

Bài tập tuần 03

Giới thiệu về OOAD & Lập trình GUI

Mục tiêu

- Thực hiện các bài tập (câu hỏi) về nội dung OOAD
- Phân tích các yêu cầu chức năng cho bài toán (case study), Phân rã các chức năng đã xác định thành các nhiệm vụ (task) để xây dựng một kế hoạch dự án phần mềm đơn giản
- Lập trình với giao diện đồ hoạ người dùng:
 - o Làm quen với mô hình MVC các thành phần giao diện người dùng
 - o Thiết kế giao diện GUI trên công cụ Netbeans
 - o Viết mã xử lý sự kiện

Đánh giá

- Hoàn thành các bài tập về nội dung OOAD
- Xác định các yêu cầu chức năng chính cho bài toán (case study) xây dựng được biểu đồ phân cấp chức năng và liệt kê một số yêu cầu phi chức năng
- Hoàn thành bài thực hành lập trình GUI

Phần I:

Bài 1.1

Hãy so sánh mô hình thác nước (Waterfall modell) và phương pháp Scrum:

	Mô hình thác nước (Waterfall modell)	Phương pháp Scrum
Đặc điểm chính		
Đặc điểm các pha phát triển		
Kích thước nhóm phát triển (team size)		
Phong cách quản lý (management style)		

Quan điểm về sự thay đổi trong dự án		
Vấn đề xây dựng tài liệu		
Đối phó với các nguy cơ / rủi ro (risk)		
Tình huống áp dụng phù hợp		

Bài 1.2

- Chúng ta thấy rằng các hệ thống phần mềm lớn, phức tạp thường được phát triển bởi rất nhiều cá nhân, rất ít người có được bức tranh toàn cảnh về toàn bộ dự án. Vậy, đối với một người làm công, tham gia vào một dự án mà không biết về toàn bộ chức năng của dự án đó thì có hợp lý không? Vì sao?
- Nếu không áp dụng các mô hình vòng đời phần mềm thì có phát triển được phần mềm không? Tại sao?
- Trong phương pháp Agile, việc luôn có đại diện của khách hàng trong nhóm phát triển thì có ưu điểm gì?

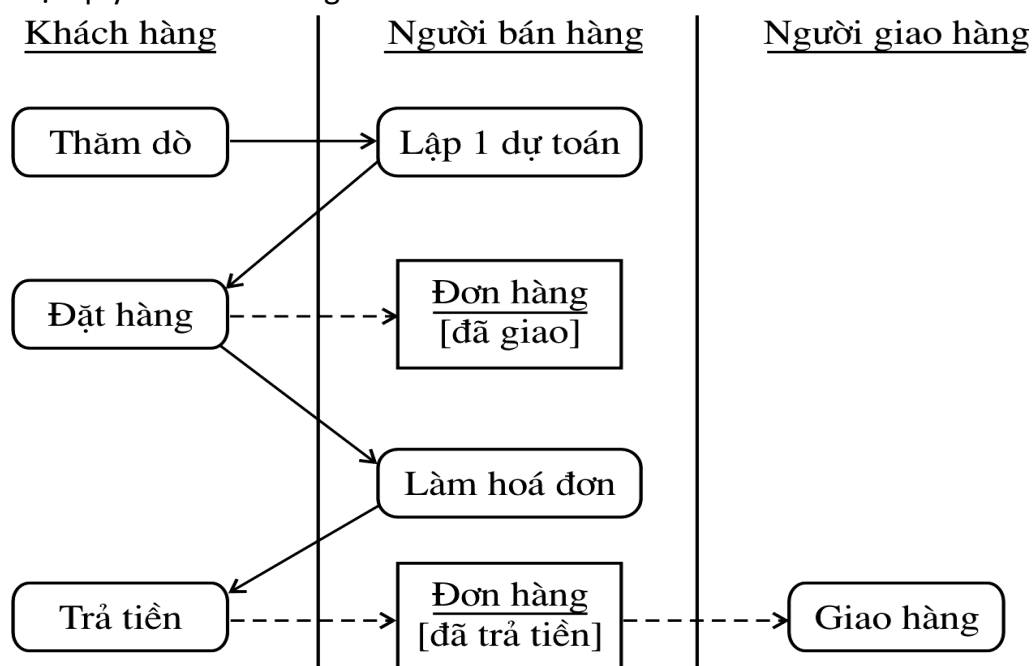
Phần II: Phân tích các yêu cầu chức năng cho bài toán (case study)

- Tiếp theo kết quả của bài tập tuần 02, các nhóm đã xác định được thông tin cơ bản: đầu vào (input), các xử lý chính (process) và kết quả (output).

- Thường thì dự án sẽ được khởi đầu bằng một bước **Nghiên cứu sơ bộ** nhằm **tìm hiểu môi trường nghiệp vụ của hệ thống tương lai**. Trong môi trường đó thì người, thiết bị, máy tính kết hợp với nhau hoạt động theo những **quy trình nghiệp vụ** nhất định. **Quy trình nghiệp vụ thường được mô tả bằng các biểu đồ hoạt động**.
- **Biểu đồ hoạt động (Activity Diagram)** là biểu đồ mô tả một nội dung hoạt động, theo các luồng đi từ việc này sang việc khác.
 - Nó không những được dùng để diễn tả một quy trình nghiệp vụ mà còn có thể dùng để diễn tả logic của một ca sử dụng, một kịch bản, một nhóm ca sử dụng, hay một thao tác phức tạp.

- Có thể nói biểu đồ hoạt động là mô hình UML tương đương với sơ đồ khối hoặc với biểu đồ luồng dữ liệu trong các phương pháp phân tích và thiết kế cũ.

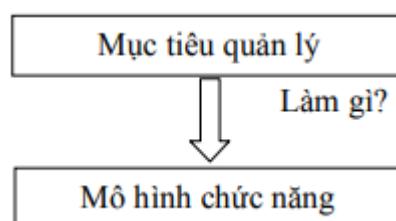
Ví dụ: một quy trình bán hàng



Yêu cầu: Hãy xây dựng biểu đồ hoạt động mô tả nghiệp vụ của bài toán (case study) mà nhóm đã lựa chọn.

Biểu đồ phân cấp chức năng (BFD - Bussiness Function Diagram): phân rã có thứ bậc đơn giản các chức năng của hệ thống thành các chức năng nhỏ hơn, cuối cùng thu được một cây chức năng.

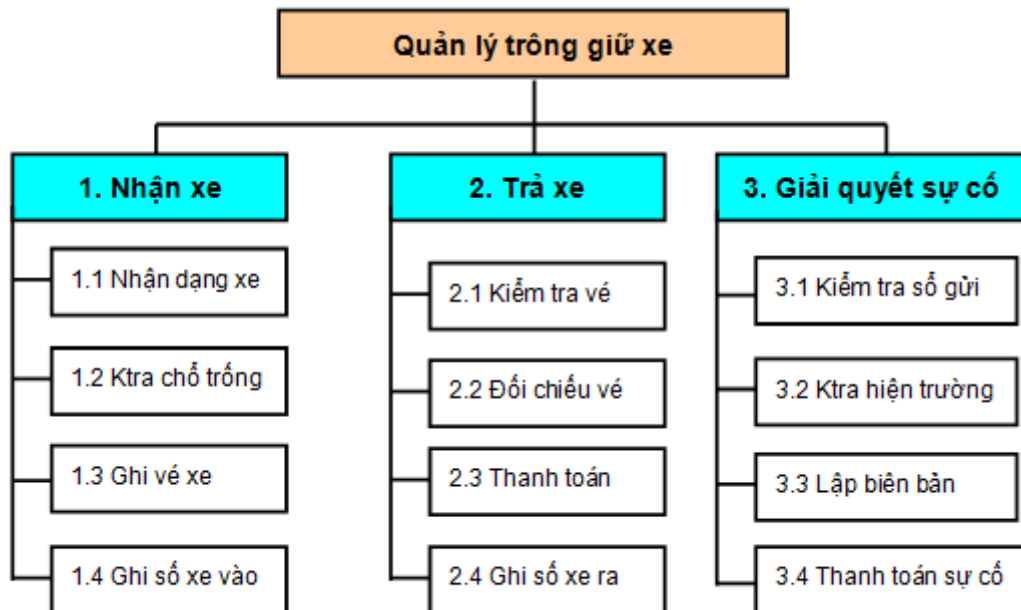
- Biểu đồ này thể hiện các đầu việc mà hệ thống cần thực hiện để đạt được mục tiêu quản lý (quản lý cái gì?), xây dựng dựa trên cơ sở của bước khảo sát và phân tích cơ bản trước đó.



- Các ký hiệu chính:

- Ký hiệu chức năng là một hình chữ nhật bên trong là tên chức năng, Tên chức năng là động từ - bổ ngữ và động từ nên ở dạng thức mệnh lệnh. Các chức năng cần đánh số thể hiện sự phân cấp.
- Liên kết các chức năng là đường thẳng

- Ví dụ:



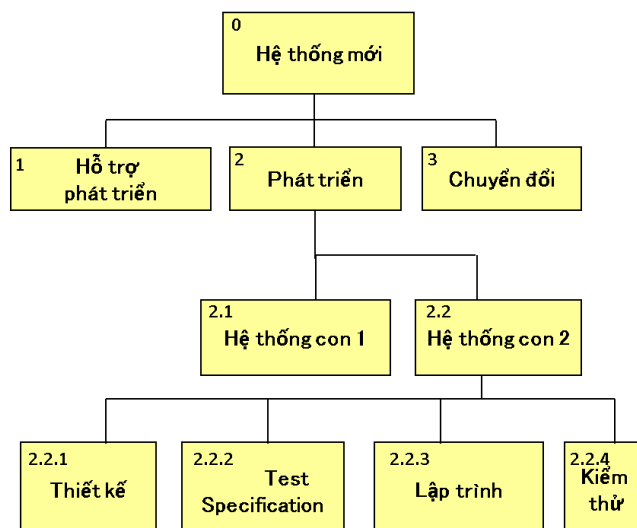
Mô tả các chức năng trong biểu đồ:

Tên chức năng	Mô tả	Đánh giá khả năng thực hiện (nhân lực, thời gian, công nghệ, môi trường)

Yêu cầu: Hãy xây dựng biểu đồ phân cấp chức năng cho bài toán (case study) mà nhóm đã lựa chọn. Mô tả các chức năng đó.

- WBS là một phân rã phân cấp (hierarchical decomposition) của toàn bộ phạm vi công việc sẽ được nhóm dự án thực hiện để hoàn thành các mục tiêu của dự án và tạo ra các sản phẩm bàn giao được yêu cầu.
- Các thành phần phân rã ở mức cuối cùng – mức lá nên thoả mãn các tiêu chí:
 - Tình trạng / tính hoàn tất của công việc có thể đo được hoặc có sản phẩm cụ thể
 - Thời gian, tài nguyên / chi phí có thể ước lượng được
 - Thời gian hoàn thành công việc trong giới hạn
 - Công việc được phân công độc lập (nghĩa là công việc không bị ngừng giữa chừng để chờ kết quả của công việc khác)

- Ví dụ:



- a) **Yêu cầu:** Nhóm sinh viên thảo luận và xây dựng WBS đối với dự án phát triển phần mềm trong bài tập môn học. Có thể phân tích theo các chức năng nghiệp vụ hoặc theo các pha trong quá trình phát triển.

Trên cơ sở biểu đồ WBS phía trên hãy xây dựng một bản kế hoạch đơn giản như sau:

Công việc	Thời gian (số giờ làm việc)	Số người
1. Công việc 1 1.1 Công việc 1.1 1.2 Công việc 1.2 ...		
2. Công việc 2 2.1 Công việc 2.1 2.2 Công việc 2.2 ...		
...		

- b) **Yêu cầu:** Xây dựng một bảng quản lý đơn giản các rủi ro với dự án phát triển phần mềm trong bài tập môn học.

Ví dụ:

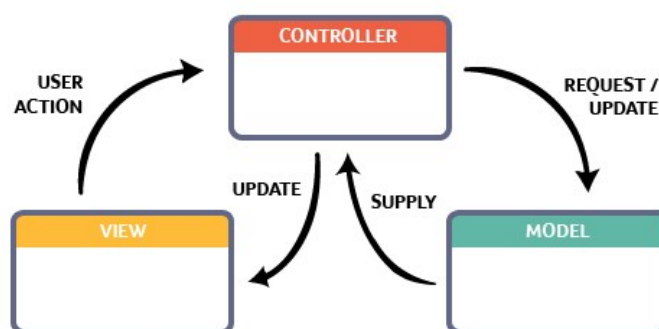
Công việc / Hoạt động	Xác định rủi ro			Quản lý rủi ro	
	Mối nguy	Rủi ro	Mức độ	Chiến lược	Biện pháp
Mua hàng	Hàng bị hư hại khi vận chuyển	Không có nguyên liệu sản xuất	Trung bình	Giảm thiểu	Xây dựng tồn kho tối thiểu

(Mức độ có thể đánh giá theo: thấp / trung bình / cao)

(Chiến lược có thể lựa chọn: Giảm thiểu rủi ro / Chuyển rủi ro / Loại bỏ rủi ro)

Phần III: Thực hành lập trình giao diện đồ họa người dùng GUI

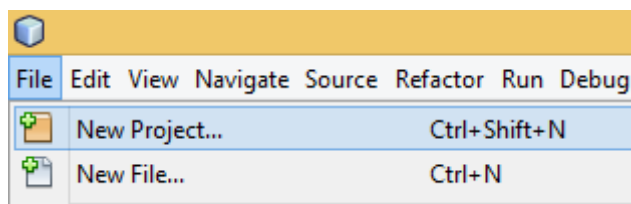
Background: Model – View – Controller (MVC) là một khuôn mẫu kiến trúc phần mềm. Mẫu thiết kế này cho phép phân tách giữa logic ứng dụng và giao diện người dùng. MVC giúp cho người phát triển phần mềm cô lập các xử lý nghiệp vụ và giao diện người dùng một cách rõ ràng hơn. Phần mềm phát triển theo mẫu MVC tạo nhiều thuận lợi cho việc bảo trì.



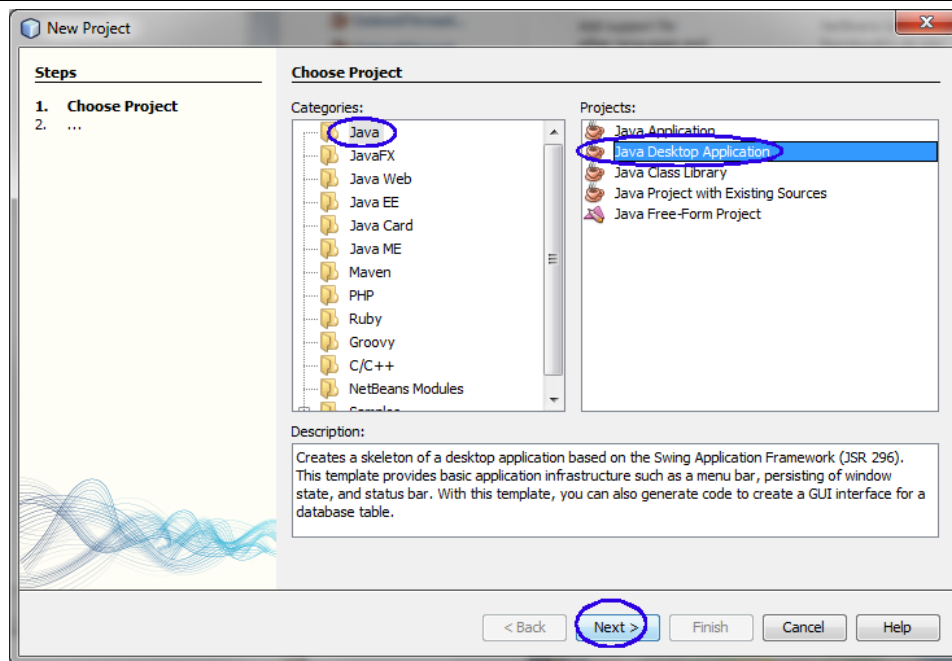
MVC chia một ứng dụng thành ba phần tương tác được với nhau để tách biệt giữa cách thức mà thông tin được xử lý nội hàm và phần thông tin được trình bày và tiếp nhận từ phía người dùng:

- **model** thành phần dữ liệu của chương trình
- **view** bao gồm các thành phần của giao diện người dùng
- **controller** quản lý sự trao đổi giữa dữ liệu và các xử lý nghiệp vụ trong các thao tác liên quan đến mô hình

Bước 1: Khởi động Netbeans -> File -> New Project ...

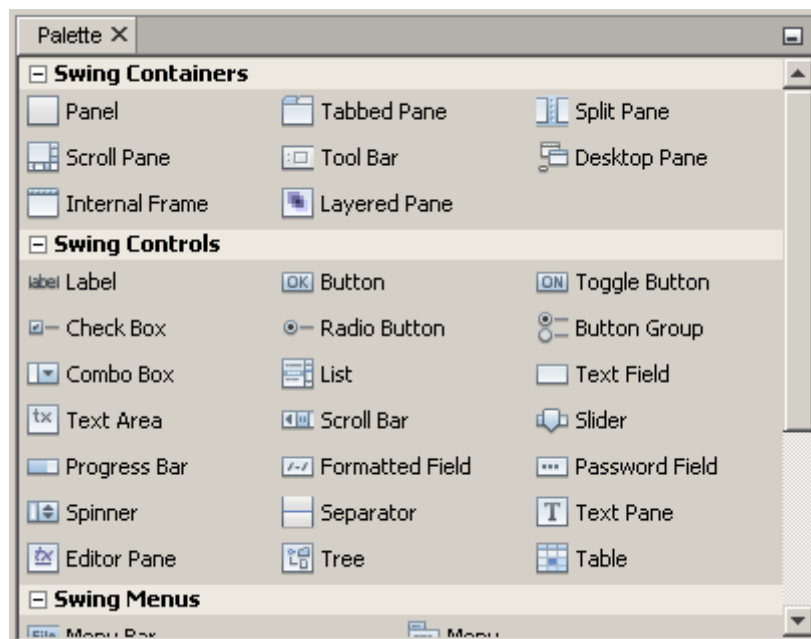


- Chọn **Java** trong mục **Categories** và **Java Desktop**
- Nhấn **Next**

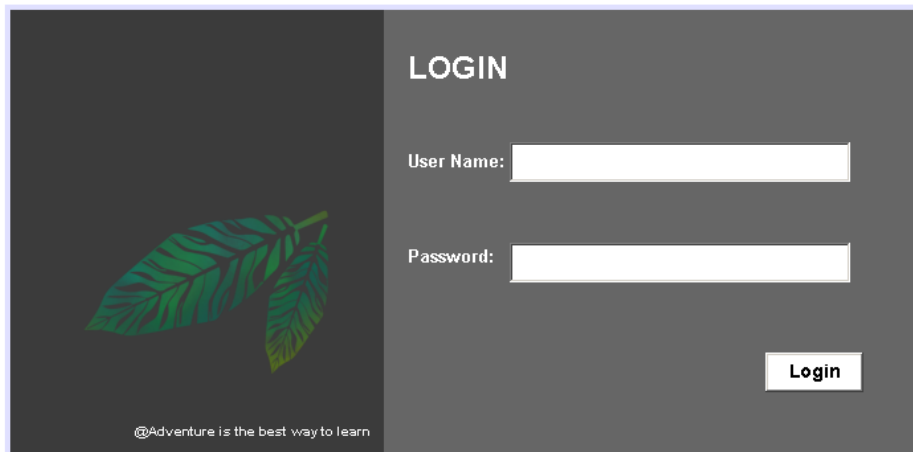


Bước 2: Tạo JFrame Form

- Nhấn chuột phải **Project** -> chọn **New** -> chọn **JFrame Form** -> nhập **LoginUI** tại **Class Name** -> chọn **Finish**
- Netbeans cung cấp các điều khiển UI trong cửa sổ **Palette**, kéo thả các điều khiển này vào khung thiết kế để xây dựng giao diện cho ứng dụng



- Trong bài tập này chúng ta sẽ thiết kế một form đăng nhập đơn giản:

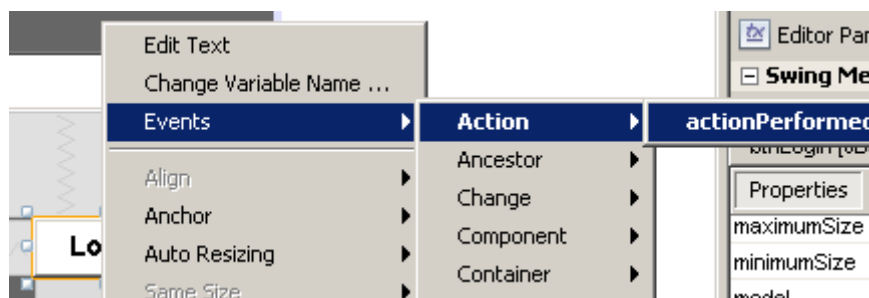


- Chú ý đặt tên các điều khiển cho phù hợp để viết mã xử lý dữ liệu trong form.
Trong giao diện này có 3 điều khiển: 2 ô textbox và một button.

txtUserName, txtPasswd và btnLogin

Bước 3: Đăng ký sự kiện

- Nhấn chuột phải vào nút **Login** -> chọn **Events** -> chọn **Action** -> chọn **actionPerformed**



- Viết code để xử lý sự kiện

```
private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {  
    this.login();  
}
```

- Xây dựng phương thức **login()** kết nối cơ sở dữ liệu và kiểm tra đăng nhập


```

private void login() {
    //Lấy thông tin nhập vào trong hai ô textbox
    String userName = txbUserName.getText();
    String password = String.valueOf(txbPasswd.getPassword());
    try {
        //Viết code kết nối cơ sở dữ liệu và kiểm tra đăng nhập
        if (/*Đăng nhập thành công*/) {
            //Hiện thị thông báo đăng nhập thành công
        } else {
            JOptionPane.showMessageDialog(rootPane, "Sai thông tin đăng nhập",
                "Warning", JOptionPane.WARNING_MESSAGE);
        }
    } catch (SQLException | ClassNotFoundException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(null, "Có lỗi xảy ra!! Vui lòng kiểm tra lại!",
            "Warning!!", JOptionPane.ERROR_MESSAGE);
    }
}

```

- Bổ sung thêm xử lý khi nhấn phím Enter cũng kiểm tra đăng nhập:

```

// xử lý sự kiện nhấn enter
private void keyListener(JTextField jtf) {
    jtf.addKeyListener(new KeyAdapter() {
        @Override
        public void keyPressed(KeyEvent e) {
            // nếu keycode == 10 ~ enter
            if (e.getKeyCode() == 10) {
                login();
            }
        }
    });
}

```

Chạy thử chương trình và kiểm tra kết quả.

Nội dung bài tập tự làm

- Thực hành xây dựng chương trình java với giao diện đồ họa: Liệt kê danh sách người dùng sau khi đăng nhập thành công:



Id	First Name	Last Name	Age
1	FNA	LNA	10
2	FNB	LNB	20
3	FNC	LNC	30
4	FND	LND	40
5	FNE	LNE	50
6	FNF	LNH	60

HẾT