

Hình ảnh minh họa

LỜI NÓI ĐẦU

Ngày nay, với sự phát triển không ngừng của khoa học, công nghệ; con người đã và đang từng bước hòa nhập, phát triển trên đà Công nghiệp hóa – Hiện đại hóa. Đi đôi với sự phát triển về kinh tế, chính trị là sự phát triển không ngừng của công nghệ. Với thời đại 4.0 hiện nay, con người từng bước số hóa công cuộc, cách thức sử dụng và tiếp cận tri thức. Ngôn ngữ lập trình đang ngày một lớn mạnh vì nó là công cụ chính trong việc phát triển công nghệ hiện đại ngày nay. Và Java, một trong những ngôn ngữ được tạo ra với tiêu chí “Viết một lần, thực thi khắp nơi” (“Write Once, Run Anywhere”). Với việc thích nghi và có khả năng chạy dưới mọi nền tảng (platform) khác nhau, Java đã trở thành một ngôn ngữ rất phổ biến và mạnh mẽ về bảo mật, phù hợp cho việc phát triển phần mềm, ứng dụng máy tính cũng như di động, ...

Với mục tiêu được tự tìm hiểu, trau dồi khả năng làm việc nhóm, tự học, sáng tạo đồng thời tự tay tạo ra được một sản phẩm hoàn chỉnh. Qua thời gian làm việc và học tập cùng nhau, nhóm chúng em đã hoàn thành trong việc tạo nên ứng dụng “*Trợ lý đa năng i.Sekai*”, đảm bảo được các yêu cầu và mục tiêu Đồ án môn học hướng đến.

NHÓM SINH VIÊN THỰC HIỆN (Nhóm 1)

Phan Thanh Tùng	1613240
Nguyễn Hồng Sỹ Nguyên	1613124
Tổng Hải Đăng	1613026



LỜI CẢM ƠN

Sau hơn 6 tuần học tập, nghiên cứu và làm việc cùng nhau, nhóm 1 chúng em đã học được rất nhiều điều bổ ích, cần thiết cho một sinh viên chuyên ngành Vật lý Tin học. Khoảng thời gian tuy không quá dài, nhưng đủ để chúng em hiểu như thế nào là làm việc nhóm, như thế nào là tự học, tự tìm hiểu và các kỹ năng cần thiết của một con người năng động để hòa vào lối sống hội nhập, hiện đại. Đó không chỉ là những gì trên giấy mực, mà cả là những gì thiết thực nhất trong cuộc sống như việc: làm thế nào sắp xếp thời gian hợp lý? Kỹ năng nói, thuyết phục người khác, kỹ năng tự học, tự tìm tòi, áp dụng các kiến thức cơ bản để biến nó thành sản phẩm thực tiễn như thế nào? Đây chắc chắn sẽ là hành trang quý báu để chúng em có được nhiều kỹ năng làm việc tốt hơn, năng động hơn khi vào đời.

Qua quãng thời gian đó, chúng em rất biết ơn và gửi lời cảm ơn chân thành đến Bộ môn Vật lý Tin học, luôn là nơi sẵn sàng mở cửa, tạo điều kiện thuận lợi cho chúng em được học tập, nghiên cứu dễ dàng hơn. Chúng em xin gửi lời cảm ơn đến Cô Nguyễn Anh Thư, giảng viên môn Lập trình Java, là người trang bị cho chúng em đầy đủ kiến thức cần thiết, tận tâm trên những giờ trên lớp và sẵn lòng giúp đỡ khi chúng em gặp khó khăn trong quá trình làm đồ án.

Ngoài ra, chúng em cũng rất biết ơn bạn bè cùng lớp, các anh chị khóa K15 luôn kề vai, sát cánh để giúp đỡ khi chúng em gặp khó khăn, luôn đưa ra những lời khuyên quý báu, chia sẻ những kinh nghiệm cần thiết, những lời động viên chân thành để chúng em hoàn thành được sản phẩm của Đồ án này.

Chúng em xin chân thành cảm ơn!

TP.Hồ Chí Minh, tháng 05 năm 2019

Nhóm 1

[illegible]



MỤC LỤC

Trang

LỜI MỞ ĐẦU	i
LỜI CẢM ƠN.....	ii
NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN	iii
MỤC LỤC	iv
DANH MỤC CÁC HÌNH ẢNH.....	vi
DANH MỤC CÁC SƠ ĐỒ	viii
CHƯƠNG I. GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI.....	9
1.1 Tổng quan về ý tưởng đề tài	9
1.2 Nhiệm vụ đề tài	10
1.3 Các thành phần – chức năng cơ bản	10
1.4 Phân chia công việc trong nhóm.....	11
CHƯƠNG II. LÝ THUYẾT.....	12
2.1 Giới thiệu về Java	12
2.2 Giới thiệu về IDE lập trình Java – Netbeans	14
2.3 Giới thiệu về Java Swing	15
2.4 Các thành phần cơ bản sử dụng và cách thức tạo lập	21
2.5 Cơ sở dữ liệu và lưu trữ thông tin.....	32
CHƯƠNG III. THIẾT KẾ VÀ THỰC HIỆN	38
3.1 Kế hoạch thiết kế và thực hiện.....	38
3.2 Các công cụ hỗ trợ khác.....	39
3.3 Giới thiệu các thành phần chính	41
3.4 Giới thiệu các thành phần phụ	44
CHƯƠNG IV. CÁCH THỨC HOẠT ĐỘNG – CHỨC NĂNG VÀ HƯỚNG DẪN SỬ DỤNG.....	46

**Trang**

4.1	Cách thức hoạt động	46
4.2	Các chức năng cơ bản chính	46
4.3	Các chức năng phụ - công cụ hỗ trợ khác.....	47
4.4	Hướng dẫn sử dụng chung	47
CHƯƠNG V. KẾT QUẢ THỰC HIỆN.....		49
CHƯƠNG VI: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....		50
6.1	Kết luận.....	50
6.2	Ưu điểm – Hạn chế	50
6.3	Hướng phát triển	50
TÀI LIỆU THAM KHẢO		lii



DANH MỤC CÁC HÌNH ẢNH

Hình 1.1 Trợ lý ảo trên thiết bị di động iOS – Siri.....	9
Hình 1.2 Giao diện cơ bản của “Trợ lý đa năng i.Sekai”.....	10
Hình 1.3 “i.Sekai” khi thu nhỏ ở dạng widget	10
Hình 2.1 James Gosling – cha đẻ ngôn ngữ lập trình Java.....	12
Hình 2.2 Logo của IDE Netbeans.....	14
Hình 2.3 Hình minh họa cho Java Swing	15
Hình 2.4 GUI Framework của Java.....	16
Hình 2.5 Ví dụ về việc tạo lập JComponent.....	21
Hình 2.6 Các thể hiện JFrame rỗng trên nền tảng Unix, Mac và Windows.....	23
Hình 2.7 Các hình ảnh khác về JFrame trên MacOS và Windows	24
Hình 2.8 Hình ảnh về ví dụ của JPanel	24
Hình 2.9 Mô tả cấu trúc sắp xếp để tạo nên một khung giao diện ứng dụng	26
Hình 2.10 Ví dụ về việc tạo lập nút bấm từ JButton	27
Hình 2.11 Ví dụ về việc tạo lập thanh Menu chức năng từ các lớp Menu của Java Swings	30
Hình 2.12 Hình ảnh Demo giao diện “i.Sekai”	31
Hình 2.13 Logo của Apache Derby	32
Hình 2.14 Kết quả sau khi sử dụng Apache Derby cho việc tạo lập CSDL phản hồi...34	
Hình 2.15 Ví dụ về việc sử dụng file JSON để thêm các cặp giá trị “key” và “value” 36	
Hình 2.16 Kết quả sau khi sử dụng JSON cho việc tạo lập chức năng cho ứng dụng ..37	
Hình 3.1 Thực hiện việc lên kế hoạch chi tiết – phân công chi tiết, rõ ràng.....	38



Hình 3.2 Logo của Github40

Hình 3.3 Hình ảnh mô tả nút nhấn chức năng gợi ý Black Hole Button.....45



DANH MỤC CÁC SƠ ĐỒ

Sơ đồ 2.1 Sơ đồ cho thấy quá trình biên dịch đa nền tảng của Java	12
Sơ đồ 2.2 Sơ đồ mô tả quá trình biên dịch và tiêu chí của ngôn ngữ Java “Write once, run anywhere”	13
Sơ đồ 2.3 Sơ đồ Cây phân lớp của Java Swing	20
Sơ đồ 2.4 Sơ đồ lớp của Container “gốc” trong swing	22
Sơ đồ 2.5 Sơ đồ lớp của JLabel	25
Sơ đồ 2.6 Sơ đồ lớp của JButton	27
Sơ đồ 2.7 Sơ đồ lớp của JMenu	28
Sơ đồ 2.8 Sơ đồ lớp của JMenuItem	29
Sơ đồ 3.1 Sơ đồ mô tả quá trình hoạt động, thực hiện và làm việc của nhóm.	39
Sơ đồ 3.2 Mô hình tổ chức dữ liệu phân tán trên Git.	40
Sơ đồ 3.3 Các thành phần chính cơ bản của “i.Sekai”.	41
Sơ đồ 3.4 Sơ đồ mô tả hệ thống thành phần của Menu Taskbar	44



CHƯƠNG I

GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI



1.1. Tổng quan về ý tưởng đề tài:



Hình 1.1 Trợ lý ảo trên thiết bị di động iOS – Siri.

Ngày nay, khi công nghệ phát triển ngày một hiện đại, không khó để chúng ta bắt gặp trong đời sống một trợ lý ảo trên các nền tảng thân thiện, thường dùng trên android hoặc iOS. Windows có Cortana; Google có Google Assistant; iOS có Siri; Amazone có Alexa; ... Nắm bắt được xu thế đó, đồng thời cũng là mong muốn

được biến ước mơ thành hiện thực cùng với việc tìm hiểu thêm về cách thức tạo ra một ứng dụng bằng ngôn ngữ Java trên máy tính, nhóm chúng em đã hướng đến ý tưởng tạo ra một trợ lý ảo trên chính nền tảng Windows. “i.Sekai” là ứng dụng trợ lý ảo đa năng được nhóm viết bằng chính ngôn ngữ Java cùng với các thư viện liên quan, bằng IDE lập trình Java Netbeans và những kiến thức về Java được học tại lớp cũng như tự tìm hiểu thông qua sách vở, Internet, ... “i.Sekai” đã được nhóm chúng em cho ra đời với giao diện thân thiện, nhỏ gọn, dễ dàng sử dụng như trên chính chiếc smartphone của mỗi người nhưng nay được hiện hữu trực tiếp trên màn hình làm việc của hệ điều hành Windows. Với “i.Sekai”, người dùng có thể tùy biến tạo ra một người bạn, một trợ lý thân thiện với tên và hình ảnh đại diện được tùy biến sẵn, ứng dụng sẽ giúp người dùng có những trải nghiệm mới lạ, tiện lợi trong cách sử dụng máy tính, đặc biệt hơn, bằng cách khai thác dữ liệu từ Internet, trải nghiệm khả năng khai thác dữ liệu từ việc lướt web cũng sẽ trở nên dễ dàng, tiện dụng, mới lạ hơn chỉ với vài câu lệnh đơn giản. Đặc biệt hơn, bằng sự kết hợp khéo léo các ứng dụng hỗ trợ học tập khác, “i.Sekai” giúp hỗ trợ tối đa cho các bạn sinh viên khi sử dụng cần lấy tài liệu hoặc thể hiện các vấn đề bài toán về “Cấu trúc dữ liệu” được tương quan, thú vị, đặc sắc hơn. Khả năng phát triển dành cho ứng dụng lớn, tùy thuộc vào sự sáng tạo của nhà phát triển, ứng dụng dễ dàng thêm được các chức năng tiện nghi, tối ưu khác như: điều khiển bằng giọng nói, dùng “i.Sekai” để điều khiển thiết bị trong nhà, ...

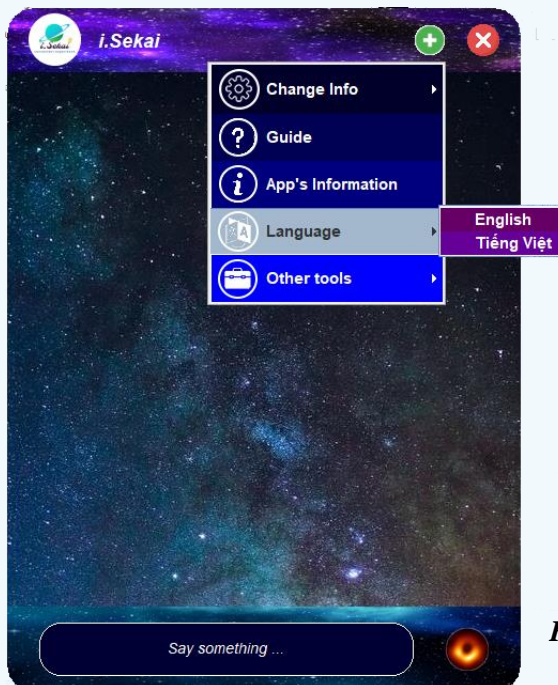


1.2. Nhiệm vụ đề tài:

- Tạo nên được giao diện làm việc cho ứng dụng, các thành tố, đối tượng của cửa sổ làm việc ứng dụng.
- Tạo được các chức năng cơ bản gắn với việc sử dụng máy tính cho ứng dụng.
- Tạo được các chức năng tùy biến khả năng khai thác dữ liệu từ Internet.
- Hoàn thiện ứng dụng với các thành phần, chức năng khác.
- Tối ưu hóa khả năng làm việc của ứng dụng.

1.3. Các thành phần – chức năng cơ bản:

Việc phát triển “Trợ lý đa năng – i.Sekai” được dựa trên ý tưởng từ những trợ lý nổi tiếng trên thiết bị di động hiện tại, với việc tạo nên một ứng dụng gọn nhẹ, thân thiện, dễ sử dụng. “i.Sekai” có giao diện tương tự như ứng dụng trò chuyện “Messenger” trên thiết bị hệ điều hành Android với việc có thể thu nhỏ cửa sổ điều khiển thành “icon bong bóng kéo thả” (đối với người dùng Windows gọi là **widget**) tự do trên màn hình làm việc. Các thành phần chính của ứng dụng có thể kể đến như: vùng hội thoại (được thiết kế nhỏ gọn, linh hoạt cho nhiều câu thoại như “Siri”), vùng gõ câu lệnh, menu taskbar công cụ, khung gợi ý, ...



Hình 1.2 Giao diện cơ bản của “Trợ lý đa năng i.Sekai”.



Hình 1.3 “i.Sekai” khi thu nhỏ ở dạng widget



Một số chức năng cơ bản mà nhóm phát triển cho ứng dụng như:

- Tối ưu hóa khả năng truy cập, sử dụng máy tính bằng câu lệnh ngắn gọn.
- Hệ thống gợi ý ngay khi gõ các kí tự tìm kiếm đầu tiên.
- Khả năng khai thác dữ liệu từ Internet nhanh, gọn.
- Tạo cho người dùng trải nghiệm sử dụng đa phương tiện mới lạ.
- Hệ thống song ngữ (Anh – Việt) linh hoạt cho nhiều đối tượng sử dụng.
- Tr� lý ảo có cơ sở dữ liệu phản hồi một số cuộc trò chuyện đơn giản.
- Được gắn thêm chức năng công cụ tìm tài liệu học tập và hỗ trợ tương quan trong việc học “Cấu trúc dữ liệu và Giải thuật”.

1.4. Phân chia công việc trong nhóm:

- **Phan Thanh Tùng:** Điều hành - điều phối; quản lý kế hoạch; lên ý tưởng; coder phụ (các phần phụ); design chính (UI).
- **Nguyễn Hồng Sỹ Nguyên:** Coder chính (các phần chính); người lên ý tưởng; design phụ (UI).
- **Tổng Hải Đăng:** Tester (đề xuất ý kiến - UX); tìm kiếm tài liệu cần thiết; tạo lập cơ sở dữ liệu.
- Các nội dung khác:
 - + Văn bản: Phan Thanh Tùng, Nguyễn Hồng Sỹ Nguyên.
 - + Kiểm tra: Phan Thanh Tùng, Nguyễn Hồng Sỹ Nguyên.
 - + Báo cáo: Phan Thanh Tùng, Nguyễn Hồng Sỹ Nguyên, Tổng Hải Đăng.



CHƯƠNG II LÝ THUYẾT

2.1. Giới thiệu về Java:

Java là ngôn ngữ lập trình thuần hướng đối tượng được phát triển bởi **Sun Microsystems** (Sau này được **Oracle** mua lại).

Java được khởi đầu bởi **James Gosling** và đồng nghiệp, ban đầu được gọi là **Oak** (nghĩa là cây sồi do ngoài cơ quan ông **Gosling** trồng nhiều cây này).

Dự định ban đầu của **Java** là thay cho **C++** với những tính năng giống **Objective-C** (ngôn ngữ phát triển của **Apple**).

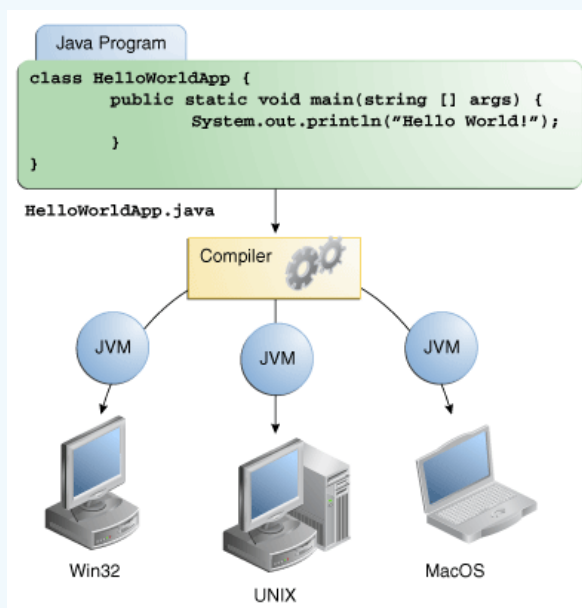


Hình 2.1 James Gosling – cha đẻ ngôn ngữ lập trình Java

Với tiêu chí “**Viết một lần, thực thi mọi nơi**” (“**Write Once, Run Anywhere**”). Chương trình phần mềm bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua môi trường thực thi với điều kiện môi trường thực thi thích hợp trên nền tảng hệ điều hành đó (ví dụ như **Sun Solaris**, **Linux**, **Mac OS**, **FreeBSD** và **Windows**).

Ngôn ngữ lập trình **Java** có những đặc trưng cơ bản sau:

- Là ngôn ngữ **hướng đối tượng**.
- Chạy bằng máy ảo **Java**. Chương trình muốn thực thi phải biên dịch ra mã máy, mà mã máy mỗi hệ điều hành là khác nhau như: Windows biên dịch dưới dạng file có đuôi **.EXE** còn Linux có dạng đuôi **.ELF**. Việc Java có thể chạy mọi hệ điều hành là do nhà phát triển **Sun Microsystems** phát triển máy ảo (JVM) chịu trách nhiệm việc này.



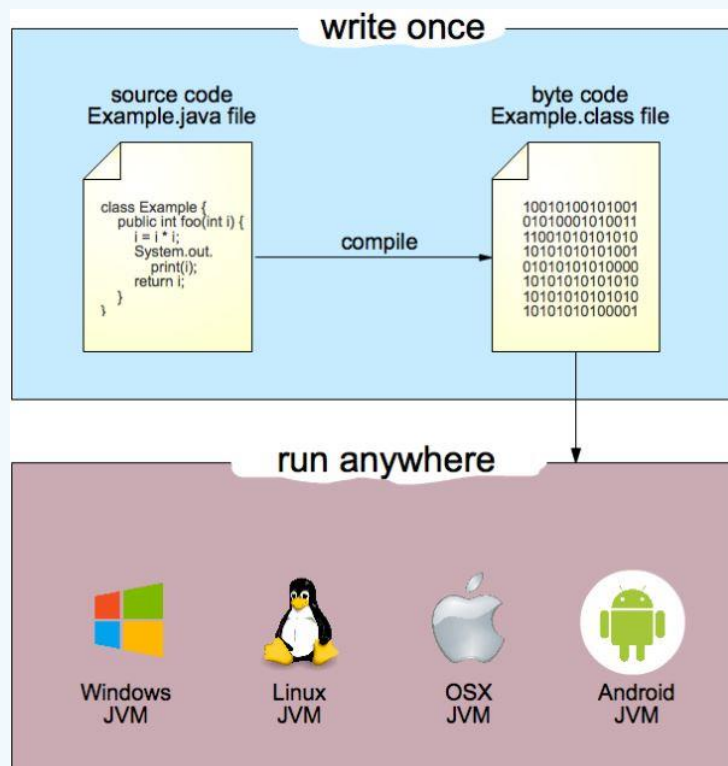
Sơ đồ 2.1 Sơ đồ cho thấy quá trình biên dịch đa nền tảng của Java



- **Đa nhiệm – Đa luồng:** Java hỗ trợ lập trình đa nhiệm, đa luồng cho phép chạy nhiều tiến trình chạy song song trong một thời điểm và tương tác lẫn nhau.
- Java **bỏ** đa kế thừa trong C++ thay bằng **sử dụng Interface**.

Những đặc điểm nổi bật có trong Java:

- **Độc lập phần cứng và hệ điều hành:** Java được thiết kế để biên dịch code thành **bytecode** và sẽ chạy trên môi trường thực thi. Nên chương trình Java có thể chạy trên nhiều thiết bị, nhiều hệ điều hành khác nhau.
- **Mạnh mẽ:** quá trình cấp phát, giải phóng bộ nhớ thực hiện tự động. Không sử dụng con trỏ hoặc phép toán con trỏ.
- **Bảo mật:** Do Java phải biên dịch qua máy ảo Java (JVM) nên sẽ được bảo mật cao hơn khi mọi đối tượng phải qua JVM mới đến hệ điều hành.
- **Phân tán:** Java hỗ trợ lập trình cho hệ thống phân tán như client-server, RMI... bằng Java Web, UDP, TC.



Sơ đồ 2.2 Sơ đồ mô tả quá trình biên dịch và tiêu chí của ngôn ngữ Java “Write once, run anywhere”



2.2. Giới thiệu về IDE lập trình Java – Netbeans:

2.2.1. IDE là gì?

IDE là cụm từ viết tắt của Integrated Development Environment, là phần mềm cung cấp cho các lập trình viên một môi trường tích hợp bao gồm nhiều công cụ khác nhau như: chương trình viết mã lệnh hay code editor, chương trình sửa lỗi hay debugger, chương trình mô phỏng ứng dụng khi chạy thực tế hay simulator, ... Nói cách khác thì IDE là một phần mềm bao gồm những gói phần mềm khác giúp phát triển một ứng dụng phần mềm bất kì. Và để thực hiện đề tài “Trợ lý đa năng – i.Sekai”, nhóm đã chọn Netbeans IDE 8.2.

2.2.2. Giới thiệu về Netbeans IDE:

Netbeans IDE là một phần mềm tạo lập môi trường lập trình miễn phí được tạo ra bởi **Sun Microsystems**. Netbeans IDE được phát triển mới mục đích ban đầu chủ yếu để giúp các lập trình viên xây dựng ứng dụng Java, tuy nhiên sau đó nó đã được mở rộng để hỗ trợ cho việc sử dụng phát triển các ứng dụng PHP, C, C++ và HTML5 (bao gồm cả Javascript và CSS).



Hình 2.2 Logo của IDE Netbeans

NetBean IDE phù hợp cho các lập trình viên chuyên nghiệp và cả những người dùng bình thường yêu thích công việc lập trình. Nó cung cấp một công cụ đầy đủ tính năng để tạo ra các sản phẩm liên quan đến ứng dụng web, máy tính để bàn và thiết bị di động bằng các ngôn ngữ hiện đại nhất. Ưu điểm của phần mềm này là hỗ trợ nhiều ngôn ngữ khác nhau, nhiều chức năng tùy chỉnh, sửa lỗi khá tiện lợi.

NetBean IDE là môi trường phát triển tích hợp rất cần thiết cho các nhà phát triển phần mềm, phần mềm này hoạt động được trên hệ điều hành khác nhau như Windows, Mac, Linux, và Solaris; là một mã nguồn mở cung cấp các tính năng cần thiết nhất nhằm tạo ra các ứng dụng web, thiết bị di động, desktop. Nó là phiên bản dành cho các ngôn ngữ lập trình hiện đại như Python, Ruby, JavaScript, Groovy, C/C++ và PHP.

2.2.3. Các tính năng chính của Netbeans IDE:

- Hỗ trợ nhiều loại ngôn ngữ lập trình, chỉnh sửa mã nguồn thông minh.
- Giao diện trực quan, dễ thao tác và sử dụng.
- Là công cụ lập trình được cả phần mềm máy tính và phần mềm trên các thiết bị di động.
- Gỡ được lỗi mạng nội bộ và từ xa.
- Thử nghiệm tính năng xây dựng giao diện đồ họa.
- Tính năng QuickSearch (Tìm kiếm nhanh), tự động biên dịch, hỗ trợ các Framework cho website, trình ứng dụng máy chủ GlassFish và Cơ sở dữ liệu.

2.2.4. Sự khác biệt của Netbeans IDE 8.2 so với các phiên bản khác:

- ECMAScript 6 và hỗ trợ ECMAScript thử nghiệm 7.
- Nâng cao HTML5 / JavaScript.
- Hỗ trợ PHP 7, Docker.
- Cải tiến các công cụ soạn thảo Java và Profiler.
- Hỗ trợ fix lỗi nhanh hơn.
- Nâng cao ngôn ngữ lập trình C/C ++.

2.3. Giới thiệu về Java Swing:



Hình 2.3 Hình minh họa cho Java Swing

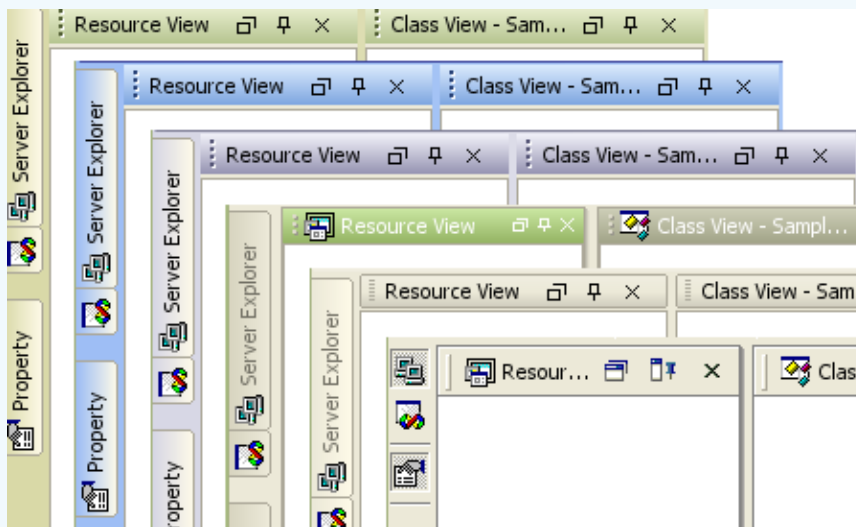
Một ứng dụng máy tính luôn không thể thiếu được thành phần quyết định người dùng có ưa chuộng nó hay không, chưa kể đến việc chức năng nó hay như thế nào thì thành phần về giao diện cũng vô cùng quan trọng. Cho đến thời điểm hiện tại, ngôn ngữ lập trình Java được phát triển rất nhiều thư viện hỗ trợ chức năng tạo ra một giao diện đẹp, bắt mắt cho một ứng dụng. Tuy nhiên, nhóm chúng em lựa chọn

Java Swing vì một phần nó là của Java Foundation Classes (JFC), được sử dụng để tạo ứng dụng cho Window-Based khá cơ bản, gần gũi, thiết thực với chương trình học và nhu cầu hiện tại.



2.3.1. Swing là gì?

Nếu truy cập vào trang chủ Java (<http://java.sun.com/>), ta sẽ nhận thấy Swing là một tập hợp các thành phần đồ họa có thể tùy chỉnh mà giao diện và hành vi xử lý (Look and Feel – L&F) có thể được quyết định tại thời điểm chạy chương trình. Tuy nhiên, trong thực tế Swing cung cấp nhiều hơn thế. Swing là bộ công cụ GUI (GUI toolkit) thế hệ kế tiếp mà Sun Microsystems tạo ra nhằm giúp lập trình viên phát triển các ứng dụng doanh nghiệp bằng ngôn ngữ Java. Phát triển phần mềm doanh nghiệp, lập trình viên có thể sử dụng Swing để tạo ra những ứng dụng Java quy mô lớn có chứa một mảng rộng những thành phần mạnh mẽ. Ngoài ra, bạn có thể dễ dàng mở rộng hoặc sửa đổi những thành phần này để kiểm soát giao diện và hành vi của chúng.



Hình 2.4 GUI Framework của Java

Swing không phải là một từ viết tắt. Swing đại diện cho sự lựa chọn hợp tác của các nhà thiết kế khi dự án được bắt tay thực hiện vào cuối năm 1996. Trên thực tế, Swing là một phần của tập hợp các sản phẩm Java có tên gọi là Java Foundation Class (JFC), JFC kết hợp nhiều tính năng của Netscape Foundation Internet Class (IFC) cũng như các khía cạnh thiết kế từ bộ phận Taligent của IBM và Lighthouse Design. Swing được phát triển tích cực từ giai đoạn beta của bản JDK (Bộ công cụ dành cho người phát triển ứng dụng Java – Java Development Kit) 1.1, khoảng mùa xuân năm 1997. Các API Swing được đưa vào bản beta vào nửa cuối năm 1997 và được phát hành tháng 3/1998. Tại thời điểm phát



hành, các thư viện Swing 1.0 chứa gần 250 lớp (class) và 80 giao diện (interface). Kể từ đó thư viện Swing vẫn tiếp tục tăng trưởng.

Mặc dù Swing được phát triển riêng biệt với lõi của JDK, nhưng để chạy được thì bản JDK tối thiểu là 1.1.5. Swing xây dựng dựa trên mô hình sự kiện (event model) của chuỗi phiên bản JDK 1.1; Ta không thể sử dụng các thư viện Swing với phiên bản JDK 1.0.2 cũ hơn. Ngoài ra, ta phải có một trình duyệt kích hoạt Java 1.1 để hỗ trợ các Swing applet. Bản phát hành Java 2 SDK 1.4 bao gồm nhiều lớp Swing cập nhật cùng với một vài tính năng mới. Swing được tích hợp đầy đủ vào cả công cụ của nhà phát triển và môi trường chạy của tất cả các phiên bản Java 2 (SDK 1.2 và phiên bản cao hơn), bao gồm cả Java Plug-In.

2.3.2. Tính năng bổ sung của Swing:

Một số tính năng phân biệt Swing với các thành phần AWT (Abstract Windows Toolkit) cũ hơn:

- Swing có nhiều thành phần mới, như bảng (table), cây (tree), thanh trượt (slider), spinner, thanh tiến trình (progress bar), internal frame và thành phần văn bản (text).
- Thành phần Swing hỗ trợ khả năng thay thế thành phần inset bằng một số lượng tùy ý các đường bao lồng nhau.
- Thành phần Swing có thể chứa các tooltip. Tooltip là hộp văn bản xuất hiện trong giây lát khi con trỏ chuột nằm bên trong vùng hoạt động của thành phần. Tooltip có thể được sử dụng để cung cấp thêm thông tin về thành phần chưa biết.
- Ta có thể tùy ý ràng buộc sự kiện bàn phím với thành phần, định nghĩa cách chúng phản ứng với các tổ hợp phím khác nhau trong những điều kiện nhất định
- Có thêm phần hỗ trợ gỡ lỗi cho việc render các thành phần Swing lightweight của riêng bạn.

2.3.3. Gói và lớp của Swing (Package and Class):

`javax.accessibility`

Chứa các lớp và giao diện có thể được sử dụng để cho phép công nghệ hỗ trợ tương tác với thành phần Swing. Công nghệ hỗ trợ bao gồm một loạt các mục, từ trình đọc văn



bản bằng âm thanh tới màn hình phóng đại. Mặc dù về mặt bản chất thì không phải là một phần của Swing, song các lớp hỗ trợ truy cập được sử dụng rộng rãi trên các thành phần của Swing.

`javax.swing`

Chứa các thành phần Swing cốt lõi, bao gồm hầu hết giao diện mô hình và lớp hỗ trợ.

`javax.swing.border`

Chứa các định nghĩa về lớp border trừu tượng cũng như tám border (đường bao) được định nghĩa trước. Đường bao không phải là thành phần; mà là những phần tử đồ họa đặc biệt mà Swing coi như các thuộc tính và đặt xung quanh các thành phần ở vùng không gian đệm bên ngoài thành phần (inset). Nếu muốn tạo ra đường bao riêng, bạn có thể tạo lớp con của một đường bao hiện có trong gói này hoặc lập trình tạo một lớp mới từ đầu.

`javax.swing.colorchooser`

Bao gồm hỗ trợ cho thành phần JColorChooser.

`javax.swing.event`

Định nghĩa một số listener và sự kiện mới mà các thành phần Swing dùng để giao tiếp thông tin không đồng bộ giữa các lớp. Để tạo ra sự kiện riêng, bạn có thể tạo lớp con của các sự kiện khác nhau trong gói này hoặc viết sự kiện của riêng bạn.

`javax.swing.filechooser`

Có hỗ trợ cho thành phần JFileChooser.

`javax.swing.plaf`

Định nghĩa những phần tử đặc trưng tạo nên L&F pluggable cho mỗi thành phần. Những gói con của gói này được sử dụng để render các L&F riêng lẻ cho mỗi thành phần trên cơ sở đa nền tảng (platform-by-platform). (Phần thực thi cụ thể của L&F trong Window và Motif nằm trong các gói con của `com.sun.java.swing.plaf`, và L&F của MacOS nằm trong gói `com.apple.mrj.swing`).

`javax.swing.table`



Cung cấp mô hình (model) và hiển thị (view) thành phần của bảng (table), cho phép ta bố trí và xếp đặt thông tin dưới dạng lưới giống như một bảng tính (spreadsheet). Bằng việc sử dụng các lớp ở cấp thấp hơn, ta có thể sửa đổi việc lựa chọn và hiển thị bảng, cũng như cách hiển thị thông tin ở mỗi dòng, mỗi ô của bảng.

`javax.swing.text`

Cung cấp các lớp và giao diện cốt lõi liên quan đến văn bản, hỗ trợ một thiết kế chung dạng tài liệu/hiển thị (document/view).

`javax.swing.text.html`

Được sử dụng đặc biệt trong việc đọc và định dạng văn bản HTML thông qua một trình chỉnh sửa phụ trợ (editor kit).

`javax.swing.text.html.parser`

Chứa các lớp hỗ trợ phân tích mã HTML.

`javax.swing.text.rtf`

Được sử dụng đặc biệt trong việc đọc và định dạng văn bản Rich Text Formal (RTF) thông qua một trình chỉnh sửa phụ trợ.

`javax.swing.tree`

Định nghĩa các mô hình và hiển thị cho thành phần cây phân cấp, thành phần này có thể biểu diễn một cấu trúc file hoặc chuỗi các thuộc tính.

`javax.swing.undo`

Chứa các chức năng cần thiết để cài đặt các chứa năng “Undo” (hoàn tác).

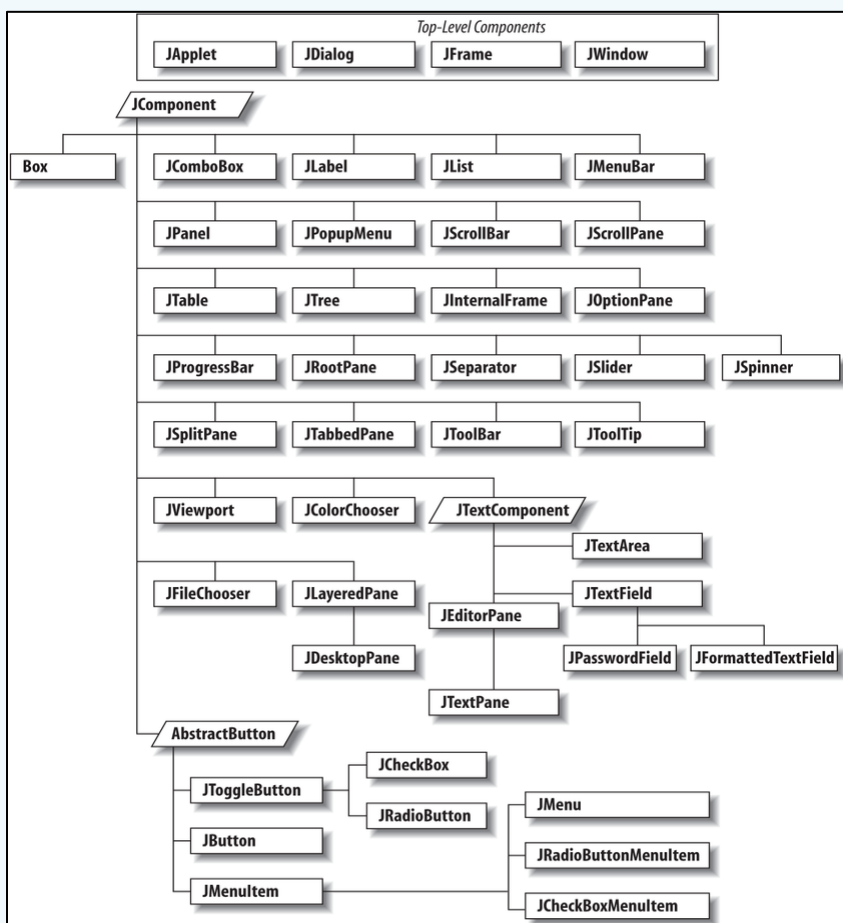
Cho tới nay thì gói được sử dụng một cách rộng rãi nhất là gói `javax.swing`. Trên thực tế, hầu hết các thành phần trong Swing – kể cả một vài lớp tiện ích (utility class) – đều được đặt trong gói này. (Ngoại lệ duy nhất là trường bao (border) và lớp hỗ trợ cho cây, bảng và thành phần văn bản). Do về sau các thành phần cung cấp khả năng mở rộng tốt hơn và thường có nhiều lớp lớn hơn để làm việc, tương tác nên các lớp này được chia thành những gói riêng biệt.



2.3.4. Phân cấp lớp (Class):

Thoạt nhìn, cây phân lớp Swing rất giống AWT. Mỗi thành phần Swing và một thành phần tương đương trong AWT có cùng tên, ngoại trừ ký tự đầu tiên trong Swing có thêm chữ J viết hoa. Trong hầu hết các trường hợp, nếu thay thế các thành phần AWT thì thành phần Swing có thể được sử dụng như một sự thay thế dạng drop-in (không cần điều chỉnh). Tuy nhiên, giữa thành phần Swing và thành phần của AWT có những điểm khác biệt cơ bản.

Ta có thể dễ dàng thấy rằng số lượng frame và pane trong Swing tăng lên. Ví dụ với các internal frame (frame bên trong). Swing hỗ trợ đặt một frame vào một frame khác – điều này thường được ám chỉ như một giao diện đa tài liệu (multiple document interface – MDI) trong Microsoft. Bạn có thể tùy ý chỉ định internal frame này vào các tầng xếp theo chiều thẳng đứng; các tầng này sẽ quyết định internal frame nào sẽ xuất hiện ở trên. Trên thực tế, kể cả frame đơn giản nhất – JFrame – cũng hỗ trợ tầng bằng cách cung cấp Layered pane (pane xếp tầng), trên đó bạn có thể chỉ định vị cho các phần tử khác nhau khi phát triển ứng dụng.



Sơ đồ 2.3 Sơ đồ Cây phân lớp của Java Swing



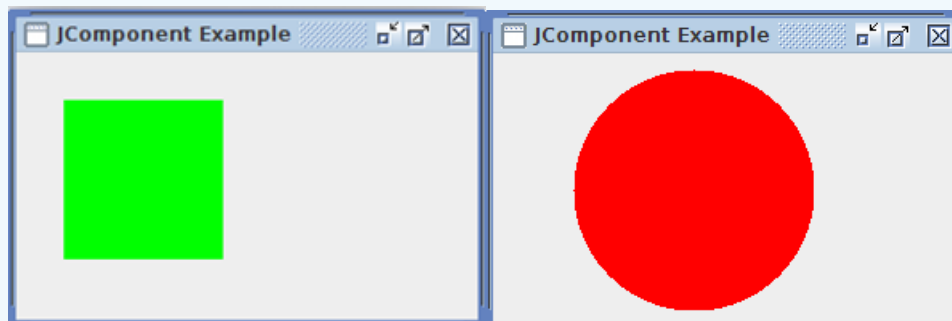
2.4. Các thành phần cơ bản sử dụng và cách thức tạo lập:

Công việc quan trọng để hoàn thành được mục tiêu đề án đó là thực hiện việc tạo lập nên giao diện sử dụng người dùng cho ứng dụng. Để thực hiện được điều này, ngoài việc phải nắm bắt được các công cụ hỗ trợ mà Java Swing mang lại, cần có sự kết hợp tỉ mỉ, lựa chọn hợp lý các lớp từ rất nhiều lớp được hỗ trợ bởi Java Swing để tạo nên được sự bắt mắt cho ứng dụng. Việc đầu tiên cần phải nắm bắt để hoàn thành mục tiêu này là chọn lựa đúng và biết cách sử dụng, tạo lập các thành phần cơ bản từ các lớp của Java Swing ảnh hưởng tới việc tạo lập giao diện ứng dụng.

2.4.1. Lớp JComponent:

JComponent là lớp trừu tượng mà hầu như mọi thành phần *Swing* đều kế thừa: *JComponent* cung cấp nhiều chức năng cơ bản phổ biến cho toàn bộ thư viện thành phần Swing (Swing component library). Cũng như ***java.awt.JComponent*** đóng vai trò là framework cho hầu hết thành phần AWT, lớp ***java.swing.JComponent*** cũng đóng vai trò tương tự thành phần Swing.

Bởi vì *JComponent* kế thừa lớp *Container* nên nhiều thành phần *Swing* có thể đóng vai trò như là container của các thành phần AWT và Swing khác. Các thành phần này có thể được thêm thông qua phương thức *add()* truyền thống của *Container*. Thêm nữa *JComponent* có thể được định vị với bất kỳ trình quản lý bố cục nào của Java ở bên trong container. Các thuật ngữ sau này vẫn giữ nguyên ý nghĩa: Con là thành phần được thêm vào một container (container này là cha của thành phần đó). Tương tự, bất kỳ thành phần nào đó ở mức cao hơn trong cây được gọi là tổ tiên, trong khi bất kỳ thành phần nào thấp hơn được gọi là con cháu.



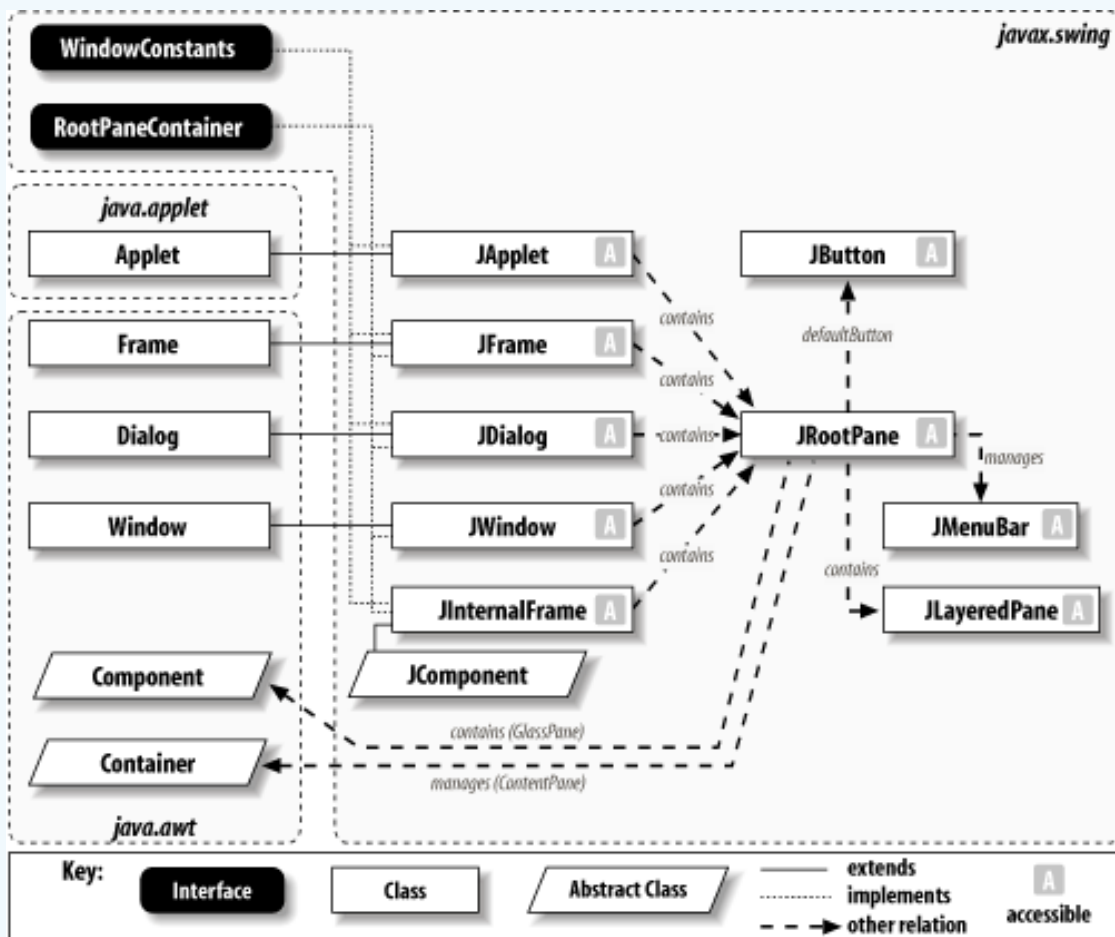
Hình 2.5 Ví dụ về việc tạo lập JComponent



2.4.2. Root Pane:

Hầu hết các container khác trong Swing (*JFrame*, *JApplet*, *JWindow*, *JDialog* và thậm chí là *JInternalFrame*) đều chứa thể hiện của một lớp khác (*JRootPane*) như thành phần duy nhất của chúng và thực thi giao diện chung là **RootPaneContainer**.

Ta hãy dành thời gian để tìm hiểu tổng thể về các lớp giao diện tạo nên container gốc (root container) của Swing. Sơ đồ 2.4 dưới đây cho thấy *JApplet*, *JFrame*, *JDialog* và *JWindow* không kế thừa lớp **JComponent** như các thành phần Swing khác, mà kế thừa từ những thành phần AWT tương ứng, đóng vai trò như cửa sổ giao diện người dùng ở mức cao nhất. Điều này ngụ ý rằng những thành phần này (không giống như thành phần lightweight Swing) có các đối tượng AWT tương ứng của nền tảng.



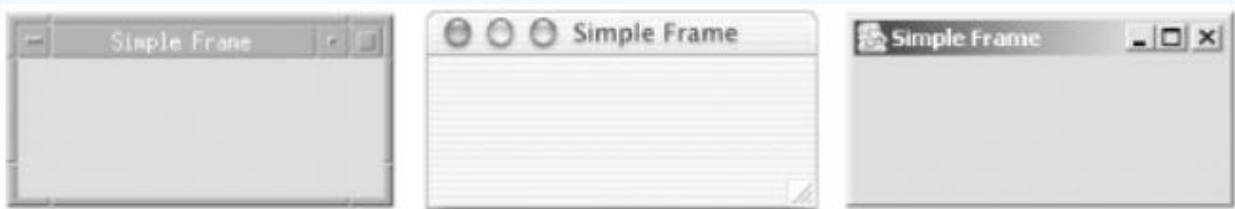
Sơ đồ 2.4 Sơ đồ lớp của Container “gốc” trong swing



Chú ý rằng Swing container này (cũng như là *JInternal Frame*) thực thi một giao diện chung là **RootPaneContainer**. Giao diện này cung cấp truy cập tới các thuộc tính của *JRootPane*. Thêm nữa là mỗi container trong năm container này đều sử dụng *JRootPane* làm container (thành phần chứa) “thực sự” cho các thành phần con mà container này quản lý.

2.4.3. Lớp JFrame:

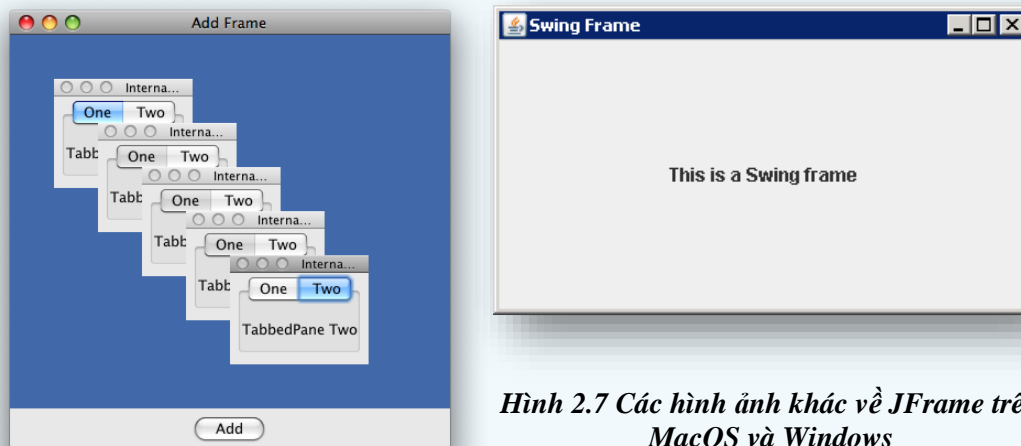
Hầu hết các Swing container của ứng dụng Java đều là lớp *JFrame*. *JFrame* cung cấp một cửa sổ ở mức cao nhất có một tiêu đề, đường bao và các yếu tố phụ thuộc nền tảng khác (ví dụ như các nút bấm thu nhỏ, phóng to và đóng). Bởi vì lớp này sử dụng *JRootPane* như là một lớp con duy nhất nên làm việc với một *JFrame* khá là khác biệt so với khi làm việc với một Frame của AWT. Một *JFrame* trông được minh họa như sau:



Hình 2.6 Các thể hiện *JFrame* trông trên nền tảng Unix, Mac và Windows

Phương thức khởi tạo:

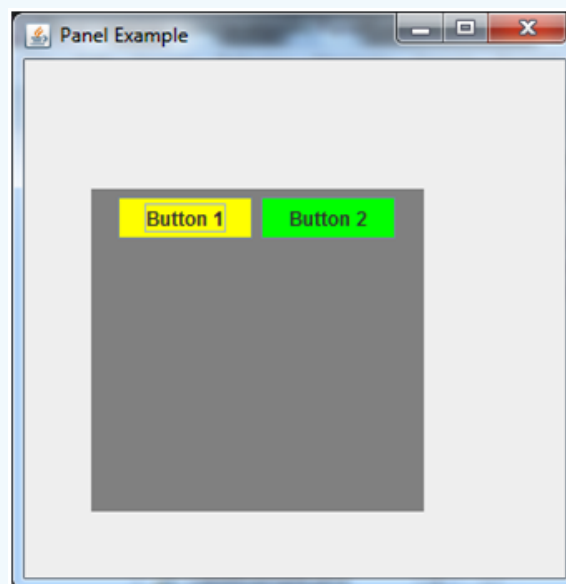
1. **JFrame()**: Xây dựng một Frame mới, ban đầu là không nhìn thấy (invisible).
2. **JFrame(GraphicsConfiguration gc)**: Tạo một Frame trong Graphics Configuration đã cho của một thiết bị màn hình và một title trống.
3. **JFrame(String title)**: Tạo một Frame mới, ban đầu là không nhìn thấy (invisible) với title đã cho.
4. **JFrame(String title, GraphicsConfiguration gc)**: Tạo một Frame với title đã cho và Graphics Configuration đã cho của một thiết bị màn hình.



Hình 2.7 Các hình ảnh khác về JFrame trên MacOS và Windows

2.4.4. Lớp JPanel:

JPanel là lớp kế thừa *JComponent* (mà *JComponent* kế thừa *java.awt.Container*) được sử dụng để nhóm các thành phần khác lại với nhau. JPanel có hầu hết các phương thức từ lớp cha. Thông thường, chúng ta sẽ khởi tạo đối tượng JPanel, thiết lập trình quản lý bố cục (có thể được thiết lập trong phương thức khởi tạo và mặc định là một *FlowLayout*) và thêm các thành phần vào JPanel thông qua phương thức *add()* được kế thừa từ *Container*.



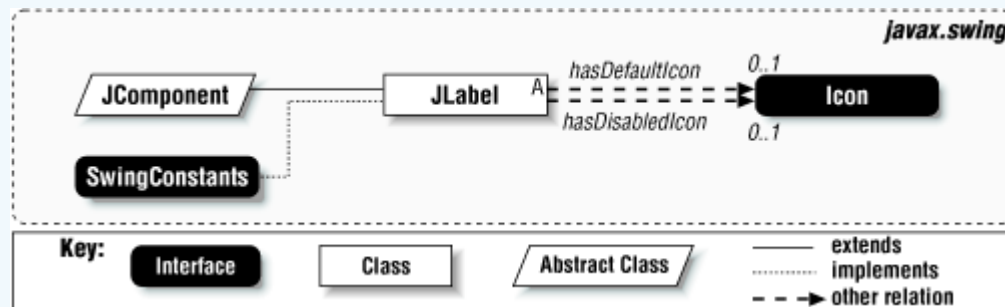
Hình 2.8 Hình ảnh về ví dụ của JPanel

**Phương thức khởi tạo:**

- JPanel():** Tạo một JPanel mới với một double buffer và một Flow Layout.
- JPanel(boolean isDoubleBuffered):** Tạo một JPanel mới với Flow Layout và trình đệm đã xác định.
- JPanel(LayoutManager layout):** Tạo một JPanel mới với Layout Manager đã cho
- JPanel(LayoutManager layout, boolean isDoubleBuffered):** Tạo một JPanel mới với Layout Manager đã cho và trình đệm đã xác định.

2.4.5. Lớp JLabel:

Swing cho phép bạn tạo ra các label (nhãn) có thể chứa văn bản, hình ảnh, hoặc cả hai. Lớp JLabel cho phép bạn thêm các nhãn cơ bản, không tương tác lên giao diện người dùng. Vì sự đơn giản vốn có của nhãn, nên không có lớp mô hình (model) cho JLabel. Sơ đồ 2.5 dưới đây cho thấy các lớp của JLabel. Sau đây, ta sẽ tìm hiểu hai mối quan hệ giữa của Icon.



Sơ đồ 2.5 Sơ đồ lớp của JLabel

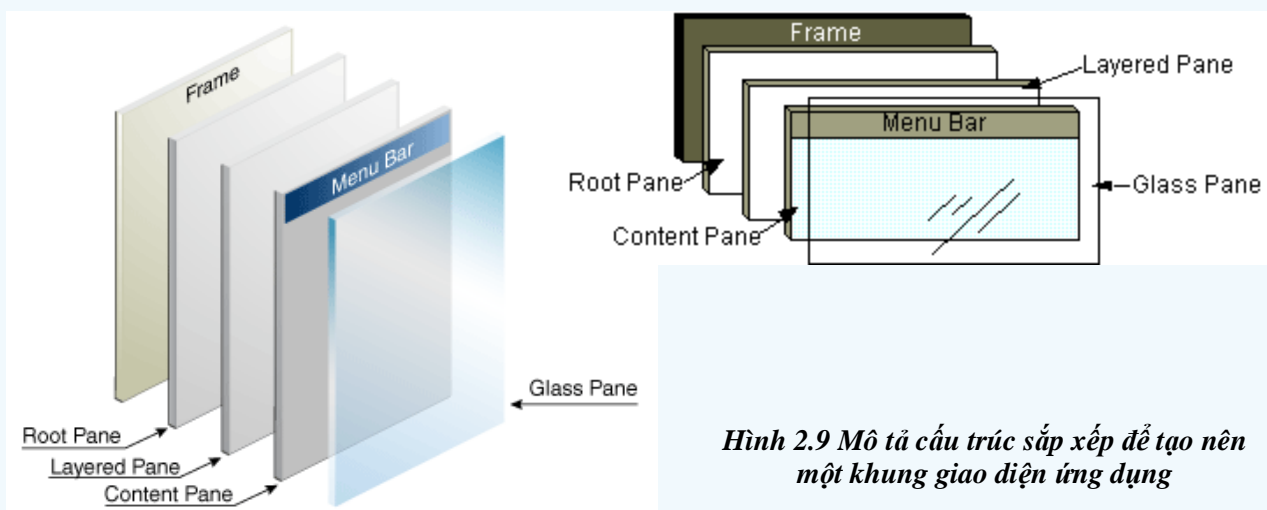
Phương thức khởi tạo:

- JLabel():** Tạo một instance của JLabel, không có hình ảnh, và với một chuỗi trống cho title.
- JLabel(Icon image):** Tạo một instance của JLabel với hình ảnh đã cho.



3. **JLabel(Icon image, int horizontalAlignment):** Tạo một instance của JLabel với hình ảnh và căn chỉnh ngang đã cho.
4. **JLabel(String text):** Tạo một instance của JLabel với text đã cho.
5. **JLabel(String text, Icon icon, int horizontalAlignment):** Tạo một instance của JLabel với text, hình ảnh, và căn chỉnh ngang đã cho.
6. **JLabel(String text, int horizontalAlignment):** Tạo một instance của JLabel với text và căn chỉnh ngang đã cho.

Qua 5 thành phần cơ bản được sử dụng như trên, ta có thể hình dung cơ bản được cấu trúc của giao diện ứng dụng bao gồm các lớp được xếp chồng và sắp xếp như thế nào. Hình ảnh dưới đây sẽ cho ta thấy tổng quan hơn về các cách sắp xếp giao diện được sử dụng từ các thành tố được nêu trên:

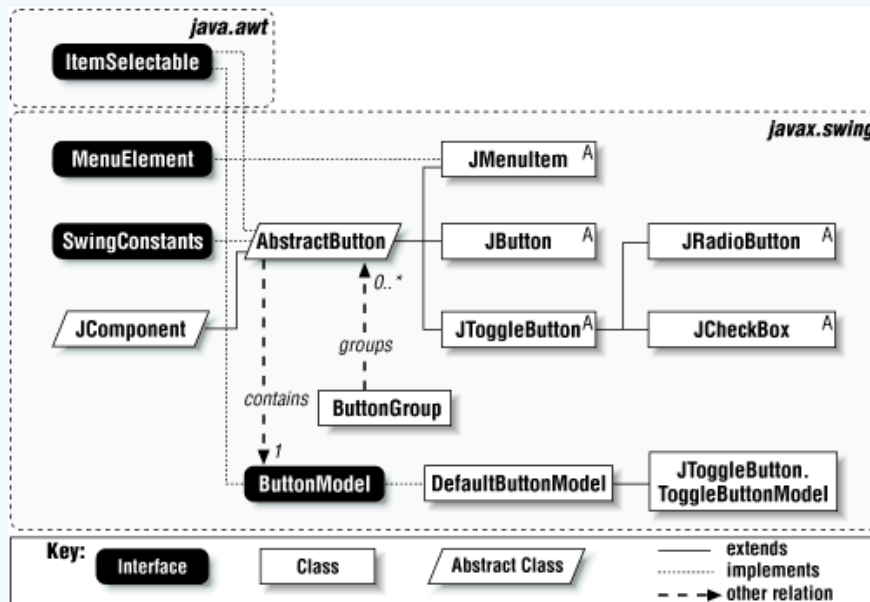


Hình 2.9 Mô tả cấu trúc sắp xếp để tạo nên một khung giao diện ứng dụng

Sau khi thiết lập được hệ thống các lớp khung tạo nên một frame giao diện, việc tiếp đến là chèn các nút bấm chức năng. Nút bấm (Button) là thành phần UI đơn giản được sử dụng để tạo ra sự kiện khi người dùng bấm. Trong Swing, nút bấm có thể hiển thị dưới dạng biểu tượng, văn bản hoặc cả hai. Vì vậy, đối tượng quan trọng cần phải tìm hiểu kế tiếp đó là lớp JButton.

2.4.6. Lớp JButton:

JButton là kiểu nút bấm đơn giản nhất, thêm rất ít so với những gì được cung cấp bởi lớp **AbstractButton**. JButton là nút mà không chuyển trạng thái từ bật sang tắt, nhưng hoạt động như nút bấm, gọi một số action (hành động) khi được nhấn vào. Sơ đồ 2.10 dưới đây trình bày hệ thống phân cấp các lớp và mối quan hệ giữa các lớp liên quan đến nút bấm trong Swing:



Sơ đồ 2.6 Sơ đồ lớp của JButton



Hình 2.10 Ví dụ về việc tạo lập nút bấm từ JButton

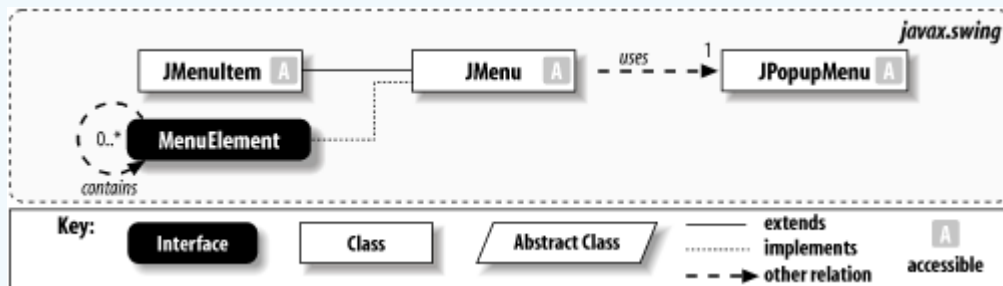
Phương thức khởi tạo:

1. **JButton():** Tạo một button mà không thiết lập text hoặc icon.
2. **JButton(Action a):** Tạo một button tại đây các thuộc tính được nhận từ Action đã cung cấp.
3. **JButton(Icon icon):** Tạo một button với một icon.
4. **JButton(String text):** Tạo một button với text.
5. **JButton(String text, Icon icon):** Tạo một button với text ban đầu và một icon.

Hệ thống nút bấm chức năng đã có, để tiếp tục đưa các nút bấm này vào một Menu chức năng tạo nên một hệ thống cài đặt – điều chỉnh có tổ chức cho ứng dụng, ta cần tìm hiểu đến các thư viện Menu của Java Swing như sau:

2.4.7. Lớp JMenu:

Lớp JMenu đại diện cho các menu được đính kèm với một JMenuBar hoặc một JMenu khác. Menu được gắn trực tiếp vào thanh menu được gọi là menu cấp cao nhất. Mặt khác, các menu con không được gắn vào thanh menu mà là một mục menu đóng vai trò là tiêu đề của nó. Tiêu đề mục menu này thường được đánh dấu bằng một mũi tên phải, cho biết menu của nó xuất hiện cùng với mục menu nếu người dùng chọn nó.



Sơ đồ 2.7 Sơ đồ lớp của JMenu

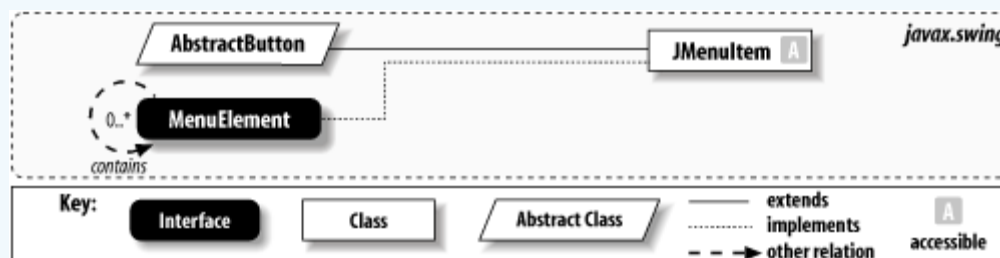
Phương thức khởi tạo:

- JMenu():** Xây dựng một JMenu mới không có text.
- JMenu(Action a):** Xây dựng một menu có các thuộc tính được nhận từ Action đã cho.
- JMenu(String s):** Xây dựng một JMenu mới với chuỗi s đã cho (như là text của nó).
- JMenu(String s, boolean b):** Xây dựng một JMenu mới với chuỗi s đã cho (như là text của nó) và một giá trị boolean để xác định có hay không một tear-off menu.

2.4.8. Lớp JMenuItem:

Một JMenuItem phục vụ như một trình bao bọc cho các chuỗi và hình ảnh được sử dụng làm các thành phần trong một menu. Lớp JMenuItem về cơ bản là một nút chuyên

dụng và mở rộng lớp **AbstractButton**. Tuy nhiên, hành vi của nó hơi khác so với các nút độc lập. Khi con trỏ chuột được kéo qua một mục menu, Swing xem xét mục menu sẽ được chọn. Nếu người dùng nhả nút chuột trong khi qua mục menu, nó được coi là được chọn và sẽ thực hiện hành động của nó.



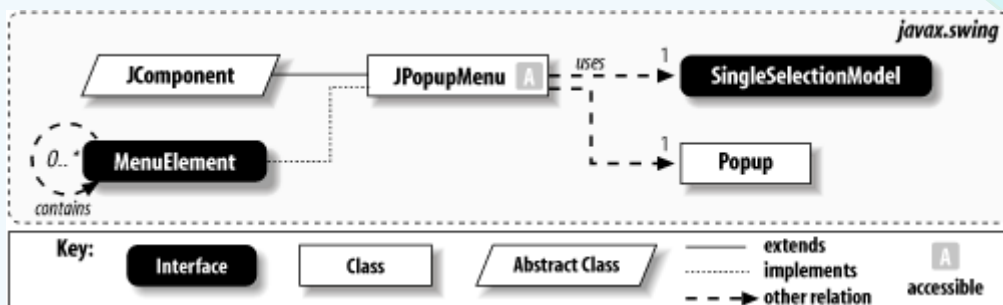
Sơ đồ 2.8 Sơ đồ lớp của JMenuItem

Phương thức khởi tạo:

1. **JMenuItem()**: Tạo một JMenuItem không có text hoặc icon.
2. **JMenuItem(Action a)**: Tạo một JMenuItem có các thuộc tính được nhận từ Action đã cho.
3. **JMenuItem(Icon icon)**: Tạo một JMenuItem với icon đã cho.
4. **JMenuItem(String text)**: Tạo một JMenuItem với text đã cho.
5. **JMenuItem(String text, Icon icon)**: Tạo một JMenuItem với text và icon đã cho.
6. **JMenuItem(String text, int mnemonic)**: Tạo một JMenuItem với text đã cho và mnemonic.

2.4.9. Lớp JPopupMenu:

Pop-up Menu là một tính năng giao diện người dùng ngày càng phổ biến. Các menu này không được gắn vào một thanh menu; thay vào đó, chúng là các menu nổi tự do liên kết với một thành phần cơ bản. Thành phần này được gọi là invoker. Được liên kết với các yếu tố giao diện cụ thể, các menu bật lên rất nhạy cảm với ngữ cảnh. Chúng được xuất hiện khi kích hoạt một sự kiện đã được gán cho một thành phần cơ bản. Sau khi xuất hiện, người dùng có thể tương tác như với menu bình thường.

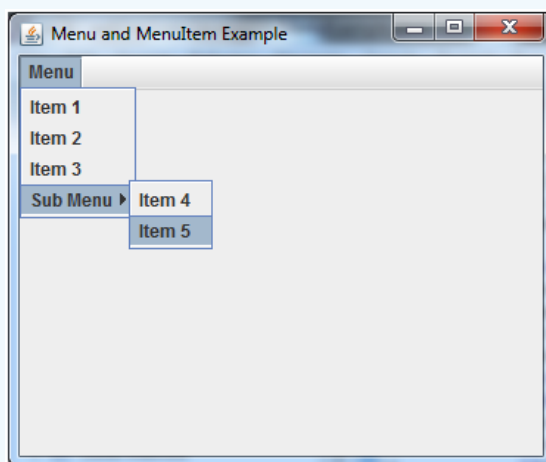


Sơ đồ 2.8 Sơ đồ lớp của JPopupMenu

Phương thức khởi tạo:

1. **JPopupMenu():** Xây dựng một JPopupMenu không có "invoker".
2. **JPopupMenu(String label):** Xây dựng một JPopupMenu với title đã cho.

Từ việc kết hợp 3 lớp Menu của Java Swing trên, chúng ta có thể tạo lập được một thanh Menu Taskbar Popup như hình ảnh sau:



Hình 2.11 Ví dụ về việc tạo lập thanh Menu chức năng từ các lớp Menu của Java Swing

Sau khi hoàn thành cơ bản về việc tạo lập giao diện và các Menu chức năng cần thiết, việc tiếp theo là tạo lập khung gõ câu lệnh để gửi lệnh cho ứng dụng thực hiện những gì chúng ta đã lập trình sẵn, phần quan trọng nhất để tạo nên một ứng dụng có tính tương tác giữa người sử dụng và trợ lý ảo để thực hiện việc ra lệnh các hành động, chức năng, ... Lớp tiếp theo hỗ trợ chúng ta hoàn thành việc này là JTextField.

2.4.10 Lớp JTextField:

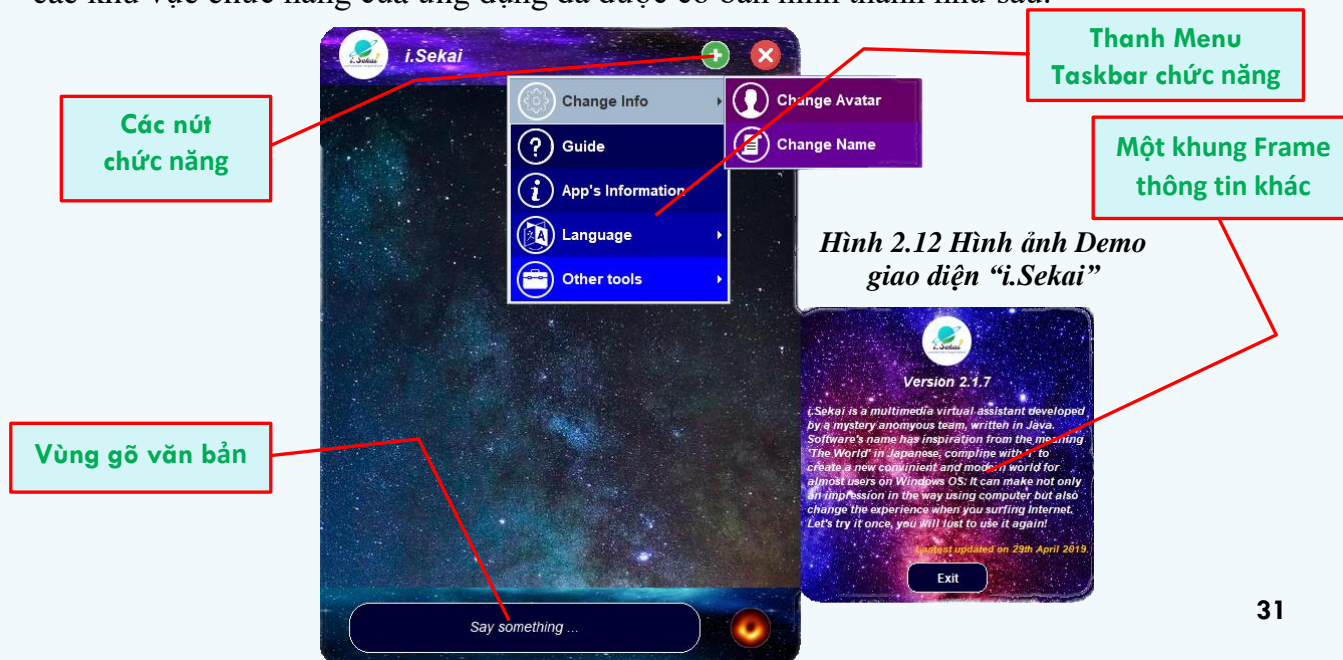
JTextField cho phép người dùng nhập một dòng văn bản, cuộn văn bản nếu kích thước của nó vượt quá kích thước vật lý của ô TextField. Một JTextField kích hoạt một ActionEvent cho bất kỳ ActionListener nào đã thiết đặt (bao gồm cả Action được đặt qua phương thức setAction(), nếu có) khi người dùng nhấn phím Enter.

JTextFields (và cả **JTextComponents**) được cài đặt một số hành vi phù hợp với L&F, do đó cắt / sao chép / dán, phím di chuyển con trỏ đặc biệt và cử chỉ chọn văn bản sẽ hoạt động mà không cần thêm bất kỳ sự can thiệp nào.

Phương thức khởi tạo:

1. **JButton()**: Tạo một button mà không thiết lập text hoặc icon.
2. **JButton(Action a)**: Tạo một button tại đây các thuộc tính được nhận từ Action đã cung cấp.
3. **JButton(Icon icon)**: Tạo một button với một icon.
4. **JButton(String text)**: Tạo một button với text.
5. **JButton(String text, Icon icon)**: Tạo một button với text ban đầu và một icon.

Như vậy, qua quá trình học tập, tìm hiểu cách hoạt động, sử dụng của 10 thành phần lớp cơ bản trên của Java Swing, bằng việc kết hợp, sự sáng tạo hợp lý, giao diện và các khu vực chức năng của ứng dụng đã được cơ bản hình thành như sau:





2.5. Cơ sở dữ liệu và lưu trữ thông tin:

Ngày nay, bất cứ ứng dụng hay trò chơi nào, trên bất cứ nền tảng nào, di động, máy tính, hay một trang web bất kì trên mạng ta đều dễ dàng bắt gặp về khái niệm Cơ sở dữ liệu (CSDL). Vậy câu hỏi đặt ra, CSDL là gì? Và tại sao khi xây dựng một hệ thống, một ứng dụng, một trò chơi đều cần sử dụng tới CSDL?

Cơ sở dữ liệu (CSDL) là một hệ thống các thông tin có cấu trúc, được lưu trữ trên các thiết bị lưu trữ nhằm thỏa mãn yêu cầu khai thác thông tin đồng thời của nhiều người sử dụng hay nhiều chương trình ứng dụng chạy cùng một lúc với những mục đích khác nhau. Với ứng dụng “Trợ lý đa năng – i.Sekai” thì CSDL có vai trò trong việc lưu trữ thông tin người dùng khi họ muốn thay đổi tên, ảnh đại diện của trợ lý ảo, thay đổi cài đặt của ứng dụng,... và quan trọng hơn cả, CSDL sẽ là hệ thống lưu toàn bộ các câu lệnh, tập lệnh phản hồi của ứng dụng, giúp ứng dụng trở nên thân thiện và giống với một trợ lý, một người bạn nhất, hỗ trợ họ trong việc trải nghiệm, sử dụng máy tính, đảm bảo được yêu cầu và mục tiêu ban đầu mà nhóm đặt ra cho đồ án.

Hiện nay có rất nhiều hệ quản trị CSDL như MySQL, SQL Server, Oracle, MS Access, ... Tuy nhiên, để phù hợp với nhu cầu sử dụng và lập trình “i.Sekai”, nhóm đã chọn sử dụng 2 thư viện hỗ trợ trong việc tạo lập CSDL phù hợp với Netbeans IDE 8.2 là Database Derby và JSON 1.1.1 cho Java.

2.5.1 Tìm hiểu về kết nối hệ quản trị CSDL Apache Derby với Java (JDBC):

Apache Derby là một hệ thống quản lý CSDL được phát triển bởi Apache Software Foundation. Đây là hệ quản trị CSDL có thể nhúng được vào các chương trình Java và thường



Hình 2.13 Logo của Apache Derby

được sử dụng để xử lý các giao dịch trực tuyến. Dung lượng của hệ thống này khá nhỏ gọn, với chỉ khoảng 3.5MB, Apache Derby được phát triển như một dự án nguồn mở với giấy



phép từ Apache 2.0. Và dĩ nhiên, với hệ thống nguồn mở này, chúng ta có thể sử dụng những hệ quản trị CSDL ấy vào Netbeans IDE 8.2 và sử dụng thuận tiện, đơn giản hơn trong việc tạo lập hệ thống phản hồi cho “i.Sekai”.

Để các ứng dụng Java làm việc với một cơ sở dữ liệu trong Java thì phải thông qua kết nối cơ sở dữ liệu trong Java (JDBC), cần phải có một thư viện trình điều khiển JDBC phù hợp trong đường dẫn lớp của ứng dụng. Thư viện trình điều khiển JDBC bao gồm các lớp Java thực hiện kết nối đến các hệ quản trị cơ sở dữ liệu. Nó nói chuyện với các ứng dụng Java thông qua JDBC API và thường đi kèm như một tệp JAR hoặc ZIP.

Java DB là cơ sở dữ liệu nguồn mở của Apache Derby và được hỗ trợ bởi Sun; gồm đầy đủ các tính năng của một CSDL cần, để dàng sử dụng, đặc biệt là dung lượng nhỏ (3.5MB) hỗ trợ cho các ứng dụng Java Desktop và J2EE.

Một số nội dung cần lưu ý để sử dụng JDBC với Netbeans, thư viện cung cấp cho người dùng rất nhiều hàm và lớp để sử dụng nhưng chúng ta chỉ chọn những nội dung phù hợp để tạo lập hệ CSDL cho ứng dụng:

1. **DriverManager:** Là một class, nó dùng để quản lý danh sách các Driver (database drivers).
2. **Driver:** Là một Interface, nó dùng để liên kết các liên lạc với cơ sở dữ liệu, điều khiển các liên lạc với database. Một khi Driver được tải lên, lập trình viên không cần phải gọi nó một cách cụ thể.
3. **Connection:** Là phiên làm việc giữa ứng dụng Java và CSDL. Đối tượng Connection được sử dụng để tạo Statement, PreparedStatement, và DatabaseMetaData. Giao diện Connection cung cấp nhiều phương thức quản lý transaction như commit(), rollback(), ...
4. **Statement:** Cung cấp các phương thức để thực thi các câu lệnh truy vấn với cơ sở dữ liệu SQL. Statement interface là một nhà máy của ResultSet, tức là nó cung cấp phương thức để tạo ra đối tượng ResultSet.
5. **ResultSet:** Duy trì một con trỏ trỏ đến một hàng của một bảng. Ban đầu, con trỏ trỏ đến hàng đầu tiên.



6. **ResultSet executeQuery(String sql):** Được sử dụng để thực hiện truy vấn SELECT. Nó trả về đối tượng của ResultSet.
7. **createStatement:** Tạo một đối tượng Statement được sử dụng để thực thi các câu truy vấn SQL.

Để tạo và lấy CSDL từ trong Apache Derby, chúng ta cần sử dụng một số câu lệnh như trong SQL Server:

Hàm tạo bảng:

CREATE TABLE <Tên bảng> (<Tên cột 1> <kiểu giá trị 1>, <Tên cột 2> <kiểu giá trị 2> ...)

Hàm Insert giá trị vào bảng:

INSERT INTO <Tên bảng> values ('<Tên cột 1>', '<Tên cột 2>', ...)

Hàm lấy giá trị từ bảng:

SELECT <Tên cột> FROM <Tên bảng> where <điều kiện>

Sau khi tạo lập được hệ thống phản hồi nhờ Apache Derby – Database (JDBC), ta thu được kết quả như ví dụ sau:



Hình 2.14 Kết quả sau khi sử dụng Apache Derby cho việc tạo lập CSDL phản hồi



2.5.2 Tìm hiểu về thư viện JSON:

Trong thế giới dịch vụ Web và IoT ngày nay, JSON đã trở thành giao thức truy cập để mang thông tin từ máy khách đến máy chủ nhanh chóng, nhỏ gọn. Họ đã thay thế XML và đây là cách được ưu tiên nhất để chuyển thông tin theo cách độc lập với nền tảng.

Thật không may, JDK không có một thư viện JSON. Tuy nhiên, có rất nhiều thư viện ngoài tốt cho phép bạn phân tích cú pháp và tạo các thông điệp JSON, như Jackson và Gson. Vậy JSON là gì? Và việc sử dụng nó như thế nào để ứng dụng vào việc tạo chức năng cho “i.Sekai”?

JSON là chữ viết tắt của **JavaScript Object Notation**, đây là một dạng dữ liệu tuân theo một quy luật nhất định mà hầu hết các ngôn ngữ lập trình hiện nay đều có thể đọc được, bạn có thể sử dụng lưu nó vào một file, một record trong CSDL rất dễ dàng. JSON có định dạng đơn giản, dễ dàng sử dụng và truy vấn hơn XML rất nhiều nên tính ứng dụng của nó hiện nay rất là phổ biến.

Thư viện **json.simple** cho phép chúng ta đọc và ghi dữ liệu JSON trong Java. Nói cách khác, chúng ta có thể mã hóa và giải mã đối tượng JSON trong java bằng cách sử dụng thư viện **json.simple**. Gói **org.json.simple** chứa các lớp quan trọng cho JSON API như:

- JSONValue
 - JSONObject
 - JSONArray
 - JsonString
 - JsonNumber
- Để sử dụng được thư viện JSON trong việc tạo nên chức năng cho ứng dụng, đầu tiên cần phải tạo ra 1 *JSONObject* với câu lệnh:

```
JSONObject obj = new JSONObject();
```

Sau đó, dùng câu lệnh **.put** để đưa các cặp giá trị “key” và “value” vào. Ví dụ:

```
obj.put("age", 19);
```



- Để tạo ra một mảng các giá trị từ JSON, ta dùng câu lệnh:

```
JSONArray list = new JSONArray();
```

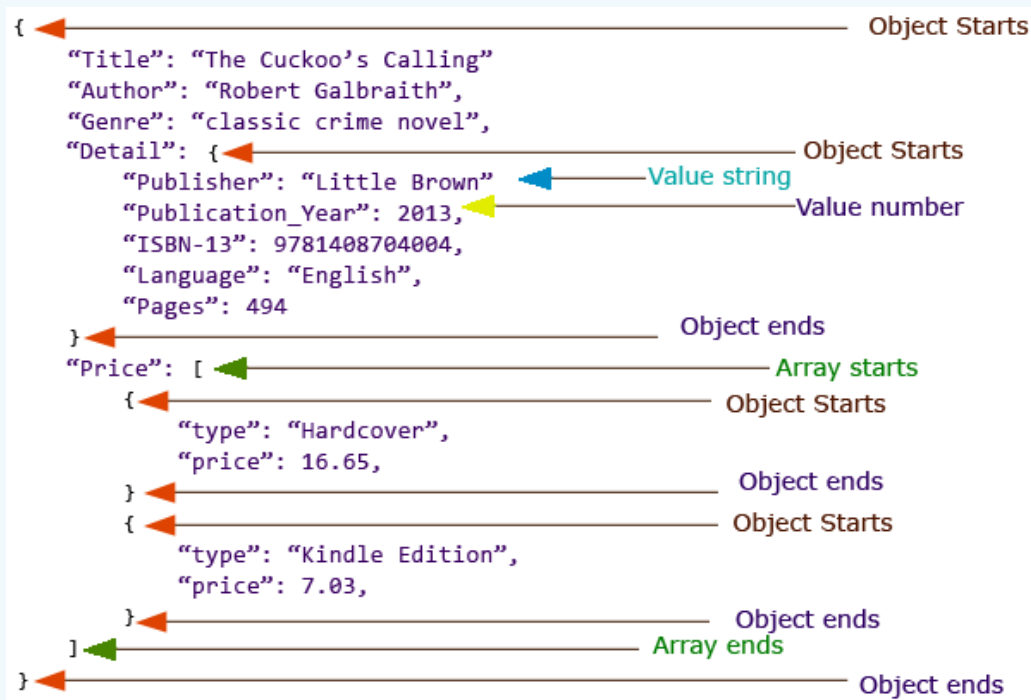
Sau đó, dùng câu lệnh `.add` để thêm các giá trị vào mảng và dùng câu lệnh `.put` để đưa mảng đã tạo vào object đã tạo. Ví dụ:

```
list.add("msg 1");
```

```
list.add("msg 2");
```

```
list.add("msg 3");
```

```
obj.put("messages", list);
```



Hình 2.14 Ví dụ về việc sử dụng file JSON để thêm các cặp giá trị “key” và “value”

- Để đưa các giá trị của JSON vào file ta dùng câu lệnh như sau:

```
try (FileWriter file = new FileWriter("đường dẫn tạo file JSON"))
```

```
{
```

```
    file.write(obj.toJSONString());
```

```
    file.flush();
```

```
} catch (IOException e) {
    e.printStackTrace();
}
```

Sau khi viết thành công file JSON, để đọc được file và truy xuất các giá trị trong file ta cần thực hiện các bước sau:

- Gọi lớp JSONParser để phân tích bằng câu lệnh:
`JSONParser parser = new JSONParser();`
- Sau đó gọi lớp object trong Java (Object là lớp cha của tất cả các lớp trong Java. Nói cách khác, nó là lớp cao nhất của Java. Lớp Object rất lợi ích nếu bạn muốn tham chiếu bất cứ đối tượng nào có kiểu mà bạn không biết):
`Object obj = parser.parse(new FileReader("đường dẫn tới file JSON"));`
- Tiếp theo, ta gọi JSONObject và bắt đầu đọc các “values” từ “key” có trong file JSON bằng câu lệnh `.get`. Ví dụ:

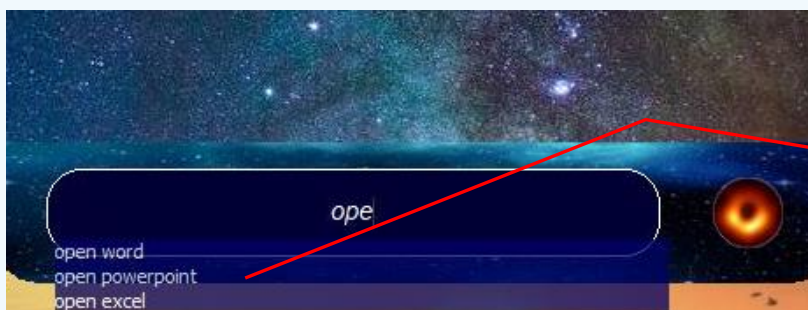
```
Object obj = parser.parse(new FileReader("D:\\test.json"));
```

```
JSONObject jsonObject = (JSONObject) obj;
```

```
String name = (String) jsonObject.get("Name"); // Lấy value có key bằng “Name”
```

```
System.out.println(Name); // In ra giá trị có key bằng “Name”
```

Bằng việc sử dụng thư viện JSON, ta có thể thu được kết quả như hình minh họa sau:



Tạo lập được phần gợi ý
câu lệnh cho ứng dụng

Hình 2.15 Kết quả sau khi sử dụng JSON cho việc
tạo lập chức năng cho ứng dụng

CHƯƠNG III

THIẾT KẾ VÀ THỰC HIỆN

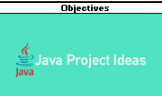



3.1. Kế hoạch thiết kế và thực hiện:

Một số thành phần, đối tượng cần thiết cho việc thiết kế, sắp xếp trên ứng dụng chính cần phải kể đến như:

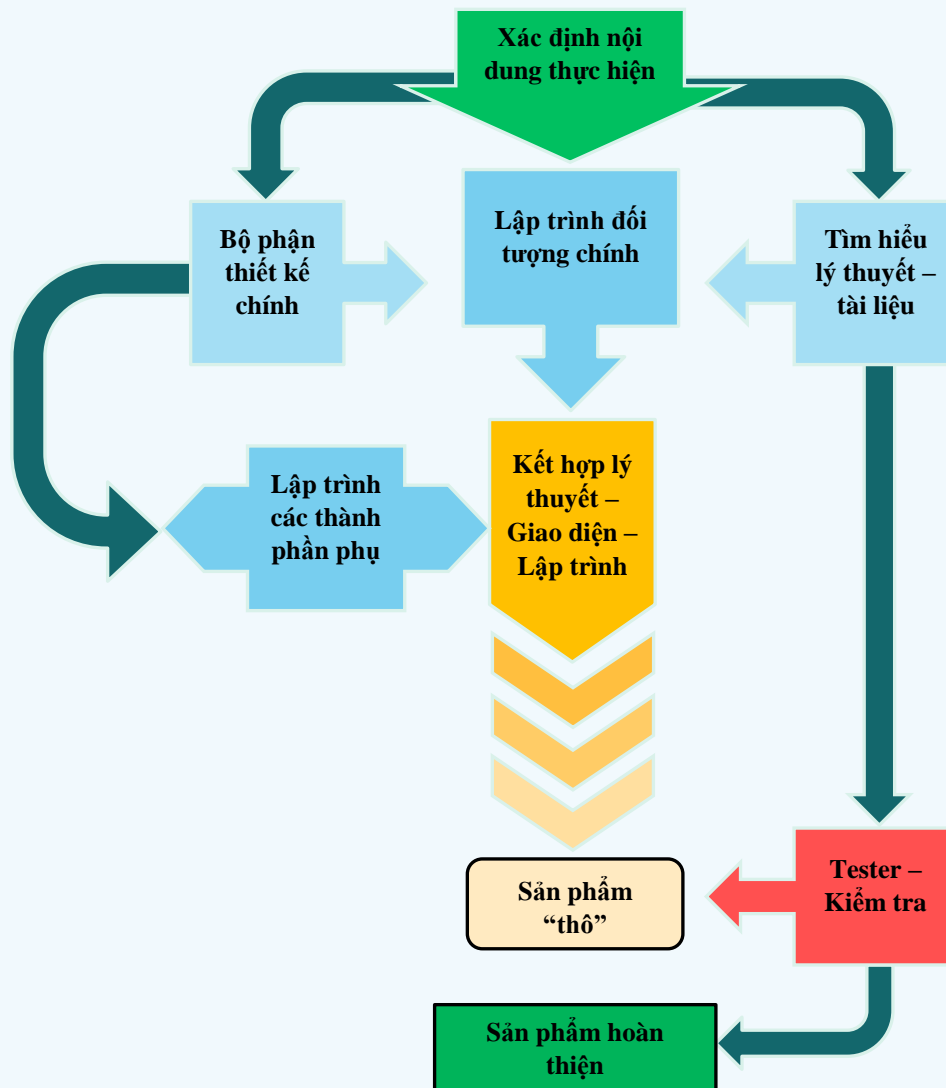
- Hình ảnh đại diện, logo ứng dụng.
- Intro khi mở ứng dụng.
- Giao diện chung, sắp xếp bố cục trình bày.
- Icon ứng dụng.
- Hệ thống hình ảnh các nút chức năng trên giao diện ứng dụng.
- Và các thành phần, chi tiết khác, ...

Vì công việc thiết kế đòi hỏi một chút đa nhiệm và thẩm mỹ, nên ngay từ khi lên ý tưởng đề tài, nhóm đã phân công công việc theo một lộ trình rõ ràng để mỗi người nắm một mảng riêng và có thể hoàn thành công việc tạo nên “i.Sekai” được nhanh chóng, kết hợp hoàn hảo và đúng mục tiêu đồ án.

Việc thiết kế các đối tượng, lập trình chức năng cho giao diện và tìm hiểu tài liệu được chia làm 3 cho các thành viên cùng hoàn thành, thực hiện và kết hợp song song với nhau. Với việc bám sát với công việc được giao cho từng cá nhân trong từng tuần học, nhóm luôn đạt được việc hoàn thành mục tiêu và hoàn thành sản phẩm (chi tiết) trong từng tuần, từng giai đoạn.

Project Leader: Phan Thanh Tùng		Project: Java Programming		Date: 23/03/2019	
Project Objective: Ứng dụng trợ lý đa năng "i.Sekai"					
Objectives	Major Tasks	Project Completed By: Group		Owner / Priority / Note	
	Tuần 0: Thành lập nhóm, xác định kế hoạch ban đầu				
	0.1 Thu thập thông tin thành viên nhóm (nhóm 3 thành viên)				
	0.2 Đặt nội quy, lịch họp hàng tuần cho nhóm.				
	0.3 Định hình kế hoạch - công việc các tuần.				
	0.4 Cùng nhau lên ý tưởng, thống nhất ý kiến về ứng dụng muốn làm.				
	0.5 Lên form giao diện, tìm cách tạo ứng dụng bằng Java, chọn IDE sử dụng thống nhất.				
	Tuần 1: Lên khung giao diện, tìm hiểu cách code và tạo các chức năng cơ bản				
	22/03: Thống nhất họp nhóm định kỳ hàng tuần vào sáng thứ Sáu.				
	1.1 Phân công công việc chính cho từng thành viên trong Đồ án.				
	+ Điều hành/Điều phối quản lý kế hoạch; lên ý tưởng; Coder (các phần phụ); thiết kế phụ (UI).				
	+ Coder chính (phần xương sống chính); lên ý tưởng; tìm kiếm tài liệu, kiến thức; thiết kế khung (UI).				
	+ Tester (UX); để xuất ý kiến; thiết kế phụ (UI); quản lý dữ liệu; hỗ trợ tài liệu, tìm kiếm kiến thức.				
	1.2 Tìm hiểu về Java Swing.				
	1.3 Thống nhất sử dụng NetBeans IDE để code ứng dụng.				
	1.4 Tạo khung ứng dụng cơ bản, các nút nhấn và một số chức năng offline cơ bản.				
	Tuần 2: Tạo chức năng tìm kiếm online, thống nhất tên và hình ảnh đại diện trợ lý ảo;				
	25/03: Họp nhóm.				
	2.1 Họp thống nhất tên và hình ảnh đại diện cho trợ lý ảo.				
	2.2 Thiết kế hình ảnh trợ lý.				
	2.3 Thiết kế giao diện ứng dụng.				
	2.4 Tìm hiểu cách thức nạp địa chỉ URL hỗ trợ khả năng truy cập Internet và trình duyệt, tìm kiếm online cho ứng dụng.				
	29/03: Họp nhóm định kỳ.				
	2.5 Tạo, tham gia Github hỗ trợ việc phát triển từng mảng của từng thành viên để dễ dàng hơn.				

Hình 3.1 Thực hiện việc lên kế hoạch chi tiết – phân công chi tiết, rõ ràng.



Sơ đồ 3.1 Sơ đồ mô tả quá trình hoạt động, thực hiện và làm việc của nhóm.

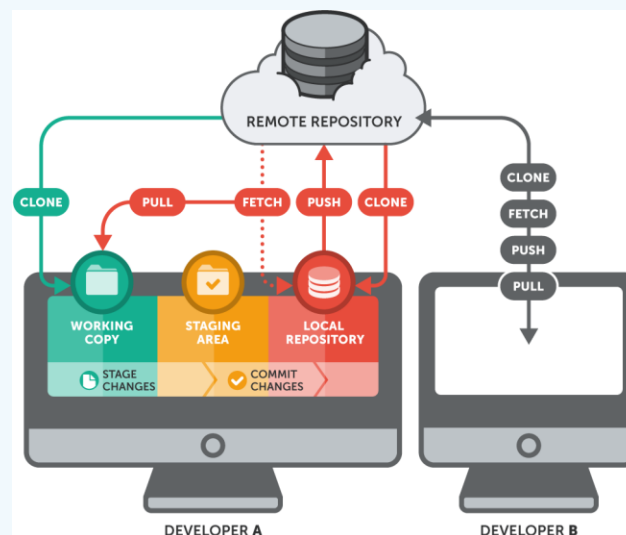
3.2. Các công cụ hỗ trợ khác:

Ngoài việc sử dụng thành thạo các công cụ cơ bản để thực hiện đồ án như Microsoft Office, Netbeans IDE 8.2. Nhóm cũng cần sử dụng đến một số công cụ hữu ích trong việc làm việc nhóm chuyên nghiệp cho việc lập trình là ***Github***.



Hình 3.2 Logo của Github

Git là một hệ thống quản lý phiên bản phân tán (Distributed Version Control System – DVCS) ra đời vào năm 2005 và hiện được dùng rất phổ biến. So với các hệ thống quản lý phiên bản tập trung khi tất cả mã nguồn và lịch sử thay đổi chỉ được lưu một nơi là máy chủ thì trong hệ thống phân tán, các máy khách không chỉ "check out" phiên bản mới nhất của các tập tin mà là sao chép (mirror) toàn bộ kho mã nguồn (repository). Như vậy, nếu như máy chủ ngừng hoạt động, thì bạn hoàn toàn có thể lấy kho chứa từ bất kỳ máy khách nào để sao chép ngược trở lại máy chủ để khôi phục lại toàn bộ hệ thống. Mỗi checkout thực sự là một bản sao đầy đủ của tất cả dữ liệu của kho chứa từ máy chủ.



Sơ đồ 3.2 Mô hình tổ chức dữ liệu phân tán trên Git

Github giúp bạn:

- ✓ Lưu lại được các phiên bản khác nhau của mã nguồn dự án phần mềm.
- ✓ Khôi phục lại mã nguồn từ một phiên bản bất kỳ.
- ✓ Dễ dàng so sánh giữa các phiên bản.
- ✓ Phát hiện được ai đã sửa phần nào làm phát sinh lỗi.
- ✓ Khôi phục lại tập tin bị mất.



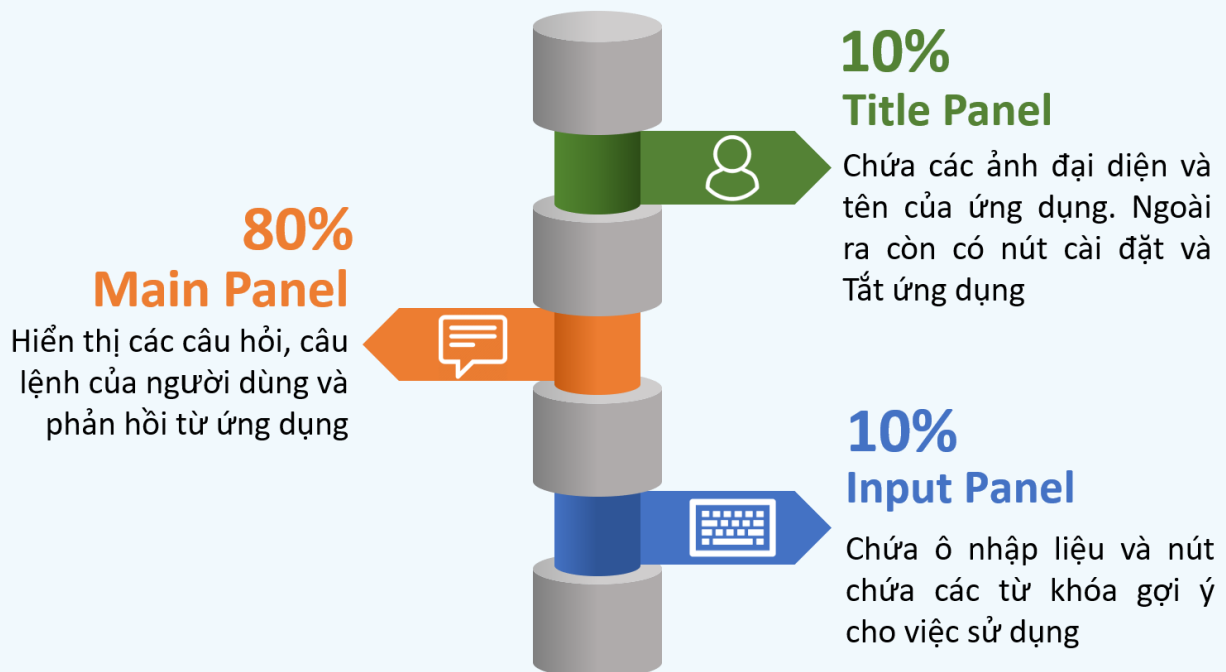
✓ Dễ dàng thử nghiệm, mở rộng tính năng của dự án mà không làm ảnh hưởng đến phiên bản chính (master branch).

✓ Giúp phối hợp thực hiện dự án trong nhóm một cách hiệu quả.

Ngoài ra, để thiết kế ra một giao diện người dùng bắt mắt, thân thiện, dễ sử dụng, phải kể đến đó là: *Adobe Photoshop*, *Adobe Illustrator*, *Proshows*, ... cùng một số công cụ xử lý hình ảnh, âm thanh, video khác.

3.3. Giới thiệu các thành phần chính:

Về cơ bản, “i.Sekai” có những thành phần chính cơ bản được tạo lập như sau:



Sơ đồ 3.3 Các thành phần chính cơ bản của “i.Sekai”



3.3.1 Title Panel:

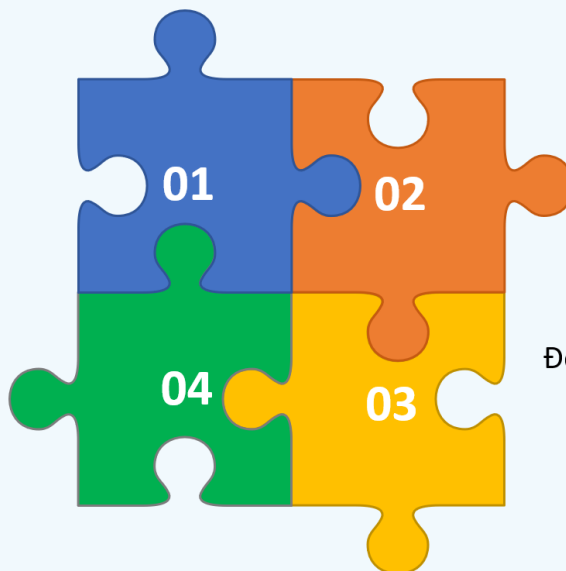


Ảnh đại diện

Là hình đại diện cho trợ lý ảo. Người dùng có thể thay đổi tùy theo sở thích cá nhân.

Nút thoát

Chỉ đơn giản là bấm vào và thoát ứng dụng



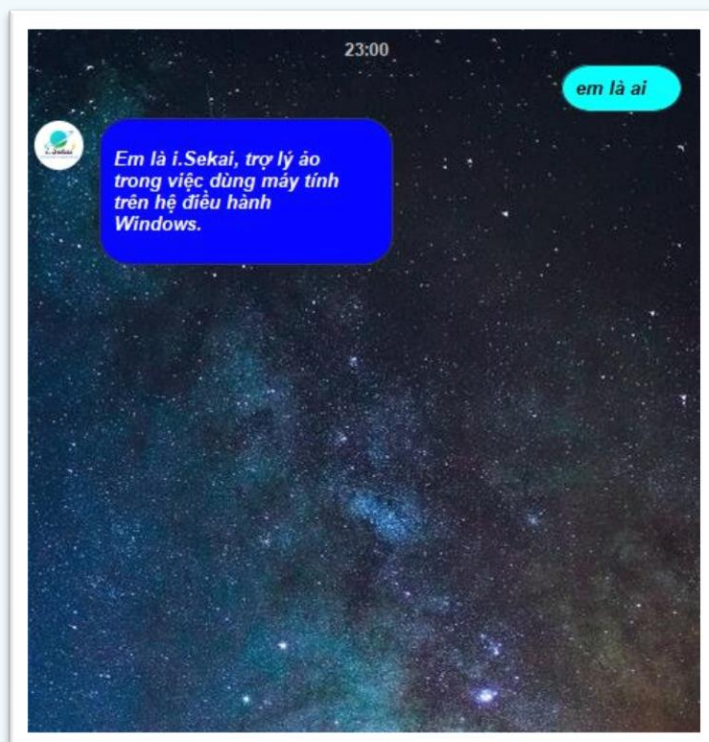
Tên trợ lý ảo

Là tên để gọi trợ lý ảo. Người dùng cũng có thể thay đổi theo mong muốn

Nút cài đặt

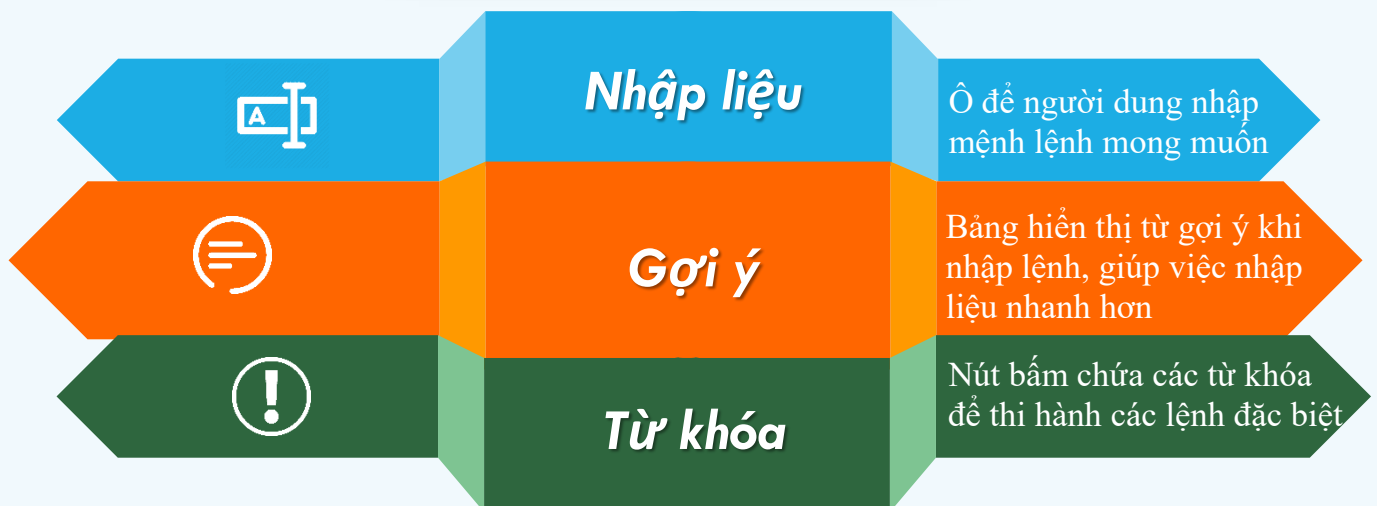
Để mở menu cài đặt. Tại đây người dùng có thể thay đổi ảnh, tên, ngôn ngữ hoặc các thông tin ứng dụng

3.3.2 Main Panel:





3.3.3 Input Panel:



3.4. Giới thiệu các thành phần phụ:

3.4.1. Hệ thống Menu Taskbar:



Sơ đồ 3.4 Sơ đồ mô tả hệ thống thành phần của Menu Taskbar

3.4.2 Hệ thống gợi ý chức năng Back Hole Button:



Chỉ với việc nhấn vào biểu tượng “Lỗ đen” bên cạnh vùng nhập câu lệnh, một danh sách các từ gợi ý cách sử dụng ứng dụng sẽ hiện lên giúp người dùng mới có thể làm quen với “i.Sekai” dễ dàng hơn, dễ tự tìm hiểu và sử dụng lại nhiều lần hơn.



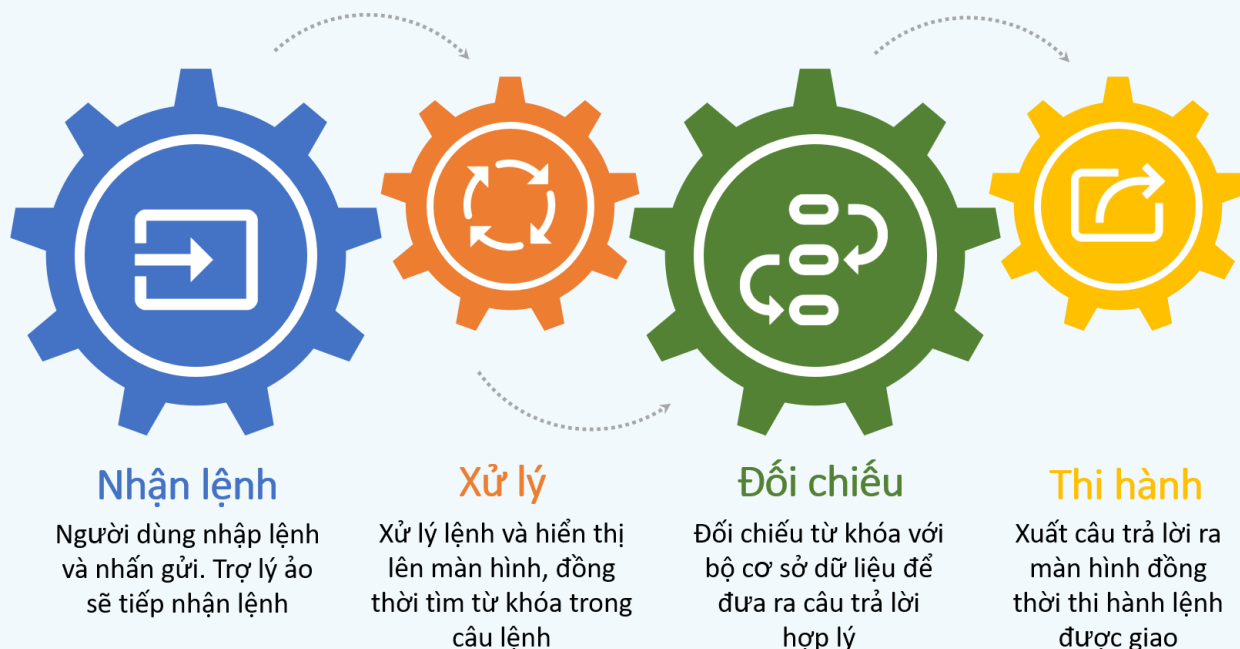
Hình 3.3 Hình ảnh mô tả nút nhấn chức năng gợi ý Back Hole Button



CHƯƠNG IV

CÁCH THỨC HOẠT ĐỘNG – CHỨC NĂNG VÀ HƯỚNG DẪN SỬ DỤNG

4.1. Cách thức hoạt động:



4.2. Các chức năng cơ bản chính

- Tối ưu hóa khả năng truy cập, sử dụng máy tính bằng các câu lệnh ngắn gọn (các từ khóa được gắn liền với hệ thống gợi ý “Lỗ đen” cho người mới sử dụng dễ dàng).
- Hệ thống gợi ý ngay khi gõ các ký tự tìm kiếm đầu tiên.
- Khả năng khai thác dữ liệu từ Internet nhanh, gọn (tìm tài liệu, ảnh, video, tìm đường nhanh).
- Tạo cho người dùng trải nghiệm sử dụng đa phương tiện mới lạ.
- Có khả năng phản hồi cuộc trò chuyện với nội dung đơn giản.



4.3. Các chức năng phụ - công cụ hỗ trợ khác:

- Hệ thống song ngữ (Anh – Việt) linh hoạt cho nhiều đối tượng sử dụng.
- Có thể thay đổi được ảnh đại diện và tên của trợ lý ảo tùy ý người dùng.
- Có một hệ thống “Guide” hướng dẫn sử dụng bằng clip một cách tỉ mỉ, chi tiết, rõ ràng.
- Có trang thông tin về ứng dụng, số hiệu phiên bản.
- Được gắn thêm chức năng công cụ tìm tài liệu học tập và hỗ trợ tương quan trọng việc học “Cấu trúc dữ liệu và Giải thuật”.

4.4. Hướng dẫn sử dụng chung:

Ứng dụng được xây dựng một hệ thống hỗ trợ người mới dùng dễ dàng biết cách sử dụng “i.Sekai” với một số hình ảnh gắn liền với 3 phương thức sử dụng cụ thể như sau:


1. Cách sử dụng cơ bản:





2. Sử dụng với việc dùng các từ khóa:

2. Using with keywords:




Step 1: Click on the black hole button to see keywords searching.

Step 2: Type and choose a keyword with content that you want to find into search box.

3. Sử dụng mở ứng dụng từ máy tính:

3. Open an application:



Step 1: Type “open” with name of PC’s application that you want to open.



CHƯƠNG V

KẾT QUẢ THỰC HIỆN

- **Kết quả thực hiện:**

- Hoàn thành được sản phẩm đúng yêu cầu.
- Kiểm tra ứng dụng “Trợ lý đa năng – i.Sekai” đã hoạt động đúng theo mục tiêu ban đầu mà nhóm đã đặt ra.
- Hoàn thành được báo cáo tổng thể và chuẩn bị được phần trình bày và báo cáo.

- **Đánh giá kết quả làm việc nhóm:**

- Các thành viên nhóm tích cực tham gia hoạt động chung.
- Chủ động tìm tòi, học hỏi và sắp xếp thời gian vì công việc chung.
- Biết cách phân công, chia việc hợp lý cho từng thành viên.
- Các thành viên trong nhóm cơ bản hoàn thành tốt công việc được phân công.



CHƯƠNG VI

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1. Kết luận:

Sau thời gian học tập, làm việc nhóm cùng nhau, chúng em rút ra được rất nhiều kinh nghiệm trong việc tự học, tự tìm tòi và hiểu rõ hơn về cách viết một ứng dụng bằng ngôn ngữ Java, đặc biệt là được tự mình học một bộ công cụ và các thư viện của nó để tạo ra sản phẩm. Được trực tiếp tự tay tạo ra từng thành phần, đối tượng nhỏ của sản phẩm mà không thông qua nhóm hay bên thiết kế riêng.

Ngoài ra, qua quá trình học cũng đã gắn kết các bạn lại với nhau, trau dồi khả năng giao tiếp, các kỹ năng mềm như tìm tài liệu, đọc tài liệu bằng tiếng Anh, quản lý thời gian, quản lý kế hoạch và khả năng sáng tạo, biến ước mơ thành hiện thực.

4.2. Ưu điểm – Hạn chế:

Ưu điểm: “Trợ lý đa năng – i.Sekai” là sản phẩm sáng tạo giúp tiết kiệm thời gian và mang đến nhiều trải nghiệm mới lạ cho người dùng Windows, với việc được tiếp xúc với một trợ lý ảo thân thiện như trong ứng dụng di động nhưng hiện hữu ngay trên chiếc laptop, hoặc máy tính bàn, ...

Hạn chế:

- Trong quá trình sử dụng còn một số lỗi kỹ thuật về giao diện hoặc hiển thị.
- CSDL phản hồi vẫn chưa được đa dạng như các trợ lý ảo hiện đại khác.

4.3. Hướng phát triển:

Đồ án của nhóm không chỉ là ứng dụng hạn chế sử dụng cho đối tượng các bạn sinh viên trong lớp, mà nó còn có thể phát triển hơn nữa:

- Chính vì Java là ngôn ngữ đa nền tảng, nên ứng dụng có tiềm năng phát triển trên nhiều hệ thống, nhiều hệ điều hành khác nhau.
- Ứng dụng là bước đầu trong việc phát triển về AI và Machine Learning trong việc hỗ trợ giao tiếp giữa con người và trợ lý ảo.



- Dễ dàng phát triển thêm tính năng cho ứng dụng vượt xa hệ thống máy tính gò bó như: điều khiển các thiết bị trong nhà, liên kết IoT, ...
- , ...



TÀI LIỆU THAM KHẢO

- [1] Marc Loy, Robert Eckstein, Dave Wood, James Elliott, Brian Cole, “Java Swing”.
- [2] Robert C. Martin, “Clean Code: A Handbook of Agile Software Craftsmanship”.
- [3] Stack Overflow, <https://stackoverflow.com>
- [4] Wikipedia, <https://en.wikipedia.org/wiki>

