh viên nắm được

- Các khái niệm về thiết kế phần mềm
- Các phương pháp thiết kế
- Phương pháp thiết kế hướng chức năng.

- **Hình thức tổ chức dạy học:** Lý thuyết, thảo luận, tự học, tự nghiên cứu

- **Thời gian:** Lý thuyết, thảo luận: 6t; Tự học, tự nghiên cứu: 12t

- **Địa điểm:** Giảng đường do P2 phân công.

- **Nội dung chính:**

I. Các khái niệm trong thiết kế phần mềm

Hoạt động thiết kế bắt đầu khi có tài liệu về các yêu cầu đối với phần mềm cần phát triển và kiến trúc phần mềm được thiết kế. Trong khi thiết kế kiến trúc phần mềm tiếp tục được cải tiến. Thông thường thiết kế tập trung xây dựng các module.

1.1 Khái niệm một bản thiết kế

Một bản thiết kế là một sơ đồ các module.

1.2 Khái niệm Module

Một module có thể là một gói, một lớp, một thủ tục, một phương thức, một tập các hàm hay một tập các lớp.

1.3 Các kiểu quan hệ giữa các module

- Một module là một phần của một module khác
- Một module phụ thuộc vào một module khác
- Một module là khái quát hóa của một module khác

1.4 Quy trình thiết kế

- Thiết kế hệ thống (high-level)
- Thiết kế chi tiết

1.5 Khái niệm phương pháp thiết kế

Phương pháp thiết kế là một cách tiếp cận hệ thống nhằm tạo ra một bản thiết kế thông qua việc áp dụng một tập các kỹ thuật và hướng dẫn.

1.6 Các tiêu chí đánh giá một bản thiết kế

Một bản thiết kế là đúng nếu nó đáp ứng được các yêu cầu đặt ra. Trên thực tế một hệ thống có thể có nhiều bản thiết kế chấp nhận được. Để chọn ra một bản thiết kế phải dựa vào các tiêu chí đánh giá một bản thiết kế.

- Khả năng module hóa

Một hệ thống được coi là có tính module nếu nó chứa các module riêng rẽ. Mỗi module được xây dựng riêng và thay đổi của một module có ít ảnh hưởng nhất tới các module khác.

- Mối liên kết giữa các module

Hai module được coi là độc lập nếu một module có thể hoạt động mà không dựa vào module kia. Rõ ràng nếu hai module độc lập, chúng có thể được giải quyết và thay đổi mà riêng rẽ. Tuy nhiên các module trong một hệ thống không thể hoàn toàn độc lập với nhau, vì chúng phải tương tác để đạt được những hành vi mong muốn của hệ thống. Mối liên kết giữa các module (Coupling) thể hiện mức độ kết nối giữa các module.

- Mối liên kết giữa các thành phần trong một module

Sự liên kết giữa các module giảm khi mối quan hệ giữa các phần tử trong các module khác nhau giảm. Nghĩa là sự liên kết giảm khi các phần tử ở các module khác nhau có ít hoặc không có quan hệ. Một cách khác để đạt được điều này là tăng cường mối quan hệ giữa các phần tử trong cùng module.

- Nguyên tắc mở để cập nhật với những yêu cầu mới và đóng với thiết kế cũ.

Nguyên tắc này muốn nói đến khả năng mở của phần mềm cho việc mở rộng, và hạn chế việc thay đổi đối với các module đã được xây dựng.

II. Các phương pháp thiết kế phần mềm

Tạo ra một bản thiết kế cho một hệ thống là công việc chính trong giai đoạn thiết kế. Nhiều kỹ thuật thiết kế đã được đề xuất để đưa ra một phương pháp trong xử lý sự phức tạp khi thiết kế các hệ thống lớn. Mục đích của các phương pháp thiết kế không phải để biến quá trình thiết kế thành một chuỗi các bước máy móc mà để đưa ra những hướng dẫn hỗ trợ cho người thiết kế trong quá trình thiết kế.

III. Thiết kế hướng chức năng

3.1 Biểu đồ cấu trúc

- Khái niệm biểu đồ cấu trúc
- Biểu diễn biểu đồ cấu trúc
- Xây dựng biểu đồ cấu trúc cho một chương trình
- Phân loại module

3.2 Phương pháp thiết kế hướng cấu trúc

3.3 Ý tưởng, mục tiêu của phương pháp thiết kế hướng cấu trúc

3.4 Các bước trong phương pháp thiết kế hướng cấu trúc

- Phát biểu lại bài toán dưới dạng sơ đồ luồng dữ liệu
- Xác định các thành phần dữ liệu vào và ra ở mức trừu tượng nhất
- Phân cấp module mức một
- Tiếp tục phân cấp các module

- Yêu cầu SV chuẩn bị:

+ Đọc trước: Mục 6.3, Tài liệu 1. Tài liệu 3.

+ Bài tập:

- Xây dựng biểu đồ cấu trúc cho một chương trình
- Vẽ sơ đồ luồng dữ liệu của một hệ thống
- Xác định dữ liệu vào và ra ở mức trừu tượng nhất trong DFD
- Xác định sơ đồ phân cấp mức một
- Xác định các sơ đồ thức cấp

+ Câu hỏi ôn tập:

- Một bản thiết kế phần mềm là gì?
- Module là gì?
- Các kiểu quan hệ giữa các module?
- Một bản thiết kế hệ thống/ chi tiết là gì?
- Phương pháp thiết kế là gì?
- Liệt kê các đặc tính để đánh giá một bản thiết kế?
- Sơ đồ cấu trúc là gì?
- Liệt kê các loại module?

Bài giảng 3: Giới thiệu các công cụ phát triển phần mềm hướng đối tượng

Chương 3 Giới thiệu các công cụ phát triển phần mềm hướng đối tượng

Tiết thứ: 11 – 16 Tuần thứ: 3, 4

- **Mục đích, yêu cầu:** Giúp sinh viên nắm được

- Những ưu điểm và khái niệm cơ bản của phương pháp phát triển phần mềm hướng đối tượng
- Công cụ mô hình hóa hợp nhất UML
- Các mô hình UML trong thiết kế phần mềm hướng đối tượng
- Môi trường phát triển phần mềm hướng đối tượng Visual C#.

- **Hình thức tổ chức dạy học:** Lý thuyết, thảo luận, tự học, tự nghiên cứu

- **Thời gian:** Lý thuyết, thảo luận: 6t; Tự học, tự nghiên cứu: 12t

- **Địa điểm:** Giảng đường do P2 phân công.

- **Nội dung chính:**

I. Thiết kế phần mềm hướng đối tượng

Các cách tiếp cận hướng đối tượng để phát triển phần mềm đã trở nên ngày càng phổ biến trong những năm gần đây. Nhiều hệ thống mới ngày nay được xây dựng sử dụng các kỹ thuật hướng đối tượng và ngôn ngữ lập trình hướng đối tượng. Các hệ thống hướng đối tượng có nhiều lợi ích. Một mô hình đối tượng có thể biểu diễn sát bài toán đặt ra, giúp chúng ta có thể tạo ra và hiểu các thiết kế dễ hơn. Các đối tượng trong hệ thống ít bị thay đổi khi các yêu cầu thay đổi, do đó cho phép việc thay đổi dễ dàng hơn. Mối liên kết có tính thừa kế và tính đóng của các đối tượng trong thiết kế các thực thể của bài toán, tạo điều kiện thuận lợi cho việc sử dụng lại. Các ứng dụng mới có thể sử dụng các module đã có hiệu quả hơn, từ đó giúp giảm chi phí và thời gian phát triển phần mềm.

II. Ngôn ngữ mô hình hóa hợp nhất (UML)

2.1 Giới thiệu về UML

UML là ký hiệu đồ họa để biểu diễn các thiết kế hướng đối tượng. UML được gọi là ngôn ngữ mô hình hóa. UML không chỉ có khả năng biểu diễn các thiết kế, nó còn cho phép biểu diễn nhiều khía cạnh khác của hệ thống. Với một bản thiết kế hướng đối tượng, đặc tả của các lớp của hệ thống có thể là đã đủ. Tuy nhiên, trong khi mô hình hóa, người thiết kế cũng muốn hiểu các lớp khác nhau liên hệ với nhau như thế nào và chúng tương tác với nhau như thế nào để đạt được chức năng mong muốn. Khía cạnh này của việc mô hình hóa giúp tạo ra những thiết kế có khả năng đáp ứng được các yêu cầu của hệ thống. Nhờ khả

năng tạo ra các mô hình khác nhau, UML đã trở thành công cụ hỗ trợ cho việc hiểu hệ thống, thiết kế hệ thống, và cung cấp các ký hiệu cho việc biểu diễn một bản thiết kế.

2.2 Sơ đồ lớp

Sơ đồ lớp của UML là thành phần trung tâm của một bản thiết kế hay một mô hình. Các sơ đồ lớp mô tả các lớp trong bản thiết kế. Vì một chương trình hướng đối tượng chủ yếu chứa các lớp, các sơ đồ lớp có mối quan hệ rất chặt với chương trình cuối cùng của một hệ thống. Một sơ đồ lớp định nghĩa các lớp có trong hệ thống, biểu diễn sự liên kết và mối quan hệ giữa các lớp.

- Mô hình hóa các lớp
- Mô hình hóa liên kết các lớp
- Mô hình hóa kiểu thứ cấp và kiểu cơ sở

2.3 Sơ đồ tương tác

Các sơ đồ lớp biểu diễn các trúc tĩnh của hệ thống. Các sơ đồ này biểu diễn cấu trúc chương trình sẽ được xây dựng và biểu diễn các lớp khác nhau liên hệ với nhau như thế nào trong chương trình. Tuy nhiên, các sơ đồ này không biểu diễn hành vi động của hệ thống. Để biểu diễn điều này ta dùng các sơ đồ tương tác.

- Sơ đồ trình tự
- Sơ đồ cộng tác

2.4 Các sơ đồ và khả năng biểu diễn khác của UML

UML là một hệ thống ký hiệu mô hình hóa có thể mở rộng và khá tinh vi. Ngoài khả năng biểu diễn hai loại sơ đồ chung nhất khi mô hình hóa một hệ thống là: sơ đồ lớp và sơ đồ tương tác, UML còn cho phép biểu diễn nhiều loại mô hình khác.

- Mô hình hóa chức năng
- Sơ đồ trạng thái
- Sơ đồ hoạt động
- Sơ đồ các hệ thống con
- Sơ đồ các gói
- Sơ đồ các thành phần
- Biểu diễn giá trị đính kèm
- Biểu diễn các khái niệm mới dựa vào những biểu diễn đã có

III. Môi trường phát triển phần mềm (Visual C#)

3.1 Giới thiệu

Microsoft Visual C# là một ngôn ngữ lập trình, được thiết kế để xây dựng nhiều loại ứng dụng khác nhau chạy trên .NET Framework. C# là một ngôn ngữ mạnh, đơn giản, và hướng đối tượng. C# cho phép phát triển nhanh các ứng dụng,

trong khi vẫn duy trì được những điểm mạnh của các ngôn ngữ lập trình dựa trên C.

3.2 Những nguyên tắc cơ bản của ngôn ngữ C#

3.3 Các lớp và các đối tượng

3.4 Tính thừa kế và tính đa hình

3.5 Ứng dụng Windows

3.6 Truy cập dữ liệu sử dụng ADO.net

- Yêu cầu SV chuẩn bị:

+ Đọc trước: Chương 2, Tài liệu 2.

+ Bài tập:

- Định nghĩa lớp, lớp thừa kế trong C#.
- Tạo ứng dụng Console trong C#
- Tạo ứng dụng Windows trong C#.

+ Câu hỏi ôn tập:

- UML là gì?
- Có những thành phần nào trong biểu diễn một lớp?
- Cách biểu diễn các mối quan hệ giữa các lớp?
- Nêu cách biểu diễn sơ đồ lớp?
- Nêu cách biểu diễn sơ đồ tương tác?
- Nêu cách biểu diễn một khái niệm mới?
- Nêu cách biểu diễn thành phần đính kèm?
- Nêu các khả năng biểu diễn khác của UML?

Bài giảng 4: Các yêu cầu của hệ thống

Chương 4 Các yêu cầu của hệ thống

Tiết thứ: 17 – 20 Tuần thứ: 5

- **Mục đích, yêu cầu:** Giúp sinh viên nắm được

- Hệ thống thực được dùng để nghiên cứu trong môn học
- Giải thích được vai trò của kỹ thuật lập yêu cầu hệ thống trong quy trình phát triển hệ thống
- Mô tả được các giai đoạn chính trong kỹ thuật lập yêu cầu
- Nhận ra và sử dụng một số kỹ thuật phổ biến để tổng hợp yêu cầu.

- **Hình thức tổ chức dạy học:** Lý thuyết, thảo luận, tự học, tự nghiên cứu

- **Thời gian:** Lý thuyết, thảo luận: 4t; Tự học, tự nghiên cứu: 8t

- **Địa điểm:** Giảng đường do P2 phân công.

- **Nội dung chính:**

I. Giới thiệu hệ thống thực dùng để nghiên cứu (case study)

1.1 Thực trạng của cửa hàng thuê xe đạp Wheels

1.2 Lý do xây dựng một hệ thống mới.

II. Kỹ thuật lập bản yêu cầu của hệ thống

2.1 Định nghĩa kỹ thuật lập bản yêu cầu

2.2 Lý do cần phải lập bản yêu cầu

2.3 Các giai đoạn trong kỹ thuật lập bản yêu cầu

III. Tập hợp yêu cầu của hệ thống (Requirement Elicitation)

3.1 Khái niệm

3.2 Các kỹ thuật

- Phỏng vấn (interview)
- Bảng hỏi (questionnaires)
- Lập kịch bản
- Quan sát công việc hàng ngày của mọi người

IV. Đặc tả yêu cầu của hệ thống

4.1 Khái niệm đặc tả yêu cầu

4.2 Định nghĩa bài toán

4.3 Danh sách các vấn đề và yêu cầu

V. Xác nhận yêu cầu

5.1 Khái niệm xác nhận yêu cầu

5.2 Các cách xác nhận yêu cầu

VI. Danh sách các yêu cầu của hệ thống được nghiên cứu

- *Yêu cầu SV chuẩn bị:*

+ Đọc trước: Chương 3, Tài liệu 2.

+ Bài tập:

- Định nghĩa bài toán cho hệ thống thực; lập danh sách các vấn đề và yêu cầu.

+ Câu hỏi ôn tập:

- Nêu các giai đoạn chính trong kỹ thuật lập bản yêu cầu của hệ thống?
- Nêu những tài liệu do người phát triển tạo ra trước khi và sau khi phỏng vấn khách hàng và người sử dụng của hệ thống?
- Bảng hỏi có ích khi nào?
- Cho biết kịch bản dùng để tập hợp yêu cầu là gì?
- Nêu những phần điển hình trong định nghĩa của một bài toán?
- Những đặc tính nào của một yêu cầu nên được ghi lại trong danh sách các bài toán và danh sách yêu cầu?
- Mục đích của việc kiểm tra?

Bài giảng 5: Các ca sử dụng

Chương 5 Các ca sử dụng

Tiết thứ: 21 – 24 Tuần thứ: 6

- **Mục đích, yêu cầu:** Giúp sinh viên nắm được

- Giải thích được mục đích của mô hình use case
- Nhận ra và mô tả được các thành phần của mô hình use case như: sơ đồ use case, mô tả tác nhân, mô tả use case, và các kịch bản
- Xác định các use case và tác nhân của hệ thống
- Vẽ sơ đồ use case
- Viết mô tả use case và tác nhân
- Giải thích được quy mô của một use case
- Lập kịch bản và hiểu được các kịch bản liên hệ với các use case như thế nào
- Chỉ ra được các use case được dùng cho phát triển phần mềm hướng đối tượng trong giai đoạn nào.

- **Hình thức tổ chức dạy học:** Lý thuyết, thảo luận, tự học, tự nghiên cứu

- **Thời gian:** Lý thuyết, thảo luận: 4t; Tự học, tự nghiên cứu: 8t

- **Địa điểm:** Giảng đường do P2 phân công.

- **Nội dung chính:**

I. Khái niệm use case

II. Lược đồ tình huống sử dụng (use case)

2.1 Các thành phần trong lược đồ use case

2.2 Lược đồ tình huống sử dụng của hệ thống Wheels

III. Cách xác định use case

3.1 Xác định use case thông qua tác nhân

3.2 Xác định use case thông qua các kịch bản

IV. Mô tả use case

4.1 Mô tả tổng quát

4.2 Mô tả chi tiết

4.3 Điều kiện tiên quyết trong mô tả use case

V. Mô tả tác nhân của hệ thống

- Định nghĩa tác nhân
- Cách xác định tác nhân
- Xác định tác nhân của hệ thống Wheels

VI. Các mối quan hệ của use case

6.1 Quan hệ giữa tác nhân và use case (communication association)

6.2 Quan hệ giữa các use case

- Quan hệ bao gồm (include)
- Quan hệ mở rộng (extend)
- Lập mô tả use case có quan hệ bao gồm và mở rộng

VII. Đường biên biểu diễn use case

VIII. Sử dụng mô hình tình huống sử dụng để phát triển hệ thống

8.1 Kiểm tra hệ thống sử dụng mô hình use case

8.2 Ước lượng quy trình phát triển sử dụng use case

8.3 Use case là cơ sở cho các sơ đồ tương tác

8.4 Use case làm điểm bắt đầu để xác định các lớp

- Yêu cầu SV chuẩn bị:

+ Đọc trước: Chương 4, Tài liệu 2.

+ Bài tập:

- Xác định lược đồ use case của hệ thống Wheels; xác định các mối quan hệ của use case trong hệ thống Wheels

+ Câu hỏi ôn tập:

- Use case mô hình khía cạnh nào của hệ thống?
- Nêu tên các thành phần bổ sung của một mô hình use case?
- Nêu hai cách xác định các use case?
- Trình bày quan hệ giữa các kịch bản và các use case?
- Lập kịch bản cho tất cả các dãy sự kiện có thể là không thực tế. Trình bày tập kịch bản cần có cho phát triển phần mềm?
- Mô tả những nội dung tiêu biểu của một bản mô tả use case mức cao và một bản miêu tả mở rộng.
- Nêu mối quan hệ giữa một tác nhân và một use case.
- Chúng ta có thể dùng bốn loại quan hệ trong sơ đồ use case. Liệt kê và mô tả từng mối quan hệ.
- Mô hình use case còn thể dùng làm gì?

Bài giảng 6: Đối tượng và lớp: Các khái niệm cơ bản

Chương 6 Đối tượng và lớp: Các khái niệm cơ bản

Tiết thứ: 25 – 28 Tuần thứ: 7

- **Mục đích, yêu cầu:** Giúp sinh viên nắm được

- Nhận ra những hạn chế của các cách phát triển phần mềm truyền thống
- Giải thích được cách tiếp cận hướng đối tượng giải quyết các vấn đề này như thế nào
- Mô tả những đặc trưng chính của một đối tượng và lợi ích của nó trong xây dựng phần mềm
- Áp dụng khái niệm lớp và giải thích quan hệ giữa các đối tượng và các lớp
- Xác định được mối quan hệ giữa các đối tượng
- Áp dụng được các tính chất thừa kế và đa hình.

- **Hình thức tổ chức dạy học:** Lý thuyết, thảo luận, tự học, tự nghiên cứu

- **Thời gian:** Lý thuyết, thảo luận: 4t; Tự học, tự nghiên cứu: 8t

- **Địa điểm:** Giảng đường do P2 phân công.

- **Nội dung chính:**

I. Giới thiệu

Chương này xem xét những vấn đề tồn tại của cách tiếp cận phát triển phần mềm truyền thống và việc cách tiếp cận hướng đối tượng giải quyết các vấn đề này như thế nào. Đối tượng được đề cập như là nền móng của một phần mềm hướng đối tượng. Chương này sau đó giải thích khái niệm lớp và mối quan hệ giữa các đối tượng và các lớp. Cuối cùng, các mối quan hệ khác nhau có thể tồn tại giữa các đối tượng được xem xét và ý nghĩa của những mối quan hệ này trong hệ thống đang được phát triển được giải thích.

II. Tại sao cần phát triển phần mềm hướng đối tượng?

2.1 Những hạn chế của cách tiếp cận hướng cấu trúc

- Phân rã chức năng
- Duy trì phần mềm
- Khả năng module hóa hạn chế
- Những hạn chế trong kiểm thử (testing)
- Những hạn chế về khả năng sử dụng lại phần mềm
- Cách phát triển phần mềm hướng chức năng không mạnh bằng cách tiếp cận dựa trên dữ liệu

2.2 Những điểm mạnh của phát triển phần mềm hướng đối tượng

2.3 Phân rã use case và phân rã hướng đối tượng

III. Định nghĩa đối tượng

3.1 Hoạt động (behaviour) của đối tượng

3.2 Trạng thái của đối tượng

3.3 Thành phần nhận diện của đối tượng (Identity)

3.4 Tính đóng gói và khả năng giấu dữ liệu

- Giao diện và các thông điệp công khai

- Những hạn chế của mỗi quan hệ phụ thuộc

IV. Định nghĩa lớp

V. Môi quan hệ giữa các lớp

5.1 Môi quan hệ liên kết (Association)

5.2 Môi quan hệ chứa trong (Aggregation)

5.3 Môi quan hệ thừa kế và khái quát hóa

5.4 Các lớp trừu tượng

5.5 Tính đa hình

- ***Yêu cầu SV chuẩn bị:***

+ Đọc trước: Chương 5, Tài liệu 2.

+ Bài tập:

- Xác định tên lớp và các thuộc tính cho các nhóm đối tượng, xác định các lớp có thể cho một hệ thống, vẽ sơ đồ quan hệ giữa các lớp.

+ Câu hỏi ôn tập:

- Nêu những vấn đề của cách tiếp cận hướng cấu trúc
- Nêu bốn phẩm chất cần có trong xây dựng phần mềm
- Phát triển không gián đoạn (seamless development) là gì?
- Nêu sự khác biệt giữa lớp và đối tượng
- Điểm khác biệt trong mô tả một lớp và một đối tượng trong UML?
- Hoạt động của một đối tượng là gì?
- Trạng thái của một đối tượng ảnh hưởng tới hoạt động của nó như thế nào?
- Trình bày tính đóng gói của các đối tượng
- Các đối tượng giao tiếp với nhau như thế nào?

- Giao diện công khai (public) của một đối tượng nghĩa là gì?
- Liệt kê và mô tả ba dạng quan hệ giữa các lớp.
- Tại sao chúng ta có thể quyết định mô hình một lớp như là một lớp con của một lớp khác?
- Lớp trừu tượng (abstract class) là gì? Tại sao các lớp này được dùng?
- Sự khác biệt giữa một hoạt động và một phương thức? Nêu điểm mạnh của tính đa hình?
- Ràng buộc động là gì?

Bài giảng 7 Biểu đồ lớp

Chương 7 Biểu đồ lớp

Tiết thứ: 29 – 32 Tuần thứ: 8

- **Mục đích, yêu cầu:** Giúp sinh viên nắm được

- Giải thích được vai trò và mục đích của biểu đồ lớp trong những giai đoạn đầu phát triển phần mềm;
- Nhận diện được các giai đoạn khác nhau trong xây dựng một biểu đồ lớp;
- Xác định được các đối tượng, các lớp và các thuộc tính của chúng trong một hệ thống;
- Xây dựng được biểu đồ lớp;
- Lập từ điển dữ liệu hỗ trợ biểu đồ lớp;
- Sử dụng đóng góp để gộp nhóm các lớp liên quan.

- **Hình thức tổ chức dạy học:** Lý thuyết, thảo luận, tự học, tự nghiên cứu

- **Thời gian:** Lý thuyết, thảo luận: 4t; Tự học, tự nghiên cứu: 8t

- **Địa điểm:** Giảng đường do P2 phân công.

- **Nội dung chính:**

I. Giới thiệu

Biểu đồ lớp là trung tâm của quá trình phân tích và thiết kế. Biểu đồ lớp định nghĩa cả kiến trúc phần mềm (tức là cấu trúc tổng thể của hệ thống), và cấu trúc của mọi đối tượng trong hệ thống. Biểu đồ lớp được dùng để mô hình các lớp và mối quan hệ giữa các lớp, đồng thời cũng để mô hình các cấu trúc tổng thể bao gồm tập hợp các lớp được nhóm lại trong các gói. Biểu đồ lớp xuất hiện lặp đi lặp lại trong tất cả các giai đoạn của quá trình phát triển. Đầu tiên, nó được dùng để mô hình mọi điều trong lĩnh vực ứng dụng như một phần của bản đặc tả yêu cầu. Tiếp theo, khi các lớp được bổ sung thêm cho phân giải pháp, biểu đồ lớp được dùng để thiết kế một giải pháp. Cuối cùng các lớp (như lớp buttons, windows, mouse-listeners) được thêm vào để tạo điều kiện thuận lợi cho quá trình thực hiện, nó được dùng để thiết kế chương trình.

II. Các giai đoạn xây dựng biểu đồ lớp

2.1 Xác định các đối tượng và các lớp

- Xác định đối tượng thông qua phân tích các danh từ trong phần định nghĩa bài toán, và mô tả use case

- Xác định danh sách các đối tượng ứng cử viên có thể (là các danh từ hoặc các cụm danh từ)

- Loại bỏ các ứng cử viên không phù hợp, sử dụng danh sách các tiêu chí loại bỏ.

- Danh sách các đối tượng của hệ thống Wheels: bike, customer, hire transaction, specialist bikes

- Cơ sở biểu đồ lớp của hệ thống Wheels: Bike, Customer, Hire, SpecialistBike

2.2 Xác định các thuộc tính

- Các thuộc tính của lớp Bike

- Các thuộc tính của lớp Hire

- Các thuộc tính của lớp Customer

- Các thuộc tính của lớp Payment

2.3 Xác định mối quan hệ giữa các lớp

- Xác định mối quan hệ liên kết và một nhiều

- Xác định mối quan hệ khái quát hóa và thừa kế

2.4 Lập từ điển dữ liệu hỗ trợ xây dựng biểu đồ lớp

III. Các gói

IV. Sử dụng biểu đồ lớp để phát triển hệ thống

- ***Yêu cầu SV chuẩn bị:***

+ Đọc trước: Chương 6, Tài liệu 2.

+ Bài tập:

- Xác định các đối tượng và nhóm các đối tượng của một hệ thống, sử dụng kỹ thuật phân tích các danh từ trong mô tả của một hệ thống để tìm ra các đối tượng trong hệ thống, xác định các thuộc tính của các lớp, vẽ biểu đồ lớp, xây dựng từ điển dữ liệu cho một quyển sách.

+ Câu hỏi ôn tập:

- Biểu đồ lớp bắt đầu với các lớp (là một phần của bài toán). Nêu các loại lớp được thêm vào trong các giai đoạn phát triển?
- Quá trình thực hiện một use case là gì?
- Nêu các giai đoạn chính trong xây dựng biểu đồ lớp?
- Liệt kê bốn nhóm đối tượng.
- Liệt kê tám lý do để loại bỏ các đối tượng khi tìm kiếm các lớp

- Nêu lý do mô hình hóa một khai niệm như một lớp hơn là một quan hệ
- Mục đích của từ điển dữ liệu là gì?
- Đóng gói là gì?

Bài giảng 8 Nhận diện chức năng sử dụng thẻ CRC và biểu đồ tương tác

Chương 8 Nhận diện chức năng sử dụng thẻ CRC và biểu đồ tương tác

Tiết thứ: 33 – 36 Tuần thứ: 9

- **Mục đích, yêu cầu:** Giúp sinh viên nắm được

- Mô tả vai trò của các thẻ CRC trong xác định định nhiệm vụ và phân nhiệm vụ cho các lớp;
- Xác định các hoạt động riêng lẻ trong một nhiệm vụ;
- Giải thích được mục đích của biểu đồ tương tác;
- Vẽ được biểu đồ tuần tự/tương tác đơn giản;
- Viết được đặc tả tiến trình để mô tả chức năng của một hoạt động.

- **Hình thức tổ chức dạy học:** Lý thuyết, thảo luận, tự học, tự nghiên cứu

- **Thời gian:** Lý thuyết, thảo luận: 4t; Tự học, tự nghiên cứu: 8t

- **Địa điểm:** Giảng đường do P2 phân công.

- **Nội dung chính:**

I. Giới thiệu

II. Xác định các hoạt động sử dụng công cụ CRC card

2.1 Trách nhiệm và hoạt động của lớp

2.2 Thẻ CRC

2.3 Xác định các hoạt động từ các tránh nhiệm

III. Sơ đồ tương tác

3.1 Sơ đồ tuần tự

3.2 Sơ đồ hợp tác

3.3 Sử dụng sơ đồ tuần tự và hợp tác

3.4 Sự thống nhất của mô hình

3.5 Sử dụng các gói trong sơ đồ tương tác

IV. Xác định các hoạt động

V. Sử dụng CRC cards và các sơ đồ tương tác để phát triển hệ thống

- **Yêu cầu SV chuẩn bị:**

+ Đọc trước: Chương 7, Tài liệu 2.

+ Bài tập: 1-10 Chương 6, Tài liệu 2.

+ Câu hỏi ôn tập:

- Thẻ CRC nắm giữ khía cạnh nào của một lớp?

- Nêu hai loại sơ đồ thuộc sơ đồ tương tác?
- Sơ đồ tương tác mô hình điều gì?
- Ngoài định dạng ra, điểm khác biệt giữa sơ đồ tuần tự và sơ đồ kịch bản là gì?
- Các thông điệp được biểu diễn trong sơ đồ tuần tự và hợp tác như thế nào?
- Liệt kê những điểm khác giữa sơ đồ hợp tác và sơ đồ tuần tự.
- Nêu các đặc trưng của một hoạt động được định nghĩa trong đặc tả?
- Khi nào một cây hoặc một bảng quyết định được dùng để định rõ một hoạt động

Bài giảng 9 Các sơ đồ trạng thái

Chương 9 Các sơ đồ trạng thái

Tiết thứ: 37 – 40 Tuần thứ: 10

- **Mục đích, yêu cầu:** Giúp sinh viên nắm được

- Giải thích được vai trò và mục đích của sơ đồ trạng thái trong phát triển hệ thống hướng đối tượng;
- Sử dụng sơ đồ trạng thái xác định cách một lớp phản ứng đối với các sự kiện;
- Vẽ được sơ đồ trạng thái đơn giản.

- **Hình thức tổ chức dạy học:** Lý thuyết, thảo luận, tự học, tự nghiên cứu

- **Thời gian:** Lý thuyết, thảo luận: 4t; Tự học, tự nghiên cứu: 8t

- **Địa điểm:** Giảng đường do P2 phân công.

- **Nội dung chính:**

I. Giới thiệu

Chúng ta đã biết các mô hình hóa việc tổ chức và cấu trúc của dữ liệu trong hệ thống sử dụng sơ đồ lớp, và cách mô hình chuỗi các tương tác giữa các đối tượng sử dụng các sơ đồ tương tác. Trong chương này chúng ta sẽ xem xét: Một lớp bị ảnh hưởng như thế nào trong hệ thống bởi các use case khác nhau, và các đối tượng của một lớp phản ứng như thế nào đối với các sự kiện tác động lên chúng. Mô hình minh họa tất cả các hành vi của các đối tượng trong một lớp được gọi là sơ đồ trạng thái. Trong chương này, chúng ta sẽ xem xét các thành phần khác nhau của một sơ đồ trạng thái, cách chúng kết hợp, và cách các sơ đồ sử dụng để phát triển hệ thống.

II. Các trạng thái và sự kiện

Trạng thái của một đối tượng ám chỉ tình trạng của đối tượng, khi đối tượng thỏa mãn một điều kiện nào đó. Sự kiện là một điều gì đó xảy ra có ý nghĩa đối với hệ thống và ảnh hưởng tới ít nhất một đối tượng của một lớp của hệ thống.

III. Một sơ đồ trạng thái đơn giản

IV. Xây dựng sơ đồ trạng thái

V. Sơ đồ trạng thái của hệ thống Wheels

VI. Sử dụng các sơ đồ trạng thái để phát triển hệ thống

- **Yêu cầu SV chuẩn bị:**

+ Đọc trước: Chương 8, Tài liệu 2.

+ Bài tập: Vẽ sơ đồ trạng thái biểu diễn hoạt động của: một lò vi sóng đơn giản, của một khách hàng tại một quầy báo, hoặc của một máy pha café và chè

+ Câu hỏi ôn tập:

- Sơ đồ trạng thái mô hình khía cạnh nào của hệ thống?
- Trạng thái là gì?
- Sự kiện là gì?
- Nêu 3 phần của nhãn chuyển tiếp?
- Tự chuyển tiếp trên một trạng thái là gì?
- Khi nào một siêu trạng thái cần thêm vào sơ đồ trạng thái?
- Sơ đồ trạng thái hữu ích nhất với loại hệ thống nào?

Bài giảng 10 Sơ đồ hoạt động

Chương 10 Sơ đồ hoạt động

Tiết thứ: 41 – 44 Tuần thứ: 11

- **Mục đích, yêu cầu:** Giúp sinh viên nắm được

- Giúp sinh viên giải thích được vai trò và mục đích của sơ đồ trong phát triển hệ thống hướng đối tượng;
- Vẽ được sơ đồ hoạt động đơn giản;
- Sử dụng các sơ đồ hoạt động để phân tích luồng công việc, phân tích use cases và các hoạt động của các lớp.

- **Hình thức tổ chức dạy học:** Lý thuyết, thảo luận, tự học, tự nghiên cứu

- **Thời gian:** Lý thuyết, thảo luận: 4t; Tự học, tự nghiên cứu: 8t

- **Địa điểm:** Giảng đường do P2 phân công.

- **Nội dung chính:**

I. Giới thiệu

Như chúng ta đã biết, chức năng của hệ thống ban đầu được biểu diễn thông qua các use case. Các use case biểu diễn những hoạt động chính của hệ thống từ góc độ người sử dụng. Chức năng của hệ thống cũng được xác định trong các hoạt động của mỗi lớp trong sơ đồ lớp. Các sơ đồ trạng thái xác định quá trình trao đổi thông điệp giữa các đối tượng để thực hiện một nhiệm vụ nào đó. Sơ đồ trạng thái mô hình tất cả các hành vi của các đối tượng của một lớp.

Trong chương này chúng ta sẽ xem xét sơ đồ trạng thái dùng để mô hình chi tiết các tiến trình phức tạp. Các sơ đồ hoạt động tương tự như các sơ đồ trạng thái, chúng cùng quan tâm đến các trạng thái và sự chuyển đổi giữa các trạng thái. Tuy nhiên, trong sơ đồ hoạt động, tất cả các trạng thái là các hoạt động và sự chuyển đổi trạng thái được thực thi khi một hoạt động hoàn thành chứ không phải do sự tác động của một sự kiện bên ngoài.

II. Mô hình hóa chuỗi các hoạt động

III. Mô hình hóa các tiến trình có thể của một hoạt động

IV. Mô hình hóa sự lặp lại của các hoạt động

V. Mô hình hóa các hoạt động được tiến hành song song

VI. Sơ đồ phân trách nhiệm (Swimlanes)

- **Yêu cầu SV chuẩn bị:**

+ Đọc trước: Chương 9, Tài liệu 2.

+ Bài tập: Vẽ sơ đồ hoạt động mô tả quy trình đặt mua hàng tại một công ty bán lẻ, thêm Swimlanes vào sơ đồ hoạt động cho use case “trao xe” trong hệ thống Wheels để thấy được các hoạt động của lễ tân, máy tính và khách hàng.

+ Câu hỏi ôn tập:

- Trong sơ đồ trạng thái các trạng thái khác với các hoạt động ở mặt nào?
- Các sơ đồ hoạt động mô hình những quy trình nào trong phát triển phần mềm?
- Nêu những đặc trưng của hầu hết các ngôn ngữ lập trình có thể được mô hình bởi một sơ đồ hoạt động?
- Nếu một sơ đồ hoạt động cho thấy nhiều hoạt động được thực hiện song song. Điều đó phải là các hoạt động được thực hiện cùng lúc hay theo trình tự xử lý không có vấn đề gì?
- Hình thoi trong sơ đồ hoạt động nghĩa là gì?
- Sơ đồ phân trách nhiệm thể hiện điều gì trong sơ đồ hoạt động?

Bài giảng 11 Thiết kế tổng thể

Chương 11 **Thiết kế tổng thể**

Tiết thứ: 45 – 48 Tuần thứ: 12

- **Mục đích, yêu cầu:** Giúp sinh viên nắm được

- Nhận ra được sự khác biệt giữa phân tích và thiết kế;
- Giải thích được vai trò và mục đích của kiến trúc theo lớp trong một hệ thống hướng đối tượng;
- Vẽ được một biểu đồ thành phần (component) đơn giản;
- Vẽ được biểu đồ triển khai;
- Thiết kế được giao diện người dùng; Giải thích được sự khác nhau giữa cơ sở dữ liệu quan hệ và cơ sở dữ liệu hướng đối tượng;
- Hiểu được liên kết của một chương trình hướng đối tượng với một cơ sở dữ liệu quan hệ;
- Có thể chuyển một biểu đồ lớp đơn giản thành một tập các bảng trong cơ sở dữ liệu quan hệ;
- Có thể áp dụng các mẫu thiết kế.

- **Hình thức tổ chức dạy học:** Lý thuyết, thảo luận, tự học, tự nghiên cứu

- **Thời gian:** Lý thuyết, thảo luận: 4t; Tự học, tự nghiên cứu: 8t

- **Địa điểm:** Giảng đường do P2 phân công.

- **Nội dung chính:**

I. Giới thiệu

Quá trình phân tích mô tả điều gì hệ thống phải làm, trong khi quá trình thiết kế mô tả làm cách nào để thực hiện nó. Nói cách khác các quyết định phân tích là độc lập với việc thực hiện (xây dựng chương trình), trong khi các quyết định thiết kế có xu hướng phụ thuộc vào việc thực hiện. Trong quá trình phân tích, chúng ta cần hiểu và mô hình hóa các yêu cầu của người dùng, quá trình phân tích đưa ra bản đặc tả những công việc mà hệ thống mới sẽ phải làm. Trong quá trình thiết kế chúng ta quyết định làm cách nào để xây dựng hệ thống có khả năng đáp ứng được những yêu cầu đặt ra. Một bản thiết kế là một bản trừu tượng của chương trình, nó bỏ qua nhiều chi tiết trong chương trình, nhưng đưa ra những điểm cốt yếu về cấu trúc và các tương tác của các chương trình. Các hoạt động thiết kế bao gồm các hoạt động liên quan đến thiết kế hệ thống tổng thể, và các hoạt động liên quan đến thiết kế chi tiết.

II. Kiến trúc

2.1 Kiến trúc theo tầng

- Các lớp entity, boundary, và control
- Tính minh bạch
- Các gói và tính phụ thuộc
- Địa phương hóa những thay đổi
- Các tầng

III. Biểu đồ thực hiện

3.1 Các biểu đồ thành phần (component diagrams)

3.2 Các biểu đồ triển khai (deployment diagrams)

IV. Giao diện

4.1 Giao diện chính

4.2. Các giao diện con

V. Làm việc với dữ liệu bền vững

5.1 Cơ sở dữ liệu quan hệ và hướng đối tượng

5.2 Kết nối chương trình hướng đối tượng với cơ sở dữ liệu quan hệ

5.3 Xây dựng biểu đồ lớp trong cơ sở dữ liệu quan hệ

VI. Các điểm kỹ thuật

- ***Yêu cầu SV chuẩn bị:***

+ Đọc trước: Chương 10, Tài liệu 2.

+ Bài tập:

- Xác định sự phụ thuộc giữa hai gói
- Khi khách hàng trả xe, lễ tân cần tìm và xem các thông tin về giao dịch cho thuê xe này, Thiết kế giao diện hiển thị tên khách hàng, số xe, hãng sản xuất, kiểu xe, kích thước, ngày bắt đầu thuê, thời gian thuê, tiền đặt cọc và tiền phí thuê đã trả.
- Nêu cách lớp Hire trong hệ thống Wheels được thực hiện trong một cơ sở dữ liệu quan hệ
- Cho biểu đồ quan hệ một nhiều giữa 2 lớp Customer và lớp Hire, nêu hai cách mỗi quan hệ này có thể được thực trong cơ sở dữ liệu quan hệ

+ Câu hỏi ôn tập:

- Mô tả và cho ví dụ về một: entity class, boundary class, control class

- Tính minh bạch (visibility) của một hoạt động và một thuộc tính?
Giá trị của tính minh bạch?
- Tại sao tính minh bạch quan trọng?
- Kiến trúc theo lớp là gì? Nêu những lớp và trình tự của chúng trong một kiến trúc bốn lớp mà bạn mong đợi?
- Biểu đồ thành phần mô hình điều gì?
- Biểu đồ triển khai mô hình điều gì?
- Liệt kê 6 chỉ dẫn khi thiết kế giao diện?
- JDBC dùng để làm gì?
- Kể tên và cho ví dụ về các loại mẫu phần thiết kế?

Bài giảng 12 Thiết kế chi tiết

Chương 12 **Thiết kế chi tiết**

Tiết thứ: 49 – 52 Tuần thứ: 13

- **Mục đích, yêu cầu:** Giúp sinh viên nắm được

- Giải thích vai trò của lớp biên, lớp kiểm soát và lớp thực thể;
- Hiểu được sự cần thiết của các lớp tập hợp; Chỉ ra được sự thi hành các mối quan hệ giữa các lớp;
- Chỉ ra được tính minh bạch của các thuộc tính và các hoạt động;
- Chỉ ra được đầy đủ ký hiệu cho một thuộc tính và một hoạt động;
- Thêm được các thông tin thiết kế chi tiết cho các biểu đồ tương tác.

- **Hình thức tổ chức dạy học:** Lý thuyết, thảo luận, tự học, tự nghiên cứu

- **Thời gian:** Lý thuyết, thảo luận: 4t; Tự học, tự nghiên cứu: 8t

- **Địa điểm:** Giảng đường do P2 phân công.

- **Nội dung chính:**

I. Giới thiệu

Các hoạt động thiết kế chi tiết bao gồm các đặc tả chi tiết của các lớp, mối quan hệ và tương tác giữa các lớp. Các mô hình cần chứa đủ thông tin cho phép chúng được sử dụng như một đặc tả chương trình. Chương này xem lại nhiều chủ đề đã được mô tả từ trước và thảo luận từ góc độ thiết kế. Chương này xem lại các biểu đồ lớp và bổ sung thêm các lớp như các lớp quản lý giao diện, các lớp điều khiển trình tự thực hiện. Các vấn đề không quan trọng lắm trong quá trình phân tích như chi tiết kỹ thuật của liên kết giữa các lớp, phạm vi truy cập của các lớp, thuộc tính, và các toán tử được chỉ rõ. Các biểu đồ tương tác được xem lại và thảo luận làm thế nào để mô hình việc thêm và xóa các đối tượng và làm thế nào để chỉ rõ điều kiện và điều khiển trong biểu đồ tương tác.

II. Biểu đồ lớp

2.1 Lớp biên và lớp kiểm soát (boundary and control classes)

2.2 Thiết kế các liên kết

- Liên kết một – một
- Liên kết một – nhiều

2.3 Thiết kế các thuộc tính và các hoạt động

- Thiết kế các thành phần của một thuộc tính
- Thiết kế các hoạt động

III. Các biểu đồ tương tác

- 3.1 Tạo và xóa đối tượng
- 3.2 Tương tác
- 3.3 Rẽ nhánh và hoạt động có điều kiện
- 3.4 Mô hình các thông điệp đồng bộ và không đồng bộ
- 3.5 Loại bỏ chi tiết thông điệp.
- 3.6 Biểu đồ tuần tự của hệ thống được nghiên cứu

- Yêu cầu SV chuẩn bị:

+ Đọc trước: Chương 11, Tài liệu 2.

+ Bài tập:

- Giải thích sự khác biệt của một quan hệ liên kết trong một biểu đồ lớp giai đoạn phân tích và thiết kế
- Sử dụng một biểu đồ lớp có sẵn, vẽ biểu đồ tuần tự mô tả được một hoạt động nào đó
- Nêu cách đối tượng gửi biết được ID của một đối tượng nhận, trong biểu đồ tuần tự.

+ Câu hỏi ôn tập:

- Nêu bốn hoạt động thiết kế chi tiết
- Đường đi có thể là gì?
- Khả năng định tuyến đơn hướng là gì?
- Đường đi đơn hướng có thể được xây dựng như thế nào?
- Nêu định dạng ký hiệu của một thuộc tính/hoạt động?
- Việc tạo và xóa một đối tượng trong biểu đồ tuần tự được xác định như thế nào?

Bài giảng 13 Chương trình

Chương 13 Chương trình

Tiết thứ: 53 – 56 Tuần thứ: 14

- **Mục đích, yêu cầu:** Giúp sinh viên nắm được

- Hiểu được một chương trình đơn giản;
- Hiểu được cách một biểu đồ lớp chuyển thành một chương trình như thế nào;
- Hiểu được cách một biểu đồ tuần tự chuyển thành một chương trình như thế nào;
- Hiểu được trình tự thực hiện của một chương trình.

- **Hình thức tổ chức dạy học:** Lý thuyết, thảo luận, tự học, tự nghiên cứu

- **Thời gian:** Lý thuyết, thảo luận: 4t; Tự học, tự nghiên cứu: 8t

- **Địa điểm:** Giảng đường do P2 phân công.

- **Nội dung chính:**

I. Giới thiệu

Chương này thảo luận về chương trình cho hệ thống Wheels. Đối tượng của chương này không phải là một khóa về lập trình, mà để chỉ ra cách chuyển một số mô hình sang chương trình. Chương này không cố gắng giải thích các chỉ dẫn trong xây dựng chương trình. Chương trình được xem xét trên góc độ nó liên hệ với các biểu đồ lớp và biểu đồ tuần tự như thế nào. Chương này chỉ ra những dòng lệnh để thực thi các lớp, các thuộc tính, các phương thức và các mối quan hệ trong biểu đồ lớp. Biểu đồ tuần tự được sử dụng như một bản đồ lộ trình để chỉ ra trình tự thực hiện của chương trình.

Hệ thống được xây dựng bằng ngôn ngữ lập trình hướng đối tượng C#. Để đơn giản, chương trình được tạo ra trong chương này là một phần của hệ thống Wheels system, dành cho một kịch bản của use case ‘Giao xe’.

II. Quy ước lập trình

2.1 Quy ước đặt tên

2.2 Quy ước trình bày

2.3 Quy ước chú thích

III. Chương trình

3.1 Chương trình của lớp khởi động

3.2 Chương trình của các lớp khác

IV. Biểu đồ thi hành của các lớp

4.1 Biểu đồ thi hành của lớp khởi động

4.2 Biểu đồ thi hành của các lớp khác

V. Xây dựng chương trình theo biểu đồ tuần tự

- Yêu cầu SV chuẩn bị:

+ Đọc trước: Tài liệu 4.

+ Bài tập:

- Cho chương trình của một biểu đồ lớp, xác định hàm khởi tạo của các lớp, xác định các lớp có quan hệ thừa kế, xác định kết quả khi chạy chương trình
- Cho một chương trình hướng đối tượng, vẽ biểu đồ lớp mô hình các lớp và mối quan hệ. Trong đó mô hình phải thể hiện được các thuộc tính, phương thức và mối quan hệ. Vẽ biểu đồ tuần tự để thể hiện chuỗi các thông điệp được thực hiện trong hàm main(). Xác định kết quả của hàm main().

+ Câu hỏi ôn tập:

- Phương thức nào phải có trong ứng dụng C#?
- Hai phương thức nào thường không có trong biểu đồ lớp?
- Phương thức khởi tạo là gì? Làm thế nào để xác định phương thức này?
- Làm thế nào để C# biết khi nào một lớp bắt đầu và kết thúc?
- Các chú thích trong C# được xác định như thế nào?
- Các biến thành viên là gì?
- Một thông điệp tham chiếu tới chính nó là gì?

Bài giảng 14 Quản lý cấu hình phần mềm

Chương 14 Quản lý cấu hình phần mềm

Tiết thứ: 57 – 60 Tuần thứ: 15

- **Mục đích, yêu cầu:** Giúp sinh viên nắm được

- Nắm được khái niệm quản lý cấu hình phần mềm;
- Hiểu được tại sao phải quản lý cấu hình phần mềm;
- Nắm được các cách thức quản lý cấu hình phần mềm;
- Hiểu được sự cần thiết phải quản lý cấu hình phần mềm bằng các công cụ.

- **Hình thức tổ chức dạy học:** Lý thuyết, thảo luận, tự học, tự nghiên cứu

- **Thời gian:** Lý thuyết, thảo luận: 4t; Tự học, tự nghiên cứu: 8t

- **Địa điểm:** Giảng đường do P2 phân công.

- **Nội dung chính:**

I. Khái niệm quản lý cấu hình phần mềm

II. Những cách thức tốt trong quản lý cấu hình phần mềm

2.1 Xác định và lưu trữ các sản phẩm ở nơi an toàn

2.2 Kiểm soát và kiểm tra những thay đổi của các sản phẩm

2.3 Nhóm các sản phẩm của một phiên bản thành các thành phần (component)

2.4 Nhóm các component của một phiên bản và các hệ thống con thành các hệ thống con trong một phiên bản mới

2.5 Ghi lại phiên bản của các sản phẩm tại các mốc thời gian của dự án

2.6 Ghi lại và theo dõi những yêu cầu thay đổi

2.7 Tổ chức và hợp nhất các tập hợp của các phiên bản

2.8 Duy trì môi trường làm việc ổn định

2.9 Hỗ trợ những thay đổi đồng thời đối với các sản phẩm và các thành phần

2.10 Hợp nhất sớm và thường xuyên

2.11 Đảm bảo khả năng tái sản xuất phần mềm

- **Yêu cầu SV chuẩn bị:**

+ Bài tập: Nêu phương án nhóm các sản phẩm thành các thành phần (component), và nhóm các component thành các hệ thống con trong hệ thống Wheels.

+ Câu hỏi ôn tập:

- Quản lý cấu hình phần mềm là gì?

- Liệt kê các cách thức quản lý cấu hình phần mềm?
 - Tại sao cần nhóm các sản phẩm của một phiên bản thành các component?
 - Nêu những khả năng của các công cụ quản lý cấu hình phần mềm có thể hỗ trợ việc thay đổi đồng thời đối với các sản phẩm?
 - Nêu vai trò của các công cụ quản lý cấu hình phần mềm?
- + Ôn tập thi kết thúc môn