

MỤC LỤC

MỤC LỤC	1
DANH SÁCH HÌNH VẼ	3
DANH SÁCH BẢNG.....	4
BẢNG THUẬT NGỮ	5
LỜI MỞ ĐẦU	6
CHƯƠNG 1: CƠ BẢN VỀ JAVA.....	1
1.1. PHẦN LÝ THUYẾT.....	1
1.1.1. Giới thiệu Java	1
1.1.2. Một số khái niệm	4
1.1.3. Cài đặt và Thiết lập biến môi trường trong Java.....	7
1.1.4. Chương trình Java đầu tiên	12
1.1.5. Cấu trúc chương trình Java	14
1.1.6. Các kiểu dữ liệu nguyên thủy	14
1.1.7. Tên định danh, biến số, hằng số.....	16
1.1.8. Các phép toán và các biểu thức.....	17
1.1.9. Nhập, xuất dữ liệu.....	17
1.1.10. Hàm (phương thức).....	18
1.1.11. Cấu trúc điều khiển.....	19
1.1.12. Kiểu dữ liệu mảng.....	20
1.1.13. Kiểu dữ liệu xâu ký tự	21
1.2. PHẦN BÀI TẬP.....	21
1.2.1. Bài tập mẫu	21
1.2.2. Bài tập tự giải.....	41
CHƯƠNG 2: HƯỚNG ĐÓI TƯỢNG TRONG JAVA	43
2.1. PHẦN LÝ THUYẾT.....	43
2.1.1. Định nghĩa lớp và các hàm thành phần.....	43
2.1.2. Phạm vi và các thuộc tính kiểm soát truy nhập các thành phần của lớp.....	43
2.1.3. Quan hệ kế thừa giữa các lớp.....	44
2.1.4. Giao diện và sự mở rộng quan hệ kế thừa	45
2.1.5. Nạp chồng và tính đa hình	46
2.1.6. Một số gói cơ bản	46
2.2.PHẦN BÀI TẬP.....	47
2.2.1. Bài tập mẫu	47
2.2.2. Bài tập tự giải.....	72
CHƯƠNG 3. LẬP TRÌNH GIAO DIỆN TRÊN WINDOWS	78
3.1. PHẦN LÝ THUYẾT.....	78
3.1.1. Containers	78
3.1.2. Components	78
3.1.3. Layout Manager.....	81
3.1.4. Xử lý các sự kiện	82
3.1.5. Menu	86
3.1.6. Tạo GUI với Swing.....	87
3.2. PHẦN BÀI TẬP.....	87

3.2.1. Bài tập mẫu	87
3.2. Bài tập tự giải.....	111
CHƯƠNG 4: LUỒNG VÀO VÀ TẬP TIN	114
4.1. PHẦN LÝ THUYẾT	114
4.1.1. Byte Stream.....	115
4.1.2. Character Stream.....	115
4.1.3. Standard Stream	115
4.1.4. Đọc và ghi file.....	115
4.1.5. Thư mục trong Java	117
4.1.6. ByteArrayInputStream	118
4.1.7. DataInputStream	119
4.1.8. ByteArrayOutputStream	119
4.1.9. DataOutputStream	120
4.1.10. Lớp File.....	120
4.1.11. FileReader	123
4.1.12. FileWriter	124
4.2. PHẦN BÀI TẬP.....	124
4.2.1. Bài tập mẫu	124
4.2.2. Bài tập tự giải.....	146
CHƯƠNG 5: LẬP TRÌNH CƠ SỞ DỮ LIỆU	147
5.1. PHẦN LÝ THUYẾT	147
5.1.1. Kiến trúc ODBC và JDBC	147
5.1.2. Các lớp và giao diện của JDBC API.....	148
5.1.3. Kết nối cơ sở dữ liệu qua JDBC	148
5.1.4. Kiểu dữ liệu SQL và Java	149
5.2. PHẦN BÀI TẬP.....	150
5.2.1. Bài tập mẫu	150
5.2.2. Bài tập tự giải.....	163
CHƯƠNG 6: BÀI TẬP TỔNG HỢP	167
6.1. CÂU HỎI TRẮC NGHIỆM.....	167
6.2. MỘT SỐ ĐỀ THI MẪU.....	184
6.3. DANH SÁCH ĐỀ TÀI LÀM BÀI TẬP LỚN	200
TÀI LIỆU THAM KHẢO.....	204
PHỤ LỤC 1 – CODE TRÒ CHƠI TERIS	205
PHỤ LỤC 2 – CODE CALCULATOR	212

DANH SÁCH HÌNH VẼ

Hình 1.1: Quá trình thông dịch chương trình Java trên các hệ điều hành	3
Hình 1.2: Máy ảo Java trên các hệ điều hành	5
Hình 1.3: Mối quan hệ giữa JDK, JRE và JVM	6
Hình 1.4: Các loại IDE phát triển mã nguồn	7
Hình 1.5: Biểu tượng cài đặt Java thành công	8
Hình 1.6: Minh họa mã nguồn chương trình Hello World	12
Hình 1.7: Thư mục lưu mã nguồn	12
Hình 1.8: Quá trình biên dịch và chạy chương trình Hello World	13
Hình 1.9: Quá trình biên dịch và chạy chương trình Java	14
Hình 1.10: Các kiểu dữ liệu nguyên thủy	15
Hình 3.1: Cấu trúc gói các thành phần điều khiển	78
Hình 3.2: Cấu trúc cây phân cấp các lớp của gói event	83
Hình 3.3: Cấu trúc cây phân cấp các giao diện của event listener	84
Hình 3.4: Các action listener	84
Hình 3.5: Các item listener	85
Hình 3.6: Các window listener	85
Hình 3.7: Các component listener	85
Hình 3.8: Cấu trúc lớp của Java Swing	87
Hình 4.1: Kiến trúc Input Stream	114
Hình 4.2: Kiến trúc Output Stream (Luồng xuất dữ liệu)	115
Hình 5.1: Kiến trúc JDBC	147

DANH SÁCH BẢNG

Bảng 1.1. Kiểu dữ liệu nguyên thủy	15
Bảng 1.2. Các chỉ định truy xuất và bô nghĩa	19
Bảng 1.3. Khai báo mảng và ví dụ	20
Bảng 2.1: Ý nghĩa của các bô ngũ	43
Bảng 2.2: Một số gói cơ bản.....	46
Bảng 3.1. Phương thức của Label	79
Bảng 3.2. Phương thức của TextField	79
Bảng 3.3. Phương thức của TextArea	79
Bảng 3.4. Các biến thành viên của GridBagConstraints	81
Bảng 3.5. Các biến dữ liệu tĩnh là các giá trị cho biến fill	82
Bảng 3.6. Các sự kiện	82
Bảng 4.1: Các phương thức khác nhau và mô tả tương ứng của lớp FileInputStream	116
Bảng 4.3: Các phương thức khác nhau và mô tả tương ứng của lớp ByteArrayInputStream	118
Bảng 4.4: Các phương thức khác nhau và mô tả tương ứng của lớp DataInputStream	119
Bảng 4.5: Các phương thức khác nhau và mô tả tương ứng của lớp ByteArrayOutputStream	119
Bảng 4.6: Các phương thức khác nhau và mô tả tương ứng của lớp DataOutputStream	120
Bảng 4.7: Các phương thức khác nhau và mô tả tương ứng của lớp File	121
Bảng 4.8: Các phương thức khác nhau và mô tả tương ứng của lớp FileReader	123
Bảng 4.9: Các phương thức khác nhau của File Writer và mô tả tương ứng của lớp FileWriter	124
Bảng 5.1: Một số lớp trong gói java.sql	148
Bảng 5.2: Một số giao diện trong gói java.sql.....	148
Bảng 5.3: Các kiểu dữ liệu SQL và kiểu của Java	149
Bảng 6.1: Đáp án câu hỏi trắc nghiệm.....	183

BẢNG THUẬT NGỮ

Từ viết tắt	Từ gốc	Giải nghĩa - tạm dịch
GUI	Graphical User Interface	Giao diện đồ họa
JVM	Java Virtual Machine	Máy ảo Java
JRE	Java Runtime Environment	Môi trường chạy Java
API	Application Programming Interface	Giao diện lập trình ứng dụng
JDK	Java Development Kit	Bộ công cụ phát triển ứng dụng bằng ngôn ngữ Java

LỜI MỞ ĐẦU

Ngôn ngữ lập trình Java ra đời năm 1994, từ đó đến nay, Java luôn là ngôn ngữ lập trình được nhiều lập trình viên sử dụng để phát triển các ứng dụng thương mại, đặc biệt là các ứng dụng trên các thiết bị di động. Ngôn ngữ lập trình Java luôn đứng ở tốp đầu trong nhóm các ngôn ngữ lập trình được ưa chuộng nhất trên thế giới.

Trong chương trình đào tạo ngành Hệ thống Thông tin ở trường Đại học công nghệ Giao thông vận tải, học phần "Ngôn ngữ lập trình Java" có khối lượng kiến thức là 3 tín chỉ và thuộc khối kiến thức chuyên sâu của ngành.

Để sinh viên có hệ thống bài tập thực hành nhằm củng cố kiến thức, tự học, tự luyện tập sau mỗi phần được giới thiệu trên lớp, đồng thời cung cấp cho sinh viên một số các bài tập nâng cao, nhóm tác giả đã biên soạn tài liệu tham khảo này.

Tài liệu gồm 6 chương:

Chương 1. Cơ bản về Java: Cung cấp cho sinh viên kiến thức tổng quan, nền tảng về ngôn ngữ lập trình Java

Chương 2. Hướng đối tượng trong Java: Cung cấp cho sinh viên cách xây dựng lớp, xây dựng giao diện, tính đóng gói, che dấu dữ liệu, các loại kế thừa và tính đa hình.

Chương 3. Lập trình giao diện trên Windows: Cung cấp các kiến thức cơ bản lập trình giao diện trên Windows.

Chương 5. Luồng vào ra và tập tin: Cung cấp các kiến thức cơ bản về luồng (streams) và tập tin (files)

Chương 5. Lập trình cơ sở dữ liệu: Gồm một số bài tập về kết nối với các hệ quản trị cơ sở dữ liệu SQL Server, Access với cầu nối JDBC. Chương này là phần kiến thức nâng cao, sinh viên tham khảo để làm bài tập lớn.

Chương 6. Bài tập tổng hợp: Cung cấp cho sinh viên một số câu hỏi trắc nghiệm, một số bài tập tổng hợp để sinh viên làm việc theo nhóm (bài tập lớn), một số đề thi mẫu để sinh viên luyện tập trước khi thi.

Cấu trúc trình bày từ chương 1 đến chương 5 như sau:

- *Phản lý thuyết:* Trình bày tóm tắt lý thuyết.
- *Bài tập mẫu:* Trình bày các bài tập có hướng dẫn giải chi tiết.
- *Bài tập tự giải:* Các bài tập không có lời giải, sinh viên tự viết chương trình.

Phần mã chương trình của các bài tập được viết để có thể thực thi trên môi trường lập trình Jcreator, NetBeans, Eclipse, Jbuilder, Jdeveloper, v.v. Sinh viên có thể tùy chọn môi trường lập trình.

Trong quá trình biên soạn, nhóm tác giả có tham khảo một số tài liệu như Core Java Volume I – Fundamentals, Core Java Volume II - Fundamentals tác giả Cay S. Horstmann, Gary Cornell, và Lập trình Java của tác giả Đoàn Văn Ban.

Phân công biên soạn: Chương 1, 2 do tác giả Lê Chí Luận đảm nhiệm. Chương 2, 3 do tác giả Nguyễn Thái Sơn đảm nhiệm và chương 5 do tác giả Trần Nguyên Hương đảm nhiệm. Chương 6, do tác giả Lê Thị Hường đảm nhiệm.

Nhóm tác giả xin chân thành cảm ơn PGS. TS. Phạm Ngọc Hùng – Chủ nhiệm bộ môn Công nghệ phần mềm trường Đại học Công nghệ – Đại học Quốc Gia Hà Nội, TS. Đặng Đức Hạnh bộ môn Công nghệ phần mềm trường Đại học Công nghệ – Đại học Quốc Gia Hà Nội và các đồng nghiệp trong khoa Công nghệ thông tin, trường Đại học Công nghệ Giao thông vận tải đã đóng góp ý kiến quý báu để chúng tôi xây dựng tài liệu này.

Tuy đã hết sức cố gắng nhưng chắc chắn không thể tránh được thiếu sót, nhóm tác giả rất mong nhận được các ý kiến đóng góp của bạn đọc để tài liệu ngày càng hoàn thiện hơn. Thông tin góp ý xin gửi tới địa chỉ email: luanlc@utt.edu.vn, sonnt@utt.edu.vn.

Hà Nội, tháng 06/2017

Nhóm tác giả

CHƯƠNG 1: CƠ BẢN VỀ JAVA

1.1. PHẦN LÝ THUYẾT

1.1.1. Giới thiệu Java

a. Java là gì?

Java là một ngôn ngữ lập trình hướng đối tượng, dựa trên các lớp, thường được sử dụng cho các hệ thống có tính độc lập cao. Khẩu hiệu của Java là "*write once, run everywhere*" (viết một lần, chạy mọi nơi), nghĩa là đoạn code Java sau khi được biên dịch có thể chạy được trên tất cả các nền tảng hỗ trợ Java mà không cần phải được biên dịch lại. Các ứng dụng Java sau khi đã được biên dịch thành bytecode có thể chạy trên bất kỳ máy ảo Java (JVM).

Cho đến năm 2017, Java vẫn được đánh giá là một trong những ngôn ngữ lập trình được dùng phổ biến nhất trên thế giới, đặc biệt cho các ứng dụng web client-server. Theo thống kê trên thế giới có khoảng hơn 9 triệu lập trình viên Java.

b. Lịch sử Java

Năm 1990, Sun Microsystems thực hiện dự án Green nhằm phát triển phần mềm trong các thiết bị dân dụng. James Gosling, chuyên gia lập trình đã tạo ra một ngôn ngữ lập trình mới có tên là Oak (cây sồi). Ngôn ngữ này có cú pháp gần giống như C++ nhưng bỏ qua các tính năng nguy hiểm của C++ như truy cập trực tiếp tài nguyên hệ thống, con trỏ, định nghĩa chồng các tác từ v.v.

Khi ngôn ngữ Oak trưởng thành, WWW cũng đang vào thời kỳ phát triển mạnh mẽ, Sun cho rằng đây là một ngôn ngữ thích hợp cho Internet. Năm 1995, Oak đổi tên thành Java (tên một hòn đảo ở Indonesia) và sau đó đến 1996 Java đã được xem như một chuẩn công nghiệp cho Internet. Năm 1995, tạp chí Time bình chọn Java là một trong 10 sản phẩm tốt nhất năm 1995. JDK 1.0 được công bố vào 23/1/1996.

Năm 2010 Oracle mua lại bản quyền Java từ Sun systems và tiếp tục phát triển, đã có nhiều phiên bản Java được công bố. Phiên bản Java hiện tại là Java SE 8. Hiện tại, Java được sử dụng trong lập trình mạng, các thiết bị mobile, game, các giải pháp về e-business, v.v.[4]

c. Java được sử dụng ở đâu?

Có rất nhiều thiết bị hiện tại đang sử dụng Java gồm:

- Desktop Application như media player, antivirus, reader, v.v.
- Web Application như irctc.co.in, javatpoint.com, v.v.
- Enterprise Application như các ứng dụng về xử lý nghiệp vụ ngân hàng, v.v.
- Trên các thiết bị Mobile.

d. Các loại ứng dụng Java

Có bốn loại ứng dụng chính mà có thể được tạo bởi sử dụng ngôn ngữ lập trình Java:

Ứng dụng StandAlone: Còn được biết đến với tên gọi khác là ứng dụng Desktop Application hay ứng dụng Windows-based. Một ứng dụng mà chúng ta cần cài đặt trên mỗi thiết bị như media player, antivirus, v.v. AWT và Swing được sử dụng trong Java để tạo các ứng dụng StandAlone.

Ứng dụng Web: Một ứng dụng mà chạy trên phía Server và tạo các trang động (Dynamic Page), được gọi là ứng dụng Web. Hiện tại, các công nghệ như Servlet, JSP, Struts, JSF, v.v được sử dụng để tạo ứng dụng Web trong Java.

Ứng dụng Enterprise: Một ứng dụng dạng như Banking, có lợi thế là tính bảo mật cao, cân bằng tải (load balancing) và clustering. Trong Java, EJB được sử dụng để tạo các ứng Enterprise.

Ứng dụng Mobile: Đây là loại ứng dụng được tạo cho thiết bị di động. Hiện tại thì Android và Java ME là hai công nghệ được sử dụng để tạo loại ứng dụng này.

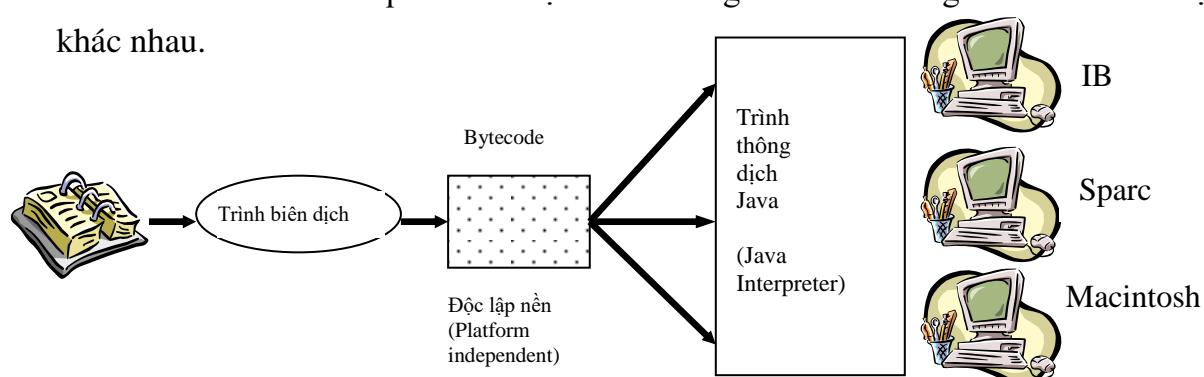
e. Các đặc trưng của Java

- *Đơn giản:* Những người thiết kế mong muốn phát triển một ngôn ngữ dễ học và quen thuộc với đa số người lập trình. Do vậy Java loại bỏ các đặc trưng phức tạp của C và C++ như thao tác con trỏ, thao tác định nghĩa chòng toán tử (operator overloading),... Java không sử dụng lệnh “goto” cũng như file header (.h). Cấu trúc “struct” và “union” cũng được loại bỏ khỏi Java.

- *Hướng đối tượng:* Java được thiết kế xoay quanh mô hình hướng đối tượng. Vì vậy trong Java, tiêu điểm là dữ liệu và các phương pháp thao tác lên dữ liệu đó. Dữ liệu và các phương pháp mô tả trạng thái và cách ứng xử của một đối tượng trong Java.

- *Độc lập phần cứng và hệ điều hành:* Đây là khả năng một chương trình được viết tại một máy nhưng có thể chạy được bất kỳ đâu. Chúng được thể hiện ở mức mã nguồn và mức nhị phân. Ở mức mã nguồn, người lập trình cần mô tả kiểu cho mỗi biến. Kiểu dữ liệu trong Java nhất quán cho tất cả các hệ điều hành và phần cứng khác nhau. Java có riêng một thư viện các lớp cơ sở. Vì vậy chương trình Java được viết trên một máy có thể dịch và chạy trên tru trên các loại máy khác mà không cần viết lại. Tính độc lập ở mức nhị phân, một chương trình đã biên dịch có thể chạy trên nhiều nền (phần cứng, hệ điều hành) khác mà không cần dịch lại mã nguồn. Tuy vậy cần có phần mềm máy ảo Java hoạt động như một trình thông dịch tại máy thực thi.

Hình sau chỉ ra quá trình thực thi chương trình viết bằng Java trên các loại máy khác nhau.



Hình 1.1: Quá trình thông dịch chương trình Java trên các hệ điều hành

Môi trường phát triển của Java được chia làm hai phần: Trình biên dịch và trình thông dịch. Không như C hay C++, trình biên dịch của Java chuyển mã nguồn thành dạng bytecode độc lập với phần cứng mà có thể chạy trên bất kỳ CPU nào.

Nhưng để thực thi chương trình dưới dạng bytecode, tại mỗi máy cần phải có trình thông dịch của Java hay còn gọi là máy ảo Java. Máy ảo Java chuyển bytecode thành mã lệnh mà CPU thực thi được.

Mạnh mẽ: Java là ngôn ngữ yêu cầu chặt chẽ về kiểu dữ liệu. Phải khai báo kiểu dữ liệu tường minh khi viết chương trình. Java kiểm tra lúc biên dịch và cả trong thời gian thông dịch vì vậy Java loại bỏ một số loại lỗi lập trình nhất định.

Java không sử dụng con trỏ và các phép toán con trỏ. Java kiểm tra tất cả các truy nhập đến mảng, chuỗi khi thực thi để đảm bảo rằng các truy nhập đó không ra ngoài giới hạn kích thước. Java kiểm tra sự chuyển đổi kiểu dữ liệu từ dạng này sang dạng khác lúc thực thi.

Trong các môi trường lập trình truyền thống, lập trình viên phải tự mình cấp phát bộ nhớ. Trước khi chương trình kết thúc thì phải tự giải phóng bộ nhớ đã cấp. Vấn đề nảy sinh khi lập trình viên quên giải phóng bộ nhớ đã xin cấp trước đó. Trong chương trình Java, lập trình viên không phải bận tâm đến việc cấp phát bộ nhớ. Quá trình cấp phát, giải phóng được thực hiện tự động, nhờ dịch vụ thu nhặt những đối tượng không còn sử dụng nữa (garbage collection).

Cơ chế bẫy lỗi của Java giúp đơn giản hóa quá trình xử lý lỗi và hồi phục sau lỗi.

Bảo mật: Viruses là nguyên nhân gây ra sự lo lắng trong việc sử dụng máy tính. Trước khi có Java, các lập trình viên phải quét virus các tệp trước khi tải về hay thực hiện chúng. Thông thường việc này cũng không loại trừ hoàn toàn virus. Ngoài ra chương trình khi thực thi có khả năng tìm kiếm và đọc các thông tin nhạy cảm trên máy của người sử dụng mà người sử dụng không hề hay biết.

Java cung cấp một môi trường quản lý thực thi chương trình. Nó cho rằng không có một đoạn mã nào là an toàn cả. Và vì vậy Java không chỉ là ngôn ngữ lập trình thuận tuý mà còn cung cấp nhiều mức độ kiểm soát tính an toàn khi thực thi chương trình.

Ở mức đầu tiên, dữ liệu và các phương thức được đóng gói bên trong lớp. Chúng chỉ được truy xuất thông qua các giao diện mà lớp cung cấp. Java không hỗ trợ con trỏ vì vậy không cho phép truy xuất bộ nhớ trực tiếp. Nó cũng ngăn chặn không cho truy xuất thông tin bên ngoài kích thước của mảng bằng kỹ thuật tràn và cũng cung cấp kỹ thuật dọn rác trong bộ nhớ. Các đặc trưng này tạo cho Java an toàn tối đa và có khả năng cơ động cao.

Trong lớp thứ hai, trình biên dịch kiểm soát để đảm bảo mã là an toàn, và tuân theo các nguyên tắc của Java.

Lớp thứ ba được đảm bảo bởi trình thông dịch. Chúng kiểm tra xem bytecode có đảm bảo các qui tắc an toàn trước khi thực thi.

Lớp thứ tư kiểm soát việc nạp các lớp vào bộ nhớ để giám sát việc vi phạm giới hạn truy xuất trước khi nạp vào hệ thống.

Phân tán: Java có thể dùng để xây dựng các ứng dụng có thể làm việc trên nhiều phần cứng, hệ điều hành và giao diện đồ họa. Java được thiết kế hỗ trợ cho các ứng dụng chạy trên mạng. Vì vậy chúng được sử dụng rộng rãi như là công cụ phát triển trên Internet, nơi sử dụng nhiều nền tảng khác nhau.

Đa luồng: Chương trình Java đa luồng(Multithreading) để thực thi các công việc đồng thời. Chúng cũng cung cấp giải pháp đồng bộ giữa các luồng. Đặc tính hỗ trợ đa này cho phép xây dựng các ứng dụng trên mạng chạy hiệu quả.

Tính động: Java được thiết kế như một ngôn ngữ động để đáp ứng cho những môi trường mở. Các chương trình Java chứa rất nhiều thông tin thực thi nhằm kiểm soát và truy nhập đối tượng lúc chạy. Điều này cho phép khả năng liên kết động mã.[1]

1.1.2. Một số khái niệm

a. Java Platform

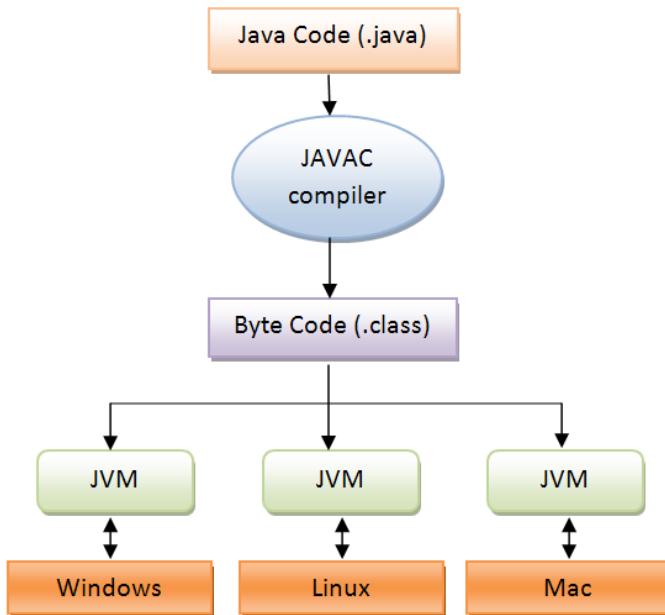
Platform là một môi trường phần cứng hoặc phần mềm để chạy chương trình . Ví dụ như Microsoft Windows, Mac OS, Linux là các platform . Java platform là tên cho một nhóm các chương trình của Sun cho phép phát triển và chạy chương trình viết bằng ngôn ngữ Java

Java platform có 2 thành phần: JVM (máy ảo java) và API (Java Application Programming Interface). API là một tập hợp lớn các phần mềm đã được làm sẵn rất hữu dụng . Các phần mềm đó được nhóm lại trong những thư viện liên quan với nhau. Những thư viện đó được gọi là các gói (packages).

b. Java Virtual Machine (JVM):

Java Virtual Machine (Máy ảo Java) là một phần mềm dựa trên cơ sở máy tính ảo. Nó có tập hợp các lệnh logic để xác định các hoạt động của máy tính.

Một chương trình viết bằng ngôn ngữ lập trình Java sẽ được biên dịch ra mã của máy ảo java (mã java bytecode). Sau đó máy ảo Java chịu trách nhiệm chuyển mã java bytecode thành mã máy tương ứng. Sun Microsystem chịu trách nhiệm phát triển các máy ảo Java chạy trên các hệ điều hành trên các kiến trúc CPU khác nhau.



Hình 1.2: Máy ảo Java trên các hệ điều hành

Một Java Virtual Machine có 3 thành phần chính là:

Class-Loader Subsystem : chuyên tìm kiếm và load các file .class vào vùng nhớ của Java.

Runtime Data Area : vùng nhớ hệ thống cấp phát cho Java Virtual Machine.

Execution Engine: chuyển các lệnh của JVM trong file .class thành các lệnh của máy, hệ điều hành tương ứng và thực thi chúng.

f. **Java Runtime Environment (JRE)** - môi trường thực thi Java: Cung cấp các Java API, máy ảo Java và các thành phần cần thiết khác để chạy các applet và ứng dụng viết bằng ngôn ngữ lập trình Java. Môi trường thực thi Java không có các công cụ và tiện ích như là các trình biên dịch hay các trình gỡ lỗi để phát triển các applet và các ứng dụng.

Java Runtime library là thư viện các class đã được compiled sẵn và đặt trong file jre\lib\rt.jar. Các class java.lang.String, java.lang.Object, v.v. nằm trong gói jar này.
JRE = JVM + Java Runtime library: bạn sẽ thấy java.exe, javaw.exe và rt.jar trong thư mục cài đặt jre.

Java Development Kit (JDK) - Bộ công cụ phát triển ứng dụng bằng ngôn ngữ Java: Là một tập hợp những công cụ phần mềm được phát triển bởi Sun Microsystems dành cho các nhà phát triển phần mềm, dùng để viết những applet Java hay những ứng dụng Java – bộ công cụ này được phát hành miễn phí gồm có trình biên dịch, trình thông dịch, trình giúp sửa lỗi (debugger, trình chạy applet và tài liệu nghiên cứu), JDK bao gồm:

- JRE (Java Runtime Environment) là một môi trường chạy ứng dụng Java.

- Javac: Một chương trình để dịch mã mà bạn viết thành mã bytecode, khi ứng dụng Java chạy nó dịch mã bytecode thành mã máy tính và thực thi, điều đó có nghĩa là bytecode chỉ là một mã trung gian.

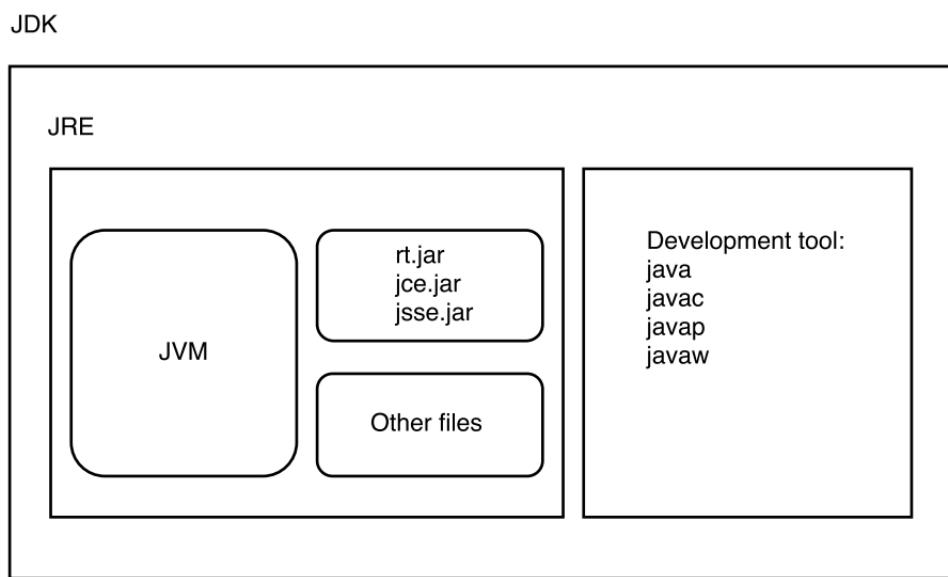
- Archive (jar): Là một chương trình nén các file thành một file duy nhất có đuôi jar. Thường dùng để đóng gói các file class.

- Javadoc: Là một công cụ tạo ra tài liệu hướng dẫn sử dụng API.

- Và các công cụ khác cần thiết cho phát triển Java.

JDK = JRE + tools (javac.exe, keytool.exe, v.v.) + document (help, samples,..) + src.zip

Mối quan hệ và sự khác nhau giữa JDK, JRE và JVM chúng ta có thể mô tả bằng hình vẽ sau:



Hình 1.3: Mối quan hệ giữa JDK, JRE và JVM

Có nhiều phiên bản JDK đã phát hành. Phiên bản đầu tiên JDK 1.0 phát hành tháng 01 năm 1996, phiên bản hiện tại JDK 8, phát hành tháng 3 năm 2014

Bộ công cụ phát triển (JDK), download tại:

<http://www.oracle.com/technetwork/java/archive-139210.html>

g. Integrated Development Environment (IDE)

Là một môi trường phát triển tích hợp (IDE), về bản chất nó là một chương trình để viết code. Chương trình này hỗ trợ nhiều tính năng tự động hóa cho người phát triển. Chẳng hạn các gợi ý khi lập trình, tự hoàn thiện mã, v.v.

Với Java bạn có nhiều sự lựa chọn IDE, nó là môi trường để viết code. Sau đây là một số IDE:



Hình 1.4: Các loại IDE phát triển mã nguồn

NotePad: Trên hệ điều hành Windows bạn có thể sử dụng trình soạn thảo text đơn giản như notePad.

NetBeans: Là một chương trình IDE mã nguồn mở mà miễn phí, bạn có thể download tại link sau: <http://www.netbeans.org/index.html>

Eclipse: Là một trình IDE được phát triển bởi cộng đồng mã nguồn mở eclipse, có thể được download tại địa chỉ <http://www.eclipse.org/>

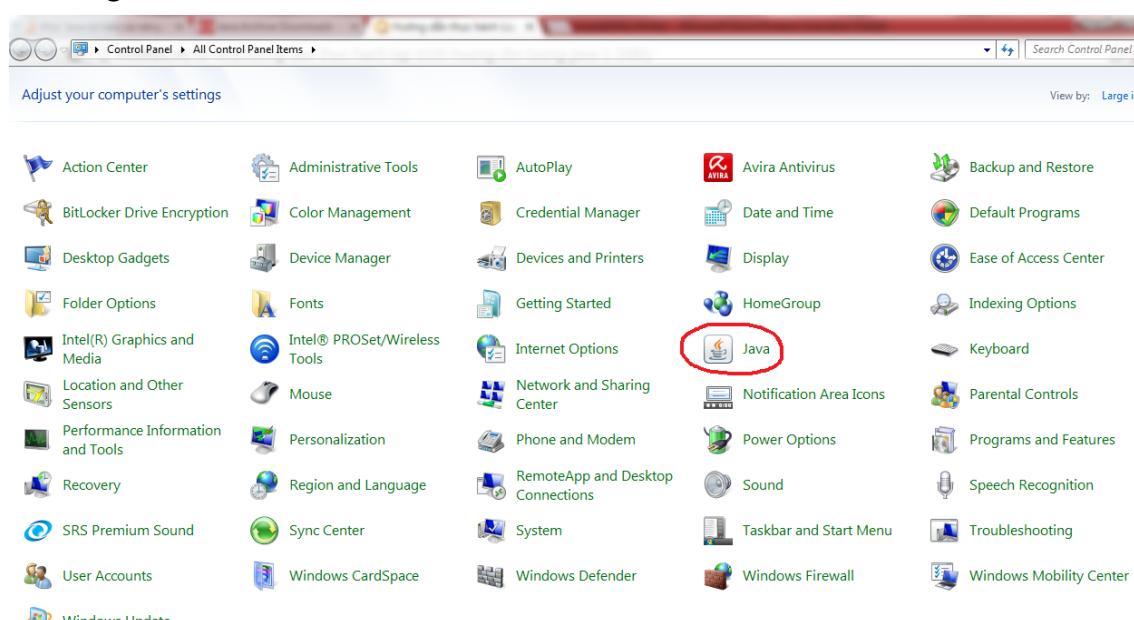
Jcreator: www.jcreator.org/download.htm

Jbuilder: www.embarcadero.com/products/jbuilder

1.1.3. Cài đặt và Thiết lập môi trường trong Java

Để lập trình bằng ngôn ngữ Java bạn cần download và cài JDK trước sau đó lựa chọn IDE để cài. Trong giới hạn giáo trình này chúng tôi sử dụng Java SE 8 và NetBeans để phát triển các ứng dụng Java.

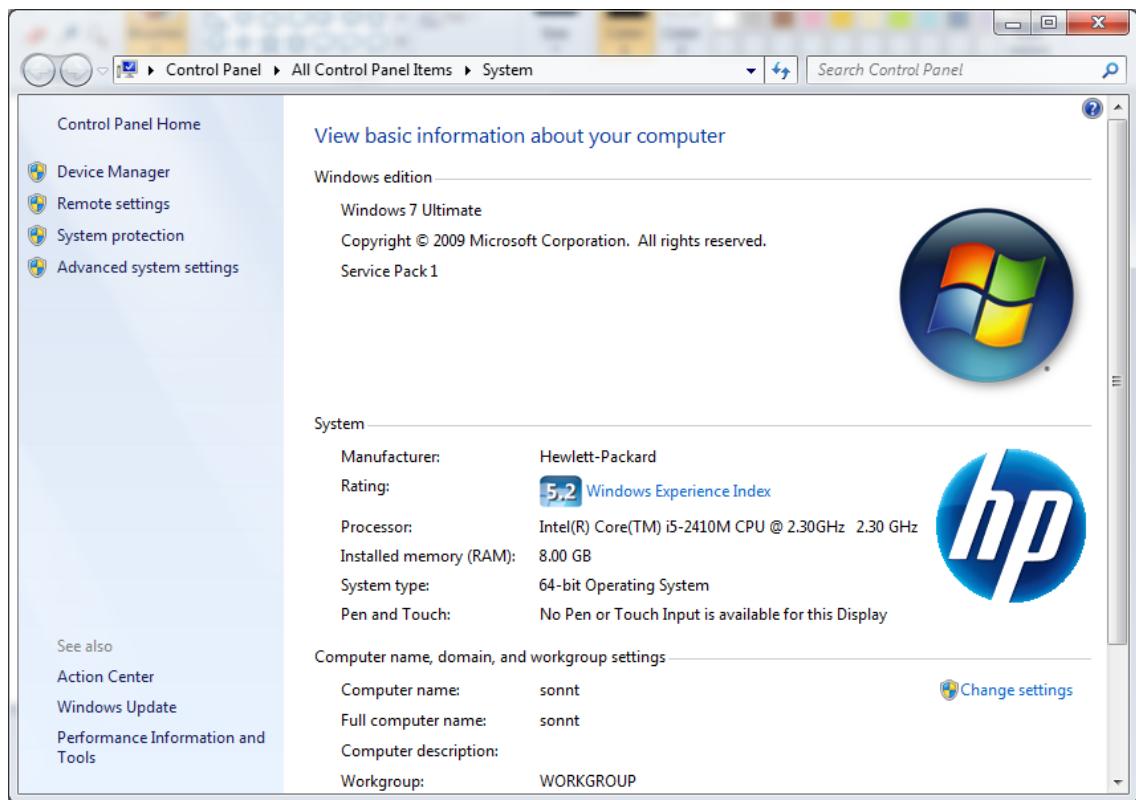
Giả sử rằng máy tính của bạn 64 bits, bạn sẽ chọn Windows x64 để tải. Sau khi tải và cài đặt thành công bạn vào Control Panel để kiểm tra xem máy ảo Java đã được cài đặt vào máy hay chưa. Nếu có biểu tượng Java như hình bên dưới thì coi như bạn đã cài đặt thành công.



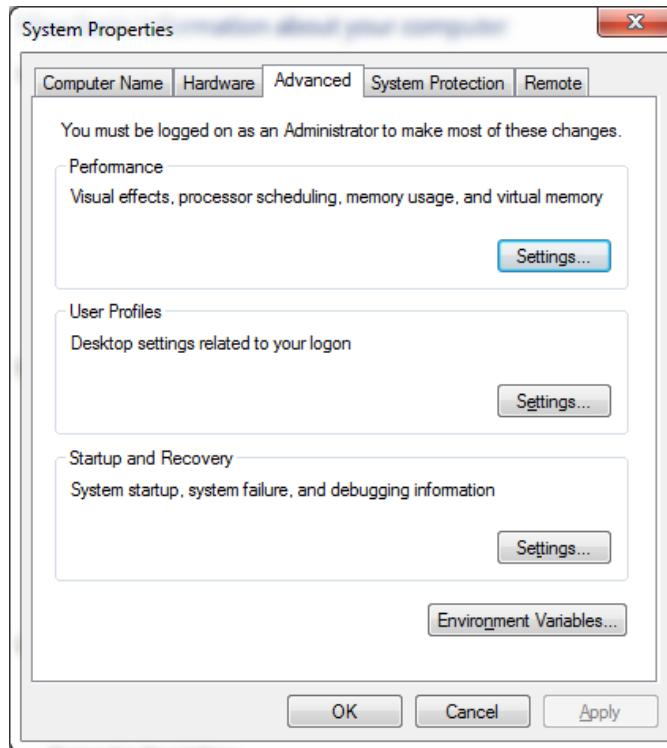
Hình 1.5: Biểu tượng cài đặt Java thành công

Thiết lập biến môi trường trong Java là rất cần thiết nếu bạn cần sử dụng các công cụ liên quan đến javac hoặc java (dùng ở cơ chế command-line). Sau đây là các bước chi tiết để thiết lập biến môi trường.

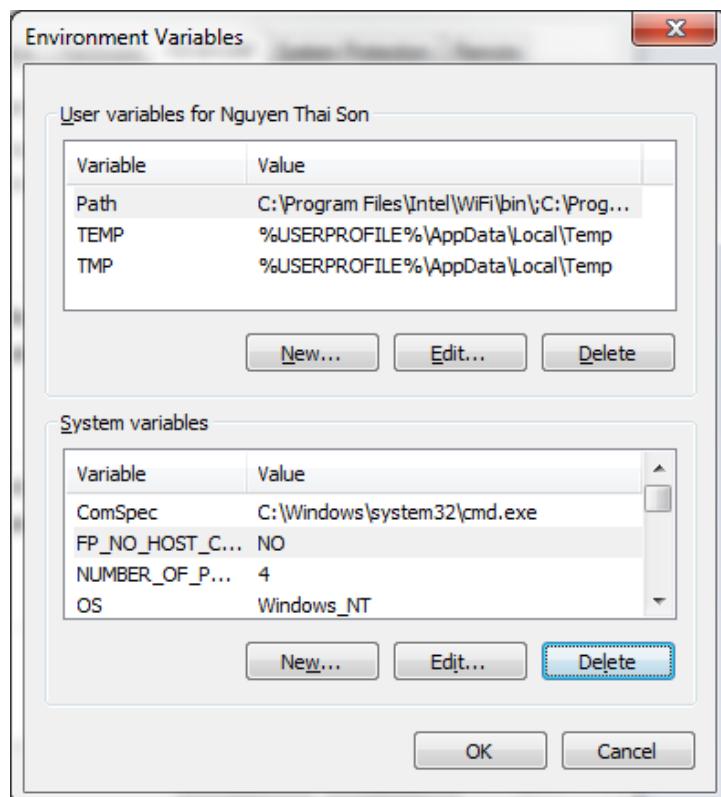
- Bấm chuột phải vào biểu tượng **Computer/ chọn Properties**



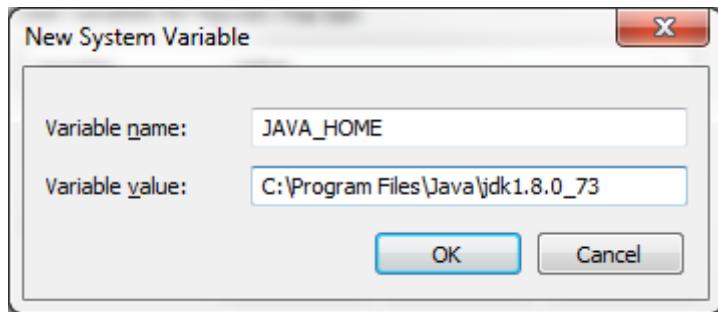
- Xuất hiện cửa sổ mới, chúng ta chọn **Advanced System Settings**



- Cửa sổ System Properties sẽ hiển thị ra, chúng ta vào tab **Advanced**, click chọn button **Environment Variables**, v.v. Cửa sổ Environment Variables sẽ hiển thị như hình bên dưới:



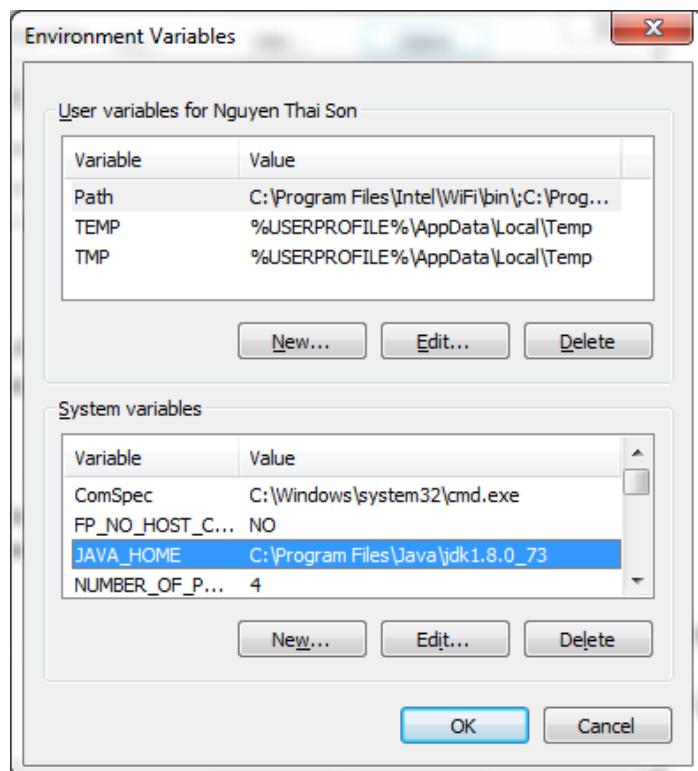
- Trong mục System variables, click chọn button **New**



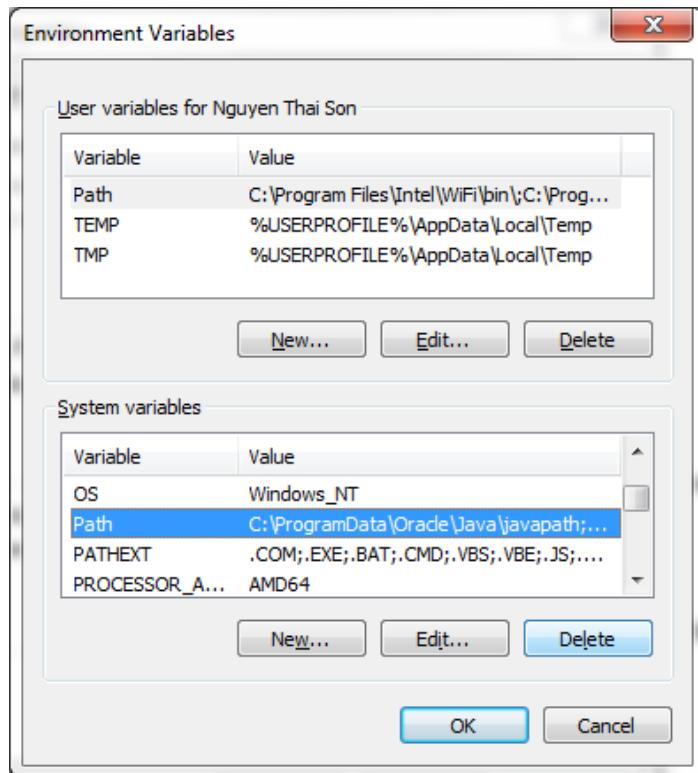
Mục **Variables name**, chúng ta nhập vào biến JAVA_HOME

Mục **Variables value**, chúng ta copy đường dẫn đã cài đặt JDK vào đây. Trong giáo trình này chúng tôi đã cài đặt JDK theo đường dẫn như hình trên.

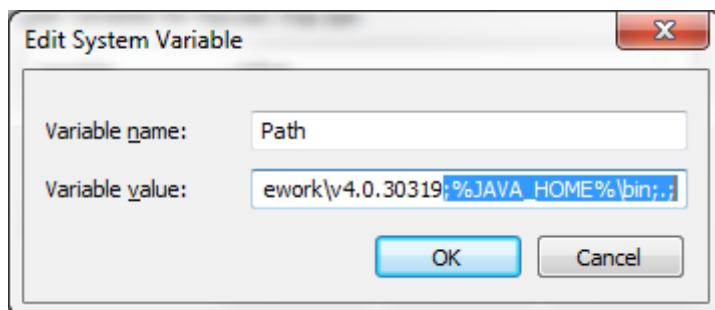
- Sau khi chọn OK, bạn quan sát vùng System Variables, biến JAVA_HOME và giá trị của nó sẽ xuất hiện như hình minh họa bên dưới.



- Bước tiếp theo là bạn tìm đến biến **Path** trong mục System variables, click chọn Edit



Trong mục Variables value, bạn di chuyển tới cuối, nhập vào các giá trị như hình minh họa: ;%JAVA_HOME%\bin;;

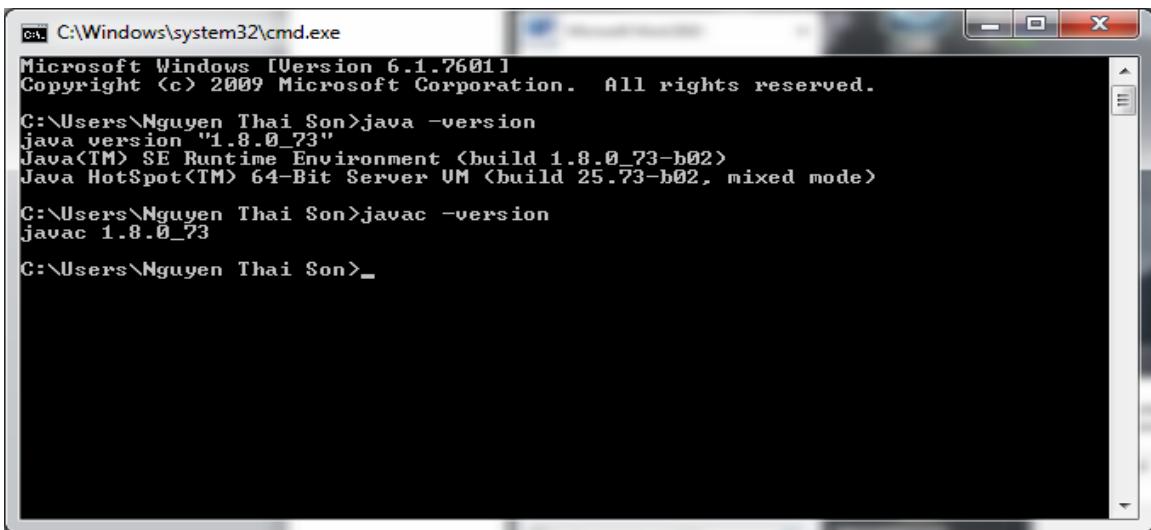


Click các button OK để kết thúc quá trình thiết lập biến môi trường.

- Kiểm tra lại cấu hình có chính xác hay không:
Để kiểm tra xem máy tính của bạn đang cài JDK version bao nhiêu, vào **Start/ Run**. Hoặc gõ tổ hợp phím Windows+R để hiện thị cửa sổ Run, trong cửa sổ này các bạn gõ vào lệnh **cmd** rồi nhấn phím Enter.

Gõ **java –version** để kiểm tra JDK version

Gõ **javac –version** để kiểm tra javac version



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Nguyen Thai Son>java -version
java version "1.8.0_73"
Java(TM) SE Runtime Environment (build 1.8.0_73-b02)
Java HotSpot(TM) 64-Bit Server VM (build 25.73-b02, mixed mode)

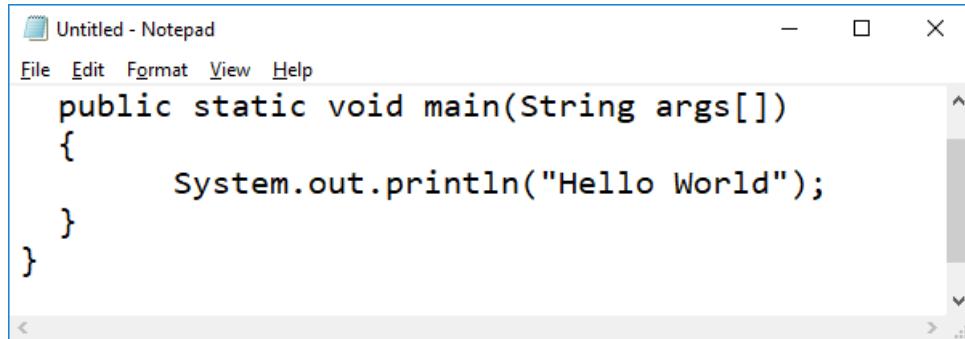
C:\Users\Nguyen Thai Son>javac -version
javac 1.8.0_73

C:\Users\Nguyen Thai Son>
```

1.1.4. Chương trình Java đầu tiên

Trong phần này, chúng ta sẽ học cách viết một chương trình Java đơn giản trong IDE notepad.

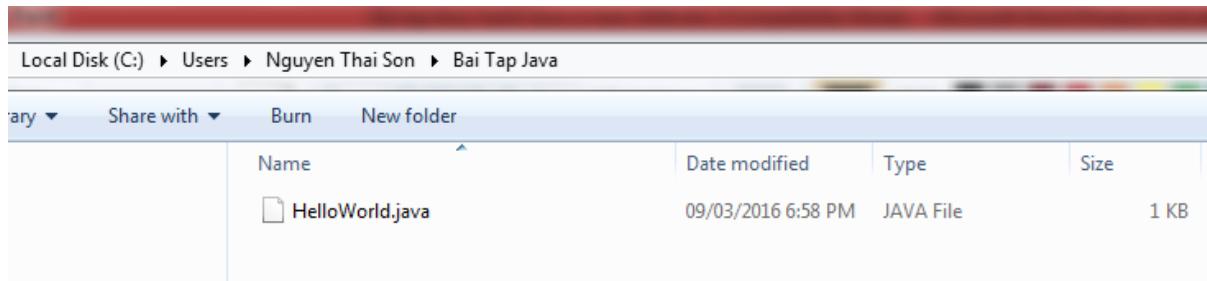
Để viết một chương trình đơn giản, bạn mở Notepad bởi **Start menu -> All Programs -> Accessories -> Notepad** và viết chương trình đơn giản Hello World như sau:



```
Untitled - Notepad
File Edit Format View Help
public static void main(String args[])
{
    System.out.println("Hello World");
}
```

Hình 1.6: Minh họa mã nguồn chương trình Hello World

Viết và lưu chương trình dưới dạng HelloWorld.java.



Hình 1.7: Thư mục lưu mã nguồn

Để biên dịch và chạy chương trình này, bạn cần mở dòng nhắc lệnh Command prompt bởi **Start menu -> All Programs -> Accessories -> Command prompt**.

Để biên dịch và chạy chương trình, đầu tiên bạn vào thư mục chứa file HelloWorld.java, với chúng tôi đó là C:\User\Nguyen Thai Son\Bai Tap Java.

- Để biên dịch, gõ: **javac HelloWorld.java**
- Để thực thi, gõ: **java HelloWorld**

Sau khi biên dịch và thực thi chương trình, kết quả như hình minh họa sau:

The screenshot shows a Microsoft Windows Command Prompt window titled "Command Prompt". The window displays the following text:
Microsoft Windows [Version 6.1.7601]
Copyright © 2009 Microsoft Corporation. All rights reserved.
C:\Users\Nguyen Thai Son>cd "Bai Tap Java"
C:\Users\Nguyen Thai Son\Bai Tap Java>javac HelloWorld.java
C:\Users\Nguyen Thai Son\Bai Tap Java>java HelloWorld
Hello World
C:\Users\Nguyen Thai Son\Bai Tap Java>

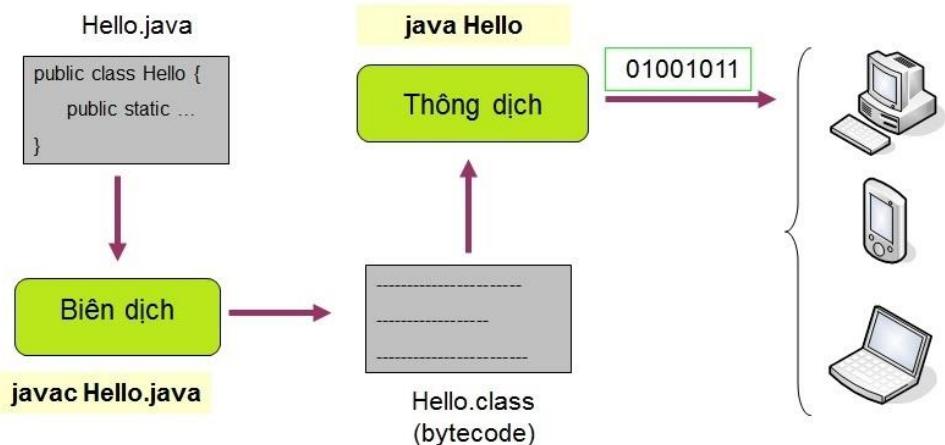
Hình 1.8: Quá trình biên dịch và chạy chương trình Hello World

Phân tích chi tiết chương trình Java đầu tiên

Dưới đây, chúng tôi sẽ giúp bạn hiểu ý nghĩa của class, public, static, void, main, String[], System.out.println().

- Từ khóa **class** được sử dụng để khai báo một lớp trong Java.
- Từ khóa **public** là một Access Modifier biểu diễn phạm vi truy xuất, nghĩa rằng nó được truy xuất ở mọi nơi.
- **static** là một từ khóa, mà nếu chúng ta khai báo bất cứ phương thức nào là static thì nó còn được gọi là phương thức tĩnh hoặc phương thức static. Lợi thế chủ yếu của phương thức static là không cần thiết tạo đối tượng để triệu hồi phương thức static. Phương thức main được thực thi bởi JVM, vì thế bạn không cần thiết tạo một đối tượng để gọi phương thức main. Việc này giúp tiết kiệm bộ nhớ.
- **void** là kiểu trả về của phương thức, nghĩa là phương thức không trả về bất cứ giá trị nào.
- **main** là hàm bắt buộc phải có trong chương trình Java, nó là điểm bắt đầu để chạy chương trình.
- **String[] args** được sử dụng cho tham số dòng lệnh..
- **System.out.println()** được sử dụng để in "HelloWorld" lên màn hình.

Quá trình biên dịch và chạy chương trình Java



Hình 1.9: Quá trình biên dịch và chạy chương trình Java

1.1.5. Cấu trúc chương trình Java

Một chương trình Java cơ bản có cấu trúc sau:

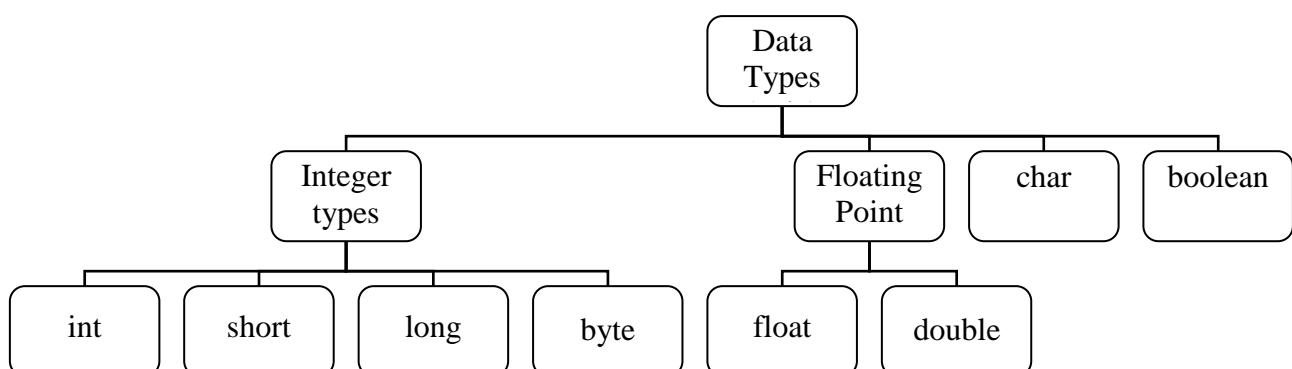
```

//Nhập các thư viện
import thư_viện;
//Xây dựng các lớp
public class clsMain{
...
    public static void main(String args[])
    {
        //điểm bắt đầu của chương trình
    }
...
    class lớp 1
    {
        ...
    }
    class lớp 2
    {
        ...
    }
}

```

Chương trình Java có thể tổ chức thành nhiều lớp, các lớp có thể đặt trên các file .java khác nhau gọi là một project. Trong các tệp tin class, bắt buộc có một tệp tin chứa hàm main()

1.1.6. Các kiểu dữ liệu nguyên thủy



Hình 1.10: Các kiểu dữ liệu nguyên thủy

Bảng 1.1. Kiểu dữ liệu nguyên thủy

Kiểu dữ liệu	Độ dài theo số bit	Phạm vi
byte	8	-128 đến 127
char	16	'\u0000' to '\uffff'
boolean	1	"True" hoặc "False"
short	16	-32768 đến 32767
int	32	-2,147,483,648 đến +2,147,483,648
long	64	-9,223,372,036,854,775,808 đến +9,223,372,036,854,775,808
float	32	-3.40292347E+38 đến +3.40292347E+38
double	64	-1,79769313486231570E+308 đến +1,79769313486231570E+308

Các qui tắc chuyển đổi kiểu

* Ép kiểu: Qui tắc ép kiểu có dạng: (<type>) <exp>

Lúc thực hiện hệ thống sẽ chuyển kết quả tính toán của biểu thức <exp> sang kiểu được ép là <type>.

Ví dụ: float f = (float)100.15; // Chuyển số 100.15 dạng kiểu double sang float

Lưu ý: Không cho phép chuyển đổi giữa các kiểu nguyên thủy với kiểu tham chiếu, ví dụ kiểu double không thể ép sang các kiểu lớp như lớp được.

Kiểu giá trị boolean (logic) không thể chuyển sang các kiểu dữ liệu số và ngược lại.

* Mở rộng và thu hẹp kiểu: Giá trị của kiểu hẹp hơn (chiếm số byte ít hơn) có thể được chuyển sang những kiểu rộng hơn (chiếm số byte nhiều hơn) mà không tổn thất thông tin. Cách chuyển kiểu đó được gọi là mở rộng kiểu.



Ví dụ:

char c = 'A';

int k = c; // mở rộng kiểu char sang kiểu int (mặc định)

Chuyển đổi kiểu theo chiều ngược lại, từ kiểu rộng về kiểu hẹp hơn được gọi là thu hẹp kiểu. Lưu ý là thu hẹp kiểu có thể dẫn tới mất thông tin.

* Một số hàm toán học: Lớp Math

Lớp final class Math định nghĩa một tập các hàm tĩnh để thực hiện các chức năng chung của toán học như các phép làm tròn số, sinh số ngẫu nhiên, tìm số cực đại, cực tiểu, v.v.

Lớp final class Math còn cung cấp những hằng số như số e (cơ số của logarithm), số pi thông qua Math.E và Math.PI.

Một số hàm thường dùng:

- *static int abs(int i) / static long abs(long l) /static float abs(float f)/ static double abs(double d)*: trả lại giá trị tuyệt đối của đối số
- *static double ceil(double d)*: trả lại giá trị nhỏ nhất kiểu *double* mà không nhỏ hơn đối số và lại bằng số nguyên.
- *static double floor(double d)*: trả lại giá trị lớn nhất kiểu *double* mà không lớn hơn đối số và lại bằng số nguyên.
- *static int round(float d)/static long round(double d)*: trả lại số nguyên gần nhất của đối số.

Các hàm lũy thừa

- *static double pow(double d1, double d2) : trả lại giá trị là lũy thừa của d1 và d2 ($d1^{d2}$)*.
- *static double exp(double d)*: trả lại giá trị là luỹ thừa cơ số *e* và số mũ *d* (e^d).
- *static double log(double d)*: trả lại giá trị là lô-ga-rit tự nhiên (cơ số *e*) của *d*.
- *static double sqrt(double d)*: trả lại giá trị là căn bậc hai của *d* ($d >= 0$).
- *static double random()*n* : sinh một số ngẫu nhiên từ 0 đến n-1

1.1.7. Tên định danh, biến số, hằng số

a. Tên định danh (identifier):

- Dùng để đặt tên cho biến, hằng, hàm, kiểu dữ liệu, lớp, giao diện v.v.
- Là dãy ký tự liên tiếp (không chứa dấu cách) và phải bắt đầu bằng chữ cái hoặc gạch dưới.
- Phân biệt kí tự in hoa và thường.
- Không được trùng với từ khóa.

b. Hằng:

- Khai báo: sử dụng từ khóa *final*
- Ví dụ:

```
final double CM_PER_INCH = 2.54;  
double paperWidth = 8.5;  
double paperHeight = 11;
```

c. Biến:

- Khai báo: <Kiểu dữ liệu> <Tên biến>;
 - Nếu khai báo nhiều biến thì mỗi biến cách nhau bởi dấu phẩy.
 - Có thể vừa khai báo vừa khởi tạo giá trị cho biến.
 - Java cho phép khai báo biến mọi nơi trong chương trình. Phạm vi hoạt động của biến từ vị trí khai báo đến hết hàm (với biến cục bộ) hoặc hết chương trình (với biến toàn cục).

Ví dụ:

```
int a=5; //Khai báo biến a kiểu int  
int x, y, z; //Khai báo nhiều biến  
double dien_tich =0, chu_vi = 1; //Khai báo và khởi tạo giá trị cho biến
```

1.1.8. Các phép toán và các biểu thức

Biểu thức: Tạo thành từ các toán tử (phép toán), toán hạng (số hạng) và các dấu mở đóng ngoặc. Toán tử tác động lên giá trị của các toán hạng và cho giá trị có kiểu nhất định.

- Các toán hạng: hằng, biến, lời gọi hàm
- Các toán tử cơ bản:
 - Toán tử số học với kiểu số nguyên: +, -, *, / (chia lấy phần nguyên), % (phép chia lấy phần dư)
 - Toán tử số học với kiểu số thực: +, -, *, / (chia lấy số thực)
 - Toán tử logic: && (phép và), || (hoặc), ! (phủ định)
 - Toán tử quan hệ: >, <, >=, <=, ==, !=.
 - Toán tử gán: =
 - Toán tử tăng, giảm: ++, -- (dạng tiền tố và hậu tố)
 - Dạng tiền tố: tăng (giảm) một đơn vị trước khi tham gia vào biểu thức tính toán.
 - Dạng hậu tố: tham gia vào biểu thức tính toán sau đó mới tăng (giảm) một đơn vị.

1.1.9. Nhập, xuất dữ liệu

a. Xuất dữ liệu:

- Xuất ra màn hình hằng văn bản:

```
System.out.print("Hằng văn bản") ;
```

hoặc System.out.println("Hằng văn bản") ;

- Xuất ra màn hình dòng thông báo và giá trị của biến số:

```
System.out.printf("Hằng văn bản" +biến số) ;
```

b. Nhập dữ liệu sử dụng lớp BufferedReader, khai báo: import java.io.*;

- Nhập xâu ký tự có chứa dấu cách:

```
BufferedReader x=new BufferedReader (new InputStreamReader(System.in)) ;  
System.out.print("Nhập một xâu s:") ;  
String s = x.readLine(); //đọc vào 1 dòng
```

- Nhập số một nguyên:

```
try  
{  
    BufferedReader x=new BufferedReader (new InputStreamReader(System.in)) ;  
    System.out.print("Nhập vào một số a=");  
    int a=Integer.parseInt (x.readLine()); //chuyển đổi kiểu  
}  
catch (Exception ex)  
{  
    System.out.println("Loi: " +ex.getMessage());  
}
```

Chuyển đổi kiểu từ xâu sang số thực double: Double.parseDouble(), số thực float : Float.parseFloat(), số nguyên long : Long.parseLong(), số nguyên byte: Byte.parseByte().

c. Nhập dữ liệu sử dụng lớp Scanner, khai báo: import java.util.*;

- Nhập xâu ký tự không chứa dấu cách:

```
System.out.println("Nhập vào 1 chuỗi:");
Scanner x=new Scanner(System.in); //x là tên biến
String s = x.nextLine(); //s là tên biến
```

- Nhập vào một số:

```
try { Scanner x=new Scanner(System.in); //x là tên biến
        System.out.print("Nhập số nguyên a=");
        int a=x.nextInt(); //nhập vào số nguyên
    }
catch (Exception ex)
{ System.out.println("Loi: " +ex.getMessage());
}
```

Để nhập các số có kiểu dữ liệu khác, ta sử dụng

```
double b=x.nextDouble(); float c=x.nextFloat(); byte d=x.nextByte();
```

1.1.10. Hàm (phương thức)

Cú pháp

<Chỉ định truy xuất> [<bổ nghĩa loại phương thức>]<kiểu dữ liệu> <Tên phương thức> (danh sách tham số)

```
{ //phần thân
}
```

Chỉ định truy xuất của phương thức:

- **public:** Phương thức có chỉ định truy xuất **public** có thể được nhìn thấy từ mọi gói hoặc mọi lớp.
- **protected:** Các lớp mở rộng từ lớp hiện hành trong cùng một gói, hoặc tại các gói khác nhau có thể truy cập các phương thức sử dụng chỉ định truy xuất này.
- **Private:** Phương thức riêng tư có thể được truy cập nhờ phương thức công cộng trên cùng một lớp.

Các bổ nghĩa loại phương thức

- **Tĩnh (static):** Các trạng thái mà phương thức có thể được thay đổi mà không cần đến đối tượng. Nó chỉ được sử dụng đối với các dữ liệu và các phương thức tĩnh.
- **Trừu tượng (abstract):** Ngu ý rằng phương thức không có một mã cụ thể (code) và nó sẽ được bổ sung ở các lớp con (subclass). Loại phương thức này được sử dụng trong các lớp kế thừa.
- **Kết thúc (final):** Phương thức không thể được thừa kế hoặc ghi đè (Overridden).
- **Tự nhiên (native):** Chỉ ra rằng phần thân của phương thức được viết trên các ngôn ngữ khác Java ví dụ C, hoặc C++.
- **Đồng bộ (synchronized):** Sử dụng với phương thức trong quá trình thực thi threads. Nó cho phép chỉ một thread được truy cập vào khối mã vào một thời điểm.

- **Linh hoạt (volatile):** Được sử dụng với các biến để thông báo rằng giá trị của biến có thể được thay đổi vài lần khi thực thi chương trình và giá trị của nó không được ghi vào thanh ghi.

Bảng dưới đây chỉ ra nơi mà các chỉ định truy xuất và bô nghĩa được sử dụng:

Bảng 1.2. Các chỉ định truy xuất và bô nghĩa

Chỉ định truy xuất/Bô nghĩa	Phương thức	Biến	Lớp
public	Yes	Yes	Yes
private	Yes	Yes	Yes (Nested class)
protected	Yes	Yes	Yes (Nested class)
abstract	Yes	No	Yes
final	Yes	Yes	Yes
native	Yes	No	No
volatile	No	Yes	No

1.1.11. Cấu trúc điều khiển

① Cấu trúc rẽ nhánh if:

- **if** dạng khuyết:
 if (<Điều kiện>) <Khối lệnh>;
- **if** dạng đầy đủ: **if ... else**
 - if** (<Điều kiện>)
 <Khối lệnh 1>;
 - else**
 <Khối lệnh 2>;

② Cấu trúc lựa chọn switch

```
switch (<Biểu thức lựa chọn> )
{
    case <Biểu thức 1>:
        <Khối lệnh 1>; break;
    case <Biểu thức 2>:
        <Khối lệnh 2>; break;
    .....
    [default:
        <Khối lệnh n>]
}
```

③ Lặp có số bước lặp xác định for

```
for (<Khởi tạo>; <Điều kiện>; <Biến điều khiển>)
    <Khối lệnh>;
```

④ Cấu trúc lặp không xác định

<p>Lặp với điều kiện trước:</p> <p>while (<Điều kiện>) { <Khối lệnh> }</p>	<p>Lặp với điều kiện sau:</p> <p>do { <Khối lệnh> }</p> <p>while (<Điều kiện>);</p>
---	---

⑤. Lặp “for each”

```
for (variable : collection) { statements; }
```

Ví dụ:

```
int[] x = { 2, 3, 5, 7, 11, 13 };
for (int element : x) System.out.println(element);
```

⑥. Lệnh thoát khỏi cấu trúc lặp và hàm

- **break**: thoát khỏi vòng lặp **while**, **do while**, **for**, và rẽ nhánh **switch**.
- **continue**: bỏ qua phần còn lại của vòng lặp **for**, **while**, **do while** và nhảy sang vòng lặp tiếp theo.
- **return**: trả về giá trị cho hàm, thoát khỏi hàm.

⑦. Cấu trúc try...catch

```
try {
    ... // khối lệnh cần được giám sát lỗi
}
catch (Exception_Type_1 exOb) {
    ... // đoạn mã xử lý khi kiểu ngoại lệ/lỗi Exception_Type_1 xảy ra
}
catch (Exception_Type_2 exOb) {
    ... // đoạn mã xử lý khi kiểu ngoại lệ/lỗi Exception_Type_2 xảy ra
}
...
finally {
    // khối lệnh cần thực hiện trước khi kết thúc
}
```

trong đó `Exception_Type_1`, `Exception_Type_2` ... là kiểu ngoại lệ vừa xảy ra,

1.1.12. Kiểu dữ liệu mảng

* Mảng 1 chiều

Mảng có thể được khai báo bằng ba cách :

Bảng 1.3. Khai báo mảng và ví dụ

Cách khai báo	Cú pháp	Ví dụ
Chỉ đơn thuần khai báo mảng	Datatype identifier[]	<code>char ch[];</code> khai báo mảng char có tên ch
Khai báo và cấp phát bộ nhớ cho các phần tử mảng sử dụng từ “new”	Datatype identifier[] <code>=new datatype [size]</code>	<code>char ch[]=new char [10];</code> Khai báo một mảng ch và lưu trữ 10 ký tự
Khai báo mảng, cấp phát bộ nhớ cho nó và gán các giá trị ban đầu cho các phần tử của mảng.	Datatype identifier[] <code>= {value1,value2...valueN };</code>	<code>char ch [] = {'A','B','C','D' };</code> khai báo mảng ch và lưu 4 chữ cái kiểu ký tự

Chỉ số của mảng bắt đầu từ 0. Để truy cập vào phần tử của mảng ta viết `tên_mảng[chỉ số]`

* Mảng 2 chiều

Ví dụ :

```
int[][] matrix;
matrix = new int[4][5];
for (int row=0; row < 4; row++)
```

```

    {   for (int col=0; col < 5; col++)
        { matrix[row][col] = row+col; }
    }

```

Ngoài ra, java còn cung cấp các mảng nhiều chiều và mảng jagged.

1.1.13. Kiểu dữ liệu xâu ký tự

Khai báo: String <tên biến> = new String;

Các phương thức:

- int length(): cho độ dài xâu
- char charAt(int index): cho ký tự thứ index của xâu,
- boolean equals(Object obj): So sánh các xâu,
- String substring(int startIndex): cho kết quả là một xâu con được lấy ra từ vị trí startIndex đến cuối của xâu.
- boolean startWith(String str), boolean endWith(String str): Hai hàm này cho kết quả true nếu xâu bắt đầu (kết thúc) bằng đối số str.

1.2. PHẦN BÀI TẬP

1.2.1. Bài tập mẫu

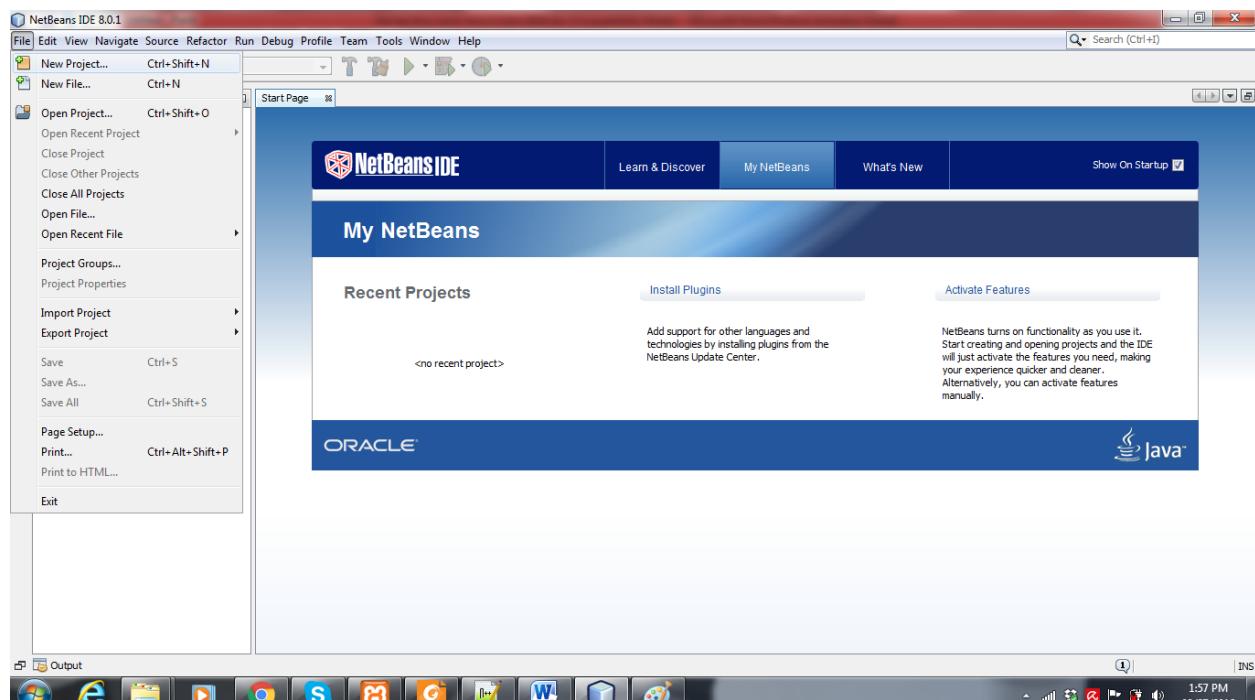
Bài 1.1. Hướng dẫn sử dụng NetBeans IDE

Viết chương trình in lên màn hình dòng chữ “Hello world!”

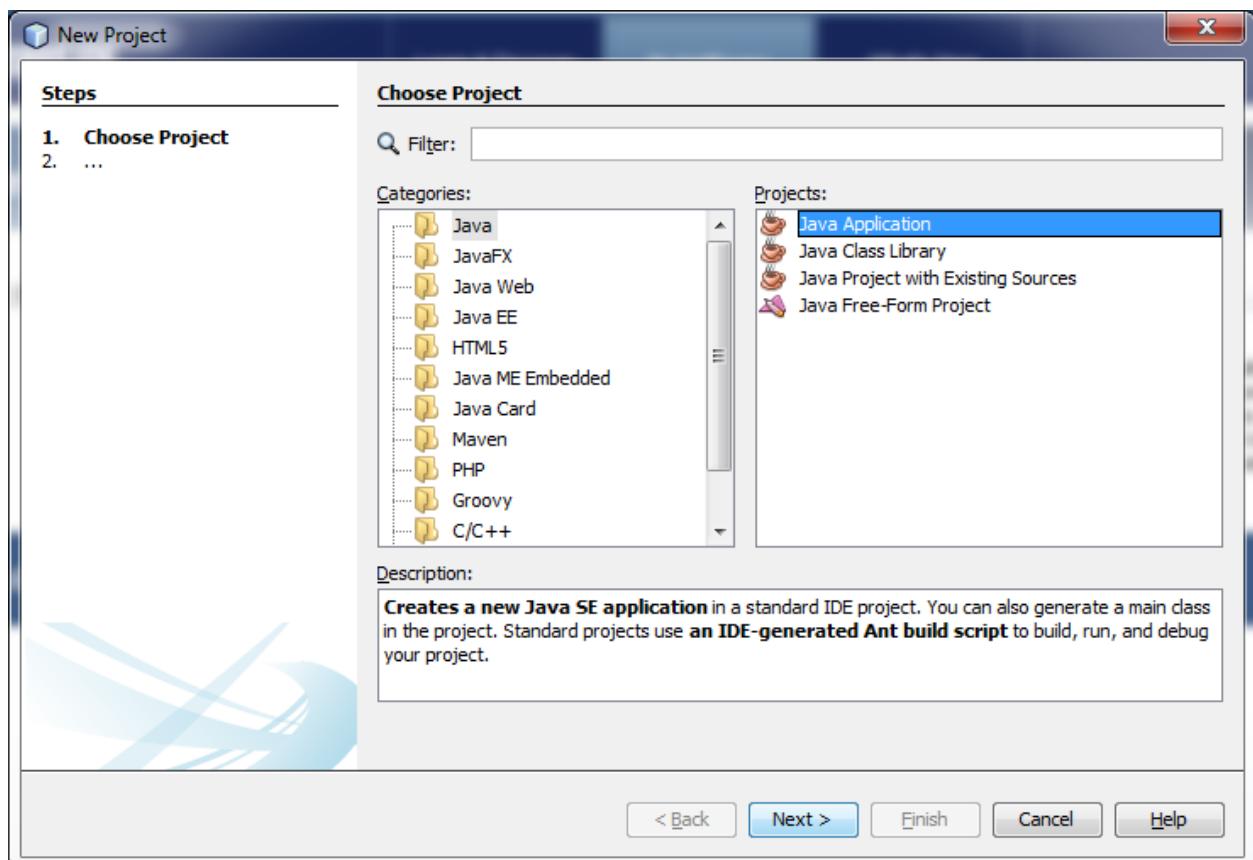
Hướng dẫn

* Sử dụng NetBeans IDE

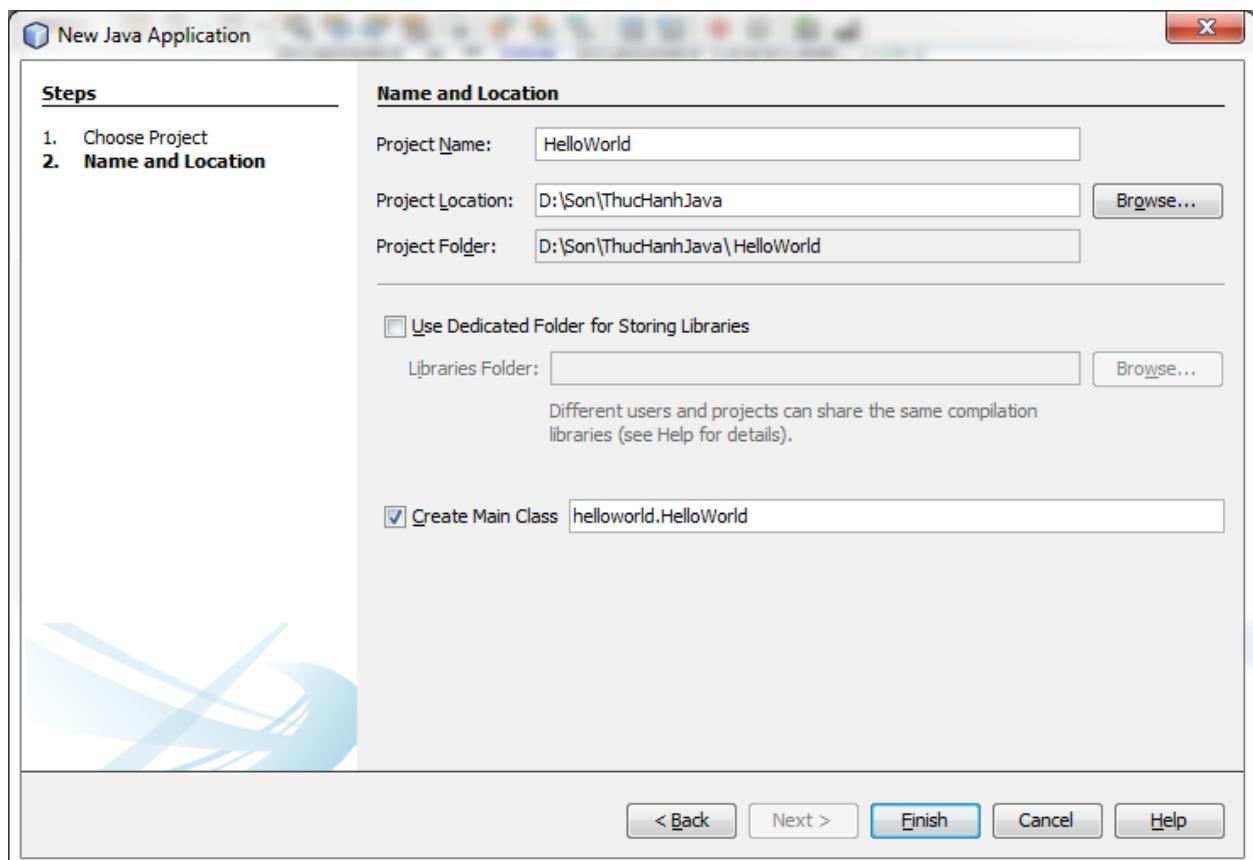
Khởi động NetBeans IDE, chọn File / New Project (hoặc Ctrl+Shift+N)



xuất hiện cửa sổ:



Trong Categories chọn Java, trong Project, chọn Java Application, sau đó chọn Next, xuất hiện cửa sổ



Đặt tên cho Project trong Project Name (chẳng hạn HelloWorld), có thể chọn đường dẫn chứa Project và một số công việc khác. Sau đó chọn Finish.

```

1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package helloworld;
7
8  /**
9  *
10 * @author Nguyen Thai Son
11 */
12 public class HelloWorld {
13
14     /**
15      * @param args the command line arguments
16     */
17     public static void main(String[] args) {
18         // TODO code application logic here
19     }
20
21 }
22

```

Trong hàm **main()**, gõ `System.out.println("Hello World!");`
Nhấn F6 để dịch và chạy chương trình. Tại cửa sổ Output xuất hiện dòng chữ Hello World!.

HelloWorld là tên lớp, trùng với tên của Project và tên file chương trình nguồn `HelloWorld.java`.

Trong **class** `HelloWord` có một hàm bắt buộc là hàm **main()**, dòng lệnh `System.out.println("Hello World!");` để in lên màn hình (cửa sổ General Output) dòng chữ Hello World!

* Sử dụng Jcreator và Eclipse

Đây là môi trường tích hợp để viết chương trình java, C++,v.v tương đối thuận tiện, phổ biến bạn đọc tự tìm hiểu.

* **Chú ý:** Các bài tập trong tài liệu này đã chạy được trên các môi trường Net Bean IDE ver 8.0.1, các IDE khác các bạn tự kiểm chứng.

■ Bài 1.2. Viết chương trình nhập vào 2 số nguyên a, b. Tính tổng S=a+b. Sử dụng đối tượng của lớp BufferedReader để nhập

a. Hướng dẫn :

- Khai báo import `java.io.*` ;
- Sử dụng đối tượng của lớp `BufferedReader` và `InputStreamReader` để nhập
- Sử dụng hàm chuyển đổi kiểu sang kiểu nguyên `Integer.parseInt()`

b. Chương trình mẫu:

```

package tinhcong;
import java.io.*;
public class TinhTong {
    public static void main(String[] args) {
        try {
            BufferedReader x = new BufferedReader(new

```

```

InputStreamReader(System.in));
        System.out.print("NHAP A=");
        //chuyển đổi kiểu
        int a = Integer.parseInt(x.readLine());
        System.out.print("NHAP B=");
        //chuyển đổi kiểu
        int b = Integer.parseInt(x.readLine());
        int S = a + b;
        System.out.println("Tong = " + S);
    } catch (Exception e) {
        System.out.println("Lỗi: " + e.getMessage());
    }
}
}

```

█ Kết quả chạy chương trình:

```

NHAP A=10
NHAP B=5
Tong =15

```

 **Bài 1.3. Sử dụng cấu trúc if, if else. Sử dụng đối tượng Scanner để nhập**

Chương trình nhập vào hai số a và b, tìm giá trị lớn nhất (max) của hai số rồi in kết quả lên màn hình.

a. Hướng dẫn :

- Khai báo import java.util.* ;
- Sử dụng đối tượng của Scanner để nhập
- Sử dụng hàm nextInt() để chuyển đổi từ xâu sang số nguyên

b. Chương trình mẫu:

```

package maxnumber;
import java.util.*;
public class MaxNumber {
    public static void main(String[] args) {
        try {
            Scanner x = new Scanner(System.in);
            System.out.print("NHAP A=");
            int a = x.nextInt();
            System.out.print("NHAP B=");
            int b = x.nextInt();
            int max = a;
            if (max < b) {
                max = b;
            }
            System.out.println("Gia tri lon nhat = " + max);
        } catch (Exception e) {
            System.out.println("Lỗi:" + e.getMessage());
        }
    }
}

```

█ Kết quả chạy chương trình:

```
NHAP A=10
NHAP B=5
Gia tri lon nhat = 10
```

❑ **Bài 1.4. Sử dụng cấu trúc if else lồng nhau**

Chương trình giải phương trình bậc 2: $ax^2+bx+c=0$

a. Hướng dẫn:

- Giải phương trình theo định lý Viết
- Để tính căn bậc 2 của số thực d, ta sử dụng Math.sqrt(d)
- Hiện thông báo lỗi trong khôi try...catch(Exception e), sử dụng lệnh System.out.print(e.getMessage())

b. Chương trình mẫu:

```
package gaiiptb2;
import java.util.*;
public class Gaiiptb2 {
    public static void main(String[] args) {
        try {
            Scanner input= new Scanner(System.in);
            System.out.print("NHAP A=");
            double a= input.nextDouble();
            System.out.print("NHAP B=");
            double b= input.nextDouble();
            System.out.print("NHAP C=");
            double c= input.nextDouble();
            double delta=b*b-4*a*c;
            if (delta>0)
            {
                double x1 = (-b+Math.sqrt(delta))/(2*a);
                double x2 = (-b-Math.sqrt(delta))/(2*a);
                System.out.println("x1="+x1+" x2="+x2);
            }
            else if (delta==0)
            {
                double x = (-b)/(2*a);
                System.out.println("Nghiệm kép x1=x2="+x);
            }
            else
                System.out.println("PT Vô nghiệm");
        }
        catch(Exception e)
        {
            System.out.println("Loi "+e.getMessage());
        }
    }
}
```

❑ Kết quả chạy chương trình:

```
NHAP A=1
NHAP B=2
NHAP C=-3
x1=1.0 x2=-3.0
```

Trường hợp PT có nghiệm kép

```
NHAP A=1
NHAP B=2
NHAP C=1
x1=x2=-1.0
```

Bài 1.5. Sử dụng cấu trúc switch.

Viết chương trình cho phép người dùng nhập vào 2 số nguyên a và b và một trong các phép toán gồm +, -, *, / (Chia lấy phần nguyên), % (chia lấy phần dư). Sau đó hiển thị kết quả tương ứng. Ví dụ nếu nhập 2 số 10, 20 và phép toán là + thì thông báo "Kết quả là 30" v.v.

a. Hướng dẫn:

Việc cộng, trừ, nhân hay chia a với b còn phụ thuộc vào phép toán (toán tử) mà người dùng nhập vào là gì. Do vậy, để ra quyết định là thực hiện phép toán nào lên 2 toán hạng a và b đó, cần sử dụng cấu trúc **switch** để kiểm tra toán tử nhập vào.

b. Chương trình mẫu:

```
// Calculator.java
package calculator;
import java.util.*;
import java.io.*;
public class Calculator {
    public static void main(String[] args) {
        try {
            Scanner input = new Scanner(System.in);
            System.out.print("NHAP A=");
            int a = input.nextInt();
            System.out.print("NHAP B=");
            int b = input.nextInt();
            int KetQua;
            System.out.print("Nhập phép toán:");
            char PhepToan = (char) System.in.read();
            switch (PhepToan) {
                case '+':
                    KetQua = a + b;
                    System.out.print("Kết quả=" + KetQua);
                    break;
                case '-':
                    KetQua = a - b;
                    System.out.print("Kết quả=" + KetQua);
                    break;
                case '*':
                    KetQua = a * b;
                    System.out.print("Kết quả=" + KetQua);
                    break;
                case '/':
                    KetQua = a / b;
                    System.out.print("Kết quả=" + KetQua);
                    break;
                case '%':
                    KetQua = a % b;
                    System.out.print("Kết quả=" + KetQua);
                    break;
            }
        } catch (Exception e) {
            System.out.println("Lỗi: " + e.getMessage());
        }
    }
}
```

```
        System.out.print("Ket qua=" + KetQua);
        break;
    default:
        System.out.print("Ban nhap sai phep toan roi");
        break;
    }
} catch (Exception e) {
    System.out.println("Loi" + e.getMessage());
}
}

}
}
```

💻 Kết quả chạy chương trình:

```
NHAP A=10
NHAP B=20
Nhập phép toán: +
Kết quả = 30
```

📘 **Bài 1.6. Sử dụng cấu trúc switch**

Lập chương trình phân loại học sinh theo điểm sử dụng cấu trúc **switch**. Điểm nhập vào là các số nguyên từ 0 đến 10. Phân loại như sau:

- Điểm 0, 1, 2, 3: Xếp loại Kém
- Điểm 4: Loại Yếu
- Điểm 5, 6: Loại Trung bình
- Điểm 7, 8: Loại Khá
- Điểm 9, 10: Loại Giỏi

a. Hướng dẫn

Khi thực hiện một lựa chọn, nếu không sử dụng lệnh **break** thì chương trình sẽ thực hiện lựa chọn tiếp theo. Sử dụng lợi thế này để viết chương trình trong trường hợp một công việc có nhiều lựa chọn.

b. Chương trình mẫu

```
//XepLoai.java
package xeploai;
import java.util.*;
public class XepLoai {
    public static void main(String[] args) {
        try {
            Scanner in = new Scanner(System.in);
            System.out.print("\nVào địa điểm:");
            System.out.print("\nĐiểm=");
            int diem = in.nextInt();
            switch (diem) {
                case 0:
                case 1:
                case 2:
                case 3:
                    System.out.print("Kém");
                case 4:
                    System.out.print("Yếu");
                case 5:
                case 6:
                    System.out.print("Trung bình");
                case 7:
                case 8:
                    System.out.print("Khá");
                case 9:
                case 10:
                    System.out.print("Giỏi");
            }
        }
    }
}
```

```

        break;
    case 4:
        System.out.print("Yeu");
        break;
    case 5:
    case 6:
        System.out.print("TB");
        break;
    case 7:
    case 8:
        System.out.print("Kha");
        break;
    case 9:
    case 10:
        System.out.print("Gioi");
        break;
    default:
        System.out.print("Vao sai");
    }
} catch (Exception e) {
    System.out.println("Loi" + e.getMessage());
}
}

```

 **Kết quả chạy chương trình:**

```

Vao du lieu:
Diem=9
Gioi

```

 **Bài 1.7. Sử dụng cấu trúc lặp for**

Viết chương trình liệt kê các số nguyên dương có 3 chữ số và có từ 20 ước số trở lên.

a. Hướng dẫn:

Cho vòng **for** chạy từ 100 đến 999. Với mỗi số i trong phạm vi này, ta chia i cho các số j (với j chạy từ 1 đến i) và đếm số ước số của i (j là ước số của i nếu $i \% j == 0$).

b. Chương trình mẫu:

```

//UocSo.java
package uocso;
import java.util.*;
public class UocSo {
    public static void main(String[] args) {
        int i, j, dem;
        for (i = 100; i < 999; i++) {
            dem = 0;
            for (j = 1; j <= i; j++) {
                if (i % j == 0) {
                    dem++;
                }
            }
            if (dem >= 20) {

```

```

        System.out.print(i + "\t");
    }
}
}
}
}
}
}
}
}
```

 **Kết quả chạy chương trình:**

240	336	360	420	432	480	504	528	540	560	576	600
624	630	648	660	672	720	756	780	792	810	816	840
864	880	900	912	924	936	960	990				

 **Bài 1.8. Sử dụng cấu trúc lặp while**

Có 1 triệu đồng, hỏi phải gửi tiết kiệm vào ngân hàng trong thời gian mấy năm để có số tiền lớn hơn 2 triệu đồng. Biết rằng lãi suất hàng năm là 8%.

a. Hướng dẫn:

- Vì không thể biết được là sau bao nhiêu năm thì tổng số tiền lớn hơn 2.000.000, mà chỉ biết rằng mỗi năm sẽ tăng thêm một lượng nào đó. Do vậy, ở đây ta sẽ sử dụng vòng lặp không xác định và mỗi lần lặp ta sẽ kiểm tra xem đã được số tiền cần thiết hay chưa? Nếu đủ rồi thì thoát và số lần thử chính là số năm cần tìm.

Nhưng ở đây tại sao ta lại sử dụng vòng lặp **while** mà không là **do ...while?** Sở dĩ sử dụng vòng lặp **while** là vì số tiền gửi vào ban đầu rất có thể đã lớn hơn số tiền kỳ vọng!

- Tính theo công thức lãi kép, lãi nhập vốn:

1. Lãi = gốc*(1+lãi suất)ⁿ - gốc
2. Số tiền rút ra = gốc*(1+lãi suất kỳ)ⁿ. (công thức của FV) với n là số kỳ
- Sử dụng hàm Math.pow(x, y) để tính x^y, hàm Math.round(x) để làm tròn số thực x.

b. Chương trình mẫu:

```

//TinhTien.java
package tinhkiem;
import java.util.*;
public class TinhTien {
    public static void main(String[] args) {
        int SoNam;
        double TongTien;
        SoNam = 0;
        TongTien = 1000000;
        while (TongTien < 2000000) {
            SoNam = SoNam + 1;
            TongTien = Math.pow((1 + 0.08), SoNam) * TongTien;
        }
        System.out.println("So nam:" + SoNam);
        System.out.println("So tien la:" + Math.round(TongTien));
    }
}
```

 **Kết quả chạy chương trình:**

```

So nam:4
So tien la:2158925
```

 **Bài 1.9. Sử dụng cấu trúc lặp do...while**

Nhập vào một dãy các số nguyên (gồm cả số âm và dương) và đếm xem có bao nhiêu số âm, bao nhiêu số dương. Việc nhập kết thúc nếu số nhập vào là 0.

a. Hướng dẫn:

Vì không biết người dùng nhập bao nhiêu số do vậy ta có thể dùng vòng lặp không xác định để tiến hành công việc nhập. Ngoài ra, cũng cần có 2 biến để lưu số lượng các số âm và số lượng các số dương. Điều kiện kết thúc vòng lặp sẽ là số nhập vào bằng 0.

b. Chương trình mẫu:

```
// DemSoWhile.java
package demsowhile;
import java.util.Scanner;
public class DemSoWhile {
    public static void main(String[] args) {
        int n, i, DemSoAm, DemSoDuong;
        DemSoAm = 0;
        DemSoDuong = 0;
        Scanner in = new Scanner(System.in);
        do {
            System.out.print("Nhập vào một số nguyên : ");
            n = in.nextInt();
            if (n > 0) {
                DemSoDuong = DemSoDuong + 1;
            }
            if (n < 0) {
                DemSoAm = DemSoAm + 1;
            }
        } while (n != 0);
        System.out.println("Có " + DemSoDuong + " số Dương");
        System.out.println("Có " + DemSoAm + " số Am");
    }
}
```

█ Kết quả chạy chương trình:

```
Nhập vào một số nguyên : 7
Nhập vào một số nguyên : -4
Nhập vào một số nguyên : 3
Nhập vào một số nguyên : -2
Nhập vào một số nguyên : 8
Nhập vào một số nguyên : 0
Có 3 số Dương
Có 2 số Am
```

█ **Bài 1.10. Hàm không có tham số. Viết chương trình giải phương trình bậc 2. $ax^2 + bx + c=0$ (a khác 0)**

a. Hướng dẫn:

Chương trình không sử dụng hàm đã viết trong bài tập 1.4 của chương này, toàn bộ phần code để nhập dữ liệu và tính nghiệm được viết trong hàm **main**. Trong bài này, ta chia thành 2 hàm:

- Hàm **Nhap**: để nhập vào 3 số a, b, c, hàm này có kiểu trả về là **void** và không có tham số.

- Hàm **TinhNghiem**: để tính nghiệm của phương trình theo định lí Viet, hàm này có kiểu **void**, không tham số.

- Để không phải gọi các thuộc tính và phương thức qua đối tượng thì các thuộc tính a, b, c, các phương thức Nhập, TinhNghiem phải khai báo là **static**.

Chương trình chính gọi hàm **Nhap** và hàm **TinhNghiem**, thân hàm **main()** rất gọn.

b. Chương trình mẫu:

```
//GiaiPTB2.java
package giaiptb2;
import java.io.*;
public class GiaiPTB2 {
    static double delta, a, b, c;
    public static void Nhap() {
        try {
            Scanner input = new Scanner(System.in);
            System.out.print("NHAP A=");
            a = input.nextDouble();
            System.out.print("NHAP B=");
            b = input.nextDouble();
            System.out.print("NHAP C=");
            c = input.nextDouble();
        } catch (Exception e) {
            System.out.println("Loi" + e.getMessage());
        }
    }
    public static void TinhNghiem() {
        delta = b * b - 4 * a * c;
        if (delta > 0) {
            double x1 = (-b + Math.sqrt(delta)) / (2 * a);
            double x2 = (-b - Math.sqrt(delta)) / (2 * a);
            System.out.println("x1=" + x1 + " x2=" + x2);
        } else if (delta == 0) {
            double x = (-b) / (2 * a);
            System.out.println("Nghiem kep x1=x2=" + x);
        } else {
            System.out.println("PT Vo nghiem!");
        }
    }
    public static void main(String[] args) {
        Nhap();
        TinhNghiem();
    }
}
```

💻 Kết quả chạy chương trình:

```
NHAP A=1
NHAP B=2
NHAP C=-3
x1=1.0 x2=-3.0
```

Trường hợp phương trình có nghiệm kép

```
NHAP A=1
NHAP B=2
```

```
NHAP C=1  
x1=x2=-1.0
```

■ **Bài 1.11. Hàm có tham số giá trị. Tính tổng dãy số**

Cho số tự nhiên N bất kỳ.

Tính tổng $S = 1 + 1/(1+2!) + 1/(1+2!+3!) + \dots + 1/(1+2!+3!+\dots+N!)$

a. Hướng dẫn:

Viết hàm sum(k) để tính tích các số $1+2!+3!+\dots+k!$ nằm ở mảng số. Hàm này có tham số kiểu giá trị.

Thực hiện vòng lặp **for** để tính tổng các phân số và lưu vào biến S

b. Chương trình mẫu:

```
//TinhTong.java  
package tinhtong;  
public class TinhTong {  
    public static void main(String[] args) {  
        int N = 3, k;  
        float S = 0;  
        for (k = 1; k <= N; k++) {  
            S += 1 / sum(k);  
        }  
        System.out.println("Kết quả là " + S);  
        System.out.print("Hay thay doi gia tri cua N de duoc ket qua  
khac");  
    }  
    public static float sum(int k) {  
        float tong = 0;  
        for (int x = 1; x <= k; x++) {  
            tong += tich(x);  
        }  
        return tong;  
    }  
    public static float tich(int x) {  
        int t = 1;  
        for (int j = 1; j <= x; j++) {  
            t = t * j;  
        }  
        return t;  
    }  
}
```

■ Kết quả chạy chương trình:

Kết quả là 1.4444445

Hay thay giá trị của N để được các kết quả khác
(trong chương trình N=3)

■ **Bài 1.12. Hàm có tham số giá trị. Tìm UCLN và BCNN của 2 số tự nhiên**

a. Hướng dẫn:

Viết hàm UCLN để tìm ước số chung lớn nhất của 2 số tự nhiên a, b
Viết hàm nhap() để nhập vào giá trị của một số. Hàm nhap() có kiểu int.

b.Chương trình mẫu:

```
/UCLN.java
package ucln;
import java.util.*;
public class UCLN {
    public static int nhap() {
        Scanner input = new Scanner(System.in);
        boolean check = false;
        int n = 0;
        while (!check) {
            System.out.print(" ");
            try {
                n = input.nextInt();
                check = true;
            } catch (Exception e) {
                System.out.println("Ban phai nhap so! hay nhap lai...");  

                input.nextLine();
            }
        }
        return (n);
    }
    public static int UCLN(int a, int b) {
        while (a != b) {
            if (a > b) {
                a = a - b;
            } else {
                b = b - a;
            }
        }
        return (a);
    }
    public static void main(String[] args) {
        System.out.println("Nhập a=");
        int a = nhap();
        System.out.println("Nhập b=");
        int b = nhap();
        System.out.println("UCLN của " + a + " và " + b + " là:  

+UCLN(a,b));
        System.out.println("BCNN của " + a + " và " + b + " là: " +
((a * b) / UCLN(a, b)));
    }
}
```

█ Kết quả chạy chương trình:

```
Nhap a=20
Nhap b= 28
UCLN cua 20 va 28 la 4
BCNN cua 20 va 28 la 140
```

Bài 1.13. Hàm có tham số giá trị. Liệt kê n số fibonaci

Hãy viết chương trình liệt kê n số fibonaci đầu tiên với n nhập từ bàn phím

a. Hướng dẫn:

Dãy số Fibonacci được định nghĩa như sau: $F(0) = 1$, $F(1) = 1$; $F(n) = F(n-1) + F(n-2)$ với $n \geq 2$.

Ví dụ: với $n=8$, ta có các số 1 1 2 3 5 8 13 21

Trong bài này, chúng ta sử dụng phương pháp quy hoạch động để giải quyết bài toán. Ta xây dựng mảng F để lưu giá trị của các số trong dãy Fibonacci. (ngoài phương pháp này, bạn đọc có thể sử dụng phương pháp đệ quy)

b. Chương trình mẫu:

```
//Fibonacci.java
package fibonacci;
import java.util.Scanner;
public class Fibonacci {
    public static int nhap() {
        Scanner input = new Scanner(System.in);
        boolean check = false;
        int n = 0;
        while (!check) {
            System.out.print(" ");
            try {
                n = input.nextInt();
                check = true;
            } catch (Exception e) {
                System.out.println("Ban phai nhap so! hay nhap lai...");  
                input.nextLine();
            }
        }
        return (n);
    }
    public static void main(String[] args) {
        System.out.print("Nhập n = ");
        int n = nhap();
        int[] f = new int[n + 1];
        f[0] = 1;
        f[1] = 1;
        for (int i = 2; i <= n; i++) {
            f[i] = f[i - 1] + f[i - 2];
        }
        for (int i = 0; i <= n; i++) {
            System.out.println("So Fibonacci thu " + i + " la: f[" + i + "] = " + f[i]);
        }
    }
}
```

Kết quả chạy chương trình:

```
Nhap n = 5
So Fibonacci thu 0 la: f[0]= 1
```

```

So Fibonanci thu 1 la: f[1]= 1
So Fibonanci thu 2 la: f[2]= 2
So Fibonanci thu 3 la: f[3]= 3
So Fibonanci thu 4 la: f[4]= 5
So Fibonanci thu 5 la: f[5]= 8

```

Bài 1.14. Mảng một chiều

Viết chương trình nhập vào vào mảng A có n phần tử, các phần tử là những số nguyên được nhập vào từ bàn phím. Thực hiện các chức năng sau:

- Tìm phần tử lớn nhất và lớn thứ 2 trong mảng cùng chỉ số của các số đó.
- Sắp xếp mảng theo thứ tự tăng dần .
- Nhập một số nguyên x và chèn x vào mảng A sao cho vẫn đảm bảo tính sắp xếp tăng dần.

Lời giải:

a. Hướng dẫn

- Viết các hàm để nhập số phần tử của mảng, nhập mảng, tìm phần tử lớn nhất, lớn thứ 2, chèn phần tử x vào mảng.
- Sử dụng Arrays.sort(a) để sắp xếp mảng a tăng dần. Khai báo java.util.Scanner

b. Chương trình mẫu:

```

//Mang1chieu.java
package mang1chieu;
import java.util.Arrays;
import java.util.Scanner;
public class Mang1chieu {
    public static int nhap() {
        Scanner input= new Scanner(System.in);
        boolean check= false;
        int n=0;
        while(!check) {
            System.out.print(" ");
            try{
                n= input.nextInt();
                check= true;
            }catch(Exception e){
                System.out.println("Ban phai nhap so! hay nhap lai....");
                input.nextLine();
            }
        }
        return (n);
    }
    public static int viTriMaxInt(int a[], int n) {
        int max= a[0];
        int key= 0;
        for(int j=0 ; j<n ; j++) {
            if(max<a[j]) {
                max= a[j]; key= j;
            }
        }
        return (key);
    }
}

```

```

public static void inArray(int[] a, int begin , int end) {
    System.out.println();
    int i;
    for(i=begin ; i<end ; i++){
        System.out.print(" "+a[i]);
    }
    System.out.println();
}
public static int viTriMax2(int[] a,int n){
    int i,key=0,Max2=0;
    for(i=0 ; i<n ; i++){
        if(a[i]>Max2 && a[i]!= a[viTriMaxInt(a, n)]){
            Max2= a[i];key= i;
        }
    }
    return (key);
}
public static void themPhanTu(int[] a,int n,int pt){
    a[0]= pt;
    Arrays.sort(a);
}
public static void main(String[] args) {
    System.out.print("Nhập n= "); int n= nhap();
    int[] a= new int[n+1];
    int i;
    for(i=0 ; i<n ; i++){
        System.out.print("\n Nhập phần tử thứ "+i+" = ");
        a[i]= nhap();
    }
    for(i=0 ; i<n ; i++){
        if(a[i]== a[viTriMax2(a, n)])
            System.out.println(" Phần tử thứ "+i+" là phần tử 2
trong mảng a["+i+"] = "+a[i]);
    }
    System.out.print("\n -Mảng đã sắp xếp: ");
    Arrays.sort(a); inArray(a,1,n+1);
    System.out.print("Nhập phần tử muốn thêm pt= ");
    System.out.print("\n -Mảng sau khi thêm: ");
    int pt= nhap(); themPhanTu(a,n+1,pt); inArray(a,0,n+1);
}
}

```

■ Kết quả chạy chương trình:

```

Nhập n= 5
Nhập phần tử thứ 0 = 4
Nhập phần tử thứ 1 = 1
Nhập phần tử thứ 2 = 7
Nhập phần tử thứ 3 = 9
Nhập phần tử thứ 4 = 3
Phần tử lớn thứ 2 trong mảng a[2]= 7
-Mảng đã sắp xếp:
1 3 4 7 9
Nhập phần tử muốn thêm pt= 5
-Mảng sau khi thêm:
1 3 4 5 7 9

```

■ Bài 1.15. Mảng hai chiều

Viết chương trình nhập vào vào ma trận A có n dòng, m cột, các phần tử là những số nguyên lớn hơn 0 và nhỏ hơn 100 được nhập vào từ bàn phím. Thực hiện các chức năng sau:

- a) Tìm phần tử lớn nhất của ma trận cùng chỉ số của số đó.
- b) Tìm và in ra các phần tử là số nguyên tố của ma trận (các phần tử không nguyên tố thì thay bằng số 0).
- c) Sắp xếp tất cả các cột của ma trận theo thứ tự tăng dần và in kết quả ra màn hình.

Lời giải:

a .Hướng dẫn

Viết các hàm nhập, in mảng, hàm kiểm tra tính nguyên tố của một số tự nhiên. Viết hàm sortColum() để sắp xếp cột theo chiều tăng dần.

b. Chương trình mẫu:

```
//Mang2chieu.java
package mang2chieu;
import java.util.Scanner;
public class Mang2chieu {
    public static int nhap() {
        Scanner input= new Scanner(System.in);
        boolean check= false;
        int n=0;
        while(!check) {
            System.out.print(" ");
            try{
                n= input.nextInt();
                check= true;
            }catch(Exception e){
                System.out.println("Ban phai nhap so! hay nhap lai...");
                input.nextLine();
            }
        }
        return (n);
    }
    public static boolean checkSNT(int n) {
        if(n>1) {
            for(int i=2;i<=Math.sqrt(n);i++) {
                if(n%i==0) return false;
            }
            return true;
        }
        else return false;
    }
    public static void inMT(int[][] A, int n, int m) {
        int i,j;
        for(i=0 ; i<n ; i++){
            System.out.print("\n");
            for(j=0 ; j<m ; j++) System.out.print(" "+A[i][j]);
        }
    }
    public static int findMaxMT(int[][] A, int n, int m) {
        int Max= A[0][0];
```

```

    for(int i=0 ; i<n ; i++) {
        for(int j=0 ; j<m ; j++) {
            if(Max<A[i][j]) Max= A[i][j];
        }
    }
    return (Max);
}

//Tim nhung phan tu la SNT
public static void phanTuSNT(int[][] A, int n, int m) {
    int count=0,i,j;
    System.out.println("\nCac phan tu la SNT :");
    for(i=0 ; i<n ; i++){
        System.out.print("\n");
        for(j=0 ; j<m ; j++) {
            if(checkSNT(A[i][j])){
                count++;
                System.out.print(" "+A[i][j]);
            }
            else System.out.print(" "+0);
        }
    }
    System.out.println("Co "+count+" phan tu la so nguyen to");
}

//Sap xep cac cot theo thu tang dan
public static void sortColum(int[][] A, int n, int m) {
    int i,j,temp;
    for(j=0 ; j<m ; j++){
        for(i=1 ; i<n ; i++){
            if(A[i-1][j]>A[i][j])
            {
                temp= A[i-1][j];
                A[i-1][j]= A[i][j];
                A[i][j]= temp;
            }
        }
        inMT(A, n, m);
    }
}

public static void main(String[] args) {
    System.out.print("Nhập số hàng n=");
    int n= nhap();
    System.out.print("Nhập số cột m=");
    int m= nhap();
    int [][] A= new int[n][m];
    int i,j;
    for(i=0 ; i<n ; i++){
        for(j=0 ; j<m ; j++){
            System.out.print("Nhập phan tu thu
A["++(i+1)+"]["+(j+1)+"]= ");
            A[i][j]= nhap();
        }
    }
    System.out.print("Mã trận nhập vào: ");
    inMT(A, n, m);
    for(i=0 ; i<n ; i++){
        for(j=0 ; j<m ; j++){
            if(A[i][j]==findMaxMT(A, n, m))
                System.out.println("Phan tu o hang "+i+" cot
"+j+" dat Max: A["+i+"]["+j+"]= "+A[i][j]);
        }
    }
}

```

```

        }
    }
    phanTuSNT(A, n, m);
    System.out.println("Sau khi sap xep theo cot tang dan:");
    sortColum(A, n, m);
}
}

```

Kết quả chạy chương trình:

```

Nhập số hàng n= 3
Nhập số cột m= 3
Nhập phần tử thu A[1][1]= 5
Nhập phần tử thu A[1][2]= 6
Nhập phần tử thu A[1][3]= 8
Nhập phần tử thu A[2][1]= 7
Nhập phần tử thu A[2][2]= 1
Nhập phần tử thu A[2][3]= 3
Nhập phần tử thu A[3][1]= 9
Nhập phần tử thu A[3][2]= 8
Nhập phần tử thu A[3][3]= 6
Mã trận nhập vào:
 5 6 8
 7 1 3
 9 8 6
Phần tử ở hàng 2 cột 0 đặt Max: A[2][0]= 9
Các phần tử là SNT :
 5 0 0
 7 0 3
 0 0 0
Có 3 phần tử là số nguyên tố
Sau khi sắp xếp theo cột tăng dần
 5 1 3
 7 6 6
 9 8 8

```

Bài 1.16. Xâu ký tự

Một số được gọi là số thuận nghịch (hay số Palindrome) nếu ta đọc từ trái sang phải hay từ phải sang trái số đó ta vẫn nhận được một số giống nhau. Hãy liệt kê tất cả các số thuận nghịch có sáu chữ số (ví dụ số: 558855).

Lời giải:

a. Hướng dẫn:

Xây dựng hàm **public static boolean testSoThuanNghich(int n)** để kiểm tra số n có là số thuận nghịch hay không. Sử dụng các hàm append(), reverse(): đảo xâu, equals(): So sánh 2 xâu

b. Chương trình mẫu:

```

//Palindrome.java
package palindrome;
public class Palindrome {
    public static boolean testSoThuanNghich(int n) {
        StringBuiler xau= new StringBuiler();
        String str= ""+n;

```

```

        xau.append(str);
        String check= ""+xau.reverse();
        if(str.equals(check)) return true;
        else return false;
    }
    public static void main(String[] args) {
        int n,count=0;
        for(n=100000 ; n<= 999999 ; n++) {
            if(testSoThuanNghich(n)) {
                System.out.print(n+" ");
                count++;
            }
        }
        System.out.println("\nCo "+count+" so thuan nghich co 6 chu so.");
    }
}

```

💻 Kết quả chạy chương trình: Có 900 số, sau đây là 27 số đầu tiên được in ra:

```

100001 101101 102201 103301 104401 105501 106601 107701 108801
109901 110011 111111 112211 113311 114411 115511 116611 117711
118811 119911 120021 121121 122221 123321 124421 125521 126621
.....
Co 900 so thuan nghich co 6 chu so.

```

Bài 1.17. Xâu ký tự.

Cho trước một xâu ký tự là họ tên người đầy đủ. Hãy tách ra tên của người này.

Lời giải:

a. Hướng dẫn

- Sử dụng các hàm để làm việc với xâu: trim(), length(), equal(), substring()

b. Chương trình mẫu

```

//TachTen.java
package tachten;
public class TachTen {
    public static void main(String[] args) {
        String S = new String(" Nguyen Van An ");
        String S1 = new String();
        S=S.trim();
        int k;
        for (k=S.length()-1;k>=0;k--)
        {
            S1=S.substring(k,k+1);
            if(S1.equals(" ")) break;
        }
        System.out.print("Ten cua nguoi do la:"+S.substring(k+1));
    }
}

```

💻 Kết quả chạy chương trình:

Ten cua nguoi do la:An

1.2.2. Bài tập tự giải

I. CÁC LỆNH NHẬP, XUẤT, GÁN

Bài 1.18. Viết chương trình in ra màn hình dòng chữ “ WELCOME TO JAVA”

Bài 1.19. Viết chương trình cho máy làm những việc sau:

- Nhập vào chiều dài và rộng của một hình chữ nhật
- In ra màn hình chu vi của hình chữ nhật có chiều dài và rộng vừa nhập
- In ra màn hình diện tích của hình chữ nhật có chiều dài và rộng vừa nhập

Bài 1.20. Viết chương trình cho máy làm những việc sau:

- Nhập vào bán kính của một hình tròn
- In ra màn hình chu vi của hình tròn có bán kính vừa nhập
- In ra màn hình diện tích của hình tròn có bán kính vừa nhập

II. CÁC LỆNH IF, SWITCH, BREAK

Bài 1.21. Viết chương trình nhập vào một số từ bàn phím. Nếu số nhập vào là chẵn thì in ra thông báo “Đây là số chẵn”, còn ngược lại in ra thông báo “Đây là số lẻ”.

Bài 1.22. Viết chương trình đưa vào 3 số a,b,c từ bàn phím, kiểm tra xem ba số đó có phải là ba độ dài của ba cạnh của một tam giác không? Nếu có thì hãy tính diện tích của tam giác theo công

$$\text{thúc Hê Rông } S = \sqrt{p(p-a)(p-b)(p-c)} \text{ với } p = \frac{a+b+c}{2}$$

Bài 1.23. Viết chương trình cho máy làm những việc sau:

- Nhập vào điểm trung bình của một sinh viên
- In ra điểm trung bình và phân loại sinh viên theo mô tả sau
 - o Loại “Kém” nếu DTB < 5
 - o Loại “Trung bình” nếu $5 \leq DTB < 7$
 - o Loại “Khá” nếu $7 \leq DTB < 8$
 - o Loại “Giỏi” nếu DTB ≥ 8

Bài 1.24. Viết chương trình nhập chỉ số điện kế tháng trước (chiso1), chỉ số điện kế tháng này (chiso2), tính tiền điện tháng này cho hộ biết rằng:

- Mỗi KW trong 60 KW đầu tiên có đơn giá là 500 đ
- Từ KW thứ 61 đến KW thứ 160 có đơn giá là 800 đ
- Từ KW thứ 161 trở lên có đơn giá là 1000 đ

III. LỆNH LẶP FOR

Bài 1.25. Viết chương trình in ra các số từ 1 đến 10, in ngược lại từ 10 đến 1.

Bài 1.26. Nhập n là số nguyên dương. Viết chương trình đếm xem có bao nhiêu số chẵn nhỏ hơn n, bao nhiêu số lẻ nhỏ hơn n.

Bài 1.27. Viết chương trình tính tổng các số chia hết cho 3 và có tận cùng là 6 trong khoảng từ 1 đến 100.

Bài 1.28. Viết chương trình in ra các số nguyên tố nằm trong khoảng từ 1 đến 500.

IV. LỆNH WHILE, DO WHILE, BREAK, CONTINUE

Bài 1.29. Viết chương trình xác định số nguyên dương n nhỏ nhất để tổng các số từ 1 đến n lớn hơn một giá trị S cho trước.

Bài 1.30. Viết chương trình tìm ước chung lớn nhất của 2 số nguyên dương A và B. Trong chương trình yêu cầu người sử dụng phải nhập 2 số nguyên dương nếu nhập sai yêu cầu nhập lại đến khi nhập đúng thì tính ước chung lớn nhất.

Bài 1.31. Viết chương trình tính tổng các số chia hết cho 3 và có tận cùng là 6 trong khoảng từ 1 đến 100 bằng lệnh **while**.

Bài 1.32. Viết chương trình tìm giá trị lớn nhất của một dãy số có n phần tử nhập vào từ bàn phím (không sử dụng mảng một chiều).

V. HÀM (PHƯƠNG THỨC)

Bài 1.33. Viết chương trình giải phương trình bậc nhất $ax+b=0$, sử dụng hàm nhập để nhập vào 2 số a, b.

Bài 1.34. Viết chương trình tính diện tích tứ giác bất kỳ khi biết độ dài bốn cạnh và một đường chéo, bằng cách xây dựng hàm tính diện tích của tam giác khi biết độ dài ba cạnh.

Bài 1.35. Xây dựng 1 hàm để kiểm tra một số có phải là số hoàn hảo hay không? (x là số hoàn hảo nếu tổng các ước số (khác x) của x bằng với x. Ví dụ số 6 là hoàn hảo vì $6=1+2+3$).

Bài 1.36. Viết chương trình tìm ước số chung lớn nhất của 3 số bằng cách xây dựng hàm tìm ước số chung lớn nhất của 2 số.

VI. KIỂU MẢNG, XÂU KÍ TỰ

Bài 1.37. Viết chương trình cho máy làm những công việc sau:

- Nhận vào mảng A có N phần tử là số tự nhiên lớn hơn 5 và nhỏ hơn 100 được nhập vào từ bàn phím ($N > 10$).
- Tính và in ra tổng các phần tử của mảng.
- Tính và in ra số phần tử chẵn, số phần tử lẻ của mảng.
- Tìm và in ra phần tử lớn nhất của mảng (in ra cả vị trí và giá trị).
- Tìm tất cả các phần tử của mảng A chia hết cho 7 và có tận cùng là 6.
- Tìm tất cả những phần tử của A có từ 4 ước số trở lên.
- Sắp xếp các phần tử của mảng theo thứ tự tăng dần

Bài 1.38. Viết chương trình nhập vào một ma trận vuông cấp $n \times n$, sau đó tính tổng các phần tử dương trên đường chéo chính.

Bài 1.39. Lập trình nhập vào một xâu ký tự gồm các ký tự là số. Tạo mảng một chiều mà mỗi phần tử của mảng là một nhóm gồm 3 số cạnh nhau trong xâu ký tự.

Ví dụ: nhập xâu $S='35465423213562313243'$ thì mảng $A[0]=354$, $A[1]=654$, $A[2]=232$, $A[3]=135$, $A[4]=623$, $A[5]=132$, $A[6]=43$.

Bài 1.40. Viết chương trình cộng 2 số nguyên dương lớn bất kỳ và in kết quả ra màn hình (sử dụng xâu ký tự).

CHƯƠNG 2: HƯỚNG ĐỐI TƯỢNG TRONG JAVA

2.1. PHẦN LÝ THUYẾT

2.1.1. Định nghĩa lớp và các hàm thành phần

* *Định nghĩa lớp*

```
[<Phạm vi hoặc kiểm soát truy nhập>] class <Tên lớp>
[extends <Tên lớp cha>] [implements <Tên giao diện>]
{
    <Các thành phần của lớp>
}
```

<Các thành phần của lớp> gồm: thuộc tính, phương thức, các lớp con, các giao diện...

* *Định nghĩa hàm (phương thức) thành phần*

```
<Chỉ định truy xuất> [<bổ nghĩa loại phương thức>]<kiểu dữ liệu> <Tên phương thức>
(danh sách tham số)
{ //phần thân
}
```

2.1.2. Phạm vi và các thuộc tính kiểm soát truy nhập các thành phần của lớp

a. Phạm vi của các thành phần

Phạm vi của các thành phần có các loại:

- Phạm vi lớp của các thành phần,
- Phạm vi khối của các biến cục bộ (local),
- Phạm vi gói (**package**)

* Phạm vi lớp: Phạm vi lớp xác định những thành phần được truy nhập bên trong của một lớp (kể cả lớp được kế thừa). Quyền truy nhập của chúng thường được xác định thông qua các bô ngữ (modifier): **public, protected, private**.

* Phạm vi khối: Trong chương trình, các lệnh khai báo và các lệnh thực hiện tuân tự có thể gộp lại thành từng khối (block) bằng cách sử dụng {}, {}.

Một biến được khai báo ở trong một khối có phạm vi xác định bên trong khối và không xác định ở bên ngoài khối đó.

Chú ý 2.1:

- Trong các khối thì lệnh khai báo là tự do, thứ tự không quan trọng, muốn khai báo ở chỗ nào cũng được miễn là phải khai báo trước khi sử dụng.
- Có thể có nhiều khối lồng nhau nhưng không được cắt nhau và những biến khai báo ở khối ngoài đều có phạm vi xác định ở trong mọi khối bao bên trong nó.

b. Các thuộc tính kiểm soát truy nhập các thành phần của lớp

Khi thiết kế các thành phần của lớp, chúng ta có thể sử dụng những bô ngữ sau: public, protected, mặc định (không sử dụng thêm bô ngữ khi định nghĩa lớp), private, static, final, abstract, synchronized, native, transient, volatile

Bảng 2.1: Ý nghĩa của các bô ngữ

Tùy khóa	Lớp	Hàm thành phần	Biến thành phần	Biến cục bộ
abstract	yes	yes	-	-
static	-	yes	yes	-
public	yes	yes	yes	-
protected	yes	yes	yes	-
private	-	yes	yes	-
final	yes	yes	yes	yes
synchronized	-	yes	-	-
native	-	yes	-	-
transient	-	-	yes	-
volatile	-	-	yes	-

trong đó chữ ‘yes’ nghĩa là được phép sử dụng, dấu ‘-’ là không được phép sử dụng.

2.1.3. Quan hệ kế thừa giữa các lớp

Cú pháp qui định quan hệ kế thừa trong Java là sự mở rộng của lớp cha, có dạng: <Tên lớp con> **extends** <Tên lớp cha> {
 // Các thuộc tính dữ liệu bổ sung
 // Các hàm thành phần bổ sung hay viết đè
 }

Lưu ý 2.2:

Mọi đối tượng của lớp con cũng sẽ là đối tượng thuộc lớp cha. Do vậy việc gán một đối tượng của lớp con sang cho biến tham chiếu đối tượng của lớp cha là sự mở rộng kiểu và do đó không cần ép kiểu.

Ngược lại gán một đối tượng của lớp cha cho biến tham chiếu đối tượng thuộc lớp con sẽ phải thực hiện ép kiểu. Lưu ý khi đó sẽ có thể bị tổn thất thông tin.

Ví dụ 2.1:

```
class SuperClass { /* ... */ }
class SubClass { /* ... */ }
class UserClass {
    //
    public static main(String args[]){
        SuperClass super1 = new SuperClass(); // Tạo ra đối tượng lớp cha
        SubClass sub1 = super1; // Mở rộng kiểu
        SuperClass super2 = (SubClass) sub1; // Thu hẹp kiểu nên phải ép kiểu
    }
}
```

* Toán tử móc xích giữa các lớp kế thừa this() và super()

Các toán tử tạo lập không thể viết đè ở các lớp dẫn xuất (lớp con). Chúng có thể được nạp chồng nhưng phải trong cùng lớp.

Trong Java có 2 toán tử tạo lập đặc biệt có tên là **this()** và **super()** được sử dụng để móc xích giữa các lớp có quan hệ kế thừa với nhau.

Toán tử tạo lập **this()**: Toán tử tạo lập này được sử dụng để tạo ra đối tượng của lớp hiện thời.

Toán tử tạo lập **super()**: Ta dùng từ khóa **super** mỗi khi lớp con cần sử dụng thuộc tính hay phương thức của lớp cha. **super** có hai dạng ứng với hai mục đích khác nhau:

- dùng để gọi hàm khởi tạo của lớp cha,
- dùng để truy xuất thành phần của lớp cha bị thành phần cùng tên ở lớp con che đi

2.1.4. Giao diện và sự mở rộng quan hệ kế thừa

Java không hỗ trợ trực tiếp kế thừa bội, song để tránh những trở ngại trên Java cung cấp khái niệm interface và cho phép kế thừa bội đối với các interface.

* Định nghĩa interface

interface định nghĩa mẫu hợp đồng thông qua việc phác thảo các hàm mẫu (prototype) và không cài đặt nội dung thực hiện.

```
interface <Tên interface> {  
    <Nội dung của interface>  
}
```

<Nội dung của interface> thường chứa danh sách các hàm mẫu và các hằng.

Giao diện interface là loại kiểu lớp đặc biệt, trong đó tất cả các hàm thành phần đều là trừu tượng (mặc định có phạm vi **public**), do vậy không thể khởi tạo được giá trị, nghĩa là không thể tạo ra đối tượng của interface. Những hàm trong interface sẽ được cài đặt ở những lớp xây dựng mới theo dạng:

```
class <Tên lớp> implements <Tên interface> {  
    // Bổ sung các thành phần;  
    // Cài đặt các hàm mẫu đã cho trong <Tên interface>;  
}
```

Sự mở rộng (kế thừa) của các interface

Giống như cấu trúc lớp, interface cũng có thể được mở rộng từ nhiều interface khác bằng mệnh đề **extends**. Được phép đa kế thừa giao diện.

Ví dụ 2.2:

```
class MyClass implements Interface1, Interface2 {  
    // Cài đặt một số hàm mẫu của Interface1, Interface2;  
}  
interface Interface1{  
    // Khai báo các hằng và hàm mẫu;  
}  
interface Interface2{  
    // Khai báo các hằng và hàm mẫu;  
}
```

Lưu ý 2.3:

- Các hàm mẫu của interface được mặc định là abstract và không được khai báo **static**;

- Một lớp có thể chọn một số hàm mẫu để cài đặt, nghĩa là có thể chỉ cài đặt một phần của giao diện interface;
- Một giao diện có thể kế thừa từ nhiều hơn một giao diện; một giao diện có thể kế thừa từ nhiều hơn một giao diện và lớp;

2.1.5. Nạp chồng và tính đa hình

- *Nạp chồng các hàm thành phần*: Trong cùng một lớp cho phép có nhiều hàm cùng tên gọi nhưng kiểu trả về và danh sách tham số khác nhau.
- *Viết đè các hàm thành phần*: Một lớp con có thể viết đè (overriding), thay đổi nội dung thực hiện của những hàm cùng tên được thừa kế từ lớp cha. Định nghĩa mới của hàm viết đè phải có cùng định danh (tên gọi và danh sách tham số) và cùng kiểu trả lại giá trị. Các hàm **final**, **static** của lớp cha không được phép viết đè.

- *Cơ chế che bóng của các biến*

Lớp con không được phép viết đè các biến thành phần của lớp cha, nhưng có thể bị che khuất chúng tương tự như biến cục bộ trong lớp con.

- *Tính đa hình*: Nếu các hàm thành phần cùng tên có ở nhiều lớp khác nhau trong kế thừa sẽ tạo ra tính đa hình.

2.1.6. Một số gói cơ bản

* **Package**: Một gói trong Java được coi như một thư mục, là nơi tổ chức các lớp và các giao diện. Một gói có thể chứa các gói con.

Mục đích: Tổ chức các lớp thành các đơn vị nhỏ hơn, có thể xác định rõ ràng vị trí. Tránh việc đặt tên xung đột khi có quá nhiều lớp.

Cách tạo:

Ở dòng đầu tiên của file, sử dụng lệnh khai báo: **package <tên package>;**

Tên gói phải bắt đầu bằng chữ thường và không có khoảng trắng

Sử dụng gói trong lớp:

import <tên package>.<tên lớp>; //sử dụng một lớp trong gói

import <tên package>.*; //sử dụng toàn bộ gói

Lệnh **import** phải đặt dưới lệnh khai báo **package** và trên lệnh định nghĩa class.

Bảng 2.2: Một số gói cơ bản

Gói	Mô tả
java.lang	Không cần phải khai báo nhập. Gói này luôn được nhập cho bạn.
java.io	Bao gồm các lớp để trợ giúp cho bạn tất cả các thao tác vào ra.
java.applet	Bao gồm các lớp để bạn cần thực thi một applet trong trình duyệt.
java.awt	Các thành phần để xây dựng giao diện đồ họa (GUI).
java.util	Cung cấp nhiều lớp và nhiều giao diện tiện ích khác nhau, như là các cấu trúc dữ liệu, lịch, ngày tháng, v.v.
java.net	Cung cấp các lớp và các giao diện cho việc lập trình mạng TCP/IP.

2.2.PHẦN BÀI TẬP

2.2.1. Bài tập mẫu

❑ *Bài 2.1. Minh họa khai báo lớp*

Viết chương trình tính tổng 2 số nguyên a, b.

a. Hướng dẫn

Xây dựng lớp TinhTong gồm:

- Thuộc tính a, b khai báo là **static**

- Phương thức: Nhập để nhập 2 số, phương thức Xuất để in ra màn hình tổng 2 số khai báo **static**.

b. Chương trình mẫu

```
//TinhTong.java
package tinh tong;
import java.util.*;
public class TinhTong {
    public static int a, b;
    public static void Nhap() {
        Scanner in = new Scanner(System.in);
        System.out.print("a=");
        a = in.nextInt();
        System.out.print("b=");
        b = in.nextInt();
    }
    public static void Xuat() {
        int S = a + b;
        System.out.print("Tong = " + S);
    }
    public static void main(String[] args) {
        Nhap();
        Xuat();
    }
}
```

❑ Kết quả chạy chương trình:

```
a=2
b=3
Tong = 5
```

❑ *Bài 2.2. Minh họa khai báo lớp*

Xây dựng lớp DiemThi, có các thành phần dữ liệu là điểm toán, lý, hóa; các phương thức nhập điểm, điểm trung bình của 3 môn và phương thức xuất ra màn hình điểm trung bình 3 môn.

a. Hướng dẫn

Xây dựng lớp DiemThi gồm:

- Thuộc tính **DiemToan**, **DiemLy**, **DiemHoa** kiểu số thực

- Phương thức: **NhapDiem()** để nhập điểm 3 môn

DiemTrungBinh() để tính điểm trung bình 3 môn.

Xuat() để in lên màn hình điểm trung bình 3 môn.

Xây dựng lớp Main() chứa hàm **main()**. Khai báo đối tượng lớp DiemThi, và truy cập đến các phương thức NhapDiem(), Xuat()

b.Chương trình mẫu

```
package diemthi;
import java.util.*;
public class DiemThi {
    float DiemToan, DiemLy, DiemHoa;
    void NhaphDiem() {
        Scanner in = new Scanner(System.in);
        System.out.print("Diem toan=");
        DiemToan = in.nextFloat();
        System.out.print("Diem ly=");
        DiemLy = in.nextFloat();
        System.out.print("Diem hoa=");
        DiemHoa = in.nextFloat();
    }

    float DiemTrungBinh() {
        return (DiemToan + DiemLy + DiemHoa) / 3;
    }

    void Xuat() {
        System.out.print("Diem Trung binh = " + DiemTrungBinh());
    }
}

class Main {
    public static void main(String[] args) {
        DiemThi x = new DiemThi();
        x.NhaphDiem();
        x.Xuat();
    }
}
```

💻 Kết quả chạy chương trình:

```
Diem toan=4
Diem ly=5
Diem hoa=6
Diem Trung binh = 5.0
```

Bài 2.3. Mảng đối tượng

Xây dựng lớp thuê bao **Mobile** mô tả thông tin về chủ sở hữu thuê bao điện thoại, gồm các thông tin tên khách hàng, số thuê bao, tiền cước phí.

Nhập vào danh sách n thuê bao, hiện lên màn hình danh sách thuê bao có cước phí lớn nhất.

a. Hướng dẫn

Xây dựng lớp Mobile gồm:

- Thuộc tính: Tên khách hàng, số thuê bao kiểu String.

 Tiền cước phí: kiểu số thực.

- Phương thức:

- Phương thức tạo không tham số, hàm tạo 3 tham số
- Phương thức Hien() để nhập, hiện thông tin về một thuê bao

Xây dựng lớp Main:

- Phương thức nhập danh sách n thuê bao

- Phương thức Hiển thị danh sách thuê bao vừa nhập
- Phương thức hiển thị thuê bao có cước phí lớn nhất.

Để hiện thuê bao có cước phí lớn nhất cần:

+ Duyệt qua mảng (bằng phương pháp tìm kiếm tuần tự) để tìm cước phí lớn nhất (max)

+ Duyệt mảng lần 2, nếu thuê bao nào có cước phí = max thì hiện thông tin về thuê bao này (sử dụng hàm HIEN của lớp Mobile).

b. Chương trình mẫu

```

package mobile;
import java.io.*;
public class Mobile {
    String soThueBao;
    String hoten;
    double cuocPhi;
    Mobile() {
        soThueBao = null;
        hoten = null;
        cuocPhi = 0;
    }
    Mobile(String ht, String tb, double cp) {
        soThueBao = tb;
        hoten = ht;
        cuocPhi = cp;
    }

    public void hien() {
        System.out.print("Hoten:" + hoten + ",\tcuocPhi:" + cuocPhi);
        System.out.print(",\tsoThueBao:\t" + soThueBao);
    }
}

class Main {
    public static void nhap(int n, Mobile a[]) throws Exception {
        BufferedReader in = new BufferedReader(new
InputStreamReader(System.in)));
        double cp;
        String ht, tb;
        for (int i = 0; i < n; i++) {
            System.out.println("Thue bao thu " + (i + 1));
            System.out.print("-Ho va Ten : ");
            ht = in.readLine();
            System.out.print("-So thue bao : ");
            tb = in.readLine();
            System.out.print("-Cuoc phi : ");
            cp = Double.parseDouble(in.readLine());
            Mobile x = new Mobile(ht, tb, cp);
            a[i] = x;
        }
    }

    public static void hienthi(int n, Mobile a[]) {
        for (int i = 0; i < n; i++) {
            System.out.print("\n[Thue bao " + (i + 1) + "]:");
            a[i].hien();
        }
    }
}

```

```

        }
    }

    public static void hienthiCuocPhiLonNhat(int n, Mobile a[]) {
        double max = a[0].cuocPhi;
        for (int i = 1; i < n; i++) {
            if (a[i].cuocPhi > max) {
                max = a[i].cuocPhi;
            }
        }
        System.out.println("\nCac thue bao co cuoc phi lon nhat");
        for (int i = 0; i < n; i++) {
            if (a[i].cuocPhi == max) {
                a[i].hien();
            }
        }
    }

    public static void main(String[] args) throws Exception {
        BufferedReader in = new BufferedReader(new
InputStreamReader(System.in));
        Mobile a[] = new Mobile[10];
        int n = 0;
        try {
            System.out.print("\nNhap so nhan vien : ");
            n = Integer.parseInt(in.readLine());
        } catch (Exception e) {
            System.out.print("\nLoi " + e.getMessage());
        }
        nhap(n, a);
        hienthi(n, a);
        hienthiCuocPhiLonNhat(n, a);
    }
}

```

💻 Kết quả chạy chương trình:

```

Nhap so nhan vien : 2
Thue bao thu 1
-Ho va Ten : Nguyen Thai Son
-So thue bao : 0969880912
-Cuoc phi : 300000
Thue bao thu 2
-Ho va Ten : Le Chi Luan
-So thue bao : 0982080380
-Cuoc phi : 500000

[Thue bao 1]:Hoten:Nguyen Thai Son, cuocPhi:300000.0, soThueBao:0969880912
[Thue bao 2]:Hoten:Le Chi Luan, cuocPhi:500000.0, soThueBao:0982080380
Cac thue bao co cuoc phi lon nhat
Hoten:Le Chi Luan, cuocPhi:500000.0, soThueBao: 0982080380

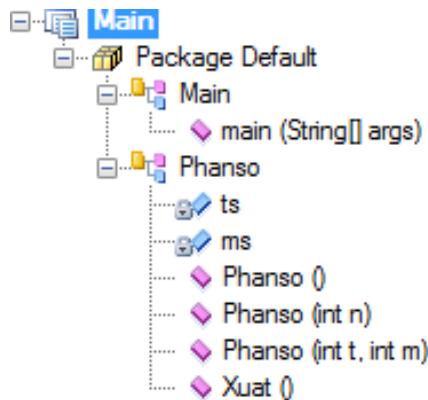
```

📘 **Bài 2.4. Minh họa chòng hàm tạo**

Xây dựng lớp Phanso mô tả phân số gồm có các hàm tạo 0, 1, 2 tham số.

a. Hướng dẫn

Xây dựng chương trình như sau:



b. Chương trình mẫu

```

//Main .java
public class PhanSo {
    private int ts, ms;
    public PhanSo() {
        ts = 0;
        ms = 1;
    }
    public PhanSo(int n) {
        ts = n;
        ms = 1;
    }
    public PhanSo(int t, int m) {
        ts = t;
        ms = m;
    }
    public void Xuat() {
        System.out.println(ts + "/" + ms);
    }
}
public class Main {
    public static void main(String[] args) {
        Phanso p = new Phanso(), q = new Phanso(6), r = new Phanso(2,
5);
        p.Xuat();
        q.Xuat();
        r.Xuat();
    }
}

```

💻 Kết quả chạy chương trình:

0/1

6/1

2/5

📘 **Bài 2.5. Quản lý sinh viên**

Viết chương trình theo phương pháp lập trình hướng đối tượng, thực hiện các chức năng xử lý sau đây trên danh sách một sinh viên:

- Nhập/Xuất danh sách nhân viên.
- Tìm kiếm một nhân viên theo mã hồ sơ, họ tên hoặc năm sinh.

Các thông tin của mỗi sinh viên trong danh sách gồm: Mã hồ sơ, họ tên, năm sinh.

Chương trình hiện ra một menu cho người dùng lựa chọn các chức năng tìm kiếm, lặp đi lặp lại quá trình tìm kiếm cho đến khi nhập vào số 0.

a. Hướng dẫn

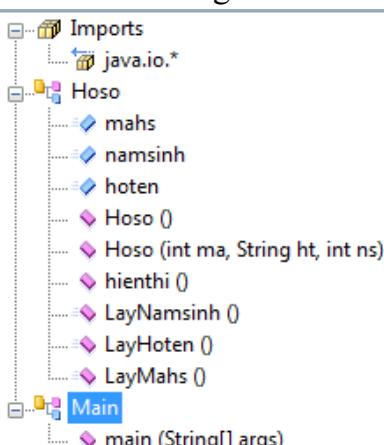
* Xây dựng lớp HoSo:

- Thuộc tính: mahs, namsinh: kiểu int, hoten: kiểu String
- Phương thức:
 - + Hàm hien thi để nhập thông tin về nhân viên.
 - + Hàm tạo không tham số, hàm tạo 3 tham số
 - + Các hàm: LayNamsinh(), LayHoten(), LayMahs() để lấy năm sinh, lấy họ tên và lấy mã hồ sơ.

* Xây dựng lớp Main chứa hàm main():

- + Khai báo mảng n hồ sơ (n nhập từ bàn phím)
- + Nhập vào n hồ sơ
- + Thực hiện các chức năng tìm kiếm theo mã hồ sơ, theo họ tên và theo năm sinh.
- + Sử dụng vòng lặp do... while lặp đi lặp lại quá trình tìm kiếm cho đến khi nhập vào số 0.

Cấu trúc chương trình như sau:



b. Chương trình mẫu

```
import java.io.*;
public class HoSo {
    int mahs, namsinh;
    String hoten;
    public HoSo() {
        mahs = 0;
        hoten = "Unknown";
        namsinh = 1900;
    }
    public HoSo(int ma, String ht, int ns) {
        mahs = ma;
```

```

        hoten = ht;
        namsinh = ns;
    }
    public void HienThi() {
        System.out.print("[ThongTin]\tMaHS:" + mahs);
        System.out.println("\tHoten:" + hoten + "\tNamSinh:" +
namsinh);
    }

    int LayNamsinh() {
        return namsinh;
    }

    String LayHoten() {
        return hoten;
    }

    int LayMahs() {
        return mahs;
    }
}

class Main {

    public static void main(String[] args) throws Exception {
        HoSo A[] = new HoSo[10];
        int n, i;
        BufferedReader in = new BufferedReader(new
InputStreamReader(System.in));
        do {
            System.out.print("Nhập số họ số, n = ");
            n = Integer.parseInt(in.readLine());
        } while ((n < 1) || (n > 10));
        //nhập họ số
        for (i = 0; i < n; i++) {
            HoSo x = new HoSo();
            System.out.println("Nhập họ số thứ : " + (i + 1));
            System.out.print("\tMa họ số : ");
            x.mahs = Integer.parseInt(in.readLine());
            System.out.print("\tHọ và Tên : ");
            x.hoten = in.readLine();
            System.out.print("\tTên sinh : ");
            x.namsinh = Integer.parseInt(in.readLine());
            A[i] = x;
        }
        //in họ số
        System.out.println("Danh Sach Họ Số Vua Nhập");
        for (i = 0; i < n; i++) {
            System.out.print("[HoSo" + (i + 1) + "] \tMaHS:" +
A[i].mahs);
            System.out.print("\tHoten:" + A[i].hoten);
            System.out.println("\tTên sinh:" + A[i].namsinh);
        }
        //tim kiem họ số
        int ch, kt;
        do {
            System.out.println("\t1.Tìm theo mã");
            System.out.println("\t2.Tìm theo họ tên");

```

```

        System.out.println("\t3.Tim theo namsinh");
        System.out.println("\t0.Thoat");
        System.out.print("Ban chon : ");
        ch = Integer.parseInt(in.readLine());
        kt = 0;
        switch (ch) {
            case 1:
                System.out.print("Nhập ma ho so : ");
                int ma = Integer.parseInt(in.readLine());
                for (i = 0; i < n; i++) {
                    if (A[i].LayMahs() == ma) {
                        kt = 1;
                        System.out.print("[HoSo" + (i + 1) +
                            "]\\tMahs:" + A[i].mahs);
                        System.out.print("\tHoten:" + A[i].hoten);
                        System.out.println("\tNam sinh:" +
                            A[i].namsinh);
                    }
                }
                if (kt == 0) {
                    System.out.println("Khong tim thay");
                }
                break;
            case 2:
                System.out.print("Nhập ho ten : ");
                String ht = in.readLine();
                for (i = 0; i < n; i++) {
                    if (ht.equals(A[i].LayHoten())) {
                        kt = 1;
                        System.out.print("[HoSo" + (i + 1) +
                            "]\\tMahs:" + A[i].mahs);
                        System.out.print("\tHoten:" + A[i].hoten);
                        System.out.println("\tNam sinh:" +
                            A[i].namsinh);
                    }
                }
                if (kt == 0) {
                    System.out.println("Khong tim thay");
                }
                break;
            case 3:
                System.out.print("Nhập nam sinh : ");
                int ns = Integer.parseInt(in.readLine());
                for (i = 0; i < n; i++) {
                    if (A[i].LayNamsinh() == ns) {
                        kt = 1;
                        System.out.print("[HoSo" + (i + 1) +
                            "]\\tMahs:" + A[i].mahs);
                        System.out.print("\tHoten:" + A[i].hoten);
                        System.out.println("\tNam sinh:" +
                            A[i].namsinh);
                    }
                }
                if (kt == 0) {
                    System.out.println("Khong tim thay");
                }
                break;
        }
    }
}

```

```
        } while (ch != 0);
    }
}
```

■ Kết quả chạy chương trình:

```
Nhap so ho so, n = 2
Nhap ho so thu : 1
    Ma ho so : 1
        Ho va Ten : Nguyen Minh
        Nam sinh : 1980
        Nhap ho so thu : 2
    Ma ho so : 2
        Ho va Ten : Giang Phuong
        Nam sinh : 1979
        Danh Sach Ho So Vua Nhap
[HoSo1]      Mahs:1      Hoten:Nguyen Minh      Nam sinh:1980
[HoSo2]      Mahs:2      Hoten:Giang Phuong      Nam sinh:1979
    1.Tim theo ma
    2.Tim theo hoten
    3.Tim theo namsinh
    0.Thoat
Ban chon : 2
Nhap ho ten : Nguyen Minh
[HoSo1]      Mahs:1      Hoten:Nguyen Minh      Nam sinh:1980
    1.Tim theo ma
    2.Tim theo hoten
    3.Tim theo namsinh
    0.Thoat
Ban chon : 1
Nhap ma ho so : 1
[HoSo1]      Mahs:1      Hoten:Nguyen Minh      Nam sinh:1980
    1.Tim theo ma
    2.Tim theo hoten
    3.Tim theo namsinh
    0.Thoat
Ban chon : 3
Nhap nam sinh : 1979
[HoSo2]      Mahs:2      Hoten:Giang Phuong      Nam sinh:1979
    1.Tim theo ma
    2.Tim theo hoten
    3.Tim theo namsinh
    0.Thoat
Ban chon : 3
Nhap nam sinh : 1996
Khong tim thay
```

Bài 2.6. Tính kế thừa. Quản lý nhân viên

a) Xây dựng lớp **Nguoi** có:

- Các thuộc tính: Họ và tên, năm sinh
- Các phương thức: khởi tạo và hiển thị

b) Xây dựng lớp **NhanVien** kế thừa từ lớp **Nguoi** và có thêm:

- Các thuộc tính: Mã nhân viên, chức vụ
- Các phương thức: khởi tạo, hiển thị, cập nhật

c) Từ các lớp đã xây dựng trên, viết chương trình thực hiện các nhiệm vụ sau:

- Nhập vào danh sách gồm N nhân viên (N nhập từ bàn phím). Các thông tin nhập vào theo thứ tự: Mã nhân viên, Họ và tên, Năm sinh, Chức vụ.
- In ra danh sách N nhân viên vừa nhập.
- Tìm kiếm thông tin về nhân viên có năm sinh nhập vào từ bàn phím.
- Cập nhật thông tin về nhân viên.

a. Hướng dẫn

Xây dựng lớp **Nguoi** có:

- Các thuộc tính: Họ và tên kiểu String, năm sinh kiểu int
- Các phương thức:
 - Khởi tạo: một tham số và ba tham số.
 - Hienthi(): hiển Họ và tên, Năm sinh

Các phương thức có phạm vi **public**

Xây dựng lớp **NhanVien** kế thừa từ lớp **Nguoi** và có:

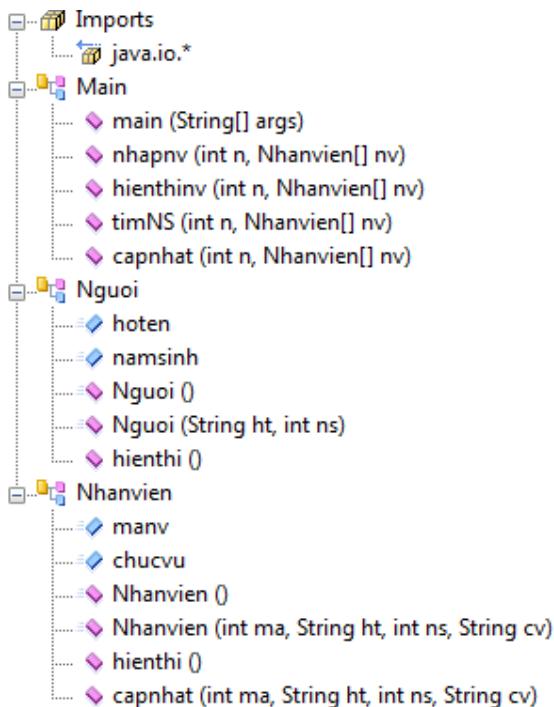
- Các thuộc tính:
 - Mã nhân viên kiểu nguyên, chức vụ: kiểu ký tự, phạm vi **private**
- Các phương thức:
 - Khởi tạo: không tham số và ba tham số
 - Hienthi: Gọi phương thức Hienthi() của lớp **Nguoi** sau đó hiện thêm các thuộc tính mã nhân viên, chức vụ. Chú ý cách trình bày: Thông tin của mỗi nhân viên viết trên một dòng, cách nhau một khoảng trắng là một tab.

Các phương thức có phạm vi **public**.

Xây dựng lớp **Main**, chứa:

- Phương thức **main()**: gọi các phương thức để thực hiện yêu cầu của chương trình.
 - **void nhapnv(int n,Nhanvien nv[])**: nhập thông tin về nhân viên
 - **void hienthinv(int n,Nhanvien nv[])**: hiển thị thông tin về nhân viên
 - **void timNS(int n,Nhanvien nv[])** : Tìm nhân viên theo năm sinh
 - **void capnhat(int n,Nhanvien nv[])** : Cập nhật thông tin về nhân viên

Cấu trúc của chương trình như sau:



a. Chương trình mẫu

```

import java.io.*;
public class QLNV {

    public static void main(String[] args) throws Exception {
        BufferedReader in = new BufferedReader(new
InputStreamReader(System.in));
        Nhanvien nv[] = new Nhanvien[10];
        int n;
        do {
            System.out.print("\nNhập số nhân viên : ");
            n = Integer.parseInt(in.readLine());
        } while ((n < 1) || (n > 10));
        nhapnv(n, nv);
        hienthinv(n, nv);
        timNS(n, nv);
        capnhat(n, nv);
        hienthinv(n, nv);
    }

    public static void nhapnv(int n, Nhanvien nv[]) throws Exception {
        BufferedReader in = new BufferedReader(new
InputStreamReader(System.in));
        int ma, ns;
        String ht, cv;
        for (int i = 0; i < n; i++) {
            System.out.println("Nhập nhân viên thứ " + (i + 1));
            System.out.print("\tMã NV : ");
            ma = Integer.parseInt(in.readLine());
            System.out.print("\tHọ và Tên : ");
            ht = in.readLine();
            System.out.print("\tTên Sinh : ");
            ns = Integer.parseInt(in.readLine());
            System.out.print("\tChức vụ : ");

```

```

        cv = in.readLine();
        Nhanvien x = new Nhanvien(ma, ht, ns, cv);
        nv[i] = x;
    }
}

public static void hienthinv(int n, Nhanvien nv[]) {
    for (int i = 0; i < n; i++) {
        System.out.print("\n[NV " + (i + 1) + "]   ");
        nv[i].hienthi();
    }
}

public static void timNS(int n, Nhanvien nv[]) throws Exception {
    BufferedReader in = new BufferedReader(new
InputStreamReader(System.in));
    int ns, i, kt = 0;
    System.out.print("\nNhap nam sinh can tim : ");
    ns = Integer.parseInt(in.readLine());
    for (i = 0; i < n; i++) {
        if (nv[i].namsinh == ns) {
            nv[i].hienthi();
            kt = 1;
        }
    }
    if (kt == 0) {
        System.out.println("Khong tim thay nhan vien nao.");
    }
}

public static void capnhat(int n, Nhanvien nv[]) throws Exception
{
    BufferedReader in = new BufferedReader(new
InputStreamReader(System.in));
    int ch;
    int ma, ns;
    String ht, cv;
    do {
        System.out.print("\nNhap so thu tu ho so muon cap nhat : ");
        ch = Integer.parseInt(in.readLine());
    } while ((ch < 0) || (ch > n));
    ch--;
    System.out.print("\tMa NV : ");
    ma = Integer.parseInt(in.readLine());
    System.out.print("\tHo va Ten : ");
    ht = in.readLine();
    System.out.print("\tNam Sinh : ");
    ns = Integer.parseInt(in.readLine());
    System.out.print("\tChuc vu : ");
    cv = in.readLine();
    nv[ch].capnhat(ma, ht, ns, cv);
}
}

class Nguoi {
    String hoten;
}

```

```

int namsinh;

Nguoi() {
    hoten = null;
    namsinh = 1900;
}

Nguoi(String ht, int ns) {
    hoten = ht;
    namsinh = ns;
}

public void hienthi() {
    System.out.print(",\tHoten:" + hoten + ",\tNamsinh:" + namsinh);
}
}

class Nhanvien extends Nguoi {

    int manv;
    String chucvu;

    Nhanvien() {
        manv = 0;
        chucvu = null;
    }

    Nhanvien(int ma, String ht, int ns, String cv) {
        super(ht, ns);
        manv = ma;
        chucvu = cv;
    }

    public void hienthi() {
        System.out.print("\nManv : " + manv);
        super.hienthi();
        System.out.print(",\tChucvu : " + chucvu);
    }

    public void capnhat(int ma, String ht, int ns, String cv) {
        super.hoten = ht;
        super.namsinh = ns;
        manv = ma;
        chucvu = cv;
    }
}

```

Kết quả chạy chương trình:

```

Nhập số nhân viên : 2
Nhập nhân viên thứ 1
    Ma NV : 1
        Họ và Tên : Trần Thành
        Năm Sinh : 1990
            Chức vụ : Giám đốc
            Nhập nhân viên thứ 2
    Ma NV : 2
        Họ và Tên : Huong Giang
        Năm Sinh : 1996

```

Chuc vu : Nhan vien

[NV 1]

Manv : 1, Hoten:Tran Thanh, Namsinh:1990,

Chucvu : Giam doc

[NV 2]

Manv : 2, Hoten:Huong Giang, Namsinh:1996,

Chucvu : Nhan vien

Nhap nam sinh can tim : 1990

Manv : 1, Hoten:Tran Thanh, Namsinh:1990,

Chucvu : Giam doc

Nhap so thu tu ho so muon cap nhat : 1

Ma NV : 1

Ho va Ten : Tran Thanh

Nam Sinh : 1990

Chuc vu : Ky su

[NV 1]

Manv : 1, Hoten:Tran Thanh, Namsinh:1990,

Chucvu : Ky su

[NV 2]

Manv : 2, Hoten:Huong Giang, Namsinh:1996,

Chucvu : Nhan vien

Process completed.

Bài 2.7. Tính Ké thừa. Sử dụng toán tử super

Xây dựng lớp Box mô tả hình hộp chữ nhật, có các thuộc tính chiều dài, chiều rộng, chiều cao, hàm tạo và hàm in ra giá trị các thuộc tính.

Xây dựng lớp BoxWeight kế thừa lớp Box mô tả hình hộp chữ nhật có trọng lượng. Ngoài thuộc tính chiều dài, chiều rộng, chiều cao còn có thuộc tính trọng lượng. Lớp này có hàm tạo và phương thức in ra giá trị các thuộc tính.

a. Hướng dẫn

Xây dựng lớp Box gồm:

+ Thuộc tính: chiều dài, chiều rộng, chiều cao

+ Phương thức:

- Hàm tạo 3 tham số

- Hàm Print1() để in ra giá trị 3 thuộc tính.

Xây dựng lớp BoxWeight kế thừa lớp Box, có thêm:

+ Thuộc tính: trọng lượng

+ Phương thức:

- Hàm tạo 4 tham số, gọi hàm tạo 3 tham số của lớp Box bằng toán tử **super**

- Hàm Print2() để in ra giá trị 4 thuộc tính.

b. Chương trình mẫu

```
class Box { // lớp cha
    double width; // chiều rộng
    double height; // chiều cao
    double depth; // chiều dài
    Box (double w, double h, double d){ // hàm khởi tạo
        width = w;
        height = h;
        depth = d;
    }
    public void Print1()
    {
        System.out.print(width+"\t"+height+"\t"+depth);
    }
}
```

```

    }
    class BoxWeight extends Box {           // lớp con của lớp Box
        double weight;                  // trọng lượng
        BoxWeight (double w, double h, double d, double m) {
            super (w,h,d);   // gọi hàm tạo của lớp cha để gán cho 3 biến
            weight = m;          // tự gán cho biến còn lại
        }
    public void Print2()
    {
        System.out.print(width+"\t"+height+"\t"+depth+"\t"+weight);
    }
}
class Main{
    public static void main(String[] args)
    {
        BoxWeight x=new BoxWeight(1,1,2,3);
        x.Print2();
    }
}

```

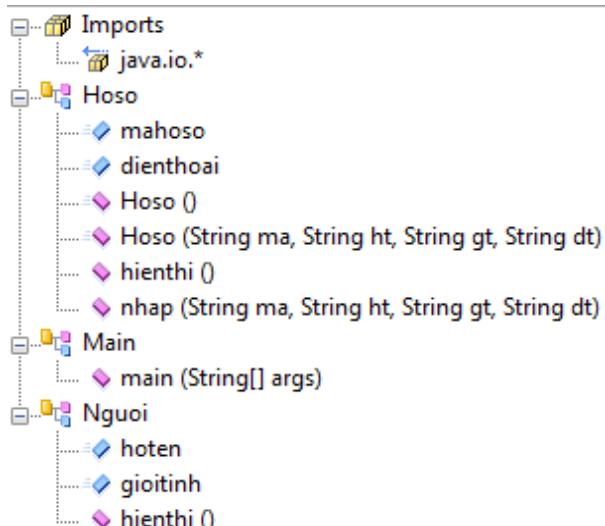
■ Kết quả chạy chương trình:

1.0 1.0 2.0 3.0

■ **Bài 2.8. Lớp trừu tượng**

- Xây dựng lớp trừu tượng Nguoi gồm có thuộc tính Họ tên, giới tính và phương thức hiển thị.
- Xây dựng lớp Hososo kế thừa từ lớp Nguoi, có thêm các thuộc tính Số điện thoại, mã hồ sơ, phương thức khởi tạo, nhập và hiển thị.

a. Hướng dẫn:



b. Chương trình mẫu:

```

import java.io.*;
abstract class Nguoi
{
    String hoten,gioitinh;
    public abstract void hienthi();
};

class Hososo extends Nguoi
{
    String mahoso,dienthoai;

```

```

Hoso()
{
    mahoso=null;
    hoten=null;
    gioitinh=null;
    dienthoai=null;
}
Hoso(String ma, String ht, String gt, String dt)
{
    mahoso=ma;
    hoten=ht;
    gioitinh=gt;
    dienthoai=dt;
}
public void hienthi()
{
    System.out.print("MaHS: " + mahoso + ",\tHoTen : " + hoten);
    System.out.print(",\tDienThoai:" + dienthoai);
    System.out.println(",\tGioiTinh:" + gioitinh);
}
public void nhap(String ma, String ht, String gt, String dt)
{
    mahoso=ma;
    hoten=ht;
    gioitinh=gt;
    dienthoai=dt;
}
}
public class Main
{
    public static void main(String[] args) throws Exception
    {
        BufferedReader in=new BufferedReader(new InputStreamReader(System.in));
        Nguoi person=new Hoso();
        person.hienthi();
        Hoso hs=new Hoso("HS0001", "Tran Nguyen Huong", "Nam", "0988180955");
        hs.hienthi();
        String ma, ht, gt, dt;
        System.out.print("Nhập mã hồ sơ : "); ma=in.readLine();
        System.out.print("Nhập Họ và Tên : "); ht=in.readLine();
        System.out.print("Nhập Giới tính : "); gt=in.readLine();
        System.out.print("Nhập Số điện thoại : "); dt=in.readLine();
        Hoso hs1=new Hoso(ma, ht, gt, dt); hs1.hienthi();
    }
}

```

Kết quả chạy chương trình:

```

MaHS: null,      HoTen : null,      DienThoai:null, GioiTinh:null
MaHS: HS0001,    HoTen : Tran Nguyen Huong, DienThoai:0988180955,
          GioiTinh:Nam
Nhập mã hồ sơ : HS0002
Nhập Họ và Tên : Tran Nguyen Thanh
Nhập Giới tính : Nam
Nhập Số điện thoại : 0987771999
MaHS: HS0002,    HoTen : Tran Nguyen Thanh, DienThoai:0987771999,
          GioiTinh:Nam

```

Bài 2.9. Tính Kế thừa (mỗi class viết trên một file)

Xây dựng lớp Rectangle tính diện tích, chu vi hình chữ nhật. Xây dựng lớp Square kế thừa lớp Rectangle tính diện tích và chu vi hình vuông.

a. Hướng dẫn

- Mỗi lớp lưu trong file *.java tương ứng, các file phải được đặt ở cùng một thư mục.
- Lớp Rectangle (hình chữ nhật) bao gồm một constructor và 2 phương thức Area (tính diện tích) và Perimeter (tính chu vi).
- Hình vuông là một trường hợp đặc biệt của hình chữ nhật nên ta xây dựng lớp Square kế thừa từ lớp Rectangle. Vì hai cạnh hình vuông bằng nhau nên cần viết lại constructor, còn hai phương thức Area và Perimeter của hình chữ nhật thì có thể tái sử dụng để tính diện tích và chu vi cho hình vuông.
- Trong lớp MainPrg, đối tượng sq kiểu Square được khởi tạo với cạnh là 1.23, sau đó được tính diện tích và chu vi thông qua phương thức sq.Area() và sq.Perimeter()

b. Chương trình mẫu:

```
// lưu trong file Rectangle.java, dịch ra file Rectangle.class
public class Rectangle {           // hình chữ nhật
    private float Width, Height; // chiều dài, chiều rộng
    public Rectangle (float w, float h) { // constructor
        Width=w; Height=h;
    }
    public float Area( ){          // diện tích
        return(Width*Height);
    }
    public float Perimeter( ){    // chu vi
        return(2*(Width+Height));
    }
}
// lưu trong file Square.java, dịch ra file Square.class
public class Square extends Rectangle{ // hình vuông
    public Square(float size) {
        super (size,size); //super gọi hàm khởi tạo (constructor) của lớp cha
    }
}
// lưu trong file MainPrg.java
public class MainPrg {
    public static void main(String argv[]) {
        Square sq=new Square(1.0);
        System.out.println("Dien tich: " + sq.Area());
        System.out.println("Chu vi: " + sq.Perimeter());
    }
}
```

💻 Kết quả chạy chương trình:

Dien tich: 3.14
Chu vi: 6.28

Bài 2.10. Tính Đa hình

Trong chương trình dưới đây chúng ta sẽ quan sát cơ chế đa hình thực hiện trên biến tham chiếu. Chúng ta biết rằng một biến tham chiếu kiểu lớp cha (biến **r** trong ví dụ dưới đây) có thể tham chiếu tới đối tượng kiểu lớp con.

```
class A {  
  
    void callme() {  
        System.out.println("phuong thuc callme cua A ");  
    }  
}  
  
class B extends A {  
  
    void callme() {  
        // nạp chồng hàm callme()  
        System.out.println("phuong thuc callme cua B");  
    }  
}  
  
class C extends A {  
  
    void callme() {  
        // nạp chồng hàm callme()  
        System.out.println("phuong thuc callme cua C");  
    }  
}  
  
class Dispatch {  
  
    public static void main(String argv[]) {  
        A a = new A();  
        B b = new B();  
        C c = new C();  
        A r; // r là một biến tham chiếu kiểu A  
        r = a;  
        r.callme();  
        r = b;  
        r.callme();  
        r = c;  
        r.callme();  
    }  
}
```

Kết quả chạy chương trình:

```
phuong thuc callme cua A  
phuong thuc callme cua B  
phuong thuc callme cua C
```

Bài 2.11. Tính đa hình

Xây dựng lớp DongHoKim, DongHoSo đa hình từ lớp DongHo.

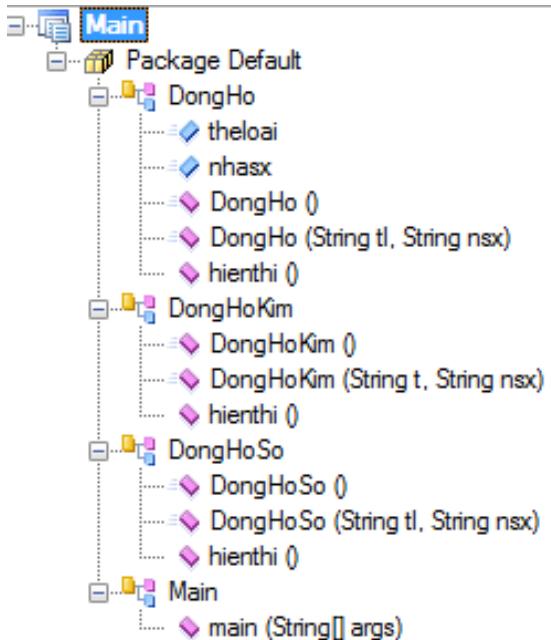
Lớp DongHo gồm có thuộc tính nhà sản xuất, thể loại các hàm tạo và hàm hiển thị.

Lớp DongHoKim kế thừa từ lớp DongHo có các hàm tạo và hàm hiển thị.

Lớp DongHoSo kế thừa từ lớp DongHo có các hàm tạo và hàm hiển thị.

a. Hướng dẫn:

Xây dựng chương trình như sau:



b. Chương trình mẫu:

```
class DongHo
{
    String theloai,nhasx;
    DongHo()
    {
        theloai="DONG HO";
        nhasx="HuongTN";
    }
    DongHo(String tl, String nsx)
    {
        theloai=tl;
        nhasx=nsx;
    }
    public void hienthi()
    {
        System.out.println("[Clock Default] Theloai : ");
        System.out.println(theloai + ",\tNhaSX : " + nhasx);
    }
}
class DongHoKim extends DongHo
{
    DongHoKim()
    {
        theloai="Dong Ho Kim";
        nhasx=null;
    }
    DongHoKim(String t, String nsx)
    {
        theloai=t;
        nhasx=nsx;
    }
    public void hienthi()
    {
        System.out.println("[DongHoKIM] Theloai : ");
        System.out.println(theloai + ",\tNhaSX : " + nhasx);
    }
}
```

```

        }
    }
class DongHoSo extends DongHo
{
    DongHoSo()
    {
        theloai=null;
        nhasx=null;
    }
    DongHoSo(String tl, String nsx)
    {
        theloai=tl;
        nhasx=nsx;
    }
    public void hienthi()
    {
        System.out.println("[DongHoSO] Theloai : ");
        System.out.println(theloai + ",\tNhaSX : " + nhasx);
    }
}
public class Main
{
    public static void main(String[] args)
    {
        System.out.print("Chuc nang 1.1 : ");
        DongHo dh=new DongHo(); dh.hienthi();
        System.out.print("Chuc nang 1.2 : ");
        dh=new DongHo("Gimiko", "Viet Nam"); dh.hienthi();

        System.out.print("Chuc nang 2.1 : ");
        DongHo dh1=new DongHoKim(); dh1.hienthi();
        System.out.print("Chuc nang 2.2 : ");
        dh1=new DongHoKim("Jeko", "Mexiko"); dh1.hienthi();
        System.out.print("Chuc nang 2.3 : ");
        DongHoKim dh2=new DongHoKim(); dh2.hienthi();
        System.out.print("Chuc nang 2.4 : ");
        dh2=new DongHoKim("Sami", "Japan"); dh2.hienthi();

        System.out.print("Chuc nang 3.1 : ");
        DongHo dh3=new DongHoSo(); dh3.hienthi();
        System.out.print("Chuc nang 3.2 : ");
        dh3=new DongHoSo("Zep", "Venezuela"); dh3.hienthi();
        System.out.print("Chuc nang 3.3 : ");
        DongHoSo dh4=new DongHoSo(); dh4.hienthi();
        System.out.print("Chuc nang 3.4 : ");
        dh4=new DongHoSo("Tockia", "Paris"); dh4.hienthi();
    }
}
}

```

Kết quả chạy chương trình:

```

Chuc nang 1.1 : [Clock Default] Theloai : DONG HO, NhaSX : HUONGTN
Chuc nang 1.2 : [Clock Default] Theloai : Gimiko, NhaSX : Viet Nam
Chuc nang 2.1 : [DongHoKIM] Theloai : Dong Ho Kim, NhaSX : null
Chuc nang 2.2 : [DongHoKIM] Theloai : Jeko, NhaSX : Mexiko
Chuc nang 2.3 : [DongHoKIM] Theloai : Dong Ho Kim, NhaSX : null
Chuc nang 2.4 : [DongHoKIM] Theloai : Sami, NhaSX : Japan
Chuc nang 3.1 : [DongHoSO] Theloai : null, NhaSX : null
Chuc nang 3.2 : [DongHoSO] Theloai : Zep, NhaSX : Venezuela
Chuc nang 3.3 : [DongHoSO] Theloai : null, NhaSX : null
Chuc nang 3.4 : [DongHoSO] Theloai : Tockia, NhaSX : Paris

```

Bài 2.12. Tính đa hình

Trong bài tập này, lớp **Figure** biểu diễn những hình hai chiều tổng quát và lưu trữ kích thước của chúng bằng hai thuộc tính **dim1** và **dim2**. Vì chưa biết rõ đó là hình gì nên không thể tính được diện tích, vì thế phương thức tính diện tích **area()** chỉ hiển thị dòng chữ: "Khong the tich duoc dien tich"

Nhưng đến hai lớp con là **Rectangle** (biểu diễn hình chữ nhật) và **Triangle** (hình tam giác vuông) thì mọi việc trở nên rõ ràng hơn. Mỗi lớp con sẽ định nghĩa phương thức **area()** theo cách tính diện tích riêng của mình.

Với **Rectangle** thì: $area = dim1 * dim2$;

còn với **Triangle** thì: $area = dim1 * dim2 / 2$;

* Chương trình mẫu:

```
class Figure {  
  
    double dim1, dim2;  
  
    Figure(double a, double b) {  
        dim1 = a;  
        dim2 = b;  
    }  
  
    double area() {  
        System.out.println("Khong the tich duoc dien tich");  
        return 0;  
    }  
}  
  
class Rectangle extends Figure {  
  
    Rectangle(double a, double b) {  
        super(a, b);  
    }  
  
    double area() {  
        System.out.println("Dien tich hinh chu nhat");  
        return dim1 * dim2;  
    }  
}  
  
class Triangle extends Figure {  
  
    Triangle(double a, double b) {  
        super(a, b);  
    }  
  
    double area() {  
        System.out.println("Dien tich tam giac");  
        return dim1 * dim2 / 2;  
    }  
}  
  
public class Polymorphism {  
  
    public static void main(String[] args) {
```

```

        Figure f = new Figure(10, 10);
        Rectangle r = new Rectangle(9, 5);
        Triangle t = new Triangle(10, 8);
        Figure fig;
        fig = r;
        System.out.println("Dientich=" + fig.area());
        fig = t;
        System.out.println("Dientich=" + fig.area());
        fig = f;
        System.out.println("Dientich=" + fig.area());
    }
}

```

Kết quả chạy chương trình:

```

Dien tich hinh chu nhat
Dientich=45.0
Dien tich tam giac
Dientich=40.0
Khong the tinh duoc dien tich
Dientich=0.0

```

Bài 2.13. Giao diện

Mỗi sản phẩm đều có tên nhà sản xuất, điện thoại nhà sản xuất và giá sản phẩm. Hãy xây dựng giao diện sanpham, sau đó xây dựng lớp xemay thực thi giao diện sản phẩm.

* Chương trình mẫu:

```

interface sanpham
{
    static final String nhasx = "Honda VN";
    static final String dienthoai = "08-8123456";
    public int gia(String s_model);
}

// khai báo 1 lớp có cài đặt interface
class xemay implements sanpham
{
    // cài đặt lại phương thức của giao diện trong lớp
    public int gia(String s_model)
    {
        if (s_model.equals("2005")) return (2000);
        else return (1500);
    }

    public String chobietnhasx()
    {
        return (nhasx);
    }
}

public class Inter {
    public static void main(String[] args) {
        xemay x=new xemay();
        System.out.print(x.chobietnhasx()+x.gia(" 2005"));
    }
}

```

Kết quả chạy chương trình:

```

Honda VN 2000

```

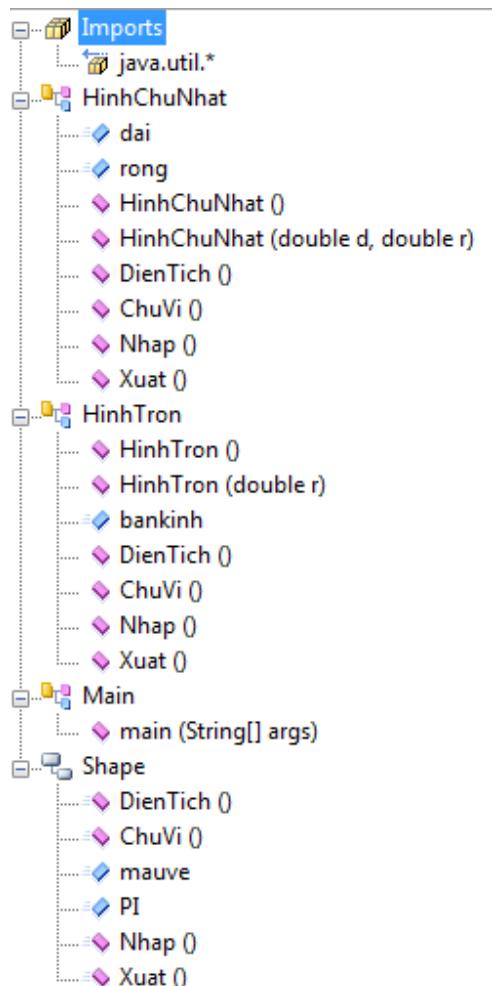
Bài 2.14. Giao diện

Xây dựng giao diện Shape để mô tả đối tượng hình học, gồm có thuộc tính màu vẽ, bán kính, hàm nhập, hàm xuất, hàm tính diện tích, chu vi.

Xây dựng lớp HinChuNhat kế thừa giao diện Shape, có các thuộc tính chiều dài, chiều rộng và cài đặt các hàm của giao diện

Xây dựng lớp HinTron kế thừa giao diện Shape, có các thuộc tính chiều dài, chiều rộng và cài đặt các hàm của giao diện

a. Hướng dẫn



b. Chương trình mẫu:

```
//Main.java
import java.util.*;
interface Shape {

    double DienTich();

    double ChuVi();
    double mauve = 0;
    double PI = 3.14;

    void Nhaph();

    void Xuat();
```

```

}

class HinhChuNhat implements Shape {

    double dai, rong;

    public HinhChuNhat() {
        dai = rong = 0;
    }

    public HinhChuNhat(double d, double r) {
        dai = d;
        rong = r;
    }

    public double DienTich() {
        return dai * rong;
    }

    public double ChuVi() {
        return 2 * (dai + rong);
    }

    public void Nhap() {
        try {
            Scanner in = new Scanner(System.in);
            System.out.print("Chieu dai=");
            dai = in.nextDouble();
            System.out.print("Chieu rong=");
            rong = in.nextDouble();
        } catch (Exception e) {
            System.out.print("Loi");
        }
    }

    public void Xuat() {
        System.out.println("Dai=" + dai);
        System.out.println("Rong=" + rong);
        System.out.println("Mau=" + mauve);
        System.out.println("Dien tich=" + DienTich());
        System.out.println("CHu vi=" + ChuVi());
    }
}

class HinhTron implements Shape {

    double bankinh;

    public HinhTron() {
        bankinh = 0;
    }

    public HinhTron(double r) {

```

```

        bankinh = r;
    }

    public double DienTich() {
        return PI * bankinh * bankinh;
    }

    public double ChuVi() {
        return 2 * PI * bankinh;
    }

    public void Nhap() {
        try {
            Scanner in = new Scanner(System.in);
            System.out.print("Ban kinh=");
            bankinh = in.nextDouble();
        } catch (Exception e) {
            System.out.print("Loi");
        }
    }

    public void Xuat() {
        System.out.println("Ban kinh=" + bankinh);
        System.out.println("Mau=" + mauve);
        System.out.println("Dien tich=" + DienTich());
        System.out.println("Chu vi=" + ChuVi());
    }
}

class Main {

    public static void main(String[] args) {
        System.out.print("Hinh chu nhat");
        HinhChuNhat a = new HinhChuNhat();
        a.Nhap();
        a.Xuat();
        System.out.print("Hinh tron");
        HinhTron b = new HinhTron(1);
        b.Xuat();
    }
}

```

Kết quả chạy chương trình:

1. Hinh chu nhat

Chieu dai=4

Chieu rong=2

Dai=4.0

Rong=2.0

Mau=0.0

Dien tich=8.0

CHu vi=12.0

2. Hinh tron

Ban kinh=1.0

```
Mau=0.0  
Dien tich=3.14  
Chu vi=6.28
```

2.2.2. Bài tập tự giải

Bài 2.15. Cho biết kết quả in ra màn hình của chương trình sau:

```
class Base{  
    int i;  
    Base() {  
        add(1);  
    }  
    void add(int n) { i+=n; }  
    void print(){  
        System.out.println(i);  
    }  
}  
class Ext extends Base{  
    Ext() { add(2); }  
    void add(int n) { i+=n*2; }  
}  
public class Q3{  
    public static void main(String[] args) {  
        bibo(new Ext());  
    }  
    static void bibo(Base b){  
        b.add(8);  
        b.print();  
    }  
}
```

Bài 2.16. Cho trước đoạn chương trình sau:

```
class A{}  
class B extends A{}  
class C extends A{}  
public class Q4{  
    public static void main(String[] args) {  
        A x=new A();  
        B y=new B();  
        C z=new C();  
        //Nơi cần chèn các lệnh hợp lệ (1)  
    }  
}
```

Hãy chọn những lệnh thích hợp sau:

- (a) x=y;
- (b) z=x;
- (c) y=(B) x;
- (d) z=(C) y;
- (e) y=(A) y;

Bài 2.17. Hãy cho biết kết quả in lên màn hình khi chạy chương trình sau:

```

public class Q5{
    int a, b;
    public void f() {
        a=b=0;
        int[] c={0};
        g(b,c);
        System.out.println(a""+b""+c[0]);
    }
    public void g(int b, int[] c) {
        a=b=1;
        c[0]=1;
    }
    public static void main(String[] args) {
        Q5 e = new Q5();
        e.f();
    }
}

```

Bài 2.18. Hãy cho biết kết quả in lên màn hình khi chạy chương trình sau:

```

class A {
int i;
}
class B extends A {
int i; // biến i này che khuất biến cùng tên của lớp cha A
B(int a, int b) {
super.i = a; // biến i của lớp cha A
i = b; // biến i của lớp con B
}
void show( ) {
System.out.println(" bien i cua lop cha = " + super.i);
System.out.println(" bien i cua lop con = " + i);
}
class UseSuper {
public static void main(String argv[]) {
B subOb = new B(1,2);
subOb.show();
}
}

```

Bài 2.19. Hãy lập chương trình xây dựng lớp các đối tượng SINHVIEN gồm:

- Các thuộc tính: họ tên, năm sinh, điểm thi
- Các phương thức: NHAP, HIEN.

Chương trình chính tạo ra 2 lớp đối tượng sinh viên, gọi phương thức NHAP và HIEN trên 2 đối tượng đó.

Bài 2.20. Xây dựng lớp thuê bao mobile gồm:

- Các thuộc tính: tên khách hàng, số thuê bao, tiền cước phí
- Các phương thức: NHAP, HIEN.

Chương trình chính: Nhập danh sách n khách hàng từ bàn phím. Sau đó hiện thuê bao có cước phí lớn nhất.

Bài 2.21: Xây dựng lớp cơ sở HOCSINH có các thông tin: Họ tên, lớp, điểm toán, lý, hóa và các phương thức nhập, xuất dữ liệu, tính điểm trung bình.

Viết chương trình chính thực hiện nhập vào một danh sách n học sinh, sau đó hiển thị danh sách những học sinh có điểm trung bình ≥ 5 .

Bài 2.22: Thông tin về một thí sinh gồm: số báo danh, họ, tên, năm sinh, điểm toán, điểm lý, điểm hóa.

Hãy xây dựng đối tượng thí sinh và viết chương trình thực hiện các thao tác xử lí sau đây trên một danh sách thí sinh:

- Nhập, xuất danh sách thí sinh.
- Tìm kiếm thí sinh có tổng điểm 2 môn cao nhất.
- In danh sách thí sinh có tổng điểm 3 môn ≥ 20 .

Bài 2.23:

- a) Hãy xây dựng lớp SinhVien có các thuộc tính riêng (private): Số báo danh, họ và tên sinh viên, địa chỉ, môn học, điểm thi học kỳ I, điểm thi học kỳ II.
- b) Viết các hàm nhập xuất thành phần dữ liệu và các hàm tạo cho lớp SinhVien
- c) Viết chương trình chính để tạo ra danh sách SinhVien và hiển thị thực đơn:
 1. Nhập vào thông tin về sinh viên
 2. Xem thông tin về sinh viên
 3. Tìm sinh viên theo điểm
 4. Kết thúc chương trình
- Viết các hàm thành phần của lớp để thực hiện các nhiệm vụ trên.

Bài 2.24: Bài tập về tính kế thừa

a) Xây dựng lớp SinhVien có cấu trúc sau:

- Thuộc tính: MaSinhVien: kiểu String
HoTen: String
DiemTrungBinh: double
XepLoai: kiểu String

Thuộc tính XepLoai sẽ được tính dựa trên giá trị của trường DiemTrungBinh như sau:

DiemTrungBinh < 5, XepLoai = Kem;
5 \leq DiemTrungBinh < 6.5, XepLoai = Trung Binh;
6.5 \leq DiemTrungBinh < 8, XepLoai = Kha;
DiemTrungBinh > 8, XepLoai = Gioi.

b) Xây dựng lớp TuyenSinh làm những việc như sau:

- *Thuộc tính:* gồm một mảng tên DanhSach có kiểu SinhVien với kích thước 20 phần tử.
- *Phương thức:*

+ Void addSinhVien(String ma, String hoten, double diemtb): Dùng để thêm sinh viên mới vào mảng DanhSach. Phải kiểm tra nếu DanhSach đầy thì đưa ra thông báo lỗi, nếu không thì đưa thông tin của sinh viên mới vào cuối danh sách.
+ Void hienThiSinhVien(): Hiển thị tất cả sinh viên có trong danh sách ra màn hình.
+ Void timKiemSinhVien(String xl): Tìm kiếm các sinh viên theo xếp loại. Ví dụ: các sinh viên loại giỏi...

c) *Chương trình chính:* chạy **class** TuyenSinh. Khi chạy chương trình có giao diện như sau:

CHUONG TRINH QUAN LY TUYEN SINH

1. Nhập sinh viên mới
2. Tìm kiếm sinh viên theo xếp loại
3. Hiển thị tất cả sinh viên
4. Thoát chương trình

Chon Menu (1-4):

Người sử dụng nhập từ 1 đến 4 để chọn menu tương ứng. Mỗi menu sẽ thực hiện một phương thức của đối tượng của lớp TuyenSinh.

Bài 2.25: Kế thừa

Xây dựng lớp CD gồm:

- Thuộc tính:

- o cdType: dữ liệu nhập vào là 1 trong 2 dạng: audio hoặc video
- o title: String
- o price: double
- o cdId: int

- Constructor:

- o Constructor mặc định: đặt các giá trị của trường là giá trị mặc định
- o Constructor 2: truyền các giá trị cho các trường

Xây dựng lớp CDList như sau:

- Thuộc tính:

- o cdList: mảng CDDetail, gồm 100 phần tử
- o cdcounter: int – đếm số cd tăng dần

- Constructor: một constructor mặc định

- Phương thức:

- o addCD(): nhập các giá trị của cd từ bàn phím. Lưu ý các điều kiện nhập dữ liệu
- o searchCD(): tìm kiếm CD theo title
- o displayCD(): hiển thị các cd trong list

Trong chương trình chính, tạo đối tượng CDList. Khi chạy chương trình, hiển thị giao diện gồm menu 4 lựa chọn:

1. Add CD to the catalog
2. Search CD by CD title
3. Display the catalog
4. Exit

Please choose option number:

Người dùng nhập vào các số từ 1 đến 4 để chọn menu. Tuỳ theo từng menu được chọn, sử dụng các chức năng tương ứng của đối tượng CDList.

Bài 2.26. Kế thừa

a) Xây dựng lớp NGUOI

- + Các thuộc tính: Tên, ngày sinh
- + Các phương thức:
 - Phương thức NHAP (nhập) để nhập các thông tin: Tên, ngày sinh
 - Phương thức HIEN (hiện) để nhập các thông tin: Tên, ngày sinh

b) Xây dựng lớp THISINH: kế thừa từ lớp NGUOI

- + Các thuộc tính: Số báo danh, điểm Toán, điểm Văn, điểm tiếng Anh, tổng điểm
- + Các phương thức:
 - Các phương thức NHAP và HIEN

Phương thức NHAP:

- o Gọi phương thức NHAP của lớp NGUOI,
- o Ngoài ra nhập các thông tin sau từ bàn phím: Số báo danh, điểm Toán, điểm Văn, điểm tiếng Anh.
- o Tổng điểm tính theo cách sau:

$$\text{Tổng điểm} = \text{điểm Toán} + \text{điểm Văn} + \text{điểm tiếng Anh}$$

Phương thức HIEN: Hiện ra các thông tin về thí sinh theo trình tự sau: Số báo danh, tên, ngày sinh, điểm Toán, điểm Văn, điểm tiếng Anh, tổng điểm.

c) Từ các lớp đã xây dựng trên, chương trình chính thực hiện:

- + Nhập vào danh sách gồm n thí sinh từ bàn phím. In ra danh sách thí sinh vừa nhập.
- + Nhập điểm chuẩn X vào từ bàn phím. Điều kiện để thí sinh viên trúng tuyển là điểm mỗi môn phải lớn hơn 0 và tổng điểm phải lớn hơn hoặc bằng X. Tìm và in ra màn hình danh sách các thí sinh trúng tuyển.
- + Sắp xếp danh sách thí sinh trúng tuyển theo tổng điểm giảm dần, nếu các thí sinh trùng tổng điểm thì sắp xếp theo tên theo thứ tự A, B, C, v.v. In kết quả vừa vừa sắp xếp ra màn hình.

Bài 2.27. Kế thừa

a) Xây dựng lớp NGUOI

- + Các thuộc tính: Tên, ngày sinh
- + Các phương thức:
 - Phương thức NHAP (nhập) để nhập Tên, ngày sinh và HIEN (hiện) để hiện tên, ngày sinh

b) Xây dựng lớp GIANGVIEN: Kế thừa lớp NGUOI

- + Các thuộc tính: Mã giảng viên, chức vụ, lương, phụ cấp, tổng lương
- + Các phương thức: NHAP và HIEN

Phương thức NHAP:

- o Gọi phương thức NHAP của lớp NGUOI
- o Tổng lương được tự động tính theo cách:

$$\text{Tổng lương} = \text{lương} + \text{phụ cấp}$$

Nếu chức vụ là “trưởng khoa” thì phụ cấp là 1000.000, “phó khoa” thì phụ cấp 800.000, “trưởng bộ môn” thì phụ cấp 500.000, còn lại phụ cấp là 0.

- o Mã giảng viên được nhập từ bàn phím

Phương thức HIEN: Hiện tất cả thông tin về giảng viên: mã giảng viên, tên, ngày sinh, chức vụ, lương, phụ cấp, tổng lương.

c) Từ các lớp đã xây dựng trên, chương trình chính thực hiện:

- + Nhập vào danh sách gồm n giảng viên từ bàn phím. In ra danh sách vừa nhập
- + Tìm kiếm một giảng viên có mã giảng viên nhập vào từ bàn phím. Hiện tất cả thông tin về giảng viên này lên màn hình.
- + Với những giảng viên có không có phụ cấp, hãy sắp xếp theo tổng lương giảm dần, nếu trùng tổng lương, hãy sắp xếp theo tên giảng viên theo thứ tự A, B, C, v.v. In danh sách vừa sắp xếp ra màn hình.

Bài 2.28. Tính đa hình: Sử dụng giao diện

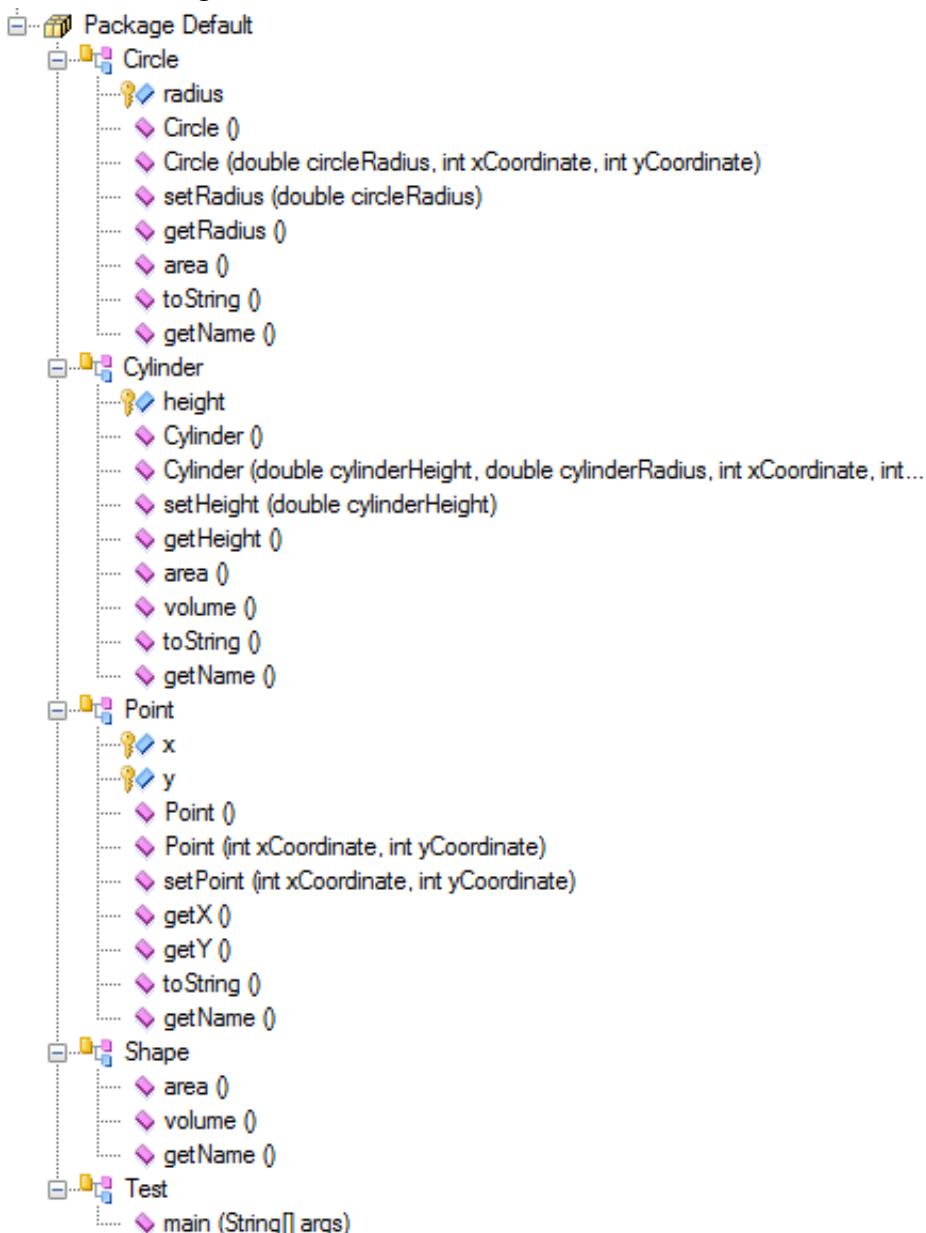
Sử dụng tính đa hình (polymorphism) trong phân cấp kế thừa thông qua việc mô tả và xử lý một số thao tác cơ bản trên các đối tượng hình học.

Yêu cầu:

- a) Định nghĩa giao diện có tên Shape có hàm tính diện tích (area) và thể tích (volume) và trả về tên của một hình (getName). Giao diện này đặt trong file Shape.java
- b) Xây dựng lớp Point kế thừa từ giao diện Shape, mô tả đối tượng điểm trong mặt phẳng. Gồm có 2 tọa độ x, y, các hàm tạo không tham số và 2 tham số, hàm lấy hoành độ (getX), hàm lấy tung độ (getY) và hàm lấy tên hình (getName). Lớp này đặt trong file Point.java

- c) Xây dựng lớp Circle kế thừa lớp Point gồm có thuộc tính bán kính (radius), tọa độ tâm, hàm tạo không tham số và ba tham số (tọa độ tâm, bán kính), hàm lấy bán kính (getRadius), hàm tính diện tích (area) và hàm trả về tên (getName) của hình tròn. Lớp này đặt trong file Circle.java
- d) Xây dựng lớp Cylinder kế thừa lớp Circle, có thêm thuộc tính chiều cao (height), các hàm tạo không tham số và hàm tạo 4 tham số (tọa độ tâm, bán kính, chiều cao), hàm lấy chiều cao (getHeight), hàm tính diện tích xung quanh (area), hàm tính thể tích (volume) và hàm trả về tên hình (getName). Lớp này đặt trong file Cylinder.java
- e) Xây dựng lớp Test để kiểm tra tính kế thừa của Point, Circle, Cylinder với giao diện Shape.

Cấu trúc chương trình như sau:



CHƯƠNG 3. LẬP TRÌNH GIAO DIỆN TRÊN WINDOWS

3.1. PHẦN LÝ THUYẾT

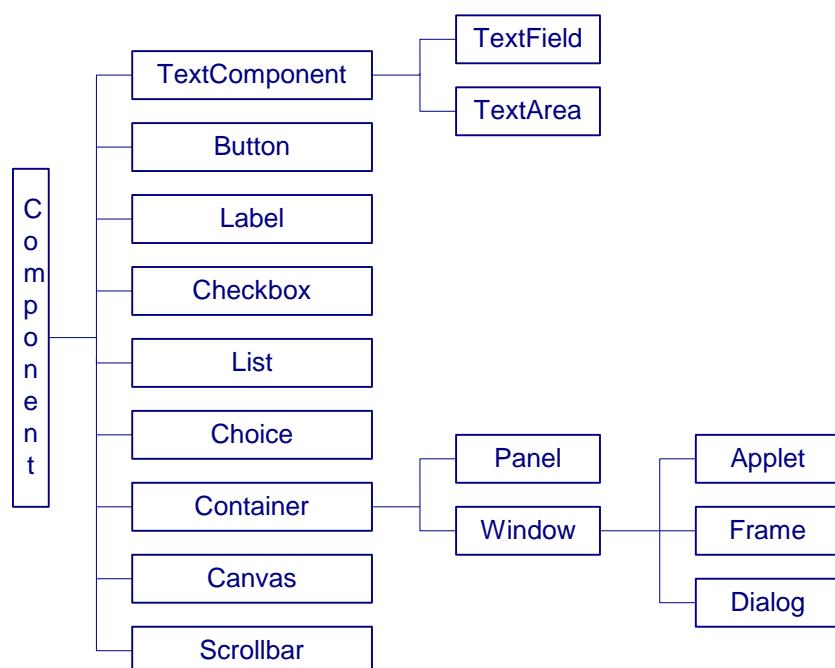
3.1.1. Containers

Gói java.awt chứa một lớp gọi là Container. Lớp này trực tiếp hay gián tiếp phát sinh ra hai container được sử dụng phổ biến nhất là Frame và Panel.

Kịch bản chung để xây dựng giao diện ứng dụng:

- a. Tạo ra một frame có tên, ví dụ “My Frame”:
`Frame guiFrame = new Frame ("My Frame");`
- b. Xây dựng một cấu trúc phân cấp các thành phần bằng cách sử dụng hàm add() để bổ sung thêm panel
`guiFrame.add(new Button ("OK")); // Đưa vào một nút (Button) có tên "OK"`
- c. Đặt lại kích thước cho frame sử dụng hàm setSize():
`guiFrame.setSize(200, 300); // Đặt lại khung frame là 200 x 300`
- d. Gói khung frame đó lại bằng hàm pack(): `guiFrame.pack();`
- e. Làm cho frame đó nhìn thấy được: `guiFrame.setVisible(true);`

3.1.2. Components



Hình 3.1: Cấu trúc gói các thành phần điều khiển

a. Nhãn (Label)

* Constructor sau đây được sử dụng để tạo một label:

- Label(): Tạo một Label trống.
- Label(String labeltext): Tạo một Label với văn bản được cho.
- Label(String labeltext, int alignment): Tạo một Label với một chế độ canh lề alignment được cho, alignment có thể là Label.LEFT, Label.RIGHT hay

Label.CENTER.

* Các phương thức được sử dụng phổ biến của label được trình bày ở bảng bên dưới:

Bảng 3.1. Phương thức của Label

Phương thức	Chức năng
setFont(Font f)	Thay đổi phông chữ đang được chọn của Label
setText(String s)	Thiết lập nhãn cho Label
getText()	Lấy nội dung hiện hành của Label

b. Ô văn bản (TextField)

* Constructor sau có thể được sử dụng để tạo một textfield:

- TextField(): Tạo một textfield mới.
- TextField(int columns): Tạo một textfield mới với số cột được cho.
- TextField(String s): Tạo một textfield mới với chuỗi văn bản được cho.
- TextField(String s, int columns): Tạo một textfield mới với nhãn và số cột được cho.

* Các phương thức thường được sử dụng của đối tượng TextField được tóm tắt trong bảng sau:

Bảng 3.2. Phương thức của TextField

Phương thức	Chức năng
setEchoChar(char)	Thiết lập các kí tự được trình bày trong dạng của một kí tự được cho.
setText(String s)	Thiết lập nhãn cho TextField.
getText()	Trả về nhãn của TextField.
setEditable(boolean)	Xác định trường có thể được soạn thảo hay không. Trường chỉ được soạn thảo khi giá trị này được đặt là True.
isEditable()	Xác định xem trường có đang trong mode soạn thảo hay không. Giá trị trả về kiểu Boolean.

c. Vùng văn bản (TextArea)

* Các constructor sau để tạo TextArea:

- TextArea(): Tạo một TextArea mới.
- TextArea(int rows, int cols): Tạo một TextArea mới với số lượng cột và dòng được cho.
- TextArea(String text): Tạo một TextArea mới với nhãn được cho.
- TextArea(String text, int rows, int cols): Tạo một TextArea mới với nhãn, số dòng và số cột được cho.

* Các phương thức thường được sử dụng nhiều nhất của TextArea:

Bảng 3.3. Phương thức của TextArea

Phương thức	Chức năng
setText(String)	Thiết lập nhãn cho TextArea.
getText()	Trả về nhãn của TextArea.
setEditable(boolean)	Xác định xem trường có thể được soạn thảo hay không. Trường có thể được soạn thảo khi giá trị này là True.

isEdiable()	Xác định xem trường có đang trong mode soạn thảo được không. Trả về giá trị là kiểu Boolean.
insertText(String, int)	Chèn String được cho vào vị trí index được cho.
replaceText(String, int, int)	Thay thế văn bản nằm giữa vị trí int, int được cho.

d. Nút bấm (Button)

* Để tạo một button, bạn làm theo các bước sau:

- 1) Tạo phần tử button với một nhãn chỉ ra mục đích của button.
- 2) Bố trí phần tử này trên màn hình.
- 3) Hiển thị phần tử trên màn hình.

* Constructor sau để tạo các button trong Java:

- Button()
- Button(String text)

Sử dụng setLabel() và getLabel() để thiết lập và nhận về nhãn của button.

e. Checkbox và RadioButton

* Làm theo các bước sau để tạo các checkbox hay radiobutton:

- 1) Tạo phần tử.
- 2) Quyết định trạng thái khởi đầu của phần tử (chọn hay không chọn).
- 3) Bố trí các phần tử trên màn hình.
- 4) Hiển thị các phần tử trên màn hình.

Thành phần checkbox có thể sử dụng một lớp phụ được gọi là CheckboxGroup để tạo ra các radiobutton.

* Các constructor sau để tạo các checkbox trong Java:

- Checkbox(): Tạo một checkbox trống.
- Checkbox(String text): Tạo một checkbox với nhãn được cho.

Để tạo các radiobutton, đầu tiên chúng ta tạo đối tượng CheckboxGroup như sau:

```
CheckboxGroup cg=new CheckboxGroup();
```

Sau đó chúng ta tạo các button, như chỉ ra dưới đây:

```
Checkbox male=new Checkbox("male", cg, true);
Checkbox female=new Checkbox("female", cg, false);
```

* Sử dụng các phương thức setState() và getState() để thiết lập và nhận về trạng thái của checkbox.

f. Danh sách chọn lựa (Choice List)

* Tạo các danh sách chọn lựa theo các bước đây:

- 1) Tạo danh sách các phần tử.
- 2) Thêm các item (có kiểu là String) vào danh sách, mỗi lần chỉ thêm được một item.
- 3) Bố trí danh sách trên màn hình.
- 4) Hiển thị danh sách trên màn hình.

* Mỗi thời điểm chỉ thêm được một item bằng cách sử dụng phương thức addItem như được chỉ ra bên dưới:

```

Choice colors=new Choice();
colors.addItem("Red");
colors.addItem("Green");

```

3.1.3. Layout Manager

a. FlowLayout manager

* FlowLayout là layout manager mặc định cho các applet và các panel. Các thành phần được xếp xép từ góc trái trên đến góc phải dưới của màn hình. Khi một số thành phần được tạo, chúng được xếp xép theo hàng, từ trái sang phải.

* Các constructor của FlowLayout:

- FlowLayout mylayout = new FlowLayout() // constructor
- FlowLayout exLayout=new FlowLayout(FlowLayout.RIGHT);
- setLayout(exLayout); //đặt layout là Flowlayout
- setLayout(new FlowLayout(FlowLayout.RIGHT));

b. GridLayout Manager

* GridLayout trợ giúp việc chia container vào trong ô lưới. Các thành phần được đặt trong các dòng và các cột. Mỗi khung lưới nên chứa ít nhất một thành phần. Một khung lưới được sử dụng khi tất cả các thành phần có cùng kích thước.

* Constructor GridLayout được tạo như sau:

```
GridLayout g1=new GridLayout(4, 3); //4 là số dòng và 3 là số cột.
```

c. GridBagLayout Manager

* GridBagLayout hiệu quả và phức tạp hơn bất cứ layout nào khác. Layout này đặt các thành phần vào vị trí chính xác. Với layout này, các thành phần không cần có cùng kích thước. Nó tương tự như GridLayout manager, khi các thành phần được xếp xép trong lưới theo dòng và cột. Tuy nhiên, thứ tự đặt các thành phần không theo nguyên tắc từ trái sang phải và từ trên xuống dưới.

```

GridBagLayout gb=new GridBagLayout()
ContainerName.setLayout(gb);

```

Để sử dụng layout này, bạn cần cung cấp thông tin về kích thước và layout của mỗi thành phần. Lớp ‘GridBagConstraints’ nắm giữ tất cả các thông tin mà lớp GridLayout cần để bố trí và định kích thước mỗi thành phần. Bảng sau liệt kê danh sách các biến thành viên của lớp GridBagConstraints:

Bảng 3.4. Các biến thành viên của GridBagConstraints

Các biến thành viên	Mục đích
weightx, weighty	Chỉ ra sự phân phối của khoảng trống trong GridBagLayout. Giá trị mặc định cho các biến này là 0.
gridwidth, gridheight	Chỉ ra số lượng các ô (cell) bắt ngang hay đi xuống trong vùng hiển thị của một thành phần.
ipadx, ipady	Chỉ ra lượng làm thay đổi chiều cao và chiều rộng tối thiểu của thành phần. Nó sẽ thêm 2*ipadx vào chiều rộng tối thiểu và 2*ipady vào chiều cao tối thiểu của thành phần. Giá trị mặc định cho cả hai là 0.

Các biến thành viên	Mục đích
Anchor	Chỉ ra cách xếp xép các thành phần trong cell. Mặc định sẽ đặt vào giữa cell. Các thành viên dữ liệu tĩnh sau đây có thể được sử dụng: <ul style="list-style-type: none"> - GridBagConstraints.NORTH - GridBagConstraints.EAST - GridBagConstraints.WEST - GridBagConstraints.SOUTH - GridBagConstraints.NORTHEAST - GridBagConstraints.SOUTHEAST
gridx, gridy	Chỉ ra cell cần đặt một thành phần. Khi thiết lập giá trị của gridx là ‘GridbagConstraints.RELATIVE’ thì thành phần được thêm sẽ nằm ở vị trí bên phải của thành phần cuối cùng.
Fill	Chỉ ra cách mà một thành phần được bố trí vào cell thế nào nếu như cell lớn hơn thành phần. Mặc định kích thước thành phần lúc đó không thay đổi.

Bảng sau đây cung cấp một danh sách các biến dữ liệu tĩnh là các giá trị cho biến fill:

Bảng 3.5. Các biến dữ liệu tĩnh là các giá trị cho biến fill

Giá trị	Mô tả
GridBagConstraints.NONE	Mặc định, không làm thay đổi kích thước của thành phần.
GridBagConstraints.HORIZONTAL	Tăng chiều rộng của thành phần theo chiều ngang (HORIZONTAL) để làm cho thành phần khớp với chiều ngang.
GridBagConstraints.VERTICAL	Tăng chiều cao của thành phần theo chiều đứng (VERTICAL) để làm cho thành phần khớp với chiều dọc.
GridBagConstraints.BOTH	Tăng chiều rộng, chiều cao của thành phần theo cả chiều ngang và chiều dọc.
Insets	Xác định khoảng cách top, bottom, left và right giữa các thành phần. Mặc định là 0.

Sử dụng phương thức ‘setConstraints()’ để thiết lập các hằng số cho mỗi thành phần. Ví dụ: `gblay.setConstraints(lbl1, gbc);`
‘gblay’ là đối tượng của lớp GridBagLayout, lbl là thành phần ‘Label’ và ‘gbc’ là đối tượng của lớp GridBagConstraints.

3.1.4. Xử lý các sự kiện

Một Event Listener lắng nghe một sự kiện nào đó mà một đối tượng thiết lập. Nghĩa là sẽ luôn phiên gọi phương thức xử lý sự kiện. Mỗi event listener cung cấp các phương thức xử lý những sự kiện này. Lớp thi hành listener cần phải định nghĩa những phương thức này. Để sử dụng mô hình này, bạn làm theo các bước sau:

- Thực hiện giao diện listener thích hợp. Cấu trúc như sau:

```
public class MyApp extends Frame implements ActionListener
```

- Xác định tất cả các thành phần tạo ra sự kiện. Các thành phần có thể là các button, label, menu item, hay window.

Bảng sau đây chỉ ra các sự kiện khác nhau và mô tả về chúng:

Bảng 3.6. Các sự kiện

Lớp sự kiện	Mô tả
ActionEvent	Phát sinh khi một button được nhấn, một item trong danh sách

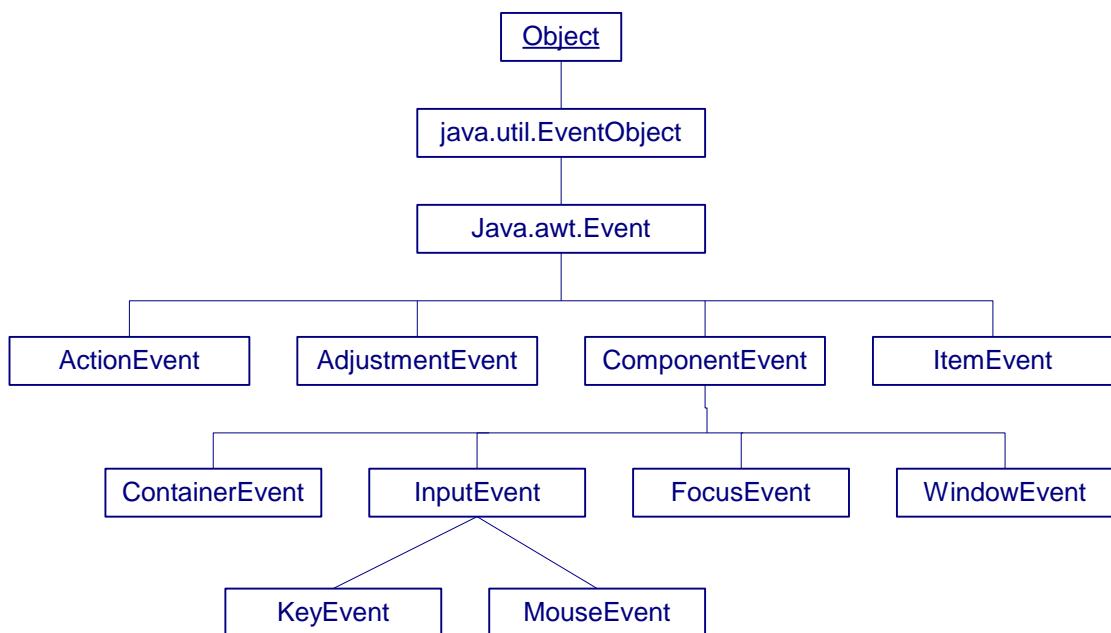
Lớp sự kiện	Mô tả
	chọn lựa được nhấp đôi hay một menu được chọn.
AdjustmentEvent	Phát sinh khi một thanh scrollbar được sử dụng.
ComponentEvent	Phát sinh khi một thành phần được thay đổi kích thước, được di chuyển, bị ẩn hay làm cho hoạt động được.
FocusEvent	Phát sinh khi một thành phần mất hay nhận focus từ bàn phím.
ItemEvent	Phát sinh khi một menu item được chọn hay bỏ chọn; hay khi một checkbox hay một item trong danh sách được click.
WindowEvent	Phát sinh khi một cửa sổ được kích hoạt, được đóng, được mở hay thoát.
TextEvent	Phát sinh khi giá trị trong thành phần text field hay text area bị thay đổi.
MouseEvent	Phát sinh khi chuột di chuyển, được click, được kéo hay bị thả ra.
KeyEvent	Phát sinh khi input được nhận từ bàn phím.

Các giao diện được thi hành để xử lý một trong số những sự kiện này là:

- ActionListener
- AdjustmentListener
- ComponentListener
- FocusListener
- ItemListener
- WindowListener
- TextListener
- MouseListener
- MouseMotionListener
- KeyListener

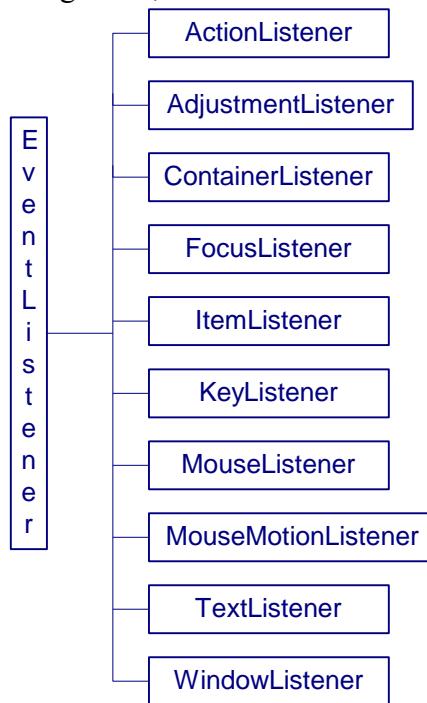
Các giao diện định nghĩa một số phương thức để xử lý mỗi sự kiện. Những phương thức này sẽ được nạp chồng trong lớp mà thi hành những giao diện này.

Hình sau đây chỉ ra một phần của cây phân cấp các lớp của gói event.



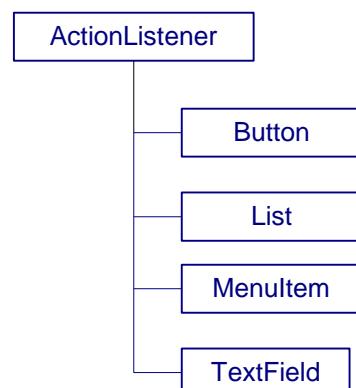
Hình 3.2: Cấu trúc cây phân cấp các lớp của gói event

Hình sau chỉ ra thứ tự phân cấp các giao diện của các event listener.



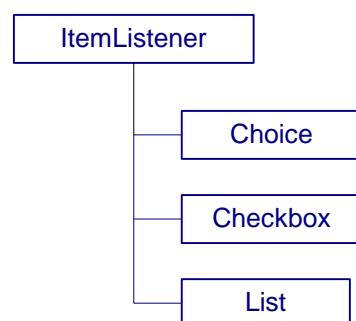
Hình 3.3: Cấu trúc cây phân cấp các giao diện của event listener

Hình sau là danh sách các listener được sử dụng cho các thành phần chỉ ra.
Action Listener



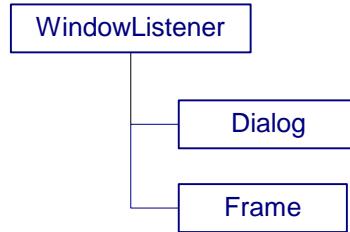
Hình 3.4: Các action listener

Item Listener



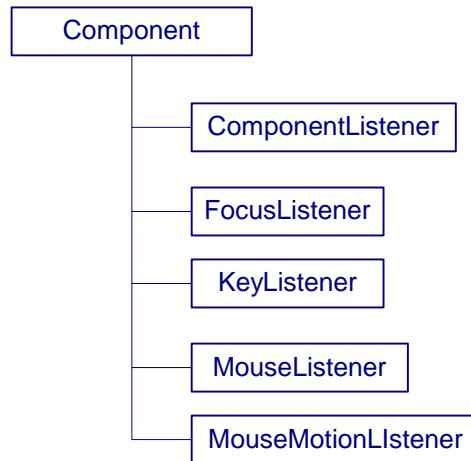
Hình 3.5: Các item listener

Window Listener



Hình 3.6: Các window listener

Các listener cho lớp Component



Hình 3.7: Các component listener

* Mẫu viết chương trình xử lý sự kiện Action Listener

```
import java.awt.*;
import java.awt.event.*;
class TÊN_LỚP extends Frame implements ActionListener
{
    //Khai báo các Component, cần bao nhiêu Component thì khai báo ở đây
    Button                                         Button(...)
    TextField           Tên_Component = new TextField(...)

    List                                         List(...)
    MenuItem                                         MenuItem(...)

    //Viết hàm tạo: Chọn các trình bày, gán sự kiện cho Component và thêm Component và Frame
    public TÊN_LỚP(String title)
    {
        super(title);
       setLayout(new FlowLayout());
        //Gán sự kiện cho component
        Tên_Component.addActionListener(this);
        // Thêm tất cả các Component vào Frame
        add(Tên_Component);
    }
    //Hàm actionPerformed: Lắng nghe sự kiện, mỗi sự kiện là một câu if
    public void actionPerformed(ActionEvent ae)
    {
        if (ae.getSource() == Tên_Component1)
```

```

    {
        //Lệnh xử lý sự kiện
    }
    if (ae.getSource() == Tên_Component2)
    {
        //Lệnh xử lý sự kiện
    }

    ...
if (ae.getSource() == ext) //Button Exit
{
    System.exit(0);
}
}

public static void main(String args[])
{
    TÊN_LỚP t = new TÊN_LỚP("Event handling");
    t.setSize(300, 200); t.show();
}
}

```

3.1.5. Menu

Lớp abstract class **MenuComponent** là lớp cơ sở cho tất cả các lớp thực hiện những vấn đề liên quan đến thực đơn (menu).

- Lớp **MenuBar** cài đặt thanh thực đơn và trong đó có thẻ chứa các thực đơn pull-down.
- Lớp **MenuItem** định nghĩa từng mục của thực đơn.
- Lớp **Menu** cài đặt các thực đơn pull-down để có thể đưa vào một thực đơn bất kỳ.
- Lớp **PopupMenu** biểu diễn cho thực đơn pop-up .
- Lớp **CheckboxMenuItem** chứa các mục được chọn để kiểm tra trong các mục thực đơn.

Việc tạo lập một thanh thực đơn cho một frame được thực hiện như sau:

1. Tạo ra một thanh thực đơn,

```
MenuBar thanhThDon = newMenuBar();
```

2. Tạo ra một thực đơn,

```
Menu thucDon = newMenu("Cac loai banh");
```

3. Tạo ra các mục trong thực đơn và đưa vào thực đơn,

```
MenuItem muc = newMenuItem("Banh day");
```

```
thucDon.add(muc); // Đưa mục vào thucDon
```

4. Đưa các thực đơn vào thanh thực đơn,

```
thanhThDon.add(thucDon); //Đưa thucDon vào thanhThDon
```

5. Tạo ra một frame và đưa thanh thực đơn vào frame đó.

```
Frame frame = newFrame("Cac mon an");
```

```
frame.add(thanhThDon); //Đưa thanhThDon vào frame
```

3.1.6. Tạo GUI với Swing

Các thành phần đồ họa trong awt được xây dựng từ các thành phần đồ họa của hệ điều hành tương ứng, do đó GUI xây dựng bằng awt thì có look-and-feel ở mỗi hệ điều hành là khác nhau.

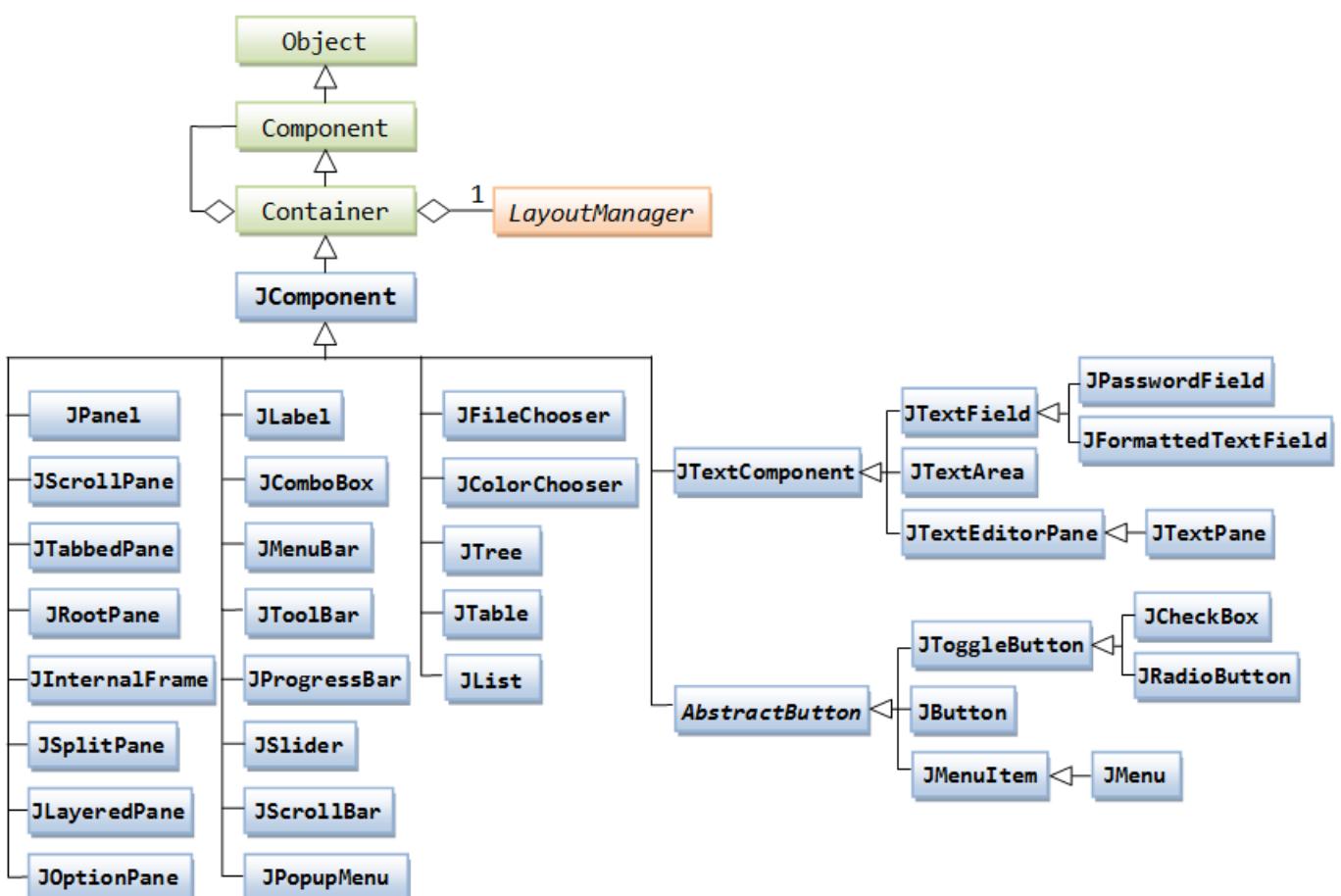
Các thành phần đồ họa trong swing thì được xây dựng từ chính java. Do đó look-and-feel của nó mềm dẻo hơn, có thể là đồng nhất trên tất cả các hệ điều hành hoặc khác nhau tùy theo ý muốn của bạn. Swing mạnh, đa năng, linh hoạt, hỗ trợ đầy đủ các component và đẹp hơn awt.

Có thể so sánh một cách hình tượng: awt là xe đạp, swing là xe máy. Cùng một chức năng là di chuyển, vậy lựa chọn như thế nào là tùy theo bạn!

* Khai báo sử dụng Swing: **import javax.swing.*;**

Sử dụng các container, component của swing tương tự awt, khi khai báo các đối tượng ta thêm tiền tố J vào trước, ví dụ JFrame, Jlabel, JButton, ...

* Cấu trúc lớp của swing:



Hình 3.8: Cấu trúc lớp của Java Swing

3.2. PHẦN BÀI TẬP

3.2.1. Bài tập mẫu

 **Bài 3.1. Minh họa cách tạo Frame**

Lời giải:

a. Hướng dẫn

Lớp được định nghĩa Framedemo là một lớp con của lớp Frame. Lớp FrameDemo này có một phương thức khởi tạo, trong phương thức khởi tạo này ta cho gọi phương thức **super()**. Nó sẽ gọi phương thức khởi tạo của lớp cha (trong trường hợp này là Frame). Mục đích của **super()** là gọi phương thức khởi tạo của lớp cha. Nó sẽ tạo một đối tượng của lớp con, lớp con này sẽ tạo Frame. Tuy nhiên, Frame vẫn không nhìn thấy được và không có kích thước. Để làm được điều này, ta sử dụng hai phương thức nằm trong phương thức **main**: **setSize()** và **setVisible()**.

b. Chương trình mẫu

```
import java.awt.*;
class FrameDemo extends Frame
{
    public FrameDemo(String title)
    {
        super(title);
    }
    public static void main(String args[])
    {
        FrameDemo f=new FrameDemo("I have been Frameed!!!");
        f.setSize(300,200);
        f.setVisible(true);
    }
}
```

💻 Kết quả chạy chương trình:



Để đóng cửa sổ Frame, chọn menu Run\Stop Build/Run (nút **X** trên Frame không đóng được).

❑ **Bài 3.2: Minh họa cách tạo Panel**

Lời giải:

a. Hướng dẫn:

Panel được sử dụng để nhóm một số các thành phần lại với nhau. Cách đơn giản nhất để tạo một panel là sử dụng phương thức khởi tạo của nó, hàm **Panel()**.

Panel không thể được nhìn thấy trực tiếp. Do đó, chúng ta cần thêm panel đến một frame. Vì vậy ta cần tạo một frame mới và thêm Panel mới được tạo này vào đó. Tuy nhiên, frame sẽ không nhìn thấy được, và không có kích thước. Chúng ta sử dụng hai



phương thức trong phương thức **main** – `setSize()` và `setVisible()` để thiết lập kích thước và hiển thị frame.

b. Chương trình mẫu:

```
import java.awt.*;
class Paneltest extends Panel
{
public static void main(String args[])
{
    Paneltest p=new Paneltest();
    Frame f=new Frame("Testing a Panel");
    f.add(p);
    f.setSize(300,200);
    f.setVisible(true);
}
public Paneltest() {}
}
```

 **Bài 3.3. Minh họa cách tạo Label**

Lời giải:

a. Hướng dẫn

Sử dụng constructor sau đây để tạo một label: `Label(String labelText)`;

Ví dụ:

```
label1=new Label("This is just a label"); // Tạo đối tượng Label
add(label1); //Thêm label1 vào Frame
```

b. Chương trình mẫu

```
import java.awt.*;
class Labeltest extends Frame
{
    Label label1=new Label("This is just a label");
    public Labeltest(String title)
    {
        super(title);
        add(label1);
    }
public static void main(String args[])
{
    Labeltest f=new Labeltest("Label");
    f.setSize(300,200);
    f.show();
}
}
```

 Kết quả chạy chương trình:



❑ Bài 3.4 . Minh họa cách tạo TextField

Lời giải:

a. Hướng dẫn

Một Textfield là một vùng chỉ chứa một dòng văn bản, trong đó văn bản có thể được hiển thị hay được nhập vào bởi người dùng. Trong Java, một trong những constructor sau có thể được sử dụng để tạo một Textfield:

- `TextField()`: Tạo một textfield mới.
- `TextField(int columns)`: Tạo một textfield mới với số cột được cho trước.
- `TextField(String s)`: Tạo một textfield mới với chuỗi văn bản được cho trước.
- `TextField(String s, int columns)`: Tạo một textfield mới với nội dung và số cột được cho trước.

Trong chương trình này, chúng ta sử dụng phương thức `setLayout()` để thay đổi cách trình bày của các thành phần trên vật chứa.

b. Chương trình mẫu

```
import java.awt.*;
class TextFieldtest extends Frame
{
    TextField tf1=new TextField(30);
    public TextFieldtest(String title)
    {
        super(title);
        setLayout(new FlowLayout());
        add(tf1);
    }
    public static void main(String args[])
    {
        TextFieldtest f=new TextFieldtest("TextField");
        f.setSize(300,200);
        f.show();
    }
}
```

❑ Kết quả chạy chương trình:



❑ Bài 3.5. Minh họa cách tạo TextArea

Lời giải:

a. Hướng dẫn

Một Textarea được sử dụng khi văn bản nhập vào có trên hai hay nhiều dòng. Textarea có một scrollbar. TextArea là một trường văn bản có thể được soạn thảo với nhiều dòng. Để tạo một Textarea, làm theo các bước sau:

- 1) Tạo một đối tượng.
 - 2) Chỉ ra số dòng, số cột đối tượng này cần có.
 - 3) Bố trí phần tử này trên màn hình.
- Sử dụng constructor sau để tạo TextArea:
- o TextArea(): Tạo một TextArea mới.
 - o TextArea(String text): Tạo một TextArea mới với nội dung được cho trước.

b. Chương trình mẫu

```
import java.awt.*;  
class TextAreatest extends Frame  
{    Label lbl=new Label("Details");  
    TextArea ta1=new TextArea();  
    public TextAreatest(String title)  
    {        super(title);  
        setLayout(new FlowLayout());  
        add(lbl);  
        add(ta1);  
    }  
    public static void main(String args[]){  
        TextAreatest t=new TextAreatest("TextArea");  
        t.setSize(300,200);  
        t.show();  
    }  
}
```

💻 Kết quả chạy chương trình:



❑ *Bài 3.6. Minh họa cách tạo Button*

Lời giải.

a. Hướng dẫn

Nút ấn hay còn gọi là nút lệnh là một phần không thể thiếu của bất kỳ GUI nào. Sử dụng button là cách dễ nhất để nhận các tác động của người dùng.

Để tạo một button, bạn làm theo các bước sau:

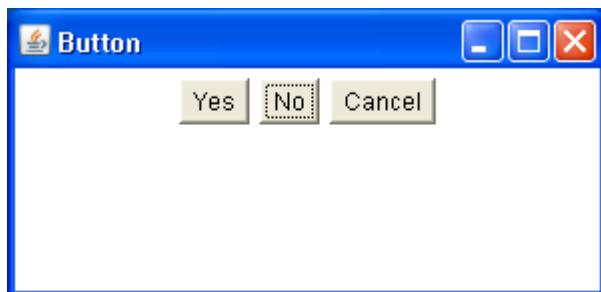
- 1) Tạo phần tử Button với một nhãn chỉ ra mục đích của Button.
- 2) Bố trí phần tử này trên màn hình.
- 3) Hiển thị phần tử trên màn hình.

Sử dụng constructor sau để tạo các button: Button(String text)
Sử dụng setLabel() và getLabel() để thiết lập và lấy giá trị nhãn của button.

b. Chương trình mẫu

```
import java.awt.*;
class Buttontest extends Frame
{
    Button b1 = new Button("Yes");
    Button b2 = new Button("No");
    Button b3 = new Button("Cancel");
    public Buttontest(String title)
    {
        super(title);
        setLayout(new FlowLayout());
        add(b1);
        add(b2);
        add(b3);
    }
    public static void main(String args[])
    {
        Buttontest t = new Buttontest("Button");
        t.setSize(300,200);
        t.show();
    }
}
```

💻 Kết quả chạy chương trình:



⇨ Bài 3.7. Minh họa cách tạo Checkbox và RadioButton

Lời giải:

a. Hướng dẫn

Làm theo các bước sau để tạo các checkbox hay radiobutton:

- 1) Tạo phần tử.
- 2) Xác định trạng thái khởi đầu của phần tử (chọn hay không chọn).
- 3) Bố trí các phần tử trên màn hình.
- 4) Hiển thị các phần tử trên màn hình.

Thành phần checkbox có thể sử dụng một lớp phụ được gọi là CheckboxGroup để tạo ra các radiobutton.

Sử dụng các constructor Checkbox(String text): Tạo một checkbox với nhãn đã cho.

Để tạo các radiobutton, đầu tiên chúng ta tạo đối tượng CheckboxGroup như sau:

```
CheckboxGroup cb=new CheckboxGroup();
```

Sau đó chúng ta tạo các đối tượng, như sau:

```
Checkbox b4=new Checkbox("small",cb,true);
Checkbox b5=new Checkbox("medium",cb,false);
```

```
Checkbox b6=new Checkbox("large",cb,false);
```

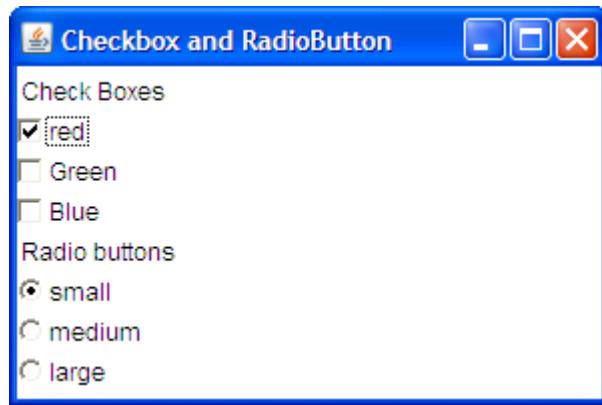
Đầu tiên chúng ta tạo một đối tượng Frame, đối tượng này hoạt động như một container sẽ chứa thành phần checkbox mà ta đã tạo. Sau đó ta tạo 6 checkbox, 02 checkbox được đánh dấu chọn. Để làm được điều này, ta đưa giá trị true như một tham số cho hàm constructor Checkbox, ngoài ra còn có một tham số String là nhãn của checkbox. Để hiển thị các điều khiển này theo dạng lưới, ta phải thiết lập cách trình bày về dạng GridLayout có 8 dòng và 1 cột. Cuối cùng, ta tạo một biểu hiện cho lớp Checkboxtest và thiết lập kích thước cho Frame. Để hiển thị nó, ta cho gọi phương thức show().

b. Chương trình mẫu

```
package checkboxtest;
import java.awt.*;
class Checkboxtest extends Frame
{
    Label l1=new Label("Check Boxes");
    Checkbox b1=new Checkbox("red",true);
    Checkbox b2=new Checkbox("Green",false);
    Checkbox b3=new Checkbox("Blue",false);
    Label l2=new Label("Radio buttons");
    CheckboxGroup cb=new CheckboxGroup();
    Checkbox b4=new Checkbox("small",cb,true);
    Checkbox b5=new Checkbox("medium",cb,false);
    Checkbox b6=new Checkbox("large",cb,false);

    public Checkboxtest(String title)
    {
        super(title);
        setLayout(new GridLayout(8,1));
        add(l1);
        add(b1);
        add(b2);
        add(b3);
        add(l2);
        add(b4);
        add(b5);
        add(b6);
    }
    public static void main(String args[])
    {
        Checkboxtest t=new Checkboxtest("Checkbox and RadioButton");
        t.setSize(300,200);
        t.show();
    }
}
```

💻 Kết quả chạy chương trình:



Bài 3.8. Minh họa cách tạo Choice List

Lời giải:

a. Hướng dẫn

Để tạo các danh sách chọn lựa, hãy làm theo các bước được cho sau đây:

- 1) Tạo danh sách các phần tử.
- 2) Thêm các mục (có kiểu là String) vào danh sách, mỗi lần chỉ thêm được một mục.
- 3) Bố trí danh sách trên màn hình.
- 4) Hiển thị danh sách trên màn hình.

Java hỗ trợ lớp Choice cho phép chúng ta tạo các danh sách chứa nhiều mục. Khi danh sách vừa được tạo ra, nó sẽ rỗng.

Choice colors=new Choice();

Mỗi thời điểm chỉ thêm được một item bằng cách sử dụng phương thức addItem như được chỉ ra bên dưới:

```
colors.addItem("Red");
colors.addItem("Green");
```

b. Chương trình mẫu

```
package choicetest;
import java.awt.*;
class Choicetest extends Frame
{
    Label l1=new Label("What is your favorite color");
    Choice colors=new Choice();
    public Choicetest(String title)
    {
        super(title);
        setLayout(new FlowLayout());
        add(l1);
        colors.addItem("White");
        colors.addItem("Red");
        colors.addItem("Orange");
        colors.addItem("Green");
        colors.addItem("Yellow");
        colors.addItem("Blue");
        colors.addItem("Black");
        add(colors);
    }
    public static void main(String args[])
    {
        Choicetest t=new Choicetest("Choice list");
        t.setSize(300,200);
        t.show();
    }
}
```

💻 Kết quả chạy chương trình:



➡ **Bài 3.9. Minh họa Flow Layout Manager**

Lời giải:

a. Hướng dẫn

'Flow Layout' là layout manager mặc định cho Applet và Panel. Các thành phần được xếp xép từ góc trái trên đến góc phải dưới của màn hình. Khi một số thành phần được tạo, chúng được xếp xép theo hàng, từ trái sang phải. Các constructor của FlowLayout:

```
FlowLayout mylayout = new FlowLayout(); // constructor
```

```
FlowLayout exLayout=new FlowLayout(FlowLayout.RIGHT);
```

```
setLayout(exLayout);
```

Các điều khiển có thể được canh về bên trái, bên phải hay ở giữa. Để canh các điều khiển về bên phải, bạn sử dụng cú pháp sau:

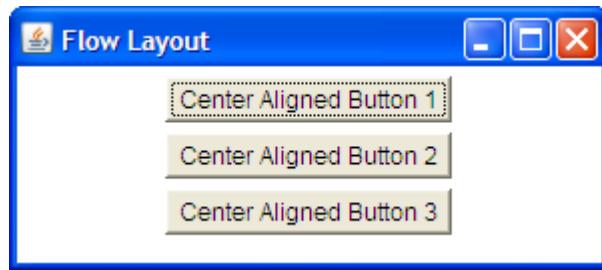
```
setLayout(new FlowLayout(FlowLayout.RIGHT));
```

b. Chương trình mẫu

```
package flowlayouttest;
import java.awt.*;
class FlowLayoutTest extends Frame
{
    Button b1=new Button("Center Aligned Button 1");
    Button b2=new Button("Center Aligned Button 2");
    Button b3=new Button("Center Aligned Button 3");

    public FlowLayoutTest(String title)
    {
        super(title);
        setLayout(new FlowLayout(FlowLayout.CENTER));
        add(b1);
        add(b2);
        add(b3);
    }
    public static void main(String args[])
    {
        FlowLayoutTest t=new FlowLayoutTest("Flow Layout");
        t.setSize(300,200);
        t.show();
    }
}
```

💻 Kết quả chạy chương trình:



❑ Bài 3.10. Minh họa GridLayout Manager

Lời giải:

a. Hướng dẫn

'GridLayout' trợ giúp việc chia container vào trong ô lưới. Các thành phần được đặt trong các ô giao của dòng và cột. Mỗi lưới nên chứa ít nhất một thành phần. Một lưới được sử dụng khi tất cả các thành phần có cùng kích thước.

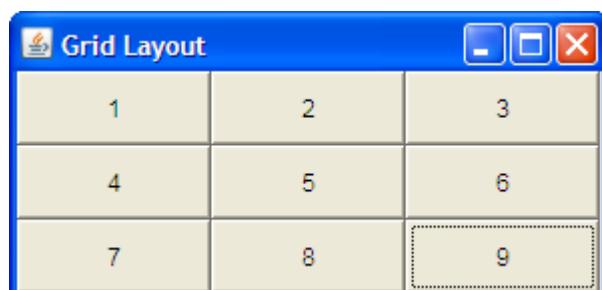
GridLayout được tạo như sau:

GridLayout g1=new GridLayout(4,3); // 4 là số dòng và 3 là số cột.

b. Chương trình mẫu

```
package gridlayouttest;
import java.awt.*;
class GridLayoutTest extends Frame
{Button btn[];
    String str[]{"1", "2", "3", "4", "5", "6", "7", "8", "9"};
    public GridLayoutTest (String title)
    {
        super(title);
        setLayout(new GridLayout(3,3));
        btn=new Button[str.length];
        for (int i=0; i<str.length;i++)
        {
            btn[i]=new Button(str[i]);
            add(btn[i]);
        }
    }
    public static void main(String args[])
    {
        GridLayoutTest t=new GridLayoutTest ("Grid Layout");
        t.setSize(300,200);
        t.show();
    }
}
```

❑ Kết quả chạy chương trình:



❑ Bài 3.11. Minh họa GridBagConstraints Manager

Lời giải:

a. Hướng dẫn

GridBag Layout giúp bố trí các component được linh hoạt hơn theo ý người dùng. Sử dụng phương thức

addComponent(Component c, int row, int col, int nrow, int ncol)

để thiết lập tọa độ đặt Component, chiều cao, độ rộng của nó.

b. Chương trình mẫu

```
package gridbaglayouttest;
import java.awt.*;
class GridBagLayoutTest extends Frame
{
    TextArea ta;
    TextField tf;
    Button b1,b2;
    CheckboxGroup cbg;
    Checkbox cb1,cb2,cb3,cb4;
    GridBagLayout gb;
    GridBagConstraints gbc;
    public GridBagLayoutTest(String title)
    {
        super(title);
        gb=new GridBagLayout();
        setLayout(gb);
        gbc=new GridBagConstraints();
        ta=new TextArea("Textarea",5,10);
        tf=new TextField("Enter your name");
        b1=new Button("TextArea");
        b2=new Button("TextField");

        cbg=new CheckboxGroup();
        cb1=new Checkbox("Bold", cbg,false);
        cb2=new Checkbox("Italic", cbg,false);
        cb3=new Checkbox("Plain", cbg,false);
        cb4=new Checkbox("Bold/Italic", cbg,true);

        gbc.fill=GridBagConstraints.BOTH;
        addComponent(ta,0,0,4,1);

        gbc.fill=GridBagConstraints.HORIZONTAL;
        addComponent(b1,0,1,1,1);

        gbc.fill=GridBagConstraints.HORIZONTAL;
        addComponent(b2,0,2,1,1);

        gbc.fill=GridBagConstraints.HORIZONTAL;
        addComponent(cb1,2,1,1,1);

        gbc.fill=GridBagConstraints.HORIZONTAL;
        addComponent(cb2,2,2,1,1);

        gbc.fill=GridBagConstraints.HORIZONTAL;
        addComponent(cb3,3,1,1,1);

        gbc.fill=GridBagConstraints.HORIZONTAL;
        addComponent(cb4,3,2,1,1);
```

```

        gbc.fill=GridBagConstraints.HORIZONTAL;
        addComponent(tf,4,0,1,3);
    }

public void addComponent(Component c, int row, int col, int nrow, int ncol)
{
    gbc.gridx=col;
    gbc.gridy=row;
    gbc.gridwidth=ncol;
    gbc.gridheight=nrow;
    gb.setConstraints(c,gbc);
    add(c);
}
public static void main(String args[])
{
    GridBagLayoutTest t=new GridBagLayoutTest("GridBag Layout");
    t.setSize(300,200); t.show();
}
}

```

💻 Kết quả chạy chương trình:



Giải thích đoạn mã trên:

gbc.fill=GridBagConstraints.BOTH;

Thành viên fill của lớp GridBagConstraints chỉ ra thành phần có thể được mở rộng theo hướng nằm ngang và thẳng đứng. Cú pháp sau mô tả thành phần chỉ được mở rộng theo hướng nằm ngang:

gbc.fill=GridBagConstraints.HORIZONTAL;

Cú pháp sau sẽ thêm vào thành phần TextArea với số dòng và số cột cần chiếm:

addComponent(ta,0,2,4,1);

0 – Khởi đầu từ dòng thứ 0

2 – Khởi đầu từ dòng thứ 2

4 – ta chiếm giữ 4 dòng

1 – ta chiếm 1 cột

Sử dụng cú pháp sau để bố trí các thành phần vào trong dòng và cột nào đó:

gbc.gridx=col;

gbc.gridy=row;

Ở đây (gridx,gridy) là cột và dòng nơi mà thành phần có thể được đặt vào.

Sử dụng cú pháp sau để chỉ ra số lượng các cột và dòng mà các thành phần có thể chiếm giữ:

gbc.gridwidth=ncol;

gbc.gridheight=nrow;

Ở đây, gridwidth xác định số lượng các cột mà một thành phần chiếm giữ và gridheight xác định số lượng các dòng mà một thành phần chiếm giữ.

Khi một container bị thay đổi kích thước và khi khoảng trống phụ tồn tại, các thành phần có chiều rộng lớn hơn sẽ chiếm giữ nhiều khoảng trống hơn là các thành phần có giá trị về chiều rộng nhỏ hơn.

Bài 3.12. Minh họa Xử lý các sự kiện

Xây dựng ứng dụng đơn giản như sau:



Khi nhập giá trị vào textField1 và nhấn button Double is (nhân đôi) thì kết quả hiện lên textField2. Nhấn nút exit thì thoát

Lời giải:

a. Hướng dẫn

Bước 1: Xây dựng giao diện như trên, tên của button Double is là btnResult, tên của button exit là ext

Bước 2: Viết code xử lý sự kiện nhấn button Double is và exit

Viết hàm:

```
public void actionPerformed(ActionEvent ae)
{
    if (ae.getSource() == btnResult)
    {
        int num = Integer.parseInt(tf1.getText()) * 2;
        tf2.setText(String.valueOf(num));
    }
    if (ae.getSource() == ext)
    {
        System.exit(0);
    }
}
```

Gán cho các button phương thức addActionListener(this); cụ thể

```
btnResult.addActionListener(this);
ext.addActionListener(this);
```

b. Chương trình mẫu

```
package evttest;
import java.awt.*;
import java.awt.event.*;
class Evttest extends Frame implements ActionListener
{
    Label lab = new Label("Enter a number");
    TextField tf1 = new TextField(5);
    TextField tf2 = new TextField(5);
    Button btnResult = new Button("Double is");
    Button ext = new Button("exit");
    public Evttest(String title)
    {
        super(title);
        setLayout(new FlowLayout());
```

```

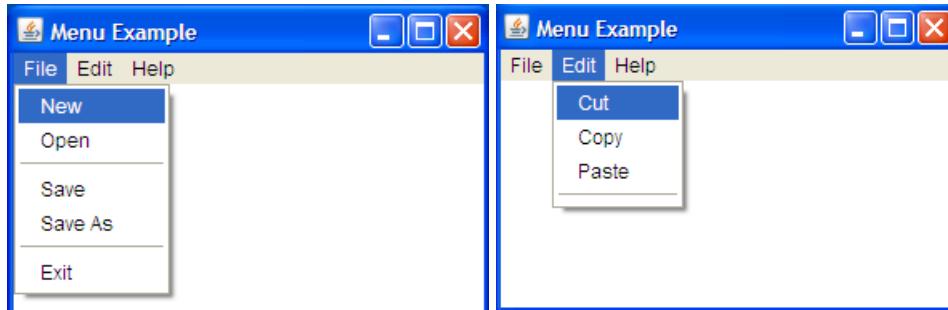
        btnResult.addActionListener(this);
        ext.addActionListener(this);
        add(lab);
        add(tf1);
        add(btnResult);
        add(tf2);           add(ext);
    }
    public void actionPerformed(ActionEvent ae)
    {
        if (ae.getSource()==btnResult)
        {   int num=Integer.parseInt(tf1.getText())*2;
            tf2.setText(String.valueOf(num));
        }
        if (ae.getSource()==ext)
        {   System.exit(0);
        }
    }
    public static void main(String args[])
    {
        Evttest t=new Evttest("Event handling");
        t.setSize(300,200);
        t.show();
    }
}

```

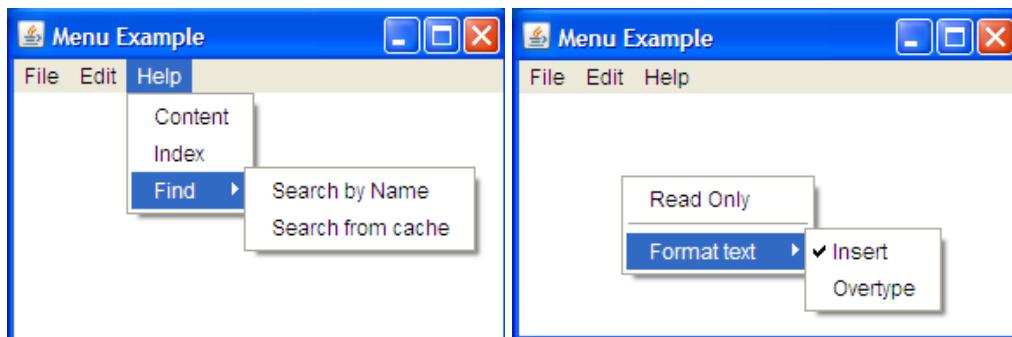
Bài 3.13. Minh họa xây dựng menu

Xây dựng hệ thống menu chính (menu – pull down), menu ngữ cảnh (pop-up menu).

Khi thực thi chương trình trên, một màn hình với các trình đơn File, Edit và Help được hiển thị. Khi click vào mục File, Edit, sẽ thấy xuất hiện như sau:



Một menu có thể chứa các menu con. Khi click vào trình đơn Help, 3 mục con có tên là Content, Index và Find sẽ xuất hiện. Trong trình đơn Find, có 2 mục con là Search by name và Search from Cache. Một khung pop-up menu sẽ hiện ra nếu nhấn chuột phải trên màn hình:



Lời giải:

a. Hướng dẫn

Sử dụng các lớp Menubar, Menu, MenuItem PopupMenu và CheckboxMenuItem để tạo menu.

b. Chương trình mẫu

```
package myframe;
import java.awt.*;
import java.awt.event.*;
class MyFrame extends Frame implements ActionListener,
MouseListener
{ MenuItem exitItem;
PopupMenu optionsMenu;
Frame frame;
public MyFrame()
{
setTitle("Menu Example");
setSize(300,200);

MenuBar mbar=newMenuBar();
setMenuBar(mbar);

Menu fileMenu=new Menu("File");
mbar.add(fileMenu);
fileMenu.addActionListener(this);
MenuItem newItem=new MenuItem("New");
fileMenu.add(newItem);
MenuItem openItem=new MenuItem("Open");
fileMenu.add(openItem);
fileMenu.addSeparator();
MenuItem saveItem=new MenuItem("Save");
fileMenu.add(saveItem);
MenuItem saveAsItem=new MenuItem("Save As");
fileMenu.add(saveAsItem);
fileMenu.addSeparator();
exitItem=new MenuItem("Exit");
fileMenu.add(exitItem);
saveAsItem.addActionListener(this);

Menu editMenu=new Menu("Edit");
mbar.add(editMenu);
editMenu.addActionListener(this);
MenuItem cutItem=new MenuItem("Cut");
editMenu.add(cutItem);
MenuItem copyItem=new MenuItem("Copy");
editMenu.add(copyItem);
MenuItem pasteItem=new MenuItem("Paste");
editMenu.add(pasteItem);
editMenu.addSeparator();

Menu helpMenu=new Menu("Help");
mbar.add(helpMenu);
helpMenu.addActionListener(this);
MenuItem contentItem=new MenuItem("Content");
helpMenu.add(contentItem);
MenuItem indexItem=new MenuItem("Index");
```

```

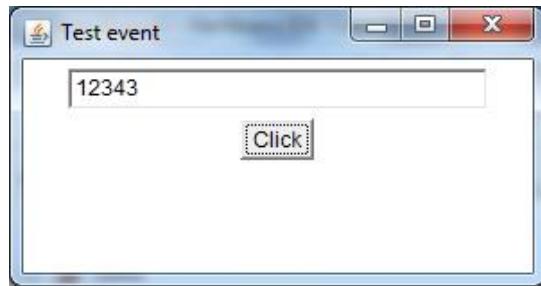
helpMenu.add(indexItem);
Menu findMenu=new Menu("Find");
helpMenu.add(findMenu);
addMouseListener(this);
MenuItem nameItem=new MenuItem("Search by Name");
findMenu.add(nameItem);
MenuItem cacheItem=new MenuItem("Search from cache");
findMenu.add(cacheItem);
optionsMenu=new PopupMenu("Options");
editMenu.add(optionsMenu);
optionsMenu.addActionListener(this);
MenuItem readItem=new MenuItem("Read Only");
optionsMenu.add(readItem);
optionsMenu.addSeparator();
Menu formatMenu=new Menu("Format text");
optionsMenu.add(formatMenu);
this.add(optionsMenu);
formatMenu.addActionListener(this);
CheckboxMenuItem insertItem=new
CheckboxMenuItem("Insert",true);
formatMenu.add(insertItem);
CheckboxMenuItem overtypeItem=new
CheckboxMenuItem("Overtype",false);
formatMenu.add(overtypeItem);
}

public void actionPerformed(ActionEvent ae)
{
    if (ae.getActionCommand().equals("Exit"))
    {
        System.exit(0);
    }
}
public void mouseEntered(MouseEvent m) {}
public void mouseExited(MouseEvent m) {}
public void mouseClicked(MouseEvent m)
{
    optionsMenu.show(this,m.getX(),m.getY());
}
public void mouseReleased(MouseEvent m) {}
public void mousePressed(MouseEvent m) {}
public static void main(String[] args)
{
    MyFrame frame=new MyFrame();
    frame.show();
}
}

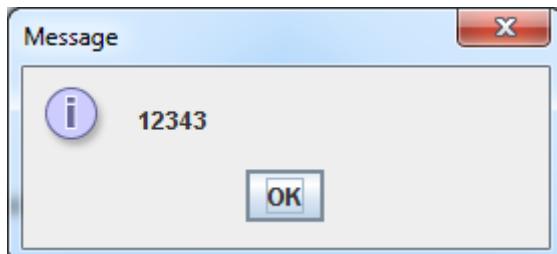
```

Bài 3.14. Java Swing. Minh họa sử dụng MessageBox

Xây dựng ứng dụng như sau:



Khi nhập giá trị số (≥ 5 số) và nhấn chọn Click thì xuất hiện thông báo



Nếu nhập số ít hơn 5 chữ số hoặc nhập các ký tự không phải là số thì xuất hiện thông báo yêu cầu nhập lại

Lời giải:

a. Hướng dẫn

Sử dụng phương thức sau trong java swing để tạo MessageBox:

```
JOptionPane.showMessageDialog(null, "Thông báo viết ở đây!");
```

b. Chương trình mẫu

```
package demoevent1;
import java.awt.Button;
import java.awt.FlowLayout;
import java.awt.Frame;
import java.awt.TextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import javax.swing.JOptionPane;

public class DemoEvent1 extends Frame implements ActionListener
{
    TextField txtName = new TextField(30);
    Button cmdTest = new Button("Click");
    public DemoEvent1()
    {
        //thiet lap layout
        setLayout(new FlowLayout());
        this.add(txtName); //dat cac component vao form
        this.add(cmdTest);

        //dang ky mot doi tuong listener dung de lang nghe va xu ly su kien
        //ActionEvent khi ta click vao nut

        cmdTest.addActionListener(this);

        //doan code nay dung de tat chuong trinh khi click vao nut X tren form
    }
}
```

```

        this.addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent we)
            {
                System.exit(0);
            }
        });
        this.setTitle("Test event");
        this.setSize(300, 300);
        this.setVisible(true);
    }

    public void actionPerformed(ActionEvent ae)
    {
        //kiem tra xem co phai user click vao nut cmdTest hay khong ?
        if(ae.getSource() == cmdTest)
        {
            //lay tat ca text trong TextField
            String name = txtName.getText();
            //kiem tra xem text trong TextField co phai trong hay khong
            //bang cach so sanh voi chuoi rong
            if(name.equals ""))
            {
                //Hien thi hop thoai thong bao
                JOptionPane.showMessageDialog(null, "Ban can nhap ten!");
                // tra con tro ban phim ve TextField
                txtName.requestFocus();
                //thoat khoi ham nay, khong xu ly gi nua neu co loi
                return;
            }
            if(name.length() < 5)
            { //Hien thi hop thoai thong bao
                JOptionPane.showMessageDialog(null, "Ban can nhap toi thieu 5 ky tu !");
                // tra con tro ban phim ve TextField
                txtName.requestFocus();
                //thoat khoi ham nay, khong xu ly gi nua neu co loi
                return;
            }
            //kiem tra bat buoc phai nhap chu so
            try
            {
                int myNumber = Integer.parseInt(name);
            }
            catch(NumberFormatException e)
            {
                //Hien thi hop thoai thong bao
                JOptionPane.showMessageDialog(null, "Ban can nhap chu so !");
                // tra con tro ban phim ve TextField
                txtName.requestFocus();
                //thoat khoi ham nay, khong xu ly gi nua neu co loi
                return;
            }
            //hien toan bo text trong TextField
            JOptionPane.showMessageDialog(null, name);
        }
    }

    public static void main(String[] args)
    {
        DemoEvent1 obj = new DemoEvent1();
        obj.setSize(200,200);
        obj.show();
    }
}

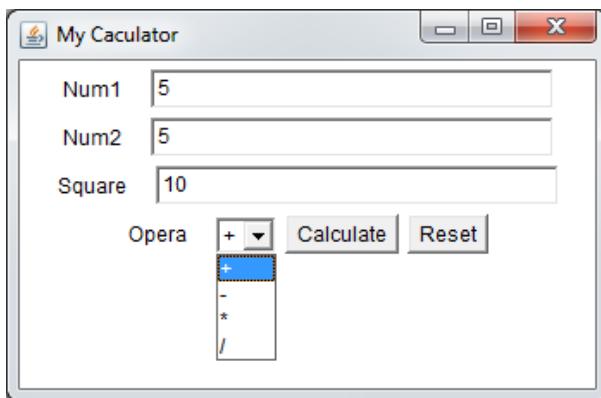
```

```

    }
}

```

-  **Bài 3.15.** Xây dựng ứng dụng cộng, trừ, nhân, chia các số như sau



Khi nhập 2 số, chọn phép toán trong Choice nhấn button Calculate sẽ thực hiện tính toán và điều kết quả vào text Square.

Nhấn nút Reset để xóa sạch các TextField

Lời giải:

a. Hướng dẫn

Khai báo đối tượng cctinh thuộc lớp Choice. Sử dụng phương thức getSelectedIndex() của đối tượng cctinh để lấy chỉ số của các phép toán. Thứ tự các phép toán là 0: phép cộng, 1: phép trừ, 2: phép nhân, 3: phép chia

b. Chương trình mẫu

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.JOptionPane;
public class May_Tinh extends Frame implements ActionListener
{
    Label lbname1 = new Label("Num1");
    TextField tfnum1 = new TextField(30);

    Label lbname2 = new Label("Num2");
    TextField tfnum2 = new TextField(30);
    Label lbsqr = new Label("Square");
    TextField tfsqr = new TextField(30);

    Button btcal = new Button("Calculate");
    Button btreset = new Button("Reset");
    Label lbope = new Label("Opera");
    Choice cctinh = new Choice();

    public May_Tinh(String title)
    {
        super(title);
        setLayout(new FlowLayout());
        add(lbname1);           add(tfnum1);
        add(lbname2);           add(tfnum2);
        add(lbsqr);             add(tfsqr);
        add(lbope);
        cctinh.add("+");
        cctinh.add("-");
        cctinh.add("*");
        cctinh.add("/");
        lbope.setAlignment(Label.CENTER);
        lbope.addActionListener(this);
        btcal.addActionListener(this);
        btreset.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource() == lbope)
        {
            int index = cctinh.getSelectedIndex();
            if(index == 0)
                tfsqr.setText((tfnum1.getText() + tfnum2.getText()));
            else if(index == 1)
                tfsqr.setText((tfnum1.getText() - tfnum2.getText()));
            else if(index == 2)
                tfsqr.setText((tfnum1.getText() * tfnum2.getText()));
            else if(index == 3)
                tfsqr.setText((tfnum1.getText() / tfnum2.getText()));
        }
        else if(e.getSource() == btcal)
        {
            tfsqr.setText("Error");
        }
        else if(e.getSource() == btreset)
        {
            tfnum1.setText("");
            tfnum2.setText("");
            tfsqr.setText("");
        }
    }
}

```

```

        add(lbope);
        cctinh.addItem("+");
        cctinh.addItem("-");
        cctinh.addItem("*");
        cctinh.addItem("/");
        add(cctinh);

        add(btcal);           add(btreset);

        btcal.addActionListener(this);
        btreset.addActionListener(this);

        tfnum1.addKeyListener(new KeyAdapter()
        {
            public void keyPressed(KeyEvent ky)
            {
                if(ky.getKeyCode() == KeyEvent.VK_ENTER)
                {
                    tfnum2.requestFocus();
                }
            }
        });//het bat loi xuong dong` 

        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent we)
            {
                System.exit(0);
            }
        });
    }

    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==btcal)
        {
            String name = tfnum1.getText();
            String name2 = tfnum2.getText();
            if(name.equals(""))
            {
                JOptionPane.showMessageDialog(null,"Num1 ko duoc bo trong");
                tfnum1.requestFocus();
                tfnum1.setText("");
                return ;
            }
            try
            {
                long num1 = Long.parseLong(name);
            }
            catch(NumberFormatException v)
            {
                JOptionPane.showMessageDialog(null,"Ban Can Phai Nhap so");
                tfnum1.requestFocus();
                tfnum1.setText("");
                return ;
            }
        if(name2.equals(""))
        {
            JOptionPane.showMessageDialog(null,"Num2 Ban ko Duoc de trong");
        }
    }
}

```

```

        tfnum2.requestFocus();
        return ;
    }
    try
    {
        long num2 = Long.parseLong(name2);
    }
    catch (NumberFormatException v)
    {
        JOptionPane.showMessageDialog(null,"Ban can Nhap Vao mot so");
        tfnum2.requestFocus();
        tfnum2.setText("");
        return ;
    }
    //Khai Bao Bien
    long num1 = Long.parseLong(tfnum1.getText());
    long num2 = Long.parseLong(tfnum2.getText());
    long sqrt;
    //Lay' cac' thuoc tinh trong Choice
    if(cctinh.getSelectedIndex()==0)
    {
        sqrt = num1+num2;
        tfsqr.setText(String.valueOf(sqrt));
    }
    if(cctinh.getSelectedIndex()==1)
    {
        sqrt = num1-num2;
        tfsqr.setText(String.valueOf(sqrt));
    }
    if(cctinh.getSelectedIndex()==2)
    {
        sqrt = num1*num2;
        tfsqr.setText(String.valueOf(sqrt));
    }
    if(cctinh.getSelectedIndex()==3)
    {
        sqrt= num1/num2;
        tfsqr.setText(String.valueOf(sqrt));
    }
}

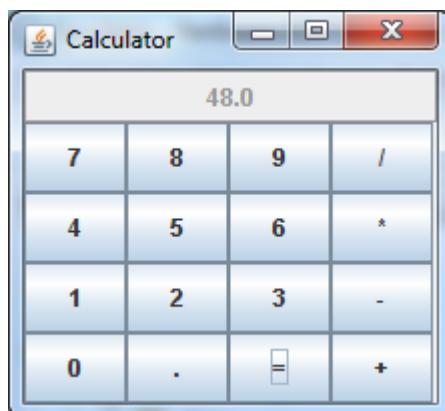
if(e.getSource()==btreset)
{
    tfnum1.setText("");
}
if(e.getSource()==btreset)
{
    tfnum2.setText("");
}
if(e.getSource()==btreset)
{
    tfsqr.setText("");
}
}

public static void main (String[] args)
{
    May_Tinh objmt = new May_Tinh("My Caculator");
    objmt.setSize(350,350);
    objmt.setVisible(true);
}
}

```

Bài 3.16 . Xây dựng Máy tính

Xây dựng máy tính với 4 phép toán +, -, *, / như sau:



Lời giải:

a. Hướng dẫn

Khai báo sử dụng JFrame, JFanel, cách trình bày BorderLayout và GridLayout...

b. Chương trình mẫu

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class Calculator
{
    public static void main(String[] args)
    {
        EventQueue.invokeLater(new Runnable()
        {
            public void run()
            {   CalculatorFrame frame = new CalculatorFrame();
                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                frame.setVisible(true);
            }
        });
    }

    class CalculatorFrame extends JFrame
    {
        public CalculatorFrame()
        {   setTitle("Calculator");
            CalculatorPanel panel = new CalculatorPanel();
            add(panel);
            pack();
        }
    }

    class CalculatorPanel extends JPanel
    {
        public CalculatorPanel()
        {   setLayout(new BorderLayout());
            result = 0;
            lastCommand = "=";
            start = true;

            // add the display
        }
    }
}
```

```

display = new JButton("0");
display.setEnabled(false);
add(display, BorderLayout.NORTH);

ActionListener insert = new InsertAction();
ActionListener command = new CommandAction();

// kích thước 4 x 4
panel = new JPanel();
panel.setLayout(new GridLayout(4, 4));

 addButton("7", insert);
 addButton("8", insert);
 addButton("9", insert);
 addButton("/", command);

 addButton("4", insert);
 addButton("5", insert);
 addButton("6", insert);
 addButton("*", command);

 addButton("1", insert);
 addButton("2", insert);
 addButton("3", insert);
 addButton("-", command);

 addButton("0", insert);
 addButton(".", insert);
 addButton("=", command);
 addButton("+", command);

 add(panel, BorderLayout.CENTER);
}

private void addButton(String label, ActionListener listener)
{
    JButton button = new JButton(label);
    button.addActionListener(listener);
    panel.add(button);
}

private class InsertAction implements ActionListener
{
    public void actionPerformed(ActionEvent event)
    {
        String input = event.getActionCommand();
        if (start)
        {   display.setText("");
            start = false;
        }
        display.setText(display.getText() + input);
    }
}

private class CommandAction implements ActionListener
{
    public void actionPerformed(ActionEvent event)
    {
        String command = event.getActionCommand();

```

```

if (start)
{
    if (command.equals("-"))
    {   display.setText(command);
        start = false;
    }
    else lastCommand = command;
}
else
{
    calculate(Double.parseDouble(display.getText()));
    lastCommand = command;
    start = true;
}
}

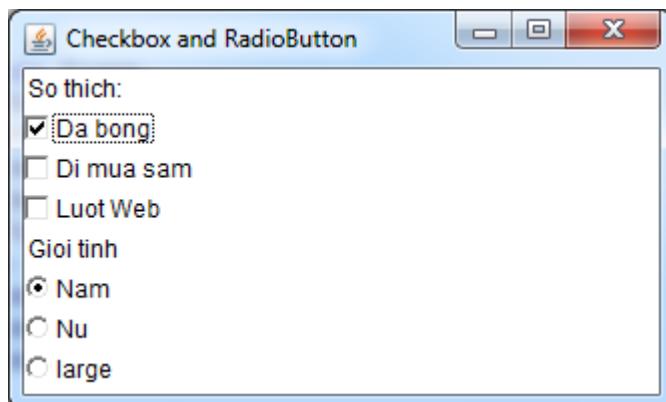
public void calculate(double x)
{
    if (lastCommand.equals("+")) result += x;
    else if (lastCommand.equals("-")) result -= x;
    else if (lastCommand.equals("*")) result *= x;
    else if (lastCommand.equals("/")) result /= x;
    else if (lastCommand.equals("=")) result = x;
    display.setText("") + result);
}
private JButton display;
private JPanel panel;
private double result;
private String lastCommand;
private boolean start;
}

```

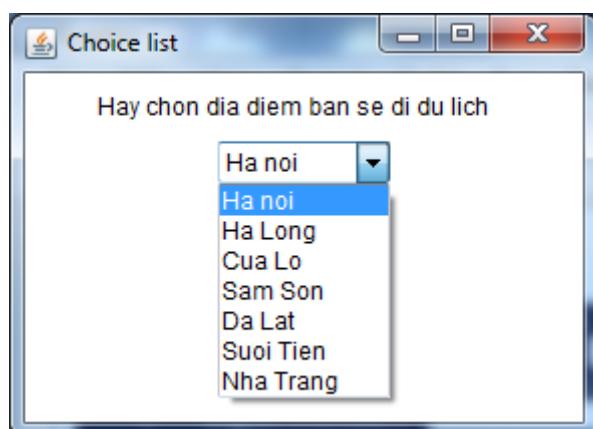
3.2. Bài tập tự giải

Bài 3.17. Viết chương trình tạo Frame có tiêu đề là HocSinh, trên Frame có một Label và một TextField: Label có phần Text là “Ho va Ten”, TextField có phần Text để trống để nhập Họ và tên.

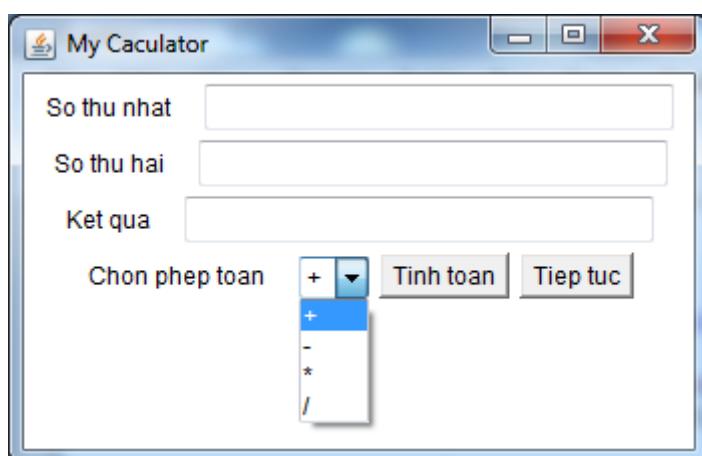
Bài 3.18. Viết chương trình tạo Checkbox và Radio button như sau:



Bài 3.19. Viết chương trình tạo Choice List như sau:



Bài 3.20. Viết chương trình tạo giao diện cho máy tính đơn giản như sau:

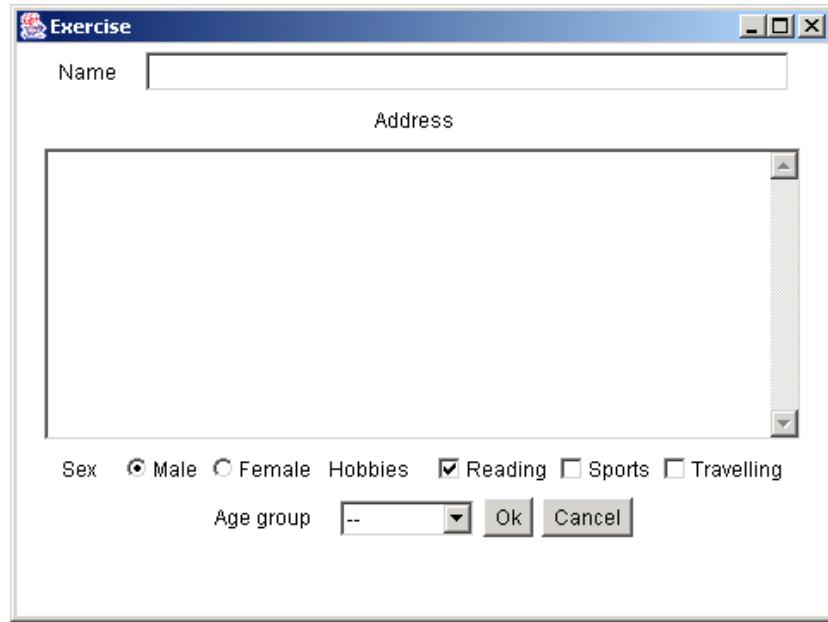


Sử dụng 3 cách trình bày:

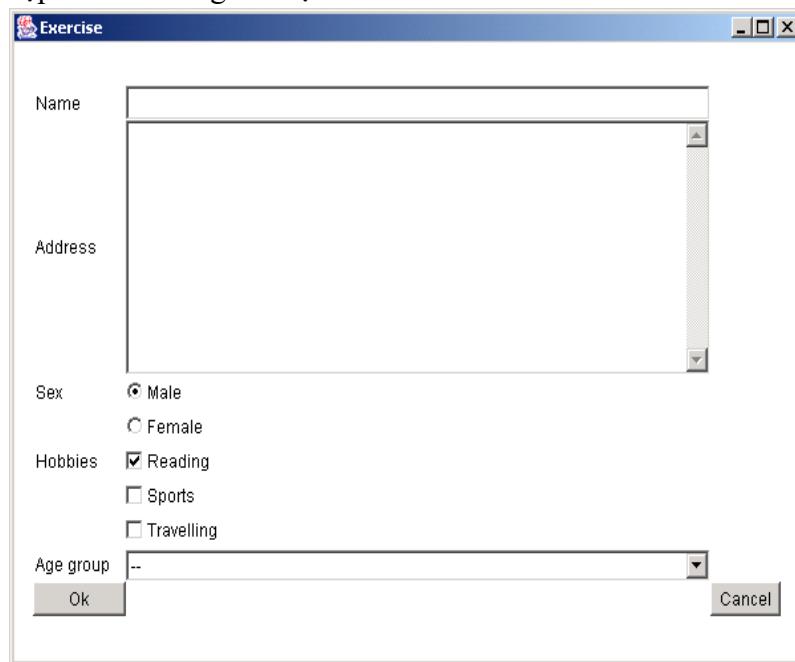
- a. FlowLayout
- b. GridLayout

c. GridBagLayout

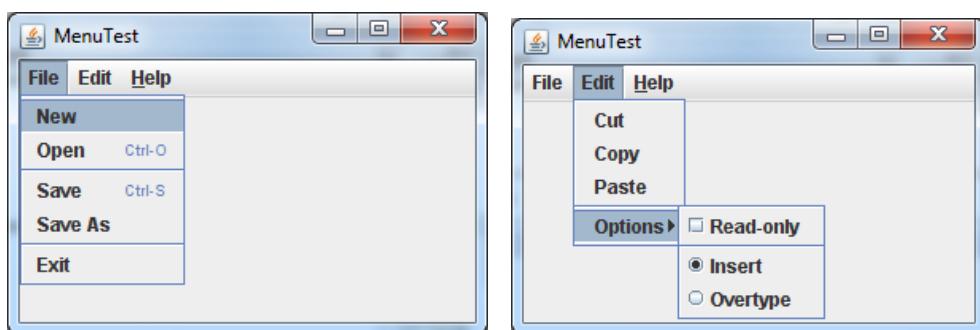
Bài 3.21. Viết chương trình Java để nhập thông tin chi tiết về người sử dụng như hình sau:

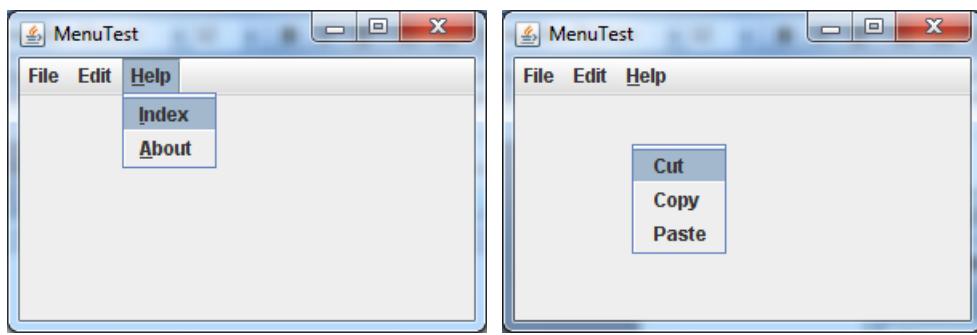


Bài 3.22. Sửa bài tập 3.21. để có giao diện như sau



Bài 3.23. Xây dựng menu như sau:





CHƯƠNG 4: LUỒNG VÀO RA VÀ TẬP TIN

4.1. PHẦN LÝ THUYẾT

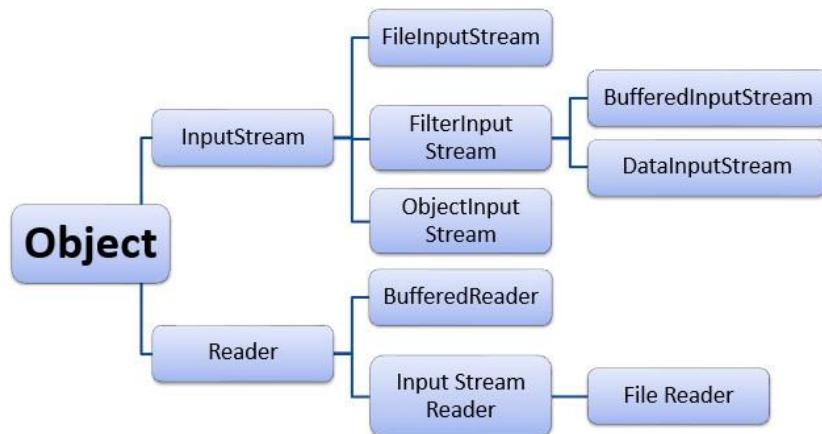
Đọc và ghi file trong java là các hoạt động nhập/xuất dữ liệu (nhập dữ liệu từ bàn phím, đọc dữ liệu từ file, ghi dữ liệu lên màn hình, ghi ra file, ghi ra đĩa, ghi ra máy in...) đều được gọi là luồng (stream).

Có 2 loại luồng dữ liệu: Input Streams (Lấy dữ liệu từ các nguồn: Files, Buffers và Sockets .v.v) và Output Streams (Ghi dữ liệu vào Files, Buffers và Sockets.v.v.)

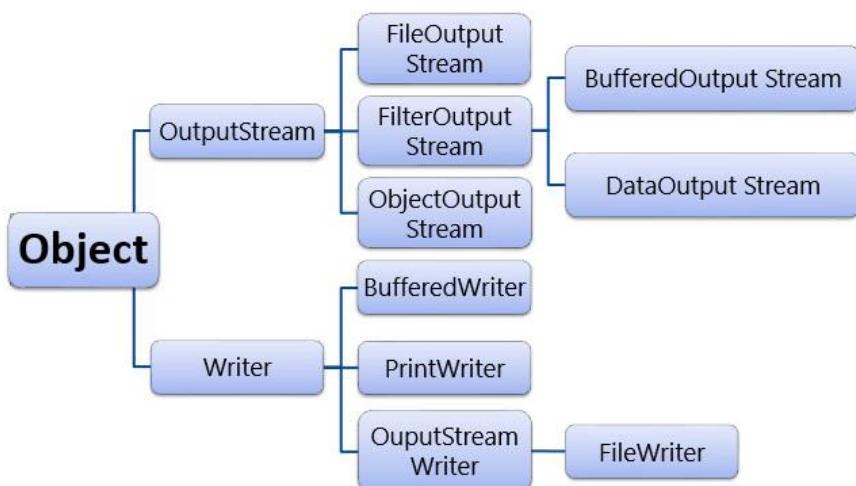
Có 2 kiểu luồng trong Java:

Luồng byte: Hỗ trợ nhập xuất dữ liệu trên byte, thường được dùng khi đọc ghi dữ liệu nhị phân. Hai class abstract là InputStream và OutputStream

Luồng character: Luồng ký tự character được thiết kế hỗ trợ việc nhập xuất dữ liệu kiểu ký tự (unicode). Hai class abstract là Reader và Writer



Hình 4.1: Kiến trúc Input Stream



Hình 4.2: Kiến trúc Output Stream (Luồng xuất dữ liệu)

Đọc và ghi file trong java – Các thao tác xử lý dữ liệu

Bước 1: Tạo đối tượng luồng và liên kết với nguồn dữ liệu.

Bước 2: Thao tác dữ liệu (đọc hoặc ghi hoặc cả hai).

Bước 3: Đóng luồng.

4.1.1. Byte Stream

Byte Stream trong Java được sử dụng để thực hiện input và output của các byte (8 bit). Mặc dù có nhiều lớp liên quan tới byte stream nhưng các lớp thường được sử dụng nhất là: FileInputStream và FileOutputStream.

4.1.2. Character Stream

Character Stream được sử dụng để thực hiện input và output cho Unicode 16 bit. Mặc dù có nhiều lớp liên quan tới character stream nhưng các lớp thường dùng nhất là FileReader và FileWriter, v.v. FileReader sử dụng FileInputStream và FileWriter sử dụng FileOutputStream, nhưng có một điểm khác biệt lớn ở đây là FileReader đọc hai byte cùng một thời điểm và FileWriter ghi 2 byte cùng một lúc.

4.1.3. Standard Stream

Java cung cấp 3 Standard Stream sau:

a. Chuẩn đầu vào (Standard Input)

Được sử dụng để truyền dữ liệu tới chương trình của người dùng, thường một bàn phím được sử dụng như là chuẩn đầu vào và được biểu diễn như System.in.

b. Chuẩn đầu ra (Standard Output)

Được sử dụng để hiển thị kết quả đầu ra từ chương trình của người dùng và thường thì một màn hình máy tính được sử dụng như là đầu ra chuẩn và được biểu diễn như là System.out

c. Lỗi chuẩn (Standard Error)

Được sử dụng để hiển thị các lỗi trong chương trình của người dùng và thường thì một màn hình máy tính được sử dụng như là lỗi chuẩn và được biểu diễn như là System.err.

4.1.4. Đọc và ghi file

a. FileInputStream

Luồng này được sử dụng để đọc dữ liệu từ các file. Các đối tượng có thể được tạo bởi sử dụng từ khóa new và có một số kiểu constructor có sẵn.

Constructor sau đây nhận tên file như là một chuỗi để tạo một đối tượng Input Stream để

đọc file:

```
InputStream f = new FileInputStream("C:/java/hello");
```

Constructor sau nhận một đối tượng File để tạo một đối tượng Input Stream để đọc file. Đầu tiên chúng ta tạo một đối tượng file bởi sử dụng phương thức File() như sau:

```
File f = new File("C:/java/hello");
```

```
InputStream f = new FileInputStream(f);
```

Khi chúng ta có đối tượng InputStream, thì khi đó có một danh sách các phương thức có thể được sử dụng để đọc stream hoặc để thực hiện hoạt động nào khác trên stream này.

Bảng 4.1: Các phương thức khác nhau và mô tả tương ứng của lớp **FileInputStream**

Phương thức	Mô tả
close()	Phương thức này đóng output stream. Giải phóng bất kỳ nguồn hệ thống nào liên kết với file. Ném một ngoại lệ IOException.
finalize()	Phương thức này xóa sự kết nối tới File đó. Bảo đảm rằng phương thức close của output stream này được gọi khi không có tham chiếu nào nữa tới stream này. Ném một IOException.
read(int r)	Phương thức này đọc byte dữ liệu đã xác định từ InputStream. Trả về giá trị kiểu nguyên. Trả về byte dữ liệu tiếp theo và -1 sẽ được trả về nếu kết thúc file.
read(byte[] r)	Phương thức này đọc r byte từ input stream vào trong một mảng. Trả về tổng số byte đã đọc. Nếu kết thúc file, -1 được trả về.
available()	Cung cấp số byte mà được đọc từ input stream này và trả về một giá trị nguyên

b. FileOutputStream

Luồng được sử dụng để tạo một file và ghi dữ liệu vào trong nó. Luồng này sẽ tạo một file, nếu nó đã không tồn tại, trước khi mở nó để ghi output.

Dưới đây là hai constructor mà có thể được sử dụng để tạo một đối tượng FileOutputStream trong Java.

Constructor sau nhận một tên file như là một chuỗi để tạo một đối tượng input stream để ghi file.

```
OutputStream f = new FileOutputStream("C:/java/hello")
```

Constructor sau nhận một đối tượng file để tạo một đối tượng output stream để ghi file.

```
File f = new File("C:/java/hello")
```

```
OutputStream f = new FileOutputStream(f)
```

Khi có đối tượng OutputStream này, thì sau đây có các phương thức có thể được

sử dụng để ghi stream hoặc để thực hiện các hoạt động khác trên stream này:

Bảng 4.2: Các phương thức khác nhau và mô tả tương ứng của lớp **FileOutputStream**

Phương thức	Miêu tả
close()	Phương thức này đóng output stream. Giải phóng bất kỳ nguồn hệ thống nào liên kết với file. Ném một IOException.
finalize()	Phương thức này xóa sự kết nối tới File đó. Bảo đảm rằng phương thức close của output stream này được gọi khi không có tham chiếu nào nữa tới stream này. Ném một IOException.
write(int w)	Phương thức này ghi byte đã xác định tới output stream.
write(byte[] w)	Ghi w byte từ mảng byte đã đề cập tới OutputStream.

4.1.5. Thư mục trong Java

Thư mục là File mà có thể giữ một danh sách các file và thư mục khác. Sử dụng đối tượng File để tạo các thư mục, hoặc liệt kê các file có sẵn trong một thư mục. Để hiểu chi tiết, cần kiểm tra các phương thức có thể gọi trên đối tượng File và những gì liên quan tới thư mục.

a. Tạo thư mục

Có hai phương thức tiện ích File hữu ích, mà có thể được sử dụng để tạo các thư mục:

- Phương thức mkdir() tạo một thư mục, trả về true nếu thành công và false nếu thất bại. Việc thất bại là do đường dẫn đã xác định trong đối tượng File đã tồn tại, hoặc là do thư mục không thể được tạo vì cả đường dẫn không tồn tại.
- Phương thức mkdirs() tạo cả một thư mục và tất cả các thư mục cha của nó.

Ví dụ sau tạo thư mục D:\tmp\user\java\bin

```
import java.io.File;
public class CreateDir {
    public static void main(String args[]) {
        String dirname = "D:/tmp/user/java/bin";
        File d = new File(dirname);
        // Bay gio tao thu muc.
        d.mkdirs();
    }
}
```

Biên dịch và thực thi code trên sẽ tạo thư mục D:\tmp\user\java\bin

b. Liệt kê thư mục

Có thể sử dụng phương thức list() được cung cấp bởi đối tượng File để liệt kê tất cả các file và thư mục có sẵn trong một thư mục như sau:

```
import java.io.File;
public class ReadDir {
    public static void main(String[] args) {
        File file = null;
```

```

String[] paths;
try{
    // Tao doi tuong file moi
    file = new File("D:/tmp");
    // mang cac file va thu muc
    paths = file.list();
    // voi moi ten trong path array
    for(String path:paths)
    {
        // in ten file va ten thu muc
        System.out.println(path);
    }
} catch(Exception e){
    // neu co bat cu error nao xuat hien
    e.printStackTrace();
}
}
}

```

Kết quả liệt kê ra các thư mục và các file có sẵn trong thư mục /tmp.

4.1.6. ByteArrayInputStream

Lớp ByteArrayInputStream cho phép một bộ đệm (buffer) trong bộ nhớ để được sử dụng như là một InputStream. Nguồn input này là một mảng byte. Có hai loại constructor cơ bản sau để tạo đối tượng ByteArrayInputStream.

Nhận một mảng byte như là tham số:

```
ByteArrayInputStream bArray = new ByteArrayInputStream(byte [] a);
```

Nhận một mảng các byte, hai đối số kiểu nguyên là off và len, với off là byte đầu tiên để được đọc và len là số byte để được đọc.

```
ByteArrayInputStream bArray = new ByteArrayInputStream(byte [] a, int off, int len);
```

Khi đối tượng ByteArrayInputStream được khởi tạo thì có một số phương thức có thể được sử dụng để đọc stream hoặc để thực hiện các hoạt động khác trên stream đó.

Bảng 4.3: Các phương thức khác nhau và mô tả tương ứng của lớp **ByteArrayInputStream**

Phương thức	Mô tả
read()	Phương thức này đọc byte kế tiếp của dữ liệu từ InputStream. Trả về một int như là byte kế tiếp của dữ liệu. Nếu là kết thúc của file thì trả về -1.
read(byte[] r, int off, int len)	Phương thức này đọc len byte bắt đầu từ off từ Input Stream vào trong một mảng. Trả về tổng số byte đã đọc. Nếu kết thúc file, thì trả về -1.
available()	Cung cấp số lượng byte mà có thể được đọc từ file input stream này. Trả về một int mà cung cấp số các byte để được đọc.
mark(int read)	Phương thức này thiết lập vị trí đã đánh dấu hiện tại trong stream. tham

Phương thức	Mô tả
	số cung cấp giới hạn tối đa của các byte mà có thể được đọc trước khi vị trí đã đánh dấu trở nên vô hiệu
skip(long n)	Bỏ qua n byte từ stream. Phương thức này trả về số byte thực sự bị bỏ qua.

4.1.7. DataInputStream

DataInputStream được sử dụng trong ngữ cảnh của DataOutputStream và có thể được sử dụng để đọc các dữ liệu gốc (nguyên thủy) trong Java. Sau đây là constructor để tạo một InputStream trong Java:

```
InputStream in = DataInputStream(InputStream in);
```

Sau khi đối tượng DataInputStream được khởi tạo thì có một số phương thức có thể được sử dụng đọc stream hoặc thực hiện các hoạt động khác trên stream đó.

Bảng 4.4: Các phương thức khác nhau và mô tả tương ứng của lớp DataInputStream

Phương thức	Mô tả
read(byte[] r, int off, int len)	Đọc len byte của dữ liệu từ input stream vào trong một mảng các byte. Trả về tổng số byte đã đọc vào trong buffer, nếu nó là kết thúc file thì trả về -1.
read(byte [] b)	Đọc một số byte từ inputstream và lưu vào trong mảng byte. Trả về tổng số byte đã đọc vào trong buffer, nếu nó là kết thúc file thì trả về -1.
readBoolean() readByte() readShort() readInt()	Các phương thức này sẽ đọc các byte từ InputStream. Trả về hai byte kế tiếp của InputStream dưới dạng kiểu dữ liệu gốc (nguyên thủy) cụ thể.
readLine()	Đọc dòng tiếp theo của text từ input stream. Đọc các byte liên tục, biến đổi mỗi byte một cách riêng rẽ thành một ký tự, tới khi nó gặp line terminator hoặc kết thúc dòng, ký tự đọc được là một String.

4.1.8. ByteArrayOutputStream

Lớp ByteArrayOutputStream tạo một buffer trong bộ nhớ và tất cả dữ liệu được gửi tới stream này được lưu trong buffer. Có nhiều loại constructor sau để tạo đối tượng ByteArrayOutputStream trong Java. Constructor sau tạo một buffer 32 byte:

```
OutputStream bOut = new ByteArrayOutputStream();
```

Constructor sau tạo một buffer có kích cỡ là int a:

```
OutputStream bOut = new ByteArrayOutputStream(int a);
```

Khi đối tượng ByteArrayOutputStream được khởi tạo, thì có một số phương thức được sử dụng ghi stream hoặc thực hiện các hoạt động khác trên stream đó.

Bảng 4.5: Các phương thức khác nhau và mô tả tương ứng của lớp ByteArrayOutputStream

Phương thức	Mô tả
reset()	Phương thức này reset số lượng valid byte của mảng byte trong output stream zero, vì thế tất cả các output tích lũy trong stream sẽ bị loại bỏ.

Phương thức	Mô tả
toByteArray()	Phương thức này tạo một mảng byte được cấp phát mới. Kích cỡ của nó sẽ là kích cỡ hiện tại của output stream và nội dung của buffer sẽ được sao chép vào trong đó. Trả về nội dung hiện tại của output stream dưới dạng một mảng byte
toString()	Biến đổi nội dung buffer thành một chuỗi. Việc biến đổi sẽ được thực hiện theo mã hóa ký tự mặc định. Trả về String đã được biến đổi từ nội dung của buffer.
write(int w)	Ghi mảng đã xác định tới output stream.
write(byte []b, int off, int len)	Ghi len byte bắt đầu từ offset off tới stream.
writeTo(OutputStream outSt)	Ghi toàn bộ nội dung của Stream này tới tham số stream đã xác định.

4.1.9. DataOutputStream

Lớp DataOutputStream cho phép ghi các kiểu dữ liệu gốc tới một nguồn output. Sau đây là constructor để tạo một DataOutputStream trong Java:

```
DataOutputStream out = DataOutputStream(OutputStream out);
```

Khi đối tượng DataOutputStream được khởi tạo thì có một số phương thức được sử dụng ghi stream này hoặc thực hiện các hoạt động khác trên stream đó.

Bảng 4.6: Các phương thức khác nhau và mô tả tương ứng của lớp DataOutputStream

Phương thức	Mô tả
write(byte[] w, int off, int len)	Ghi len byte từ mảng byte đã xác định bắt đầu từ điểm off tới stream.
write(byte [] b)	Ghi số byte hiện tại, đã được ghi tới luồng dữ liệu output này. Trả về tổng số byte ghi vào trong buffer.
writeBoolean () writeByte () writeShort () writeInt ()	Những phương thức này sẽ ghi dữ liệu có kiểu gốc cụ thể vào trong output stream dưới dạng các byte.
flush()	Xóa sạch (flush) luồng dữ liệu output.
writeBytes(String s)	Ghi chuỗi tới output stream dưới dạng một dãy các byte. Mỗi ký tự trong chuỗi được ghi liên tục.

4.1.10. Lớp File

Lớp File trong biểu diễn tên đường dẫn (pathname) của các file và thư mục theo phương thức trừu tượng. Lớp này được sử dụng để tạo các file và thư mục, để tìm kiếm các file, xóa các file, v.v.

Đối tượng File biểu diễn file/thư mục thực sự trên đĩa. Có nhiều cách để tạo một đối tượng File:

- Cách sau tạo một đối tượng File mới từ một pathname trừu tượng cha và một chuỗi pathname con.

```
File(File parent, String child);
```

- Cú pháp sau tạo một đối tượng File mới bởi việc biến đổi chuỗi pathname đã cho thành một pathname trừu tượng.

`File(String pathname)`

- Cú pháp sau tạo một đối tượng File mới từ một chuỗi pathname cha và một chuỗi pathname con.

`File(String parent, String child)`

- Cú pháp sau tạo một đối tượng File mới bởi việc biến đổi URI đã cho thành một pathname trừu tượng.

`File(URI uri)`

Khi đối tượng File được tạo thì có các phương thức sau được sử dụng thao tác các file:

Bảng 4.7: Các phương thức khác nhau và mô tả tương ứng của lớp File

Phương thức	Mô tả
<code>getName()</code>	Trả về tên của file hoặc thư mục đã được biểu thị bởi pathname trừu tượng này.
<code>getParent()</code>	Trả về chuỗi pathname của pathname trừu tượng cha, hoặc null nếu pathname này không là tên của một thư mục cha.
<code>getParentFile()</code>	Trả về pathname trừu tượng của pathname trừu tượng cha, hoặc null nếu pathname này không là tên của một thư mục cha.
<code>getPath()</code>	Biến đổi pathname trừu tượng này thành một chuỗi pathname.
<code>isAbsolute()</code>	Kiểm tra có hay không pathname trừu tượng này là absolute. Trả về true nếu là absolute, nếu không là false.
<code>getAbsolutePath()</code>	Trả về chuỗi pathname tuyệt đối của pathname trừu tượng này.
<code>canRead()</code>	Kiểm tra có hay không ứng dụng có thể đọc file, được biểu thị bởi pathname trừu tượng này. Trả về true nếu và chỉ nếu file đã xác định bởi pathname trừu tượng này là tồn tại và có thể được đọc từ ứng dụng; nếu không là false.
<code>canWrite()</code>	Kiểm tra có hay không ứng dụng có thể sửa đổi file, được biểu thị bởi pathname trừu tượng này. Trả về true nếu và chỉ nếu hệ thống file thực sự chứa một file được biểu thị bằng pathname trừu tượng này và ứng dụng được phép write tới file đó; ngược lại là false.
<code>exists()</code>	Kiểm tra có hay không file hoặc thư mục được biểu thị bởi pathname trừu tượng này là tồn tại. Trả về true nếu và chỉ nếu file hoặc thư mục được biểu thị bởi pathname trừu tượng này là tồn tại; ngược lại là false.
<code>isDirectory()</code>	Kiểm tra có hay không file được biểu thị bởi pathname trừu tượng này là một thư mục. Trả về true nếu và chỉ nếu file được biểu thị bởi pathname trừu tượng này là tồn tại và là một thư mục; ngược lại là false.
<code>isFile()</code>	Kiểm tra có hay không file được biểu thị bởi pathname trừu tượng này là một normal file. Một file là normal nếu nó không là một thư mục và ngoại ra, thỏa mãn các tiêu chuẩn khác phụ thuộc vào hệ thống. Bất kỳ một non-directory file được tạo bởi một ứng dụng Java được bảo đảm là

Phương thức	Mô tả
	một normal file. Trả về true nếu và chỉ nếu file được biểu thị bởi pathname trùu tượng này là một normal file; ngược lại là false.
lastModified()	Trả về time mà file được biểu thị bởi pathname trùu tượng này được sửa đổi lần cuối cùng. Trả về giá trị long biểu diễn thời gian mà file được sửa đổi lần cuối cùng, được tính toán bằng mili giây từ 00:00:00 GMT, 1/1/1970, hoặc 0L nếu file không tồn tại hoặc nếu một I/O error xuất hiện.
length()	Trả về độ dài của file được biểu thị bởi pathname trùu tượng này. Giá trị trả về là không rõ ràng nếu pathname này biểu thị một thư mục.
createNewFile()	Tạo một file mới, trống với tên pathname trùu tượng này nếu và chỉ nếu một file với tên này chưa tồn tại. Trả về true nếu và chỉ nếu file với tên này không tồn tại và được tạo thành công; nếu file với tên này đã tồn tại, thì trả về false.
delete()	Xóa file hoặc thư mục được biểu thị bởi pathname trùu tượng này. Nếu pathname này biểu thị một thư mục, thì thư mục phải là trống để bị xóa. Trả về true nếu và chỉ nếu file hoặc thư mục bị xóa thành công; ngược lại là false.
deleteOnExit()	Yêu cầu rằng file hoặc thư mục được biểu thị bởi pathname trùu tượng này để bị xóa khi thiết bị ảo kết thúc.
list()	Trả về một mảng các chuỗi chỉ các file và thư mục trong thư mục được biểu thị bởi pathname trùu tượng này.
list(FilenameFilter filter)	Trả về một mảng các chuỗi chỉ các file và thư mục trong thư mục được biểu thị bởi pathname trùu tượng này mà thỏa mãn filter đã cho.
listFiles()	Trả về một mảng các pathname trùu tượng biểu thị các file trong thư mục được biểu thị bởi pathname trùu tượng này.
listFiles(FileFilter filter)	Trả về một mảng các pathname trùu tượng biểu thị các file và thư mục trong thư mục được biểu thị bởi pathname trùu tượng này mà thỏa mãn filter đã cho.
mkdir()	Tạo thư mục được chỉ bởi pathname trùu tượng này. Trả về true nếu và chỉ nếu thư mục được tạo; ngược lại là false
mkdirs()	Tạo thư mục được chỉ bởi pathname trùu tượng này, bao gồm bất kỳ thư mục cha nào cần thiết nhưng không tồn tại. Trả về true nếu và chỉ nếu thư mục được tạo, cùng với tất cả thư mục cha cần thiết; ngược lại là false.
renameTo(File dest)	Đổi tên file được biểu thị bởi pathname trùu tượng này. Trả về true nếu và chỉ nếu việc đổi tên thành công; ngược lại là false.
setLastModified(long time)	Thiết lập thời gian chỉnh sửa cuối cùng của file hoặc thư mục được chỉ bởi pathname trùu tượng này. Trả về true nếu và chỉ nếu hoạt động này thành công; ngược lại là false.
setReadOnly()	Đánh dấu file hoặc thư mục được chỉ bởi pathname trùu tượng này để mà các hoạt động read-only được cho phép. Trả về true nếu hoạt động này thành công; ngược lại là false.

Phương thức	Mô tả
createTempFile(String prefix, String suffix, File directory)	Tạo một file trống trong thư mục đã cho, bởi sử dụng các chuỗi tiền tố và hậu tố đã cung cấp để tạo tên của nó. Trả về pathname trừu tượng biểu thị một file trống mới được tạo.
createTempFile(String prefix, String suffix)	Tạo một file trống trong thư mục temporary-file, bởi sử dụng tiền tố và hậu tố đã cung cấp để tạo tên của nó. Triệu hồi phương thức này tương đương với gọi phương thức createTempFile(prefix, suffix, null). Trả về pathname trừu tượng biểu thị một file trống mới được tạo.
compareTo(File pathname)	So sánh 2 pathname trừu tượng theo từ điển. Trả về 0 nếu tham số là cân bằng với pathname trừu tượng này, một giá trị nhỏ hơn 0 nếu pathname trừu tượng này nhỏ hơn tham số theo từ điển, hoặc một giá trị lớn hơn 0 nếu pathname trừu tượng này lớn hơn tham số theo từ điển.
compareTo(Object o)	So sánh pathname trừu tượng này với đối tượng khác. Trả về 0 nếu tham số là cân bằng với pathname trừu tượng này, một giá trị nhỏ hơn 0 nếu pathname trừu tượng nhỏ hơn tham số theo từ điển, hoặc một giá trị lớn hơn 0 nếu pathname trừu tượng này lớn hơn tham số theo từ điển.
equals(Object obj)	Kiểm tra pathname trừu tượng này có cân bằng với đối tượng đã cung cấp không. Trả về true nếu và chỉ nếu tham số là không null và là một pathname trừu tượng mà biểu thị file hoặc thư mục giống như với pathname trừu tượng này.
toString()	Trả về chuỗi pathname của pathname trừu tượng này. Đây là chuỗi được trả về bởi phương thức getPath().

4.1.11. FileReader

Lớp FileReader trong Java kế thừa từ lớp InputStreamReader. FileReader được sử dụng để đọc các luồng ký tự. Lớp này có một số constructor để tạo các đối tượng cần thiết.

Cú pháp sau tạo một FileReader mới trong Java, được cung cấp File để đọc từ đó.

```
FileReader(File file)
```

Cú pháp sau tạo một FileReader mới trong Java, được cung cấp FileDescriptor để đọc từ đó.

```
FileReader(FileDescriptor fd)
```

Cú pháp sau tạo một FileReader mới trong Java, được cung cấp tên file để đọc từ đó.

```
FileReader(String fileName)
```

Khi đối tượng FileReader được tạo, thì có một số phương thức sau có thể thao tác các file này..

Bảng 4.8: Các phương thức khác nhau và mô tả tương ứng của lớp FileReader

Phương thức	Mô tả
read()	Đọc một ký tự đơn. Trả về một int, mà biểu diễn ký tự đã đọc.
read(char [] c, int offset, int	Đọc các ký tự bên trong một mảng. Trả về số ký tự đã đọc.

Phương thức	Mô tả
len)	

4.1.12. FileWriter

Lớp FileWriter trong Java kế thừa từ lớp OutputStreamWriter. Lớp này được sử dụng để ghi các luồng ký tự. Lớp FileWriter có một số constructor để tạo các đối tượng cần thiết.

Cú pháp sau tạo một đối tượng FileWriter được cung cấp một đối tượng File.

`FileWriter(File file)`

Cú pháp sau tạo một đối tượng FileWriter được cung cấp một đối tượng File với một Boolean chỉ dẫn có hay không phụ thêm dữ liệu đã ghi.

`FileWriter(File file, boolean append)`

Cú pháp sau tạo một đối tượng FileWriter mà liên kết với một File Descriptor.

`FileWriter(FileDescriptor fd)`

Cú pháp sau tạo một đối tượng FileWriter được cung cấp một tên file.

`FileWriter(String fileName)`

Cú pháp sau tạo một đối tượng FileWriter được cung cấp một tên file với một Boolean chỉ dẫn có hay không phụ thêm dữ liệu đã ghi.

`FileWriter(String fileName, boolean append)`

Khi đối tượng FileWriter được tạo thì có một số phương thức có thể được sử dụng để thao tác các file này.

Bảng 4.9: Các phương thức khác nhau của File Writer và mô tả tương ứng của lớp FileWriter

Phương thức	Mô tả
<code>write(int c)</code>	Ghi một ký tự đơn
<code>write(char [] c, int offset, int len)</code>	Ghi một phần của mảng các ký tự bắt đầu từ offset và với độ dài len
<code>write(String s, int offset, int len)</code>	Ghi một phần của String bắt đầu từ offset và với độ dài len

4.2. PHẦN BÀI TẬP

4.2.1. Bài tập mẫu

Bài 4.1. So sánh File path

Viết chương trình so sánh đường dẫn của 2 tệp tin

a. Hướng dẫn

Sử dụng phương thức filename.compareTo(another_filename) của lớp File trong Java.

b. Chương trình mẫu

```
import java.io.File;
public class CompareTo {
    public static void main(String[] args) {
        File file1 = new File("C:/File/demo1.txt");
    }
}
```

```

        File file2 = new File("C:/java/demo1.txt");
        if (file1.compareTo(file2) == 0) {
            System.out.println("Both paths are same!");
        } else {
            System.out.println("Paths are not same!");
        }
    }
}

```

💻 Kết quả chạy chương trình:

Paths are not same!

Bài 4.2: Tạo một File mới

Viết chương trình tạo một file trên ổ đĩa cứng. Chương trình thông báo thành công nếu tạo được file, thông báo không thành công nếu đã tồn tại file/

a. Hướng dẫn:

Sử dụng File() constructor và phương thức file.createNewFile() của lớp File.

b. Chương trình mẫu:

```

import java.io.File;
import java.io.IOException;

public class CreateFile {

    public static void main(String[] args) {
        try {
            File file = new File("D:/java/myfile.txt");
            if (file.createNewFile()) {
                System.out.println("Success!");
            } else {
                System.out.println("Error, file already exists.");
            }
        } catch (IOException ioe) {
            ioe.printStackTrace();
        }
    }
}

```

💻 Kết quả chạy chương trình:

Nếu myfile.txt không tồn tại trong thư mục D:/java
Success!

Nếu myfile.txt tồn tại trong thư mục D:/java
Error, file already exists.

Bài 4.3. Ngày sửa đổi File

Viết chương trình xem ngày sửa đổi file gần nhất

a. Hướng dẫn

Sử dụng phương thức file.lastModified() của lớp File.

b. Chương trình mẫu

```

import java.io.File;
import java.util.Date;

public class DateModifier {

```

```

    public static void main(String[] args) {
        File file = new File("D:/java/test.java");
        Long lastModified = file.lastModified();
        Date date = new Date(lastModified);
        System.out.println(date);
    }
}

```

💻 Kết quả chạy chương trình:

Tue Oct 18 14:14:57 ICT 2016

Bài 4.4. Tạo một File trong một thư mục cụ thể

Viết chương trình tạo một thư mục trên ổ đĩa cứng

a. Hướng dẫn

Sử dụng phương thức File.createTempFile() của lớp File.

b. Chương trình mẫu

```

import java.io.File;

public class CreFileInDir {

    public static void main(String[] args)
        throws Exception {
        File file = null;
        File dir = new File("D:/java");
        file = File.createTempFile("JavaTemp", ".javatemp", dir);
        System.out.println(file.getPath());
    }
}

```

💻 Kết quả chạy chương trình:

D:\java\JavaTemp6275777186753838097.javatemp

Bài 4.5. Kiểm tra một file có tồn tại hay không ?

Viết chương trình kiểm tra một tệp tin có tồn tại trên ổ đĩa hay không?

a. Hướng dẫn

Sử dụng phương thức file.exists() của lớp File.

b. Chương trình mẫu

```

import java.io.File;

public class FileExist {

    public static void main(String[] args) {
        File file = new File("D:/java/test.java");
        System.out.println(file.exists());
    }
}

```

💻 Kết quả chạy chương trình:

true

➡ **Bài 4.6. Tạo một file là read-only**

Viết chương trình tạo một file và gán cho file thuộc tính chỉ đọc (read-only)

a. Hướng dẫn

Sử dụng phương thức file.setReadOnly() và file.canWrite() của lớp File.

b. Chương trình mẫu

```
import java.io.File;
public class FileReaOnly {
    public static void main(String[] args) {
        File file = new File("D:/java/test.java");
        System.out.println(file.setReadOnly());
        System.out.println(file.canWrite());
    }
}
```

💻 Kết quả chạy chương trình:

true
false

➡ **Bài 4.7. Tạo, đọc, ghi, xóa file**

Viết chương trình tạo một file sau đó đọc file, ghi file và xóa file vừa tạo.

a. Hướng dẫn

Sử dụng phương thức delete() để xóa file.

Sử dụng BufferedReader, FileReader để tạo và đọc file.

Sử dụng BufferedWriter, FileWriter để ghi file.

b. Chương trình mẫu

```
import java.io.*;
public class CreReaDelFile {

    public static void main(String[] args) {
        try {
            BufferedWriter out = new BufferedWriter(new
FileWriter("filename"));
            out.write("dong 1\n");
            out.close();
            boolean success = (new File("filename")).delete();
            if (success) {
                System.out.println("Xoa file thanh cong!");
            }
            BufferedReader in = new BufferedReader(new
FileReader("filename"));
            String str;
            while ((str = in.readLine()) != null) {
                System.out.println(str);
            }
        }
    }
}
```

```

        }
        in.close();
    } catch (IOException e) {
        System.out.println("Xay ra loi:" + e);
        System.out.println("File khong ton tai de doc hoac
xoai!");
    }
}
}
}
}

```

 **Kết quả chạy chương trình:**

```

Xoa file thanh cong!
Xay ra loi:java.io.FileNotFoundException: filename (The system cannot
find the file specified)
File khong ton tai de doc hoac xoai!

```

 **Bài 4.8. Ghi dữ liệu vào d:/java/data.dat với DataOutputStream**

Viết chương trình để ghi dữ liệu dạng số vào file

a. Hướng dẫn

Sử dụng DataOutputStream để đọc file.

Sử dụng FileOutputStream để ghi file.

Sử dụng phương thức writeType để ghi dữ liệu nguyên thủy.

b. Chương trình mẫu

```

package BTChuong4;
import java.io.DataOutputStream;
import java.io.FileOutputStream;
import java.io.IOException;
public class DataOutput {
    public static void main(String[] args) {
        try {
            //Bước 1: Tạo đối tượng luồng và liên kết nguồn dữ liệu
            FileOutputStream fos = new
            FileOutputStream("d:/java/data.dat");
            DataOutputStream dos = new DataOutputStream(fos);
            //Bước 2: Ghi dữ liệu
            dos.writeInt(100);
            dos.writeDouble(9.5);
            //Bước 3: Đóng luồng
            fos.close();
            dos.close();
            System.out.println("Done!");
            System.out.println("Đã ghi giá trị 100 và 9.5 vào
file data.dat");
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}

```

 **Kết quả chạy chương trình: Done!,**

Đã ghi giá trị 100 và 9.5 vào file data.dat

 **Bài 4.9. Đọc dữ liệu chứa trong tập tin d:/java/data.dat với DataInputStream**

Viết chương trình đọc dữ liệu từ file sử dụng DataInputStream

a. Hướng dẫn

- Sử dụng FileInputStream để mở file.
- Sử dụng DataInputStream để đọc file.
- Sử dụng phương thức `readType` để đọc dữ liệu nguyên thủy.
- Dữ liệu đã được ghi vào file data.dat ở bài 4.8

b. Chương trình mẫu

```

package BTChuong4;

import java.io.DataInputStream;
import java.io.FileInputStream;
import java.io.IOException;

public class DataInput {
    public static void main(String[] args) {
        try {
            //Bước 1: Tạo đối tượng luồng và liên kết nguồn dữ
            liệu
            FileInputStream fis = new
            FileInputStream("d:/java/data.dat");
            DataInputStream dis = new DataInputStream(fis);
            //Bước 2: Đọc dữ liệu
            int n = dis.readInt();
            double m = dis.readDouble();
            //Bước 3: Đóng luồng
            fis.close();
            dis.close();
            //Hiển thị nội dung đọc từ file
            System.out.println("Số nguyên: " + n);
            System.out.println("Số thực: " + m);
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}

```

█ Kết quả chạy chương trình:

Số nguyên: 100
Số thực: 9.5

Bài 4.10. Đọc và ghi dữ liệu kiểu object

Thông tin về hàng hóa gồm: Mã hàng (id), tên hàng (name), giá (price) và số lượng (quantity).

Viết chương trình:

- Tạo lớp Stock chứa hàm tạo để khởi tạo các thuộc tính về hàng hóa, hàm in ra thông tin về hàng hóa.
- Ghi danh sách hàng hóa vào file objectdata.dat
- Đọc danh sách hàng hóa từ file objectdata.dat và in dữ liệu ra màn hình.

a. Hướng dẫn

- Tạo lớp Stock và lớp này phải cài đặt giao diện Serializable.
- Tạo lớp ObjectOutputStream dùng để ghi dữ liệu kiểu object

- Tạo lớp ObjectInput dùng để đọc dữ liệu kiểu object
- Tạo project BTChuong4.DocGhiObject chứa 3 file ứng với 3 chương trình sau:

b. Chương trình mẫu

* **Chương trình 1: Tạo lớp Stock**

```
//Stock.java
//Khai báo lớp Stock
package BTChuong4.DocGhiObject;
import java.io.Serializable;
public class Stock implements Serializable {
    private int id;
    private String desc;
    private double price;
    private int quantity;
    public Stock(int id, String desc, double price, int quantity) {
        this.id = id;
        this.desc = desc;
        this.price = price;
        this.quantity = quantity;
    }

    @Override
    public String toString() {
        return id + " " + desc + " " + price + " " + quantity;
    }
}
```

* **Chương trình 2: Ghi dữ liệu vào file**

```
// ObjectOutputStream.java
package BTChuong4.DocGhiObject;

import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectOutputStream;

public class ObjectOutputStream {

    public static void main(String[] args) {
        try {
            //Bước 1: Tạo đối tượng luồng và liên kết nguồn dữ liệu
            FileOutputStream fos = new FileOutputStream("d:/java/objectdata.dat");
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            //Khai báo mảng
            Stock s[] = {
                new Stock(1, "CD ROM", 100, 2), new Stock(2, "CPU", 500, 2), new
                Stock(3, "HP Scanner", 75, 1)
            };
            //Bước 2: Ghi mảng đối tượng vào file
            oos.writeObject(s);

            //Bước 3: Đóng luồng
            fos.close();
            oos.close();
            System.out.println("Ghi dữ liệu thành công!");
        } catch (IOException ex) {
            System.out.println("Loi ghi file: " + ex);
        }
    }
}
```

```

        }
    }
}

```

█ Kết quả chạy chương trình:
Ghi dữ liệu thành công!

* Chương trình 3: Đọc dữ liệu từ file in ra màn hình

```

// ObjectInput.java
package BTChuong4.DocGhiObject;

import java.io.FileInputStream;
import java.io.ObjectInputStream;

public class ObjectInput {

    public static void main(String[] args) {
        try {
            //Bước 1: Tạo đối tượng luồng và liên kết nguồn dữ liệu
            FileInputStream fis = new
FileInputStream("d:/java/objectdata.dat");
            ObjectInputStream ois = new ObjectInputStream(fis);
            //Bước 2: Đọc dữ liệu
            Stock sArr[] = (Stock[]) ois.readObject();
            for (Stock s : sArr) {
                System.out.println(s.toString());
            }
            //Bước 3: Đóng luồng
            fis.close();
            ois.close();
        } catch (Exception ex) {
            System.out.println("Loi doc file: " + ex);
        }
    }
}

```

█ Kết quả chạy chương trình:

1-	CD ROM	00.0	2
2-	CPU	500.0	2
3-	HP Scanner	75.0	1

█ *Bài 4.11. Đọc và ghi dữ liệu kiểu object – QL sinh viên*

Thông tin về sinh viên gồm: Họ tên (name), tuổi (age).

Viết chương trình:

- Tạo lớp Student gồm hàm tạo, hàm getName(), setName(), getAge(), setAge()
- Tạo lớp ProcessStudent để xử lý danh sách sinh viên, gồm có: Hàm tạo danh sách n sinh viên (creat), hàm ghi danh sách vào file (write), hàm đọc file ra mảng (read) và hàm in thông tin sinh viên lên màn hình (show)
- Chương trình chính: Nhập vào danh sách n sinh viên, ghi danh sách vào tệp, đọc dữ liệu từ tệp và in kết quả ra màn hình.

a. Hướng dẫn

- Tạo lớp Stock và lớp này phải cài đặt giao diện Serializable.

- Tạo lớp ObjectOutputStream dùng để ghi dữ liệu kiểu object
- Tạo lớp ObjectInputStream dùng để đọc dữ liệu kiểu object
- Tạo project BTChuong4.DocGhiObject chứa 3 file ứng với 3 chương trình sau:

b. Chương trình mẫu

```

package BTChuong4.DocGhiObject;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.Scanner;

public class IOFileObject_Student
{
    public static void main(String args[]) throws Exception {
        ProcessStudent ps = new ProcessStudent();
        // tao doi tuong cua lop ProcessStudent
        Student[] student1 = null, student2 = null;
        // tao 2 doi tuong student
        student1 = ps.create(); // nhap student
        ps.write(student1); // viet vao file
        student2 = ps.read(); // doc tu file
        ps.show(student2); //hien thi kq
    }
}

class Student implements Serializable
{
    // Serializable dung de ghi va doc theo Object
    private static final long serialVersionUID = 1L;
    // ID of Serializable
    private String name;
    private int age;
    public Student(String name, int age) {
        this.name = name;
        this.age = age;
    }
    // cac ham get, set gia tri cac bien
    public String getName() {
        return name;
    }
}

```

```

public void setName(String name) {
    this.name = name;
}

public int getAge() {
    return age;
}

public void setAge(int age) {
    this.age = age;
}

}

// class xu ly thong tin
class ProcessStudent

{
    public Student[] creat() {          // Input the data from Keyboard
        int n;
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter number of student: ");
        n = Integer.parseInt(scan.nextLine()); //nhap so sinh vien
        Student[] student = new Student[n];
        for (int i = 0; i < n; i++) {
            System.out.print("Enter name: "); // nhap ten sv thu i
            String name = scan.nextLine();
            System.out.print("Enter age: "); // nhap tuoi
            int age = Integer.parseInt(scan.nextLine());
            // tranh troi lenh nhu khi dung scan.nextInt()
            student[i] = new Student(name, age);
            // khai tao doi tuong Student thu i
        }
        scan.close();
        return student;
    }

    public void write(Student[] student) { //ghi theo Object
        try { // dat try cacth de tranh ngoai le khi tao va ghi File
            // tao file f tro den student.dat
            FileOutputStream f = new FileOutputStream("e:/student.dat");
            // dung de ghi theo Object vao file f
            ObjectOutputStream oStream = new ObjectOutputStream(f);
            // ghi student theo kieu Object vao file
            oStream.writeObject(student);
        }
    }
}

```

```

oStream.close();
System.out.println("Write OK!");
} catch (IOException e) {
System.out.println("Error Write file");
}
}

public Student[] read() { // doc theo Object
Student[] student = null;
try { // dat try cacth de tranh ngoai le khi tao va doc File
// tao file f tro den student.dat
FileInputStream f = new FileInputStream("e:/student.dat");
// dung de doc theo Object vao file f
ObjectInputStream inStream = new ObjectInputStream(f);
// dung inStream doc theo Object, ep kieu tra ve la Student
student = (Student[]) inStream.readObject();
inStream.close();
} catch (ClassNotFoundException e) {
System.out.println("Class not found");
} catch (IOException e) {
System.out.println("Error Read file");
}
return student;
}

//In data doc duoc tu file ra man hinh
public void show(Student[] student) throws Exception {
try {
for (int i = 0; i < student.length; i++) {
System.out.println((i + 1) + " : " + student[i].getName() + " " +
student[i].getAge());
}
} catch (NullPointerException e) {
System.out.println("File Empty");
}
}
}

```

 Kết quả chạy chương trình:

```

Enter number of student: 3
Enter name: Tran Nguyen Huong
Enter age: 38
Enter name: Le Chi Luan
Enter age: 40
Enter name: Nguyen Thai Son
Enter age: 32

```

```

Write OK!
1 : Tran Nguyen Huong      38
2 : Le Chi Luan          40
3 : Nguyen Thai Son       32

```

❑ Bài 4.11. Vào ra mảng byte

a. Hướng dẫn

Sử dụng `ByteArrayInputStream` để nhập dòng dữ liệu vào bộ đệm dạng byte.

Sử dụng `ByteArrayOutputStream` để lấy dòng dữ liệu từ bộ đệm dạng byte.

Sử dụng phương thức `write` và `read` để ghi và đọc dữ liệu dạng byte.

b. Chương trình mẫu

```

package BTChuong4;

import java.io.*;

public class ByteArrayExam {

    public static void main(String args[]) throws IOException {
        ByteArrayOutputStream os = new ByteArrayOutputStream();
        String s = "Welcome to Byte Array Input Outputclasses";
        for (int i = 0; i < s.length(); i++) {
            os.write(s.charAt(i));
        }
        System.out.println("Output Stream is:" + os);
        System.out.println("Size of output stream is: " +
os.size());
        ByteArrayInputStream in;
        in = new ByteArrayInputStream(os.toByteArray());
        int ib = in.available();

        System.out.println("Input Stream has: " + ib + " "
available bytes);
        byte ibuf[] = new byte[ib];
        int byrd = in.read(ibuf, 0, ib);
        System.out.println("Number of Bytes read are: " + byrd);
        System.out.println("They are: " + new String(ibuf));
    }
}

```

❑ Kết quả chạy chương trình:

```

Output Stream is:Welcome to Byte Array Input Outputclasses
Size of output stream is: 41
Input Stream has: 41 available bytes
Number of Bytes read are: 41
They are: Welcome to Byte Array Input Outputclasses

```

❑ Bài 4.12. Tìm kiếm thư mục

Viết chương trình tìm kiếm xem có tồn tại thư mục nào đó trong ổ đĩa hay không

a. Hướng dẫn

Sử dụng `File` để mở tạo đối tượng file trong đường dẫn thư mục.

Sử dụng phương thức `list` để lấy các file trong thư mục.

b. Chương trình mẫu

```
import java.io.File;

public class Main {
    public static void main(String[] argv)
        throws Exception {
        File dir = new File("E:\\KHTN");
        String[] children = dir.list();
        if (children == null) {
            System.out.println("does not exist or
                is not a directory");
        }
        else {
            for (int i = 0; i < children.length; i++) {
                String filename = children[i];
                System.out.println(filename);
            }
        }
    }
}
```

█ Kết quả chạy chương trình: KHTN

████ **Bài 4.13. Ghi dữ liệu với FileWriter**

a. Hướng dẫn

Sử dụng File để mở file.

Sử dụng FileWriter để ghi file.

Sử dụng phương thức write để ghi dữ liệu character.

b. Chương trình mẫu

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;

public class FileWriterExample {

    public static void main(String[] args) {
        try {
            //Bước 1: Tạo đối tượng luồng và liên kết nguồn dữ liệu
            File f = new File("d:\\java\\data2.dat");
            FileWriter fw = new FileWriter(f);
            //Bước 2: Ghi dữ liệu
            fw.write("Ghi dữ liệu bằng luồng character");
            //Bước 3: Đóng luồng
            fw.close();
        } catch (IOException ex) {
            System.out.println("Loi ghi file: " + ex);
        }
    }
}
```

█ Kết quả chạy chương trình:

Ghi file thành công!

████ **Bài 4.14. Đọc dữ liệu với FileReader**

a. Hướng dẫn

Sử dụng File để mở file.

Sử dụng FileReader để đọc file.

Sử dụng phương thức readLine để đọc từng dòng dữ liệu character.

b. Chương trình mẫu

```

package BTChuong4;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;

public class FileReaderExample {

    public static void main(String[] args) {
        try {
            //Bước 1: Tạo đối tượng luồng và liên kết nguồn dữ liệu
            File f = new File("d:/java/data2.dat");
            FileReader fr = new FileReader(f);
            //Bước 2: Đọc dữ liệu
            BufferedReader br = new BufferedReader(fr);
            String line;
            while ((line = br.readLine()) != null) {
                System.out.println(line);
            }
            //Bước 3: Đóng luồng
            fr.close();
            br.close();
        } catch (Exception ex) {
            System.out.println("Loi doc file: " + ex);
        }
    }
}

```

💻 Kết quả chạy chương trình:

Ghi dữ liệu bằng luồng character!

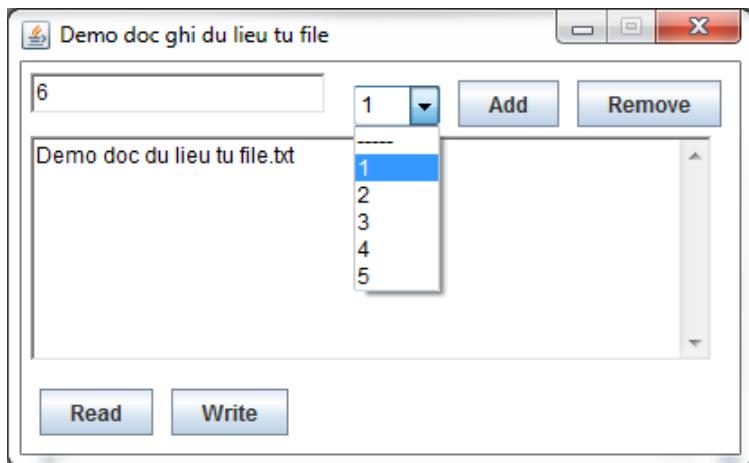
Ghi chú : *Đọc và ghi file trong java – Xử lý nhập xuất dữ liệu bằng luồng character*

Luồng byte rất mạnh mẽ và linh hoạt. Tuy nhiên nếu bạn muốn lưu trữ file chứa văn bản Unicode thì luồng character là lựa chọn tốt nhất vì ưu điểm của luồng character là nó thao tác trực tiếp trên ký tự Unicode.

Tất cả các luồng character đều được kế thừa từ 2 lớp Reader và Writer

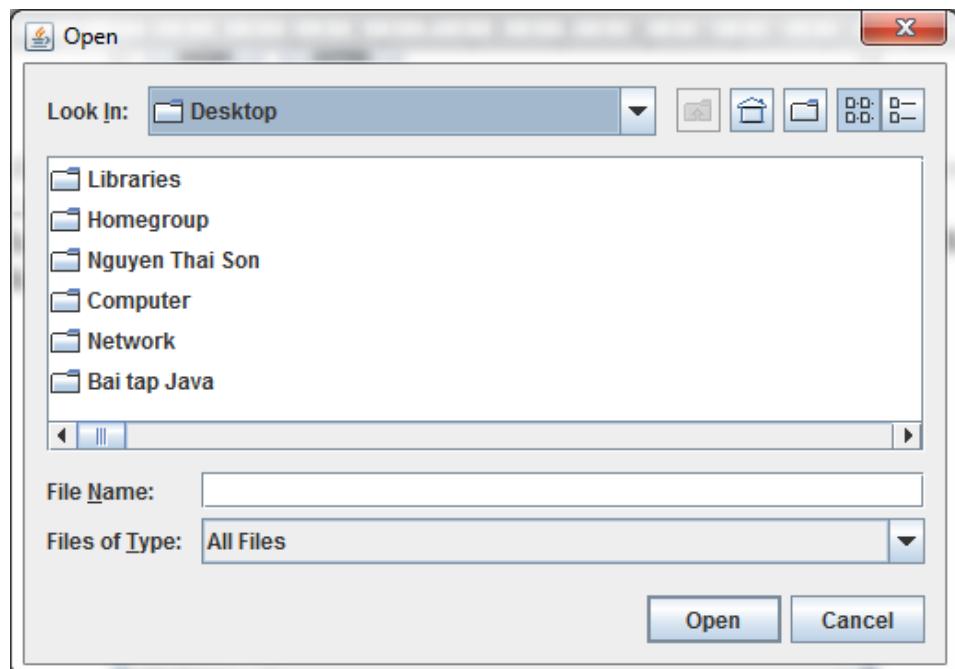
Bài 4.15. Đọc và ghi tệp (Sử dụng swing)

Xây dựng ứng dụng như sau:



Khi nhập dữ liệu vào textField và nhấn nút Add thì dữ liệu sẽ đưa vào Choice. Chọn dữ liệu trong Choice và nhấn nút Remove thì dữ liệu sẽ bị xóa khỏi Choice

Nút Read và Write để đọc, ghi dữ liệu trên TextArea vào file. Khi chọn nút Read thì xuất hiện hộp thoại Open để tìm tập tin cần mở.



Sau khi chọn được tệp tin (tệp txt) thì dữ liệu sẽ đưa vào TextArea.

Khi nhập văn bản vào TextArea và nhấn nút Write thì dữ liệu sẽ ghi vào file vừa mở và xuất hiện thông báo.



a. Hướng dẫn

- Khai báo để sử dụng các component của swing:

```
import javax.swing.*;
```

- Sử dụng phương thức JOptionPane.showMessageDialog() để hiển thị hộp thông báo
- Chương trình này khai báo sử dụng hỗn hợp các component của awt và swing.

b. Chương trình mẫu

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;

public class AddToChoice extends Frame implements ActionListener
{
    TextField txtItem = new TextField(20);
    JButton btAdd = new JButton("Add");
    JButton btRemove = new JButton("Remove");
    TextArea taContent = new TextArea(7, 30);
    GridBagLayout gb = new GridBagLayout();
    GridBagConstraints gbc = new GridBagConstraints();
    JButton btRead = new JButton("Read");
    JButton btWrite = new JButton("Write");
    Choice item = new Choice();
    String sItem[];
    int IndexArray = 0;
    public String path;
    ByteArrayOutputStream objbs = new ByteArrayOutputStream();

    AddToChoice(String txt)
    {
        super(txt);
        setLayout(gb);
        addAllComponent();
        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent e)
            {
                setVisible(false);
                System.exit(0);
            }
        });
    }

    public void AddItemToChoise(String str)
    {
        item.addItem(str);
        txtItem.setText("");
    }

    public void RemoveItemFromChoise(String str)
    {
        item.removeItem(str);
        txtItem.setText("");
    }

    public void addAllComponent()
    {
        gbc.fill = GridBagConstraints.HORIZONTAL;
        addComponent(txtItem, 0, 0, 1, 3);
        gbc.fill = GridBagConstraints.HORIZONTAL;
        item.addItem("-----");
        addComponent(item, 0, 3, 1, 1);
        gbc.fill = GridBagConstraints.HORIZONTAL;
        addComponent(btAdd, 0, 4, 1, 1);
        gbc.fill = GridBagConstraints.HORIZONTAL;
        addComponent(btRemove, 0, 5, 1, 1);
        gbc.fill = GridBagConstraints.BOTH;
        addComponent(taContent, 1, 0, 3, 6);
    }
}
```

```

        gbc.fill = GridBagConstraints.HORIZONTAL;
        addComponent(btRead, 4, 0, 1, 1);
        gbc.fill = GridBagConstraints.HORIZONTAL;
        addComponent(btWrite, 4, 1, 1, 1);
        btAdd.addActionListener(this);
        btRemove.addActionListener(this);
        btWrite.addActionListener(this);
        btRead.addActionListener(this);
    }
}

public void addComponent(Component comp, int row, int col, int nrow, int ncol)
{
    gbc.gridx      = col;
    gbc.gridy      = row;
    gbc.gridheight = nrow;
    gbc.gridwidth   = ncol;
    gbc.insets     = new Insets(5, 5, 5, 5);
    gb.setConstraints(comp, gbc);
    add(comp);
}

public boolean checkItem(String strItem)
{
    int i = 0;
    boolean flag = true;
    for(i= 0; i< item.getItemCount(); i++)
    {
        if (strItem == item.getItem(i))
        {
            JOptionPane.showMessageDialog(null,"err");
            flag = false;
            return flag;
        }
    }
    return flag;
}

public void actionPerformed(ActionEvent e)
{
    if(e.getSource() == btAdd)
    {
        if (txtItem.getText().equals(""))
        {
            JOptionPane.showMessageDialog(null,"Nhập giá trị !");
        }
        else
        {
            boolean flag = true;
            int i=0;
            for(i= 0; i< item.getItemCount(); i++)
            {
                if (txtItem.getText().equals(item.getItem(i)))
                    flag = false;
            }
            if(flag == true)
            {
                AddItemToChoise(txtItem.getText());
            }
            else
            {
                JOptionPane.showMessageDialog(null,"Giá trị đã tồn tại");
                txtItem.requestFocus();
                txtItem.selectAll();
            }
        }
    }
}

```

```

if(e.getSource() == btRemove)
{
    boolean flag = false;
    int i=0;
    if (txtItem.getText().equals(""))
    {
        JOptionPane.showMessageDialog(null,"Nhap gia tri !");
    }
    else
    {
        for(i= 0; i< item.getItemCount(); i++)
        {
            if (txtItem.getText().equals(item.getItem(i)))
            {
                flag = true;
            }
        }
        if(flag == true)
        {
            RemoveItemFromChoise(txtItem.getText());
        }
        else
        {
            JOptionPane.showMessageDialog(null,"Khong ton tai gia tri");
            txtItem.requestFocus();
            txtItem.selectAll();
        }
    }
}

if(e.getSource() == btRead)
{
try{
    JFileChooser chooser = new JFileChooser();
    int returnVal = chooser.showOpenDialog(this);
    File fl = chooser.getSelectedFile();

    String path = fl.getPath();
    File objf = new File(path);
    objbs.write(path.getBytes());
    RandomAccessFile objfi = new RandomAccessFile(path,"rw");
    long size = objfi.length();
    long fp = 0;
    String str = new String();
    while (fp < size)
    {
        str += objfi.readLine() + "\n";
        fp = objfi.getFilePointer();
    }
    taContent.setText(str);
    objfi.close();
}
catch(Exception ef)
{
    System.out.println(ef);
}
}

if(e.getSource() == btWrite)

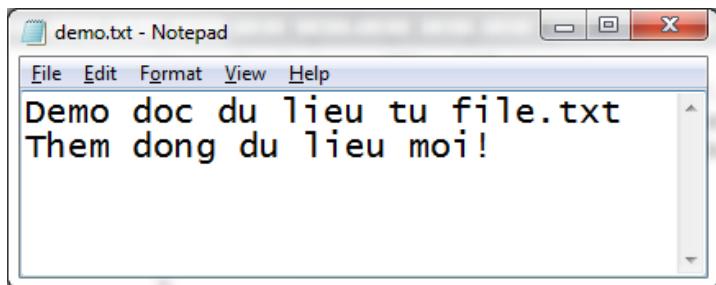
```

```

    }
    try {
        RandomAccessFile objfo = new
        RandomAccessFile(objbs.toString(),"rw");
        String content = taContent.getText();
        JOptionPane.showMessageDialog(null,objbs.toString());
        objfo.write(content.getBytes());
        objfo.close();
    }
    catch(Exception foe) {
    }
}
public static void main (String[] args) {
    AddToChoice obj = new AddToChoice("Demo doc ghi du lieu tu
file");
    obj.pack();
    obj.setResizable(false);
    obj.show();
}
}

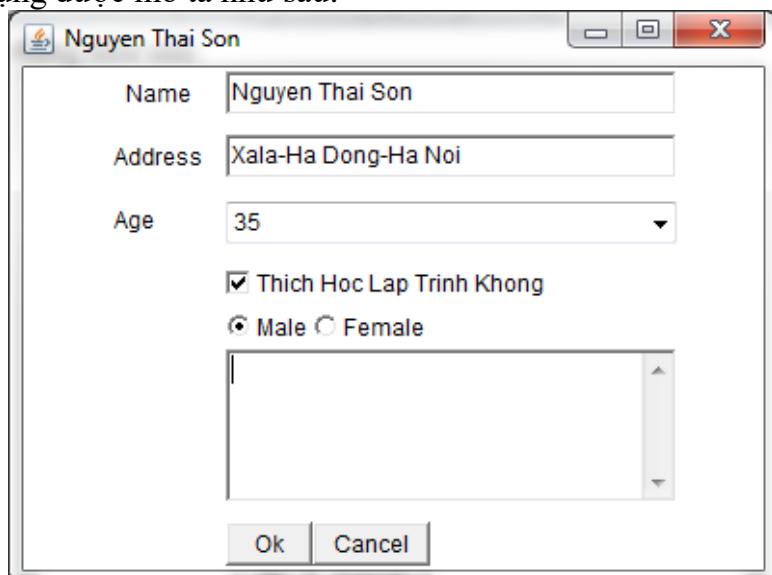
```

💻 Kết quả chạy chương trình:

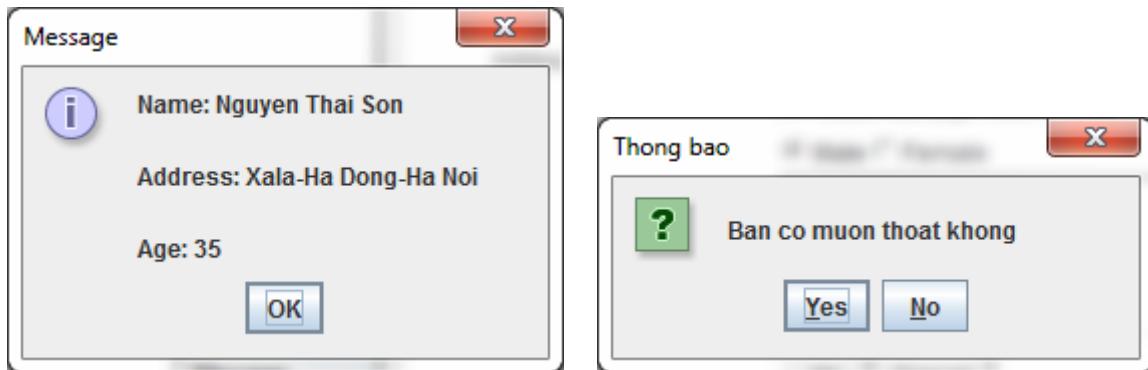


❑ Bài 4.16. Ghi dữ liệu vào file

Xây dựng ứng dụng được mô tả như sau:



Nhập đầy đủ thông tin và nhấn OK thì dữ liệu sẽ ghi vào file và xuất hiện hộp thông báo như hình sau:



Nếu nhập thiếu dữ liệu, sẽ xuất hiện các hộp thông báo nhắc nhở. Nhấn Cancel sẽ xuất hiện hộp thông báo Bạn có muốn thoát không?

a. Hướng dẫn

Để đọc hoặc ghi dữ liệu ta khai báo và sử dụng đối tượng thuộc lớp FileOutputStream.

Chẳng hạn: FileOutputStream fos = new FileOutputStream("D:\\Dulieu.text",true);

Sử dụng lệnh fos.write() để ghi dữ liệu lên file. Lệnh fos.close() để đóng file

b. Chương trình mẫu

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;

public class Demo extends Frame implements ActionListener
{
    Label lbname = new Label("Name");
    TextField tfname = new TextField(30);
    Label lbaddress = new Label("Address");
    TextField tfaddress = new TextField(30);
    Label lbage = new Label("Age");
    Choice ce = new Choice();

    CheckboxGroup cg = new CheckboxGroup();
    Checkbox ramale = new Checkbox("Male",cg,true);
    Checkbox rafemale = new Checkbox("Female",cg,false);

    TextArea ta = new TextArea(4,30);
    Button btok = new Button("Ok");
    Button btcancel = new Button("Cancel");

    Checkbox cb = new Checkbox("Thich Hoc Lap Trinh Khong",false);
    Panel pn = new Panel();
    Panel pl = new Panel();
    Panel pe =new Panel();
    Panel pnl = new Panel();

    GridBagLayout gb;
    GridBagConstraints gbc;
    FileOutputStream fos;

    public Demo(String v)
    {
        super(v);
        gb = new GridBagLayout();
    }
}

```

```

setLayout(gb);
gbc = new GridBagConstraints();

addComponent(lbname,1,0,1,1);
gbc.fill = GridBagConstraints.HORIZONTAL;
addComponent(tfname,1,1,1,4);
addComponent(pe,2,1,1,4);

addComponent(lbaddress,3,0,1,1);
gbc.fill= GridBagConstraints.HORIZONTAL;
addComponent(tfaddress,3,1,1,4);
addComponent(pl,4,1,1,4);

addComponent(lbage,5,0,1,1);
addComponent(ce,5,1,1,4);
ce.add("Tuoi");

int num;//tao ra cac phan` tu tuoi trong Choice
for( num = 1;num<=100;num++)
{
    String s = String.valueOf(num);
    ce.add(s);
}
addComponent(pn,6,1,1,4);
gbc.fill= GridBagConstraints.HORIZONTAL;
addComponent(cb,7,1,1,4);

gbc.fill = GridBagConstraints.HORIZONTAL;
addComponent(ramale,9,1,1,1);
addComponent(rafemale,9,2,1,1);

addComponent(ta,10,1,1,4);
gbc.fill = GridBagConstraints.HORIZONTAL;
addComponent(pnl,11,1,1,4);
addComponent(btok,12,1,1,1);
addComponent(btcancel,12,2,1,1);

btcancel.addActionListener(this);
btok.addActionListener(this);

//bat su kien khi nhan Enter se xuong dong
tfname.addKeyListener(new KeyAdapter()
{
    public void keyPressed(KeyEvent ky)
    {
        if(ky.getKeyCode ()==KeyEvent.VK_ENTER)
        {
            tfaddress.requestFocus();
        }
    }
});

addWindowListener(new WindowAdapter()
{
    public void windowClosing(WindowEvent we)
    {System.exit(0);
    }
});
}

```

```

public void addComponent(Component cmp,int row,int col, int nrow,int ncol)
{
    gbc.gridx = col;
    gbc.gridy = row;
    gbc.gridwidth = ncol;
    gbc.gridheight = nrow;
    gb.setConstraints(cmp,gbc);
    add(cmp);
}

public void actionPerformed( ActionEvent e)
{
    if(e.getSource() == btcancel)//khi click vào Cancel thì sẽ đóng
    {
        int tat;
        tat = JOptionPane.showConfirmDialog(null,"Bạn có muốn thoát không",
            "Thông báo",JOptionPane.YES_NO_OPTION);
        if(tat == JOptionPane.YES_NO_OPTION)
        {
            System.exit(0);
        }
        else    return ;
    }
    if(e.getSource()==btok)
    {
        String name = tfname.getText();
        String name2 = tfaddress.getText();
        String name3 = ce.getSelectedItem();
        if(name.equals(""))

        {
            JOptionPane.showMessageDialog(null,"Tên không được để trống");
            tfname.requestFocus();
            return ;
        }
        if(name2.equals(""))

        {
            JOptionPane.showMessageDialog(null,"Địa chỉ không được để trống");
            tfaddress.requestFocus();
            return ;
        }
        String v = tfname.getText();
        v = v + "\r\n" + tfaddress.getText() + "\r\n";
        //lấy dữ liệu từ 2 TextField
        try
        {
            //Xuat ra file dữ liệu
            //nếu để True thì xuất liên tiếp
            fos = new FileOutputStream("D:\\Dulieu.text",true);
            for(int index = 0;index<v.length();index++)
            {
                fos.write(v.charAt(index));
            }
            fos.close();
        }
        catch(IOException ev)
        {
        }
    }
    if(ce.getSelectedIndex()==0)//bắt sự kiện chọn tuổi
}

```

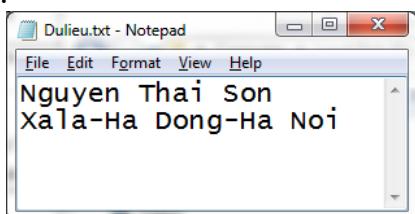
```

    {
        JOptionPane.showMessageDialog(null,"Ban phai chon tuoi");
        ce.requestFocus();
        return ;
    }
    JOptionPane.showMessageDialog(null,"Name: " + name + "\n\n" +
    "Address: " + name2 + "\n\n"+ "Age: " + name3 );
}
}

public static void main (String[] args)
{
    Demo objfd = new Demo("Nguyen Thai Son");
    objfd.setSize(400,300);
    objfd.setVisible(true);
}
}

```

💻 Kết quả chạy chương trình:



4.2.2. Bài tập tự giải

Bài 4.17. Viết chương trình so sánh path của hai file.

Bài 4.18. Viết chương trình sao chép một file vào trong file khác.

Bài 4.19. Viết chương trình kiểm tra thư mục là trống hay không.

Bài 4.20. Viết chương trình in cấu trúc cấp bậc của thư mục.

Bài 4.21. Chương trình Quản lý nhân viên

Tạo lớp NhanVien có các thuộc tính mã nhân viên, họ tên, tuổi và lương.

- Viết setter, getter, constructor và toString.
- Viết phương thức cho phép nhập thông tin cho 3 nhân viên từ bàn phím. Sử dụng FileWriter để ghi thông tin vào nhanvien.txt.
- Viết phương thức để đọc nội dung file nhanvien.txt và in những gì đọc được từ file ra màn hình.

Tạo lớp mới và thực hiện các xử lý sau:

- Viết phương thức tạo mảng 3 nhân viên, nhập thông tin cho các nhân viên, sử dụng ObjectOutputStream để ghi mảng đối tượng nhân viên vào file nhanvien.bin
- Viết phương thức để đọc file nhanvien.bin và in mảng nhân viên ra màn hình.

CHƯƠNG 5: LẬP TRÌNH CƠ SỞ DỮ LIỆU

5.1. PHẦN LÝ THUYẾT

5.1.1. Kiến trúc ODBC và JDBC

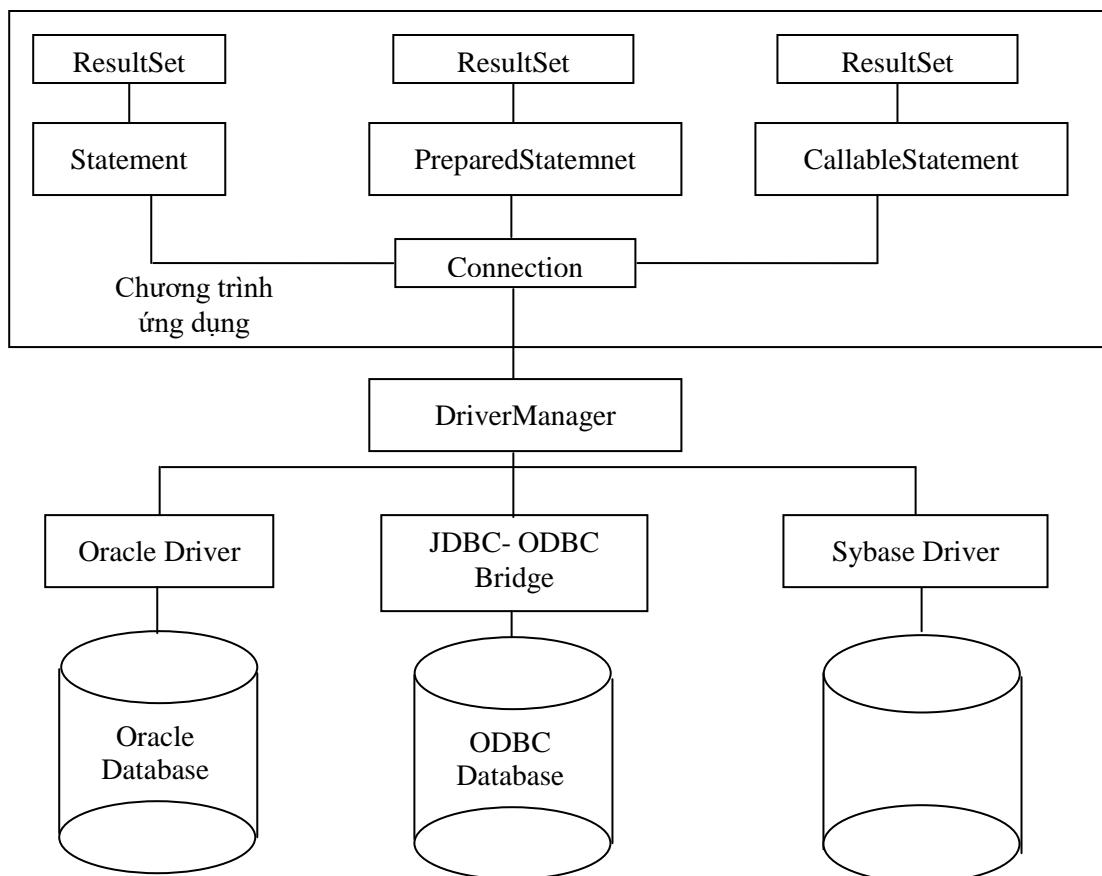
a. Kiến trúc ODBC (*Open DataBase Connectivity*) của Microsoft

ODBC là một trong những giao diện CSDL được sử dụng phổ biến nhất hiện nay trên các máy PC. ODBC cung cấp các chức năng của một ngôn ngữ lập trình để tương tác với các CSDL như: bổ sung, cập nhật, xoá những dữ liệu không cần thiết và nhận được những thông tin chi tiết về các bảng, các chỉ số hay khung nhìn của các phần dữ liệu trong một hệ CSDL[1].

Một ứng dụng ODBC thường được tổ chức theo năm tầng: tầng ứng dụng, giao diện ODBC, tầng quản lý các bộ điều khiển, bộ điều khiển và tầng nguồn dữ liệu.

b. Kiến trúc JDBC (*Java DataBase Connectivity*) của Sun

[1] Mỗi CSDL đều có một API riêng và muốn tương tác với CSDL đó thì phải biết được kiến trúc của chúng. Muốn sử dụng được các thông tin, dữ liệu từ nhiều CSDL trên mạng, chúng ta phải sử dụng những giao diện lập trình ứng dụng (API) để kết nối và truy vấn vào các CSDL[1].



Hình 5.1: Kiến trúc JDBC

JDBC tạo ra một lớp giao diện通用 giữa CSDL và Java. Với JDBC API, chúng ta có thể thực hiện được những chức năng cơ bản đối với CSDL:

- Thực hiện các truy vấn vào một CSDL.
- Xử lý kết quả từ truy vấn.
- Xác định các thông tin về cấu hình hệ thống.

Các lớp trong JDBC API được sử dụng để thực hiện những chức năng trên được tổ chức theo kiến trúc như trong hình 4.1.

5.1.2. Các lớp và giao diện của JDBC API

Một số lớp và giao diện trong gói java.sql được mô tả như bảng 4.1 và bảng 4.2

a. Một số lớp:

Bảng 5.1: Một số lớp trong gói java.sql

Tên lớp	Ý nghĩa
Date	Lớp biểu diễn kiểu DATE
DriverPropertyInfo	Chứa các thuộc tính của trình điều khiển đã nạp
Timestamp	Lớp biểu diễn cho SQL TimeTemp
DriverManager	Lớp quản lý các trình điều khiển
Time	Lớp biểu diễn kiểu DATE
Types	Lớp định nghĩa các hằng tương ứng với các kiểu dữ liệu SQL, hay còn gọi là kiểu dữ liệu JDBC

b. Một số giao diện:

Bảng 5.2: Một số giao diện trong gói java.sql

Tên giao diện	Mô tả ý nghĩa
CallableStatement	Giao diện chứa các phương thức cho phép ta làm việc với thủ tục lưu trữ nội
DatabaseMetaData	Cho phép ta xem các thông tin về CSDL
PreparedStatement	Giao diện này cho phép ta thực thi các câu lệnh SQL chứa tham số
ResultSetMetaData	
Connection	Thể hiện một kết nối đến CSDL
Driver	Giao diện mà mỗi trình điều khiển phải cài đặt
ResultSet	Thể hiện một tập các bản ghi lấy về từ CSDL
Statement	Giao diện cho phép ta thực hiện các phát biểu SQL

5.1.3. Kết nối cơ sở dữ liệu qua JDBC

Việc kết nối với CSDL bằng JDBC được thực hiện qua hai bước: đăng ký trình điều khiển JDBC; tiếp theo thực thi phương thức getConnection() của lớp DriverManager.

* Đăng ký trình điều khiển

Trình điều khiển JDBC được nạp khi mã bytecode của nó được nạp vào JVM. Một cách đơn giản để thực hiện công việc này là thực thi phương thức Class.forName("<JDBC Driver>").

Ví dụ 4.1: Để nạp trình điều khiển sử dụng câu nối ODBC do Sun cung cấp, chúng ta sử dụng câu lệnh sau Class.forName("sun.jdbc.odbc.JdbcOdbcDriver").

* **Thực hiện kết nối**

Sau khi đã nạp trình điều khiển JDBC, việc kết nối với CSDL được thực hiện với một trong các phương thức sau trong lớp DriverManager:

public static Connection getConnection(String url) throws SQLException: thực hiện kết nối với CSDL được yêu cầu. Bộ quản lý trình điều khiển sẽ tự động lựa chọn trình điều khiển phù hợp trong số các trình điều khiển đã được nạp.

public static Connection getConnection(String url, String user, String pass) throws SQLException: tiến hành kết nối tới CSDL với tài khoản user và mật mã pass.

public static Connection getConnection(String url, Properties info) throws SQLException: tương tự hai phương thức trên ngoài ra cung cấp thêm các thông tin qui định thuộc tính kết nối thông qua đối tượng của lớp Properties. Kết quả trả về của các phương thức trên là một đối tượng của lớp java.sql.Connection được dùng để đại diện cho kết nối đến CSDL.

* **Để xử lý được dữ liệu từ một CSDL, chương trình Java phải thực hiện lần lượt theo các bước sau:**

1. Trước tiên là gọi hàm *getConnection()* để nhận được đối tượng của lớp *Connection*;
2. Tạo ra một đối tượng của lớp *Statement*;
3. Chuẩn bị một đối tượng để xử lý lệnh của SQL và truy vấn vào dữ liệu theo yêu cầu; Câu lệnh SQL có thể thực hiện trực tiếp thông qua đối tượng của *Statement* hoặc có thể được biên dịch, dịch thông qua đối tượng của *PreparedStatement* hay gọi một thủ tục để lưu lại thông qua *CallableStatement*.
4. Khi hàm *executeQuery()* được thực hiện thì kết quả được cho lại là đối tượng của lớp *ResultSet* bao gồm các dòng dữ liệu và có thể sử dụng hàm *next()* để xác định các dữ liệu theo yêu cầu.

5.1.4. Kiểu dữ liệu SQL và Java

Các kiểu dữ liệu của SQL, và kiểu của Java tương ứng được mô tả như trong Bảng 5.3.

Bảng 5.3: Các kiểu dữ liệu SQL và kiểu của Java

Kiểu của SQL	Kiểu của Java	Hàm getXXX()
CHAR	String	getString()
VARCHAR	String	getString()
LONGVARCHAR	String	getString()
NUMBERIC	java.math.BigDecimal	getBigDecimal()
DECIMAL	java.math.BigDecimal	getBigDecimal()
BIT	Boolean (boolean)	getBoolean()

Kiểu của SQL	Kiểu của Java	Hàm getXXX()
TINYINT	Integer (byte)	getByte()
SMALLINT	Integer (short)	getShort()
INTEGER	Integer (int)	getInt()
BIGINT	Long (long)	getLong()
REAL	Float (float)	getFloat()
FLOAT	Double (double)	getDouble()
DOUBLE	Double (double)	getDouble()
BINARY	byte[]	getBytes()
VARBINARY	byte[]	getBytes()
LONGVARBINARY	byte[]	getBytes()
DATE	java.sql.Date	getDate()
TIME	java.sql.Time	getTime()
TIMESTAMP	java.sql.Timestamp	getTimestamp()

5.2. PHẦN BÀI TẬP

5.2.1. Bài tập mẫu

Bài 5.1. Kết nối với CSDL MS SQL Server

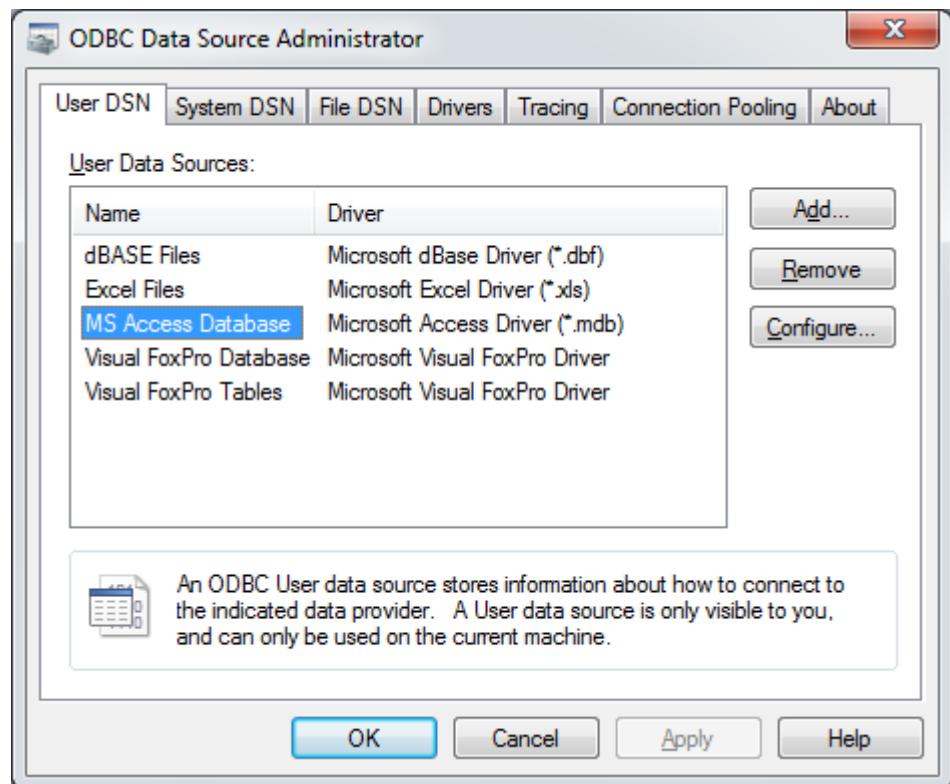
Trong **MS SQL Server** tạo Data Base có tên CSDL.MDF gồm 01 bảng như sau :

SinhVien(MaSV, Hoten, NgaySinh, DiaChi)

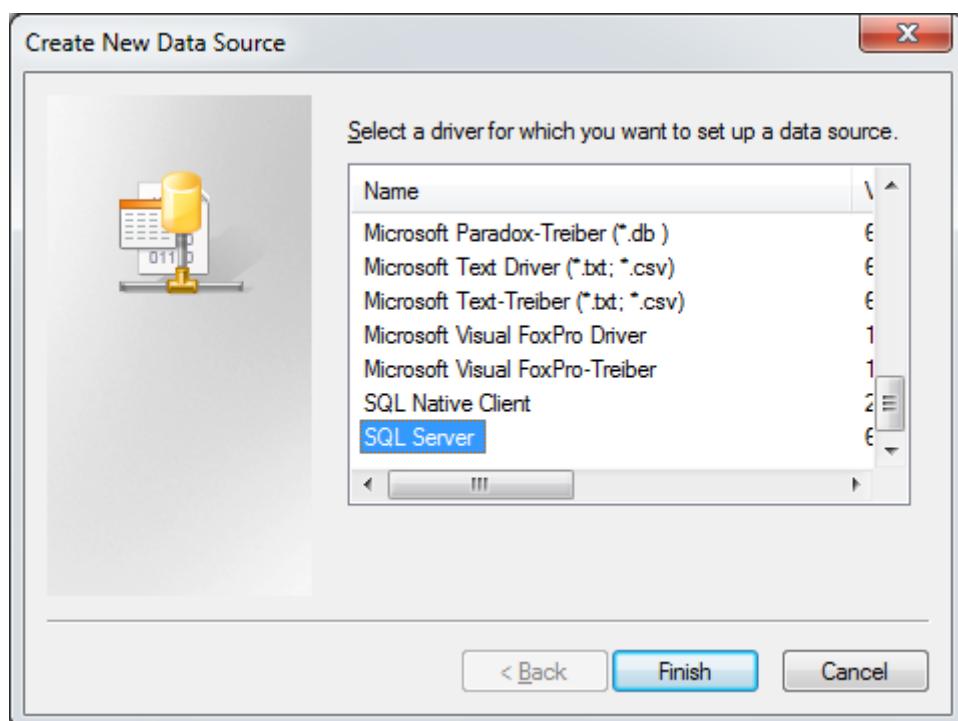
Yêu cầu: Kết nối Java với CSDL SQL Server và hiển thị dữ liệu trong bảng sinh viên lên màn hình.

a. Hướng dẫn

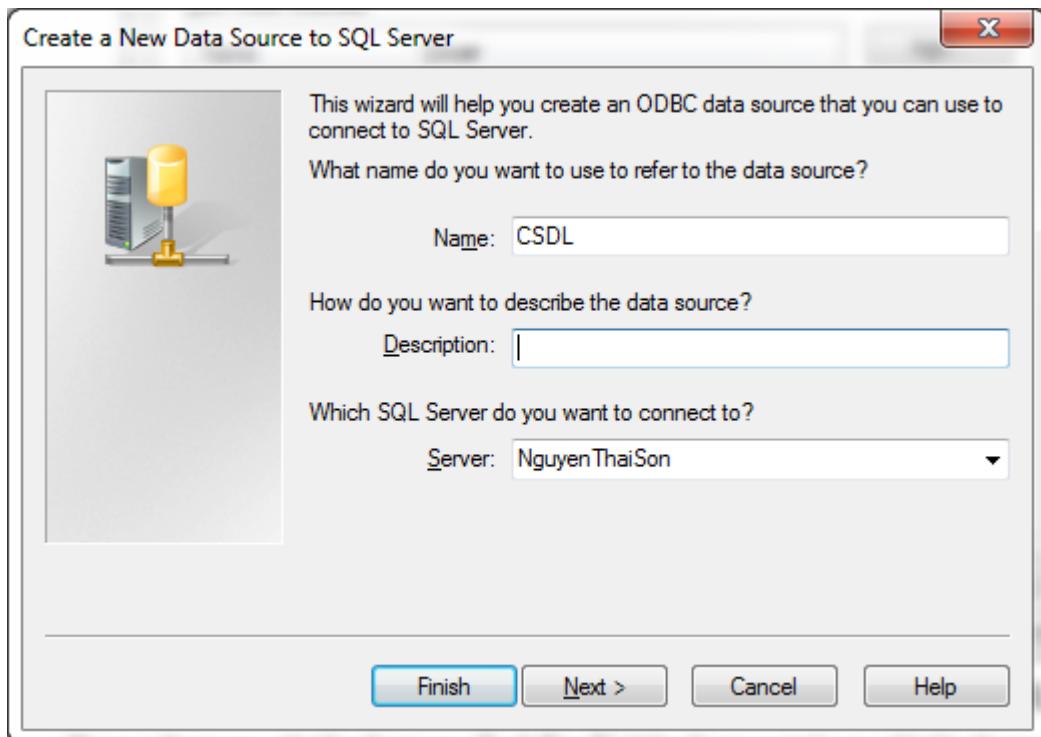
- Tạo Database như yêu cầu của đề bài
- Cấu hình trong Control Panel như sau: Control Panel\System & Security\Administrative Tools\Data Sources (ODBC), xuất hiện hộp thoại:



Chọn nút Add, xuất hiện hộp thoại, chọn SQL Server:



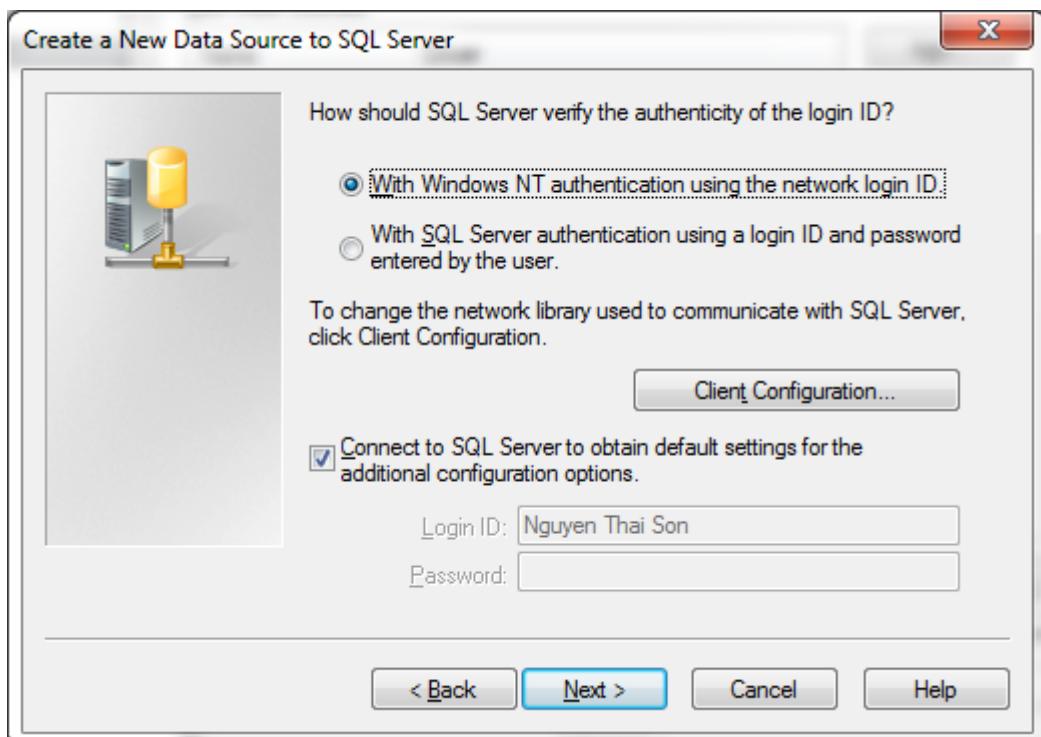
Chọn Finish, xuất hiện hộp thoại:



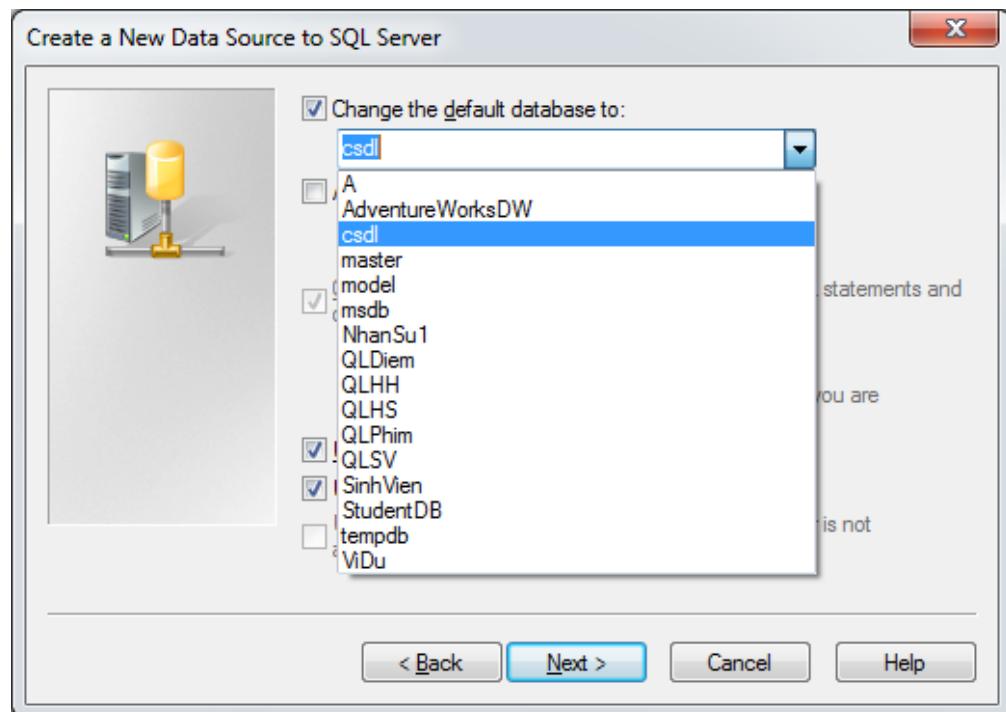
Trong Name gõ tên của cơ sở dữ liệu đã tạo, trong bài này ta gõ CSDL.

Trong Server, gõ tên Server cần kết nối (tên Server trùng với tên Server lúc đăng nhập hệ quản trị CSDL SQL Server), chẳng hạn ta gõ NguyenThaiSon

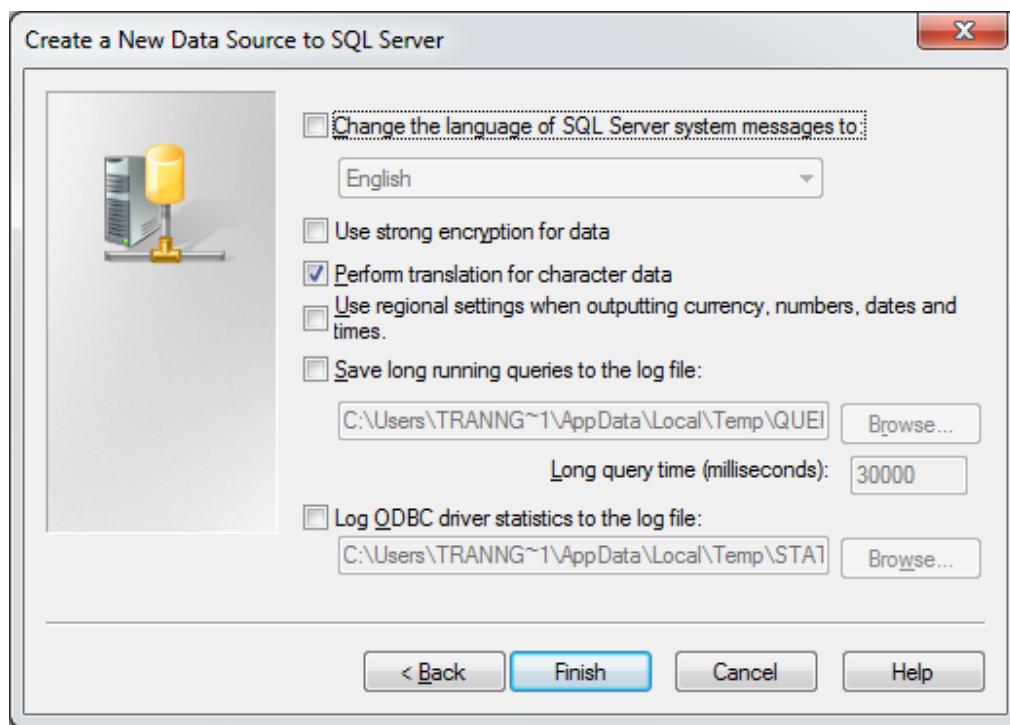
Chọn Next, xuất hiện hộp thoại:



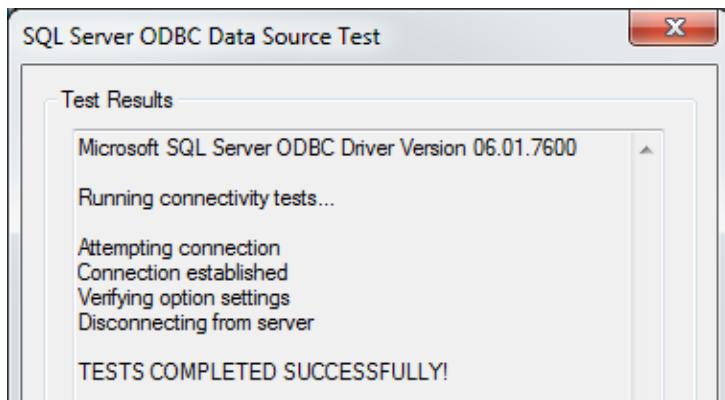
Chọn Next, xuất hiện hộp thoại



Tích chọn mục Change the default database to, sau đó chọn tên cơ sở dữ liệu cần kết nối, trong bài này ta chọn CSDL, sau đó chọn Next, xuất hiện hộp thoại



* Chọn Finish, sau đó chọn Test Data Source v.v để kiểm tra kết nối. Nếu xuất hiện hộp thoại sau thì kết nối thành công. Nhấn OK để kết thúc.



b. Chương trình mẫu

```

package qlsinhvien;
import java.io.*;
import java.sql.*;
class SinhVien
{
    String masv,hoten,ngaysinh,diachi;
    public SinhVien(String ma,String ht,String ns,String dc)
    {
        masv=ma;
        hoten=ht;
        ngaysinh=ns;
        diachi=dc;
    }
    public void hienthi()
    {
        System.out.print("MSV : " + masv + "\tHT : " + hoten);
        System.out.println("\tNS : " + ngaysinh + "\tDC : " + diachi);
    }
}
class BangSV
{
    Connection cnn;
    public BangSV(Connection cn)
    {
        cnn=cn;
    }
    public void hienthi()
    {
        try
        {
            Statement ADODB=cnn.createStatement();
            String sql="Select * from SinhVien";
            ResultSet rst=ADODB.executeQuery(sql);
            while (rst.next())
            {
                String ma=rst.getString(1);
                String ht=rst.getString(2);
                String ns=rst.getString(3);
                String dc=rst.getString(4);
                SinhVien sv=new SinhVien(ma,ht,ns,dc);
                sv.hienthi();
            }
            rst.close();
            ADODB.close();
        }
        catch (SQLException ev)
    }
}

```

```

        {
            ev.printStackTrace();
        }
    }
};

class qlSinhvien
{
    final static String jdbcURL = "jdbc:odbc:csdl";
    final static String jdbcDriver ="sun.jdbc.odbc.JdbcOdbcDriver";
    final static String table = "SinhVien";
    public static void main(String[] args)
    {
        try
        {
            Class.forName(jdbcDriver);
            Connection con = DriverManager.getConnection(jdbcURL, "", "");
            Statement stmt = con.createStatement();
            BangSV tbSV = new BangSV(con);
            tbSV.hienthi();
            con.close();
        }
        catch (Exception evt)
        {
            System.out.println("Loi ket noi CSDL"+evt.getMessage());
        }
    }
}
}

```

💻 Kết quả chạy chương trình:

```

MSV : s1      HT : Huong NS : 1990-01-01 00:00:00.000      DC : HN
MSV : s2      HT : Hoa     NS : 1990-03-02 00:00:00.000      DC : TB
Process completed.

```

▣ **Bài 5.2. Kết nối trên nhiều bảng CSDL SQL Server**

Tạo CSDL QLPhim.mdf trong SQL Server gồm có bảng:

Phim(MaPhim, TenPhim, MaTL, DaoDien) và

TheLoai(MaTL, TenTL): mã thể loại, tên thể loại.

Hãy hiển thị ra màn hình mã phim, tên phim, tên thể loại và đạo diễn.

a. Hướng dẫn

* Xây dựng CSDL trong SQL Server theo yêu cầu của đề bài, nhập dữ liệu như sau:

MaPhim	TenPhim	MaTL	DaoDien
P1	Cáo và chó săn	B	Art Stevens
P2	Ký băng hà	B	Carlos Saldanha
P3	HCM chân dung một con người	D	Bùi Đình Hạc
P4	Tây du ký	E	Ngô Thừa Ân

MaTL	TenTL
A	Khoa học
B	Hoạt hình
C	Tâm lý
D	Lịch sử
E	Phim thiếu nhi

Lưu CSDL với tên là QLPhim.MDF.

* Cấu hình trong Control Panel như sau: Control Panel\System & Security\ Aministrative Tools\Data Sources (ODBC): Các bước thực hiện tương tự bài 4.1

b. Chương trình mẫu

```

import java.io.*;
import java.sql.*;
class Phim
{
    String map,tenp,tloai,ddien;
    public Phim(String ma,String ten,String tl, String dd)
    {
        map=ma;
        tenp=ten;
        tloai=tl;
        ddiem=dd;
    }
    public String toString()
    {
        return "Maphim : " + map + ",\tTenphim : " + tenp + ",\tTheLoai : " +
        tloai + "\tDaodien : " + ddiem;
    }
}

class MoBang
{
    Connection con;
    public MoBang(Connection cn)
    {
        con=cn;
    }
    public void hienthi()
    {
        try
        {
            Statement Data=con.createStatement();
String sql="select maphim,tenphim,tentl,daodien from theloai inner
join phim on theloai.matl=phim.maTL";
            ResultSet rst=Data.executeQuery(sql);
            while (rst.next())
            {
                String ma=rst.getString(1);
                String ten=rst.getString(2);
                String tl=rst.getString(3);
                String dd=rst.getString(4);
                Phim ph=new Phim(ma,ten,tl,dd);
                System.out.println(""+ph.toString());
            }
            rst.close();
            Data.close();
        }
        catch (SQLException evt)
    }
}

```

```

        {
            evt.printStackTrace();
        }
    }
}

class qlPhim
{
    final static String jdbcURL = "jdbc:odbc:QLphim";
    final static String jdbcDriver ="sun.jdbc.odbc.JdbcOdbcDriver";
    public static void main(String[] args)
    {
        try
        {
            Class.forName(jdbcDriver);
            Connection con = DriverManager.getConnection(jdbcURL, "", "");
            Statement stmt = con.createStatement();
            MoBang tbPhim=new MoBang(con);
            tbPhim.hienthi();
            con.close();
        }
        catch (Exception evt)
        {
            System.out.println("Loi ket noi : " + evt.toString());
        }
    }
}
}

```

Kết quả chạy chương trình:

Maphim : P1,Tenphim : Cáo và chó săn, Theloai : Hoạt hình Daodien : Art Stevens
Maphim : P2,Tenphim : Kỷ băng hà, Theloai : Hoạt hình Daodien : Carlos Saldanha
Maphim : P3,Tenphim : HCM chân dung một con người, Theloai: Lịch sử Daodien: Bùi Đình Hạc
Maphim : P4,Tenphim : Tây du ký, Theloai : Phim thiếu nhi Daodien : Ngô Thùa Ân

Bài 5.3. Thêm, sửa, xóa trên SQL Server

Tạo CSDL QLHH.mdf trong SQL Server gồm có bảng HangHoa(MaHang, TenHang, Dvt)

Thực hiện các thao tác thêm, sửa, xóa dữ liệu trong CSDL trên.

a. Hướng dẫn

- * Tạo cơ sở dữ liệu QLHH.MDF như yêu cầu của đề bài
- * Cấu hình cơ sở dữ liệu trong Control Panel: như bài 4.1
- * Xây dựng menu như sau:

Danh Mục Các Chức Năng

1. Them Hang Hoa
2. Sua Hang Hoa
3. Xoa Hang Hoa
4. Hien Thi Danh Sach
0. Thoat Khoi Chuong Trinh

Bạn chọn :

Người dùng sẽ chọn các chức năng tương ứng.

- * Các thao tác:

Thêm: sử dụng insert...into...values..

Sửa: update...set...where

Xóa: delete from ... where

b. Chương trình mẫu

```
import java.io.*;
import java.sql.*;
class HangHoa
{
    String mah,tenh,dv;
    public HangHoa(String mah,String tenh,String dv)
    {
        this.mah=mah;
        this.tenh=tenh;
        this.dv=dv;
    }
    public void hienthi()
    {
        System.out.println("Mahang : " + mah + "\tTenhang : " + tenh);
        System.out.println("\tDVT : " + dv);
    }
}
class MoBang
{
    Connection cnn;
    public HangHoa A[]=new HangHoa[100];
    public int n=0;
    int i=0;
    String ma,ten,dv;
    int kt=0;

    public MoBang(Connection cn)
    {
        cnn=cn;
    }
    public void layDL()
    {
        n=0;
        i=0;
        try
        {
            Statement table=cnn.createStatement();
            String strSQL="Select * From HangHoa";
            ResultSet rst = table.executeQuery(strSQL);
            while (rst.next())
            {
                ma=rst.getString(1);
                ten=rst.getString(2);
                dv=rst.getString(3);
                HangHoa a=new HangHoa(ma,ten,dv);
                A[i]=a;
                i=i+1;
                n=n+1;
            }
            rst.close();
            table.close();
        }
        catch (SQLException eLayDL)
```

```

        {
            eLayDL.printStackTrace();
        }
    }

public void CapNhatDL()
{
    int ch=0;
    BufferedReader in=new BufferedReader(new
InputStreamReader(System.in));
    try
    {
        do
        {
            layDL();
            System.out.println("\n\tDanh Muc Cac Chuc Nang");
            System.out.println("\t1.Them Hang Hoa");
            System.out.println("\t2.Sua Hang Hoa");
            System.out.println("\t3.Xoa Hang Hoa");
            System.out.println("\t4.Hien Thi Danh Sach");
            System.out.println("\t0.Thoat Khoi Chuong Trinh");
            System.out.print("\tBan chon : ");
            ch=Integer.parseInt(in.readLine());
            switch (ch)
            {
                case 1:
                    ThemMoiHH(); break;
                case 2:
                    CapNhatHH(); break;
                case 3:
                    XoaBoHH(); break;
                case 4:
                    hienthiHH(); break;
            }
        }
        while (ch!=0);
    }
    catch (Exception evMenu)
    {
        evMenu.printStackTrace();
    }
}

public void hienthiHH()
{
    System.out.println("Danh Muc Hang Hoa");
    for (i=0;i<n;i++) A[i].hienthi();
}

public void ThemMoiHH() throws Exception
{
    System.out.println("Them Moi Hang Hoa");
    BufferedReader in=new BufferedReader(new
InputStreamReader(System.in));
    try
    {
        do
        {
            kt=0;
            System.out.print("Nhap Ma Hang : ");
            ma=in.readLine();
            //kiem tra trung ma, khi them khong cho phep nhap trung ma
            for (i=0;i<n;i++)
                if (ma.equals(A[i].mah) || ma.equals("")) kt=1;
    }
}

```

```

        }
        while (kt!=0);
        System.out.print("Nhập Tên Hàng : ");
        ten=in.readLine();
        System.out.print("Nhập Đơn Vị Tính : ");
        dv=in.readLine();
        Statement table=cnn.createStatement();
String strSQL="INSERT INTO HangHoa(MaHang,TenHang,Dvt) VALUES ('" + ma
+ "','" + ten + "','" + dv + "')";
        ResultSet rst = table.executeQuery(strSQL);
    }
catch (SQLException evt1)
{
    evt1.printStackTrace();
}
}

public void CapNhatHH() throws Exception
{
System.out.println("Cap Nhat Hang Hoa");
BufferedReader in=new BufferedReader(new
InputStreamReader(System.in));
try
{
    kt=0;
    String masua;
    System.out.print("Nhập Mã Hàng Cần Cập Nhật : ");
    masua=in.readLine();
    //kiểm tra trùng mã, khi sửa thi mã hàng phải ton tai
    for (i=0;i<n;i++)
    if (ma.equals(A[i].mah) || (ma.equals("")))) kt=1;
    if (kt==1)
    {
    do
    {
    kt=0;
    System.out.print("Nhập Mã Hàng Mới : ");
    ma=in.readLine();
    //ma hàng phải chưa ton tai hoặc trùng với masua
    for (i=0;i<n;i++)
if ((ma.equals(A[i].mah)) && (!ma.equals(masua)) && (ma.equals(""))))
        kt=1;
    if (kt==1)
        System.out.println("Mã hàng bạn nhập đã tồn tại trong CSDL");
    }while (kt!=0);
    System.out.print("Nhập Tên Hàng : ");
    ten=in.readLine();
    System.out.print("Nhập Đơn Vị Tính : ");
    dv=in.readLine();

    Statement table=cnn.createStatement();
    String strSQL="UPDATE HangHoa SET MaHang = '" + ma + "',TenHang =
'" + ten + "',dvt = '" + dv + "' WHERE mahang = '" + masua + "'";
    ResultSet rst = table.executeQuery(strSQL);
    }
else
    {
    System.out.print("Mã Hàng Không Tồn Tại Trong CSDL");
    }
}
}

```

```

        catch (SQLException evt1)
        {
            evt1.printStackTrace();
        }
    }

    public void XoaBoHH() throws Exception
    {
        System.out.println("Xoa Bo Hang Hoa");
        BufferedReader in=new BufferedReader(new
InputStreamReader(System.in));
        try
        {
            kt=0;
            System.out.print("Nhap Ma Hang Can Xoa: ");
            ma=in.readLine();
            //kiem tra ma hang can xoa phai ton tai
            for (i=0;i<n;i++)
                if (ma.equals(A[i].mah)) kt=1;
            if (kt==1)
            {
                Statement table=cnn.createStatement();
String strSQL="DELETE FROM HangHoa WHERE mahang ='" + ma + "'";
                ResultSet rst = table.executeQuery(strSQL);
            }
            else
            {
                System.out.print("Ma Hang Khong Ton Tai Trong CSDL");
            }
        }
        catch (SQLException evt1)
        {
            evt1.printStackTrace();
        }
    }
}

class qlHH
{
    public static void main(String[] args)
    {
        String jdbcURL = "jdbc:odbc:QLHH";
        String jdbcDriver = "sun.jdbc.odbc.JdbcOdbcDriver";
        String table = "hanghoa";
try
{
    Class.forName(jdbcDriver);
    Connection con =DriverManager.getConnection(jdbcURL, "", "");
    Statement stmt = con.createStatement();

    MoBang tbl=new MoBang(con);
    tbl.layDL();
    tbl.CapNhatDL();
    con.close();
}
catch (Exception evCNN)
{
    evCNN.printStackTrace();
}
}

```

}

💻 Kết quả chạy chương trình:

Danh Mục Các Chức Năng

- 1.Thêm Hàng Hóa
- 2.Sửa Hàng Hóa
- 3.Xóa Hàng Hóa
- 4.Hiện Thị Danh Sach
- 0.Thoát Khoi Chuong Trinh

Bạn chọn : 4

Danh Mục Hàng Hoá

Mahang : H1	Tenhang : Laptop	DVT : chiec
Mahang : H2	Tenhang : Tivi	DVT : Chiec
Mahang : H3	Tenhang : Gao	DVT : kg
Mahang : H4	Tenhang : Xe dap	DVT : chiec
Mahang : H5	Tenhang : Xe máy	DVT : Chiec
Mahang : H6	Tenhang : Quan áo	DVT : bo
Mahang : H8	Tenhang : Mac book	DVT : chiec

Danh Mục Các Chức Năng

- 1.Thêm Hàng Hóa
- 2.Sửa Hàng Hóa
- 3.Xóa Hàng Hóa
- 4.Hiện Thị Danh Sach
- 0.Thoát Khoi Chuong Trinh

Bạn chọn : 1

Thêm Mới Hàng Hoá

Nhập Mã Hàng : h10
Nhập Tên Hàng : Ipad
Nhập Đơn Vị Tính : chiec
Danh Mục Các Chức Năng
1.Thêm Hàng Hóa
2.Sửa Hàng Hóa
3.Xóa Hàng Hóa
4.Hiện Thị Danh Sach
0.Thoát Khoi Chuong Trinh
Bạn chọn : 2

Cập Nhật Hàng Hoá

Nhập Mã Hàng Cần Cập Nhật : H5
Nhập Mã Hàng Mới : H5
Nhập Tên Hàng : Dau goi
Nhập Đơn Vị Tính : chai
Danh Mục Các Chức Năng
1.Thêm Hàng Hoá
2.Sửa Hàng Hoá
3.Xóa Hàng Hoá
4.Hiện Thị Danh Sach
0.Thoát Khoi Chuong Trinh
Bạn chọn : 4

Danh Mục Hàng Hoá

Mahang : H1	Tenhang : Laptop	DVT : chiec
Mahang : h10	Tenhang : Ipad	DVT : chiec
Mahang : H2	Tenhang : Tivi	DVT : Chiec
Mahang : H3	Tenhang : Gao	DVT : kg
Mahang : H4	Tenhang : Xe dap	DVT : chiec
Mahang : H5	Tenhang : Dau goi	DVT : chai
Mahang : H6	Tenhang : Quan áo	DVT : bo
Mahang : H8	Tenhang : Mac book	DVT : chiec

```

Danh Muc Cac Chuc Nang
1.Them Hang Hoa
2.Sua Hang Hoa
3.Xoa Hang Hoa
4.Hien Thi Danh Sach
0.Thoat Khoi Chuong Trinh
Ban chon : 3
Xoa Bo Hang Hoa
Nhap Ma Hang Can Xoa: H8
Danh Muc Cac Chuc Nang
1.Them Hang Hoa
2.Sua Hang Hoa
3.Xoa Hang Hoa
4.Hien Thi Danh Sach
0.Thoat Khoi Chuong Trinh
Ban chon : 4
Danh Muc Hang Hoa
Mahang : Dien thoai    Tenhang : Iphone 6      DVT : chiec
Mahang : H1          Tenhang : Laptop DVT : chiec
Mahang : h10         Tenhang : Ipad   DVT : chiec
Mahang : H2          Tenhang : Tivi    DVT : Chiec
Mahang : H3          Tenhang : Gao     DVT : kg
Mahang : H4          Tenhang : Xe dap DVT : chiec
Mahang : H5          Tenhang : Dau goi   DVT : chai
Mahang : H6          Tenhang : Quan áo   DVT : bo

Danh Muc Cac Chuc Nang
1.Them Hang Hoa
2.Sua Hang Hoa
3.Xoa Hang Hoa
4.Hien Thi Danh Sach
0.Thoat Khoi Chuong Trinh
Ban chon : 0

```

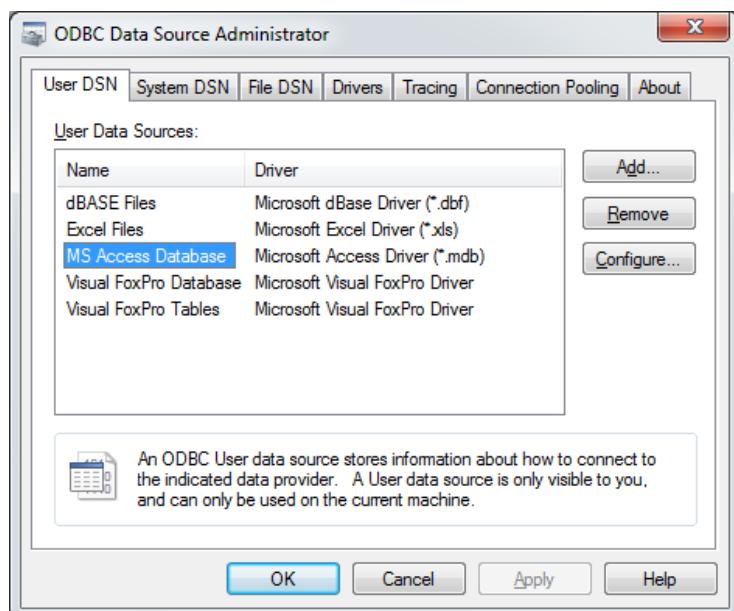
Process completed.

5.2.2. Bài tập tự giải

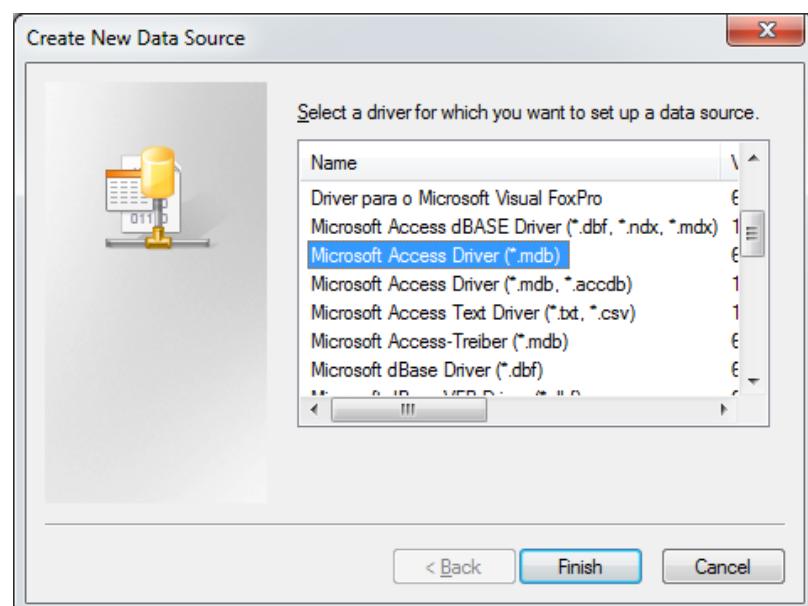
Bài 5.4. Làm lại bài tập 5.1 với cơ sở dữ liệu được tạo trong MS Access

a. Hướng dẫn:

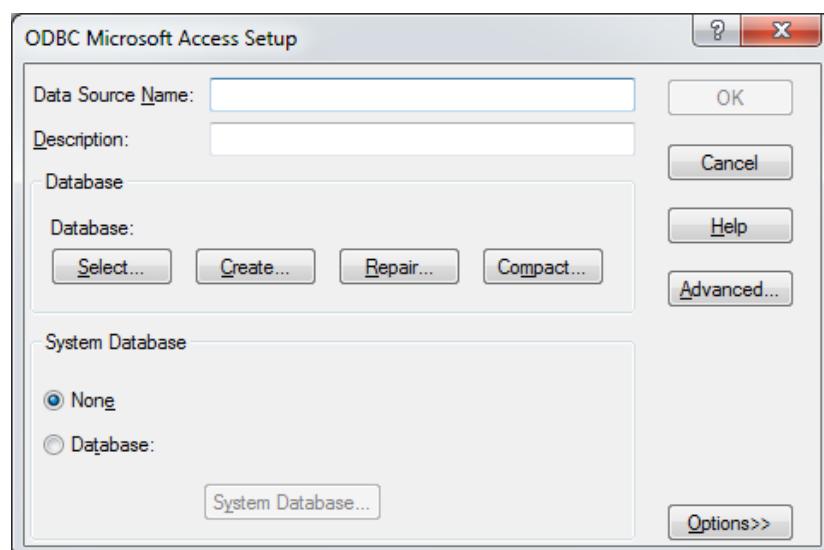
- Tạo Data base như yêu cầu của đề bài
- Cấu hình trong Control Panel như sau: Control Panel\Aministrative Tools\Daata Sources (ODBC), xuất hiện hộp thoại:



Chọn MS Access Database, chọn Add, xuất hiện



chọn Microsoft Access Driver (*.mdb), chọn Finish



Trong Data Source Name gõ CSDL.

Chọn Select, sau đó chỉ đường dẫn đến CSDL.MDB, chọn OK

* Copy cơ sở dữ liệu CSDL.MDB vào thư mục CLASSES của ứng dụng

b. Chương trình mẫu

```
//Phần đầu giống như chương trình bài 5.1
class SinhVien { };
class BangSV { };
public class qlSinhvien //Có sự thay đổi về cách kết nối
{
    public static void main(String[] args)
    {
        Connection cnn=null;
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            File f=new File("C:\\\\users\\\\Tr\\\\Desktop\\\\csdl.mdb");
            cnn=DriverManager.getConnection("jdbc:odbc: DRIVER= Microsoft Access
Driver (*.mdb);DBQ=" + f.getCanonicalFile());
            BangSV tbSV = new BangSV(cnn);
            tbSV.hienthi();
            cnn.close();
        } catch (Exception evt)
        {
            System.out.println("Loi ket noi CSDL"+evt.getMessage());
        }
    }
}
```

Bài 5.5. Làm lại bài tập 5.2 với cơ sở dữ liệu được tạo trong MS Access

a. Hướng dẫn:

```
//Phần đầu giống bài 5.2
class Phim { };
class MoBang { };
class qlPhim //Phần kết nối có thay đổi
{
    public static void main(String[] args)
    {
        Connection cnn;
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            File fl=new File("csdl.mdb");
            cnn=DriverManager.getConnection("jdbc:odbc:DRIVER = Microsoft Access
Driver (*.mdb);DBQ=" + fl.getCanonicalFile());
            MoBang tbPhim=new MoBang(cnn);
            tbPhim.hienthi();
            cnn.close();
        } catch (Exception evt)
        {
            System.out.println("Loi ket noi : " + evt.toString());
        }
    }
}
```

Bài 5.6. Làm lại bài tập 5.3. với cơ sở dữ liệu được tạo trong MS Access

Bài 5.7. Tạo cơ sở dữ liệu Quản lý sinh viên có table SINHVIEN chứa các thông tin về sinh viên như sau:

Column name	Data Type	Description
MaSV	Nchar (10)	Mã sinh viên, khoá chính
HoTen	Nchar (20)	Họ và tên sinh viên
NgaySinh	Date	Ngày sinh
GioiTinh	Nchar (4)	Giới tính
MaKH	NChar (10)	Mã khoa
HocBong	Numeric(8,0)	Học bổng của sinh viên

1. Hãy thực hiện các chức năng: Thêm, sửa, xóa các bản ghi (trong môi trường Console)
2. Hãy tạo giao diện để quản lý sinh viên như sau:

Student Information

Mã sinh viên	<input type="text" value="SV2"/>	Giới tính	<input type="text" value="Nu"/>	<input type="button" value="Add"/>
Họ và tên	<input type="text" value="Giang Thi Phuong"/>	MaKH	<input type="text" value="AV"/>	<input type="button" value="Modify"/>
Ngày sinh	<input type="text" value="3/1/1979"/>	Học bổng	<input type="text" value="200"/>	<input type="button" value="Delete"/>

- Thực hiện các chức năng thêm mới (Add), cập nhật (Modify), xóa (Delete) bản ghi.

CHƯƠNG 6: BÀI TẬP TỔNG HỢP

6.1. CÂU HỎI TRẮC NGHIỆM

Câu 1: Một chương trình JAVA có thể chạy trên bất kỳ hệ thống máy tính nào có_____?

- A. Hệ máy Intel
- B. Trình biên dịch Java
- C. Hệ điều hành Linux
- D. Máy ảo Java

Câu 2 : Package là tập hợp của:

- A. Lớp và Interface
- B. Lớp
- C. Interface
- D. Các công cụ biên dịch

Câu 3 : Phát biểu nào sau đây SAI về từ khóa static?

- A. Một thuộc tính static được dùng chung bởi tất cả các đối tượng thuộc lớp đó
- B. Một phương thức static có thể được gọi mà không cần khởi tạo đối tượng
- C. Từ khóa static không dùng trong khai báo lớp
- D. Một phương thức static có thể sử dụng tất cả các biến trong lớp

Câu 4 : Đoạn mã sau sai chỗ nào?

```
public class Question {  
    public static void main(String[] args) {  
        Boolean b = new Boolean("TRUE");  
        if (b){  
            for (Integer i=0;i<10 ;i++ ){  
                System.out.println(i);  
            }  
        }  
    }  
}
```

- A. (1) Đoạn mã không có lỗi
- B. (2) Điều kiện của câu lệnh if phải có kiểu boolean thay vì Boolean
- C. (3) Chỉ số của câu lệnh for là int thay vì Integer
- D. Câu (2) và (3) đúng

Câu 5 : Chọn khai báo đúng cho phương thức main

- A. public static void main()
- B. public void main(String[] arg)
- C. public static void main(String[] args)
- D. public static int main(String[] arg)

Câu 6 : Chọn thứ tự đúng của các thành phần trong một tập tin nguồn

- A. câu lệnh import, khai báo package, khai báo lớp
- B. Khai báo package đầu tiên; thứ tự câu lệnh import và khai báo lớp là tùy ý
- C. Khai báo package, câu lệnh import, khai báo lớp
- D. Câu lệnh import trước tiên; thứ tự của khai báo package và khai báo lớp là tùy ý

Câu 7 : Cho biết kết quả sau khi biên dịch và thi hành chương trình sau :

```
01: public class Conditional {
```

```

02: public static void main(String[] args) {
03:     int x = 4;
04:     System.out.println("value is " + ((x>4)?99.99:9));
05: }
06: };
A. Kết quả là : value is 99.99
B. Kết quả là : value is 9
C. Kết quả là : value is 9.0
D. Lỗi biên dịch tại dòng 4

```

Câu 8 : Chọn phát biểu đúng cho chương trình sau :

```

01: class StaticStuff
02: {
03:     static int x = 10;
04:
05:     static {x+=5;}
06:
07:     public static void main(String[] args)
08:     {
09:         System.out.println("x = " + x);
10:     }
11:
12:     static {x/=5;}
13: };

```

- A. Lỗi biên dịch tại dòng 5 và 12 bởi vì thiếu tên phương thức và kiểu trả về
- B. Chương trình chạy và cho kết quả x = 10
- C. Chương trình chạy và cho kết quả x = 15
- D. Chương trình chạy và cho kết quả x = 3

Câu 9 : Cho khôi lệnh như sau, kết quả in ra màn hình sẽ là

```

int k,n;
for(k=1;k<30;k++)
    if((k%3==0)||(k%7==0))
        System.out.print(k + "\t");
A. 21
B. 3,6,7,9,12,14,15,18,21,24,27 và 28
C. 3 và 7
D. Không in ra gì cả

```

Câu 10 : chọn phát biểu đúng cho chương trình sau :

```

01: class HasStatic {
02:     private static int x = 100;
03:     public static void main(String[] args) {
04:         HasStatic hs1 = new HasStatic();
05:         hs1.x++;
06:         HasStatic hs2 = new HasStatic();
07:         hs2.x++;
08:         hs1 = new HasStatic();
09:         hs1.x++;
10:         HasStatic.x++;
11:         System.out.println("x = " + x);
12:     }

```

- 13: };
A. Chương trình chạy và cho kết quả x = 102
B. Chương trình chạy và cho kết quả x = 103
C. Chương trình chạy và cho kết quả x = 104
D. Tất cả các câu đều sai

Câu 11 : Cho đoạn chương trình sau :

```
01: class Cruncher{  

02:     void cruncher(int i){  

03:         System.out.println("int version");  

04:     }  

05:     void crunch(String s){  

06:         System.out.println("String version");  

07:     }  

08:  

09:     public static void main(String []args){  

10:         Cruncher crun = new Cruncher();  

11:         char ch = 'p';  

12:         crun.crunch(ch);  

13:     }  

14: }
```

- A. Dòng 5 sẽ không biên dịch vì phương thức trả về kiểu void không để overide
B. Dòng 12 sẽ không biên dịch vì không có phiên bản nào của phương thức crunch() nhận vào tham số kiểu char
C. Đoạn mã biên dịch được nhưng sẽ phát sinh Exception ở dòng 12
D. Chương trình chạy và in kết quả int version

Câu 12 : Chọn vòng lặp đúng

- A. while (int i<7) {i++; System.out.println("i is " + i);};
B. int i = 3; while (i){ System.out.println("i is " + i);};
C. int j = 0; for (int k = 0; j+k != 10; j++,k++)
 { System.out.println("j is " + j + " k is " + k);};
D. int j = 0; do{
 System.out.println("j is " + j++);
 if (j == 3) { continue loop;}
}while (j<10);

Câu 13 : Cho biết kết xuất của đoạn mã sau :

```
01: int x = 0, y = 4, z = 5;  

02: if (x>2){  

03:     if (y<5){  

04:         System.out.println("message one");  

05:     }  

06:     else {  

07:         System.out.println("message two");  

08:     }  

09: }  

10: else if (z>5){  

11:     System.out.println("message three");  

12: }  

13: else{  

14:     System.out.println("message four");
```

- 15: }
A. message one
B. message two
C. message three
D. message four

Câu 14 : Chọn phát biểu đúng cho đoạn mã sau :

```
01:int j = 2;  
02:switch (j){  
03: case 2:  
04:     System.out.println("value is two");  
05: case 2+1:  
06:     System.out.println("value is three");  
07:     break;  
08: default:  
09:     System.out.println("value is " + j);  
10:    break;  
11:}
```

- A. Đoạn mã không hợp lệ bởi biểu thức ở dòng 5
B. Biến j trong cấu trúc switch() có thể là một trong các kiểu : byte, short, int hoặc long
C. Kết xuất của chương trình chỉ là dòng : value is two
D. Kết xuất của chương trình chỉ là dòng : value is two và value is three

Câu 15: Nếu muốn sinh ra một số nguyên ngẫu nhiên từ 1->6, biểu thức nào sau đây được dùng?

- A. (int)(Math.random()*6)+1
B. (int)(Math.random()*6)
C. (int)(Math.random()+6)
D. Math.random()*6

Câu 16: Muốn khai báo hằng số PI trong Java thì dùng câu lệnh nào sau đây?

- A. final constant double PI = 3.14159;
B. final PI = 3.14159;
C. final double PI = 3.14159;
D. constant double PI = 3.14159;

Câu 17: Lớp String thuộc package

- A. java.util
B. java.lang
C. javax.util
D. java.string

Câu 18: Để chuyển một chuỗi chứa giá trị nguyên (String) sang một số nguyên (int) thì sẽ dùng phương thức nào sau đây?

- A. toInteger()
B. parseInt()
C. toInt()
D. parseInteger()

Câu 19: Trong Java, kiểu dữ liệu nào không phải là kiểu dữ liệu tham chiếu?

- A. float
B. Array
C. String

D. BufferedReader

Câu 20: Trong Java, để đọc dữ liệu từ bàn phím vào dùng lớp nào sau đây?

- A. Scanner
- B. Scaner
- C. ScanReader
- D. ScanerReader

Câu 21: Trong Java, các câu lệnh xử lý Exception sẽ được đặt trong khối nào?

- A. try
- B. catch
- C. finally
- D. Exception

Câu 22: Để chuyển đổi kiểu dữ liệu số nguyên (int) thành một chuỗi chứa giá trị nguyên (String) thì sử dụng phương thức nào?

- A. valueOf()
- B. convert()
- C. equals()
- D. CompareTo()

Câu 23: Khai báo nào sau đây sẽ không gây ra lỗi biên dịch?

- A. abstract int i;
- B. native MyMethod();
- C. default class MyClass {};
- D. abstract final class MyClass {};

Câu 24: Cho đoạn mã sau :

```
try{  
    int i = Integer.parseInt("one");  
}
```

Khối lệnh trên sẽ phát sinh Exception gì?

- A. ClassCastException
- B. IllegalStateException
- C. NumberFormatException
- D. ArrayIndexOutOfBoundsException

Câu 25: Cho đoạn mã sau :

```
StringBuffer sbuf = new StringBuffer("Hello, ");  
sbu.append("Truongbt");  
System.out.println(sbuf);
```

Kết quả đoạn mã trên là ?

- A. Hello,
- B. Truongbt
- C. Hello, Truongbt
- D. Lỗi biên dịch

Câu 26: Để chuyển đổi một chuỗi số sang dạng số thực, ta sử dụng phương thức nào sau đây

- A. Integer.parseInt()
- B. Float.parseFloat()
- C. Double.parseDouble()
- D. Double.parseDouble()

Câu 27: Cho đoạn lệnh sau đây :

```
if("String".toString() == "String")
    System.out.println("Equal");
else
    System.out.println("Not Equal");
```

Cho biết kết quả khi thực thi chương trình ?

- A. Equal
- B. Not Equal
- C. Lỗi Biên Dịch
- D. Lỗi Thực Thi

Câu 28 : Cho chương trình sau :

```
public class TNH
{
    public void method(Object o)
    {
        System.out.println("Object Verion");
    }
    public void method(String s)
    {
        System.out.println("String Version");
    }
    public static void main(String args[])
    {
        TNH question = new TNH();
        question.method(null);
    }
}
```

Cho biết kết quả sau khi thực thi chương trình trên?

- A. Đoạn mã không biên dịch
- B. Biên dịch thành công và hiển thị kết quả "String Version"
- C. Biên dịch thành công và hiển thị kết quả "Object Verion"
- D. Đoạn mã xảy ra ngoại lệ Exception khi thực hiện

Câu 29: Câu lệnh khai báo biến : int i,j và float x,y; các phép gán dưới đây phép gán nào sai :

- A. x = Float.parseFloat("0.123");
- B. i = Integer.parseInt("123");
- C. y = Float.parseFloat("123");
- D. j = Integer.parseInt("123");

Câu 30: Từ khoá nào khi thực thi sẽ tham khảo đến bản thân lớp chứa nó?

- A. this
- B. super
- C. abstract
- D. final

Câu 31: Cho biết kết quả sau khi thực thi đoạn chương trình sau :

```
class C2 {
    public static void main(String args[]) {
        int k = 50;
        IncValue inc = new IncValue();
```

```

        System.out.println("result = " + inc.increment(k));
        System.out.println("k = " + k);
    }
}
class IncValue {
    public int increment(int v) {
        int z = v + 10;
        v = v - 3;
        return z;
    }
}
A. result = 60 và k = 50
B. result = 50 và k = 50
C. result = 47 và k = 50
D. result = 60 và k = 47

```

Câu 32: Cho đoạn chương trình sau :

```

class Value
{
    public int i = 15;
}
public class Test
{
    public static void main( String args[])
    {
        Test t = new Test();
        t.first();
    }
    public void first()
    {
        int i = 5;
        Value v = new Value();
        v.i = 25;
        second(v,i);
        System.out.print(" " + v.i);
    }
    public void second( Value v, int i)
    {
        i = 0;
        v.i = 20;
        Value val = new Value();
        v = val;
        System.out.print( v.i + " " + i );
    }
}

```

Cho biết kết quả in lên màn hình sau khi thực thi?

- A. 0 15 20
- B. 15 0 20
- C. 20 0 15
- D. 20 15 0

Câu 33: Giá trị mặc định cho một biến kiểu char là?

- A. 0
- B. 1
- C. Nothing
- D. null

Câu 34: Cho đoạn chương trình sau:

```
public class vd111 {
    public static void main(String[] args) {
        int a=7,b=4;
        double x;
        x=-(double)a/b;
        System.out.print("Nghiem la: " + x);
    }
}
```

Kết quả khi chạy chương trình là:

- A. Nghịch là: -1.4
- B. Nghịch là: -1.0
- C. Nghịch là: -1.75
- D. Chương trình báo lỗi

Câu 35: Phương thức next() trong lớp Scanner có tác dụng?

- A. Đọc 1 chuỗi ký tự từ bàn phím, kể cả dấu cách
- B. Lớp Scanner không có phương thức này
- C. Đọc 1 giá trị số nguyên từ bàn phím
- D. Đọc 1 giá trị số thực từ bàn phím

Câu 36: Lệnh khai báo

```
Scanner in = new Scanner(System.in);
```

có tác dụng gì?

- A. Khai báo đối tượng Scanner kết nối tới dữ liệu
- B. Khai báo đối tượng kiểu Scanner để in dữ liệu
- C. Khai báo đối tượng tên là in, kết nối với stream output chuẩn
- D. Khai báo đối tượng tên là in, kết nối với stream input chuẩn

Câu 37: Phương thức nextLine() trong lớp Scanner có tác dụng?

- A. Đọc 1 chuỗi ký tự từ bàn phím, không kể dấu cách
- B. Đọc 1 chuỗi ký tự từ bàn phím, kể cả dấu cách
- C. Đọc 1 giá trị số nguyên từ bàn phím
- D. Đọc 1 giá trị số thực từ bàn phím

Câu 38: Cho đoạn chương trình sau:

```
public class vd222 {
    public static void main(String[] args) {
        double a=7,b=4;
        int x=(a<b)?a:b;
        System.out.print("Nghiem la: " + x);
    }
}
```

Kết quả khi chạy chương trình là:

- A. Nghịch là: 7
- B. Nghịch là: 4.0
- C. Nghịch là: 4
- D. Nghịch là: 7.0

Câu 39: Để kiểm tra 2 chuỗi có bằng nhau hay không, sử dụng phương thức nào?

- A. string1== string2
- B. string1 = string2
- C. Không có đáp án đúng
- D. string1.equal(string2)

Câu 40: Biểu thức nào sau đây hợp lệ

- A. String x = "Hello"; int y = 9; x+=y;
- B. String x = "Hello"; int y = 9; x= x + y;
- C. String x = null; int y = (x != null) && (x.length>0)?x.length():0;
- D. Tất cả các câu đều đúng

Câu 41: Cho đoạn mã sau, kết quả của đoạn mã là?

```
class StringBufferTest {  
    public static void main(String []args) {  
        1.         StringBuffer result = new StringBuffer();  
        2.         StringBuffer s = null;  
        3.         result.append(s);  
        4.         result.insert(0,"123");  
        5.         System.out.println(result);  
    }  
}
```

- A. Lỗi biên dịch ở dòng số 3
- B. Lỗi biên dịch ở dòng số 4
- C. Lỗi biên dịch ở dòng số 1
- D. In ra kết quả "123null"

Câu 42: Đoạn mã nào sau đây in ra màn hình chữ "Equal":

- A. (1) int x = 100; float y = 100.0F; if (x==y){ System.out.println("Equal");}
- B. (2) Integer x = new Integer(100); Integer y = new Integer(100); if (x==y){ System.out.println("Equal");}
- C. (3) String x = "100"; String y = "100"; if (x==y){ System.out.println("Equal");}
- D. Câu (1) và (3) đúng

Câu 43: Cho biết kết quả sau khi thi hành chương trình sau :

```
01: public class Short {  
02:     public static void main(String[] args) {  
03:         StringBuffer s = new StringBuffer("Hello");  
04:         if ((s.length()>5) &&  
05:             (s.append("there").equals("False")))  
06:             ;//do nothing  
07:         System.out.println("value is : " + s);  
08:     }  
09: };
```

- A. Giá trị xuất là Hello
- B. Lỗi biên dịch tại dòng 4 và 5
- C. Không có giá trị xuất
- D. Thông báo NullPointerException

Câu 44: Lệnh charAt(n) có tác dụng gì?

- A. Tìm kiếm ký tự thứ n

- B. Trả về ký tự có vị trí chỉ mục n
- C. Trả về ký tự thứ n
- D. Trả về ký tự thứ n-1

Câu 45: Khai báo nào sau đây là khai báo hợp lệ của một lớp abstract?

- A. abstract class Vehicle { abstract void display(); }
- B. class Vehicle { abstract void display(); }
- C. abstract class Vehicle { abstract void display(System.out.println("Car")); }
- D. abstract Vehicle { abstract void display(); }

Câu 46: Trong Java, đa kế thừa thực hiện bằng cách nào?

- A. Thừa kế từ 2 hay nhiều lớp
- B. Thừa kế từ 1 Interface và 1 hay nhiều lớp
- C. Thừa kế từ 1 lớp và 1 hay nhiều Interface
- D. Tất cả đều đúng

Câu 47: Chọn phát biểu đúng

- A. Lớp Class là lớp cha của lớp Object
- B. Lớp Object là một lớp final
- C. Mọi lớp đều kế thừa trực tiếp hoặc gián tiếp từ lớp Object
- D. Tất cả các câu trên đều sai

Câu 48: Hãy cho biết kết xuất của đoạn chương trình sau :

```
public class S1 {
    public static void main(String[] args) {
        new S2();
    }
    S1(){
        System.out.print("S1");
    }
}
class S2 extends S1{
    S2(){
        System.out.print("S2");
    }
}
```

- A. S1
- B. S2
- C. S1S2
- D. S2S1

Câu 49: Cho biết kết xuất của đoạn chương trình sau:

```
class Question {
    String s = "Outer";
    public static void main(String[] args) {
        S2 s2 = new S2();
        s2.display();
    }
};
class S1{
    String s = "S1";
    void display(){
```

```

        System.out.print(s);
    }
}
class S2 extends S1{
    String s = "S2";
}
A. S1
B. S2
C. null
D. S1S2

```

Câu 50: Cho đoạn mã sau

```

class Q6 {
    public static void main(String[] args) {
        Holder h = new Holder();
        h.held = 100;
        h.bump(h);
        System.out.println(h.held);
    }
};
class Holder{
    public int held;
    public void bump(Holder theHolder){
        theHolder.held++;
    }
}

```

Giá trị in ra của câu lệnh dòng thứ 6 là:

- A. 0
- B. 1
- C. 100
- D. 101

Câu 51: Cho đoạn mã sau:

```

class Q7 {
    public static void main(String[] args) {
        double d = 12.3;
        Decrement dec = new Decrement();
        dec.decrement(d);
        System.out.println(d);
    }
};
class Decrement{
    public void decrement(double decMe){
        decMe = decMe - 1.0;
    }
}

```

Giá trị in ra của câu lệnh dòng thứ 6 là :

- A. 0.0
- B. -1.0
- C. 12.3
- D. 11.3

Câu 52: Chọn cách sửa ít nhất để đoạn mã sau biên dịch đúng:

```
01: final class Aaa
02: {
03:     int xxx;
04:     void yyy(){xxx = 1;}
05: }
06:
07:
08: class Bbb
09: {
10:     final Aaa finalRef = new Aaa();
11:
12:     final void yyy()
13:     {
14:         System.out.println("In method yyy()");
15:         finalRef.xxx = 12345;
16:     }
17: };
```

- A. Xóa từ final ở dòng 1
- B. Xóa từ final ở dòng 10
- C. Xóa từ final ở dòng 1 và 10
- D. Không cần phải chỉnh sửa gì

Câu 53: Cho đoạn mã sau:

```
01: class SuperDupper
02: {
03:     void aMethod(){}
04: }
05:
06: class Sub extends SuperDupper
07: {
08:     void aMethod(){}
09: }
```

Hãy chọn từ khóa chỉ phạm vi hợp lệ đứng trước aMethod() ở dòng 8

- A. default
- B. protected
- C. public
- D. Tất cả các câu đều đúng

Câu 54: Chọn phát biểu đúng cho lớp Bird trên và lớp Parrot sau:

```
package abcde;
public class Bird{
protected static int referenceCount = 0;
public Bird(){referenceCount++;}
protected void fly(){...}
static int getRefCount(){return referenceCount;}
};

01: package abcde;
02:
03: class Parrot extends abcde.Bird{
04:     public void fly(){
05:         //
```

```

06: }
07: public int getRefCount(){
08:     return referenceCount;
09: }
10: }
```

- A. Lỗi biên dịch ở dòng 4 tập tin Parrot.java vì phương thức fly() là protected trong lớp cha và lớp Bird và Parrot nằm trong cùng package.
- B. Lỗi biên dịch ở dòng 4 tập tin Parrot.java vì phương thức fly() là protected trong lớp cha và public trong lớp con
- C. Lỗi biên dịch ở dòng 7 tập tin Parrot.java vì phương thức getRefCount() là static trong lớp cha.
- D. Chương trình biên dịch thành công nhưng sẽ phát sinh Exception khi chạy nếu phương thức fly() của lớp Parrot không được gọi.

Câu 55: Chọn phát biểu đúng cho lớp Bird trên và lớp Nightingale sau:

```

package abcde;
public class Bird{
    protected static int referenceCount = 0;
    public Bird(){referenceCount++;}
    protected void fly(){...}
    static int getRefCount(){return referenceCount;}
};

01: package singers;
02:
03: class Nightingale extends abcde.Bird{
04:     Nightingale(){referenceCount++;}
05:
06:     public static void main(String[] args) {
07:         System.out.print("Before : " + referenceCount);
08:         Nightingale florence = new Nightingale();
09:         System.out.print("After : " + referenceCount);
10:         florence.fly();
11:     }
12: }
```

- A. Kết quả trên màn hình là : Before : 0 After : 2
- B. Kết quả trên màn hình là : Before : 0 After : 1
- C. Lỗi biên dịch ở dòng 4 của lớp Nightingale vì không thể override thành viên static
- D. Lỗi biên dịch ở dòng 10 của lớp Nightingale vì phương thức fly() là protected trong lớp cha.

Câu 56: Phát biểu nào sau đây đúng nhất về 2 phương thức overloading:

- A. Thuộc hai lớp khác nhau
- B. Thuộc hai lớp có quan hệ kế thừa
- C. Cùng tên khác danh sách đối số
- D. Cùng tên khác kiểu trả về

Câu 57 : Phát biểu nào sau đây đúng nhất về 2 phương thức overriding:

- A. Cùng tên khác kiểu trả về
- B. Thuộc hai lớp có quan hệ kế thừa
- C. Cùng tên khác danh sách đối số
- D. Thuộc cùng một lớp

Câu 58: Để tham chiếu đến phương thức trong lớp cơ sở, ta dùng từ khóa nào sau đây:

- A. subclass
- B. this
- C. parent
- D. super

Câu 59: Phát biểu nào sau đây đúng:

- A. Một lớp có thể thừa từ nhiều lớp khác bằng cách dùng từ khoá extends
- B. Một lớp có thể thừa kế từ nhiều interface khác bằng cách dùng từ khoá extends
- C. Một interface có thể thừa kế từ nhiều interface bằng cách dùng từ khoá extends
- D. Tất cả các phát biểu đều sai

Câu 60: Phát biểu nào sau đây là đúng:

- A. Một phương thức final thì không thể Override
- B. Một lớp có chứa phương thức final sẽ không cho phép thừa kế
- C. Một biến final chỉ có thể khai báo trong một lớp final
- D. Một biến final tương tự như một biến static

Câu 61: Cho class như sau:

```

1: public interface Comparable {
2: int compareTo(Object other);
3: }
4: class Employee implements Comparable {
5: public int compareTo(Object other) {
6: Employee other = (Employee) otherObject;
7: return 0;
8: }
9: }
```

Chương trình trên sai ở dòng nào

- A. Không sai ở dòng nào cả
- B. 2, 4
- C. 5
- D. 8

Câu 62: Phương thức init() trong Applet có mục đích gì:

- A. Phương thức được kích hoạt đầu tiên khi applet khởi động. init() chỉ chạy một lần duy nhất vào lúc bắt đầu chương trình.
- B. Phương thức khởi tạo của applet.
- C. Phương thức chạy song song với applet khi bắt đầu chương trình.
- D. Không có đáp án đúng.

Câu 63: Đối tượng nào cho phép xử lý thao tác chuột trong applet:

- A. ActionListener
- B. Không có đáp án đúng.
- C. MouseMotionListener
- D. Cả 3 đáp án.

Câu 64: Đối tượng nào cho phép xử lý tương tác hành động giữa người dùng và applet:

- A. ActionListener
- B. MouseListener
- C. MouseMotionListener
- D. Không có đáp án đúng.

Câu 66: Sau khi tạo một frame bằng câu lệnh Frame f = new Frame() và tạo menu bar bằng câu lệnh MenuBar mb = new MenuBar(), làm thế nào để gắnMenuBar tên mb vào f:

- A. f.add(mb)
- B. f.setMenu(mb)
- C. f.addMenu(mb)
- D. f.setMenuBar(mb)

Câu 66: Đôi tượng nào bên dưới có thể chứa một menubar (chọn những câu đúng) :

- A. Panel
- B. ScrollPane
- C. Frame
- D. Menu

Câu 67: Bạn có thể tạo ra cửa sổ chính của ứng dụng bằng cách gọi:

Frame f = new Frame("Main Frame");

Nhưng khi bạn chạy chương trình thì Frame không hiển thị. Dòng nào bên dưới sẽ làm hiển thị Frame

- A. f.setSize(320,200);
- B. f.setBounds(10,10,500,400);
- C. f.setForeground(Color.white);
- D. f.setVisible(true);

Câu 68: Khi có nhiều component được gắn các bộ lắng nghe của cùng một loại sự kiện thì các component nào sẽ nhận được sự kiện đầu tiên:

- A. Component đầu tiên được gắn bộ lắng nghe
- B. Component cuối cùng được gắn bộ lắng nghe
- C. Không thể xác định component nào sẽ nhận trước
- D. Không thể có nhiều hơn một bộ lắng nghe cho cùng một loại sự kiện

Câu 69: Chọn phát biểu đúng:

- A. MouseListener interface định nghĩa các phương thức để xử lý sự kiện nhấp chuột
- B. MouseMotionListener interface định nghĩa các phương thức để xử lý sự kiện nhấp chuột
- C. MouseClickListener interface định nghĩa các phương thức để xử lý sự kiện nhấp chuột
- D. Tất cả các câu trên đều đúng

Câu 70: Giả sử chúng ta có thể hiện e của bộ lắng nghe sự kiện TextEvent và thể hiện t của lớp TextArea. Cho biết cách để gắn bộ lắng nghe e vào t:

- A. t.addTextListener(e)
- B. e.addTextListener(t)
- C. addTextListener(e,t)
- D. addTextListener(t,e)

Câu 71: Cho một component comp và một container cont có kiểu trình bày là BorderLayout. Cho biết cách để gắn comp vào vị trí đầu của cont:

- A. addTop(cont,comp)
- B. comp.add("North",cont)
- C. cont.addTop(comp)
- D. cont.add(comp,BorderLayout.NORTH)

Câu 72: Cho một component comp và một container cont có kiểu trình bày là FlowLayout. Cho biết cách để gắn comp vào cont:

- A. cont.add(comp)
- B. comp.add(cont)

- C. cont.addComponent(comp)
- D. cont.addAllComponents()

Câu 73: Chọn phương thức dùng để xác định các trình bày của một khung chứa:

- A. startLayout()
- B. initLayout()
- C. layoutContainer()
- D. setLayout()

Câu 74: Chọn kiểu trình bày để đặt các component trên khung chứa dưới dạng bảng:

- A. CardLayout
- B. BorderLayout
- C. GridLayout
- D. FlowLayout

Câu 75: Chọn phát biểu đúng:

- A. Lớp CheckboxGroup dùng để định nghĩa các RadioButtons
- B. Lớp RadioGroup dùng để định nghĩa các RadioButtons
- C. tất cả các câu trên đều đúng
- D. Tất cả các câu trên đều sai

Câu 76: Chọn câu lệnh để tạo TextArea có 10 dòng và 20 cột:

- A. new TextArea(10,20)
- B. new TextArea(20,10)
- C. new TextArea(200)
- D. Tất cả các câu trên đều sai

Câu 77: Chọn câu lệnh để tạo ra một danh sách gồm năm mục chọn và cho phép thực hiện chọn nhiều mục cùng lúc:

- A. new List(5,true)
- B. new List(true,5)
- C. new List(5,false)
- D. new List(false,5)

Câu 78: Chọn phương thức để hiển thị Frame lên màn hình:

- A. setVisible()
- B. display
- C. displayFrame()
- D. Tất cả các câu trên đều sai

Câu 79: Bộ quản lý trình bày mặc định cho một khung chứa kiểu Panel là:

- A. FlowLayout
- B. BorderLayout
- C. GridLayout
- D. GridBagLayout

Câu 80: Cho biết kết quả sau khi dịch và thực thi đoạn chương trình sau:

```
1: import java.awt.*;  
2:  
3: class Test extends Frame{  
4:     Test(){  
5:         setSize(300,300);  
6:         Button b = new Button("Apply");
```

```

7:     add(b);
8: }
9:
10: public static void main(String[] args){
11:     Test f = new Test();
12:     f.setVisible(true);
13: }
14:

```

- A. Có lỗi biên dịch tại dòng 11 bởi vì constructor ở dòng 4 không được khai báo public
B. Chương trình biên dịch thành công nhưng sẽ ném ra Exception khi thực thi câu lệnh ở dòng thứ 7
C. Chương trình hiển thị một nút nhấn (button) sử dụng font chữ mặc định cho nhãn button. Button chỉ đủ lớn để bao quanh nhãn của nó
D. Chương trình hiển thị nút nhấn (button) sử dụng font chữ mặc định cho nhãn nút. Nút nhấn sẽ chiếm tất cả vùng hiển thị của frame.

Bảng 6.1: Đáp án câu hỏi trắc nghiệm

Câu	Đáp án								
1	D	17	C	33	D	49	A	65	D
2	A	18	B	34	C	50	D	66	C
3	D	19	B	35	A	51	C	67	D
4	D	20	A	36	D	52	A	68	C
5	C	21	A	37	B	53	D	69	A
6	C	22	B	38	B	54	C	70	A
7	C	23	A	39	A	55	A	71	D
8	D	24	D	40	D	56	C	72	A
9	B	25	C	41	D	57	B	73	D
10	C	26	D	42	D	58	D	74	C
11	D	27	A	43	A	59	C	75	A
12	C	28	B	44	B	60	A	76	A
13	D	29	B	45	A	61	A	77	A
14	D	30	A	46	C	62	A	78	A
15	A	31	A	47	C	63	A	79	A
16		32	B	48	C	64	A	80	D

6.2. MỘT SỐ ĐỀ THI MẪU

Thời gian làm bài thi: 60 phút

ĐỀ SỐ 01

Câu 1-01 (8 điểm) – Quản lý nhân viên

- a. (1 điểm) Hãy định nghĩa class Person với các thuộc tính: Họ tên, ngày sinh, địa chỉ, giới tính và các phương thức khởi tạo, hiển thị thông tin.
- b. (1 điểm) Định nghĩa class NhanVien kế thừa từ Person, có thêm các thuộc tính: phòng ban (nhận 1 trong các giá trị: Thiết bị, nhân sự, R&D, Kinh doanh), hệ số lương, thâm niên, lương cơ bản và các phương thức khởi tạo, tính lương thực lĩnh, hiển thị thông tin cá nhân

Trong đó: $lương\ thực\ lĩnh = Lương\ cơ\ bản * hệ\ số\ (1 + thâm\ niên/100)$

- c. (2 điểm) Tạo form cho phép nhập các thông tin nhân viên như định nghĩa ở trên.
- d. (2 điểm) Khi người dùng click nút Thêm nhân viên thì chương trình khởi tạo đối tượng nhân viên với các giá trị thuộc tính lấy từ form người dùng nhập và hiển thị thông tin lên màn hình console.
- e. (2 điểm) Khi người dùng click nút Lưu dữ liệu thì ghi toàn bộ thông tin vào file nhanvien.dat

Câu 3-01 (2 điểm)

Hãy cho biết kết quả (và lý do) của đoạn chương trình sau:

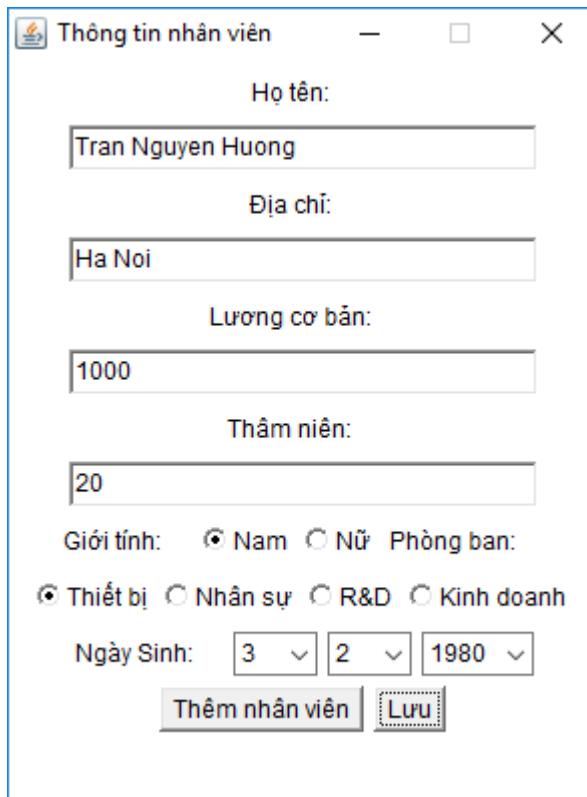
```
class Msg{  
    public Msg(){  
        System.out.println("Hello");}  
    public void sayGoodbye(){  
        System.out.println("Goodbye");}  
    }  
class TestMsg extends Msg{  
    public void sayGoodbye(){  
        System.out.println("Goodbye Java");}  
    }  
public class Demo {  
    public static void main(String args[]){  
        Msg m1 = new Msg();  
        Msg m2 = new TestMsg();  
  
        m1.sayGoodbye();  
        m2.sayGoodbye();  
  
    }  
}
```

Hướng dẫn giải đề 01

Câu 1: Quản lý nhân viên

1. Hướng dẫn

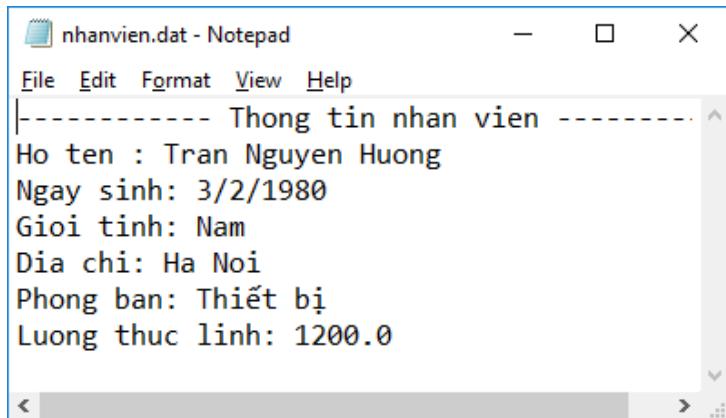
- Xây dựng lớp person, lưu vào file person.java
- Xây dựng lớp nhanvien kế thừa lớp person và lưu vào file nhanvien.java
- Xây dựng lớp form, lưu và file form.java. Giao diện thiết kế như sau:



Khi nhập dữ liệu vào form, nhấn nút **Thêm nhân viên** thì dữ liệu vừa nhập sẽ hiện thị lên màn hình console như sau:

```
-----
Họ và tên: Tran Nguyen Huong
Ngày sinh: 3/2/1980
Giới tính: Nam
Địa chỉ: Ha Noi
Phòng ban: Thiết bị
Lương thực lĩnh: 1200.0
-----
```

Nhấn nút **Lưu** thì dữ liệu trên form sẽ lưu trong file nhanvien.dat như sau:



- Tại hàm main, đặt kích thước cho frame và hiển thị frame.

2. Chương trình tham khảo

/* Tên project là DeThiSo1

```

    /* Lớp peron, file person.java
    package dethis01;
    public class person {
        protected String hoten, ngaysinh, gioitinh, diachi;
        protected person() {
            hoten = null;
            ngaysinh = null;
            gioitinh = null;
            diachi = null;
        }
        //Hàm tạo có tham số
        protected person(String hoten, String ngaysinh, String gioitinh,
        String diachi) {
            this.hoten = hoten;
            this.ngaysinh = ngaysinh;
            this.gioitinh = gioitinh;
            this.diachi = diachi;
        }
        protected void hienthi() {
            System.out.println("Họ và tên: " + hoten);
            System.out.println("Ngày sinh: " + ngaysinh);
            System.out.println("Giới tính: " + gioitinh);
            System.out.println("Địa chỉ: " + diachi);
        }
    }

```

/* Lớp nhanvien kế thừa lớp person, file nhanvien.java

```

    package dethis01;
    import java.io.FileNotFoundException;
    import java.io.FileOutputStream;
    import java.io.IOException;
    import java.io.PrintWriter;
    public class nhanvien extends person {
        //-----Khai báo thuộc tính-----
        protected String phongban;
        protected double hesoluong, luongcoban, thamnien;

```

```

//-----Phương thức khởi tạo-----
//Hàm tạo không tham số
protected nhanvien() {
    phongban = null;
    hesoluong = 0;
    luongcoban = 0;
    thamnien = 0;
}
//Hàm tạo có tham số
protected nhanvien(String hoten, String ngaysinh, String gioitinh,
String diachi, String phongban, double luongcoban, double thamnien) {
    //dùng super để gọi hàm tạo có tham số ở lớp cha (Person)
    super(hoten, ngaysinh, gioitinh, diachi);
    this.phongban = phongban;
    this.luongcoban = luongcoban;
    this.thamnien = thamnien;
}
//-----Phương thức tính lương thực lĩnh-----
protected double luongthuclinh() {
    double luongthuclinh;
    hesoluong = 1 + (thamnien / 100);
    luongthuclinh = luongcoban * hesoluong;
    return luongthuclinh;
}
//-----Phương thức hiển thị thông tin-----
@Override
protected void hienthi() {
    System.out.println("-----");
    //gọi phương thức hiển thị của lớp cha
    //dùng super để phân biệt phương thức của lớp cha với phương
    //thức của lớp con trong trường hợp lớp con ghi đè(override) phương thức
    //lớp cha
    super.hienthi();
    System.out.println("Phòng ban: " + phongban);
    System.out.println("Lương thực lĩnh: " + luongthuclinh());
    System.out.println("-----");
}
//-----
protected void luuthongtin() throws FileNotFoundException,
IOException {
    FileOutputStream fos = new FileOutputStream("nhanvien.dat",
true);
    PrintWriter pw = new PrintWriter(fos);
    pw.println("----- Thông tin nhân viên -----");
    pw.println("Họ tên : " + hoten);
    pw.println("Ngày sinh: " + ngaysinh);
    pw.println("Giới tính: " + gioitinh);
    pw.println("Địa chỉ: " + diachi);
    pw.println("Phòng ban: " + phongban);
    pw.println("Lương thực lĩnh: " + luongthuclinh());
}

```

```

        System.out.println("Luu thanh cong !!!");
        pw.close();
        fos.flush();
        fos.close();
    }
}

/* Xây dựng lớp form để thiết kế giao diện và thực hiện chức năng lưu dữ liệu và
hiển thị dữ liệu lên màn hình console, file form.java

```

```

package dethiso1;
import java.awt.Button;
import java.awt.Checkbox;
import java.awt.CheckboxGroup;
import java.awt.Choice;
import java.awt.FlowLayout;
import java.awt.Frame;
import java.awt.Label;
import java.awt.TextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;
public class form extends Frame implements ActionListener {
    //Khởi tạo các nhãn ( dùng để giải thích cho ô trống phía sau nó
    - TextField cần điền gì vào )
    Label lb1 = new Label("Họ tên: ");
    Label lb2 = new Label("Địa chỉ: ");
    Label lb3 = new Label("Giới tính: ");
    Label lb4 = new Label("Ngày Sinh: ");
    Label lb5 = new Label("Phòng ban: ");
    Label lb6 = new Label("Lương cơ bản: ");
    Label lb7 = new Label("Thâm niên: ");
    // khởi tạo các ô trống để nhập 1 chuỗi (TextField)
    TextField tf1 = new TextField(30);
    TextField tf2 = new TextField(30);
    TextField tf3 = new TextField(30);
    TextField tf4 = new TextField(30);
    //khởi tạo ô vuông checkbox để chọn giới tính, phòng ban
    //Giới tính
    CheckboxGroup cbg1 = new CheckboxGroup();
    Checkbox cb1 = new Checkbox("Nam", cbg1, true);
    Checkbox cb2 = new Checkbox("Nữ", cbg1, false);
    //Phòng ban
    CheckboxGroup cbg2 = new CheckboxGroup();
    Checkbox cb3 = new Checkbox("Thiết bị", cbg2, true);
    Checkbox cb4 = new Checkbox("Nhân sự", cbg2, false);
    Checkbox cb5 = new Checkbox("R&D", cbg2, false);
}

```

```

Checkbox cb6 = new Checkbox("Kinh doanh", cbg2, false);
Choice ngay = new Choice();
Choice thang = new Choice();
Choice nam = new Choice();
// khởi tạo 2 nút ấn
//Thêm nhân viên
Button bt1 = new Button("Thêm nhân viên");
//Lưu
Button bt2 = new Button("Lưu");

public form(String title) {
    //khởi tạo chương trình và đặt tên cho cửa sổ khi chạy chương
    trình là chuỗi t truyền vào
    super(title);
    // set kiểu Layout là Flow
    setLayout(new FlowLayout());
    //Thêm các Label và TextField vào Frame.
    //Họ tên
    add(lb1);           add(tf1);
    //Địa chỉ
    add(lb2);           add(tf2);
    //Lương cơ bản
    add(lb6);           add(tf3);
    //Thâm niên
    add(lb7);           add(tf4);
    // thêm checkbox group cbg1
    //giới tính
    add(lb3);           add(cb1);           add(cb2);
    // thêm checkbox group cbg2
    //Phòng ban
    add(lb5);
    add(cb3); // thiết bị
    add(cb4); // Nhân sự
    add(cb5); // R&D
    add(cb6); // Kinh doanh
    //Ngày, tháng, năm sinh
    // add thêm giá trị cho danh sách chọn. -> danh sách ngày,
    danh sách tháng, danh sách năm.
    //danh sách ngày
    add(lb4);
    for (int i = 1; i <= 31; i++) {
        ngay.addItem("" + i);
    }
    //danh sách tháng
    for (int i = 1; i <= 12; i++) { thang.addItem("" + i);
}
    //danh sách năm
    for (int i = 2017; i >= 1969; i--) {
        nam.addItem("" + i);
    }
}

```

```

// thêm 3 danh sách chọn vào Frame
add/ngay);           add/thang);           add/nam);
add/bt1);           bt1.addActionListener(this);
add/bt2);           bt2.addActionListener(this);
}

@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == bt1) {
        String hoten = tf1.getText();
        String diachi = tf2.getText();
        String gioitinh = null;
        if (cb1.getState() == true) {
            gioitinh = "Nam";
        }
        if (cb2.getState() == true) {
            gioitinh = "Nữ";
        }
        String ngaysinh = (" " + (ngay.getSelectedItem()) + "/" +
(thang.getSelectedItem()) + "/" + (nam.getSelectedItem()));
        String phongban = null;
        if (cb3.getState() == true) {
            phongban = "Thiết bị";
        }
        if (cb4.getState() == true) {
            phongban = "Nhân sự";
        }
        if (cb5.getState() == true) {
            phongban = "R&D";
        }
        if (cb6.getState() == true) {
            phongban = "Kinh doanh";
        }
        double luongcoban = Double.parseDouble(tf3.getText());
        double thamnien = Double.parseDouble(tf4.getText());
        nhanvien nv = new nhanvien(hoten, ngaysinh, gioitinh,
diachi, phongban, luongcoban, thamnien);
        nv.hienthi();
    }

    if (e.getSource() == bt2) {
        String hoten = tf1.getText();
        String diachi = tf2.getText();
        String gioitinh = null;
        if (cb1.getState() == true) {
            gioitinh = "Nam";
        }
        if (cb2.getState() == true) {
            gioitinh = "Nữ";
        }
    }
}

```

```

        String ngaysinh = (" " + (ngay.getSelectedItem()) + "/" +
(thang.getSelectedItem()) + "/" + (nam.getSelectedItem()));
        String phongban = null;
        if (cb3.getState() == true) {
            phongban = "Thiết bị";
        }
        if (cb4.getState() == true) {
            phongban = "Nhân sự";
        }
        if (cb5.getState() == true) {
            phongban = "R&D";
        }
        if (cb6.getState() == true) {
            phongban = "Kinh doanh";
        }
        double luongcoban = Double.parseDouble(tf3.getText());
        double thamnien = Double.parseDouble(tf4.getText());
        nhanvien nv = new nhanvien(hoten, ngaysinh, gioitinh,
diachi, phongban, luongcoban, thamnien);
        try {
            nv.luuthongtin();
        } catch (IOException ex) {

Logger.getLogger(form.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
}

/*Tạo file DeThiSo1.java, chứa hàm main để gọi form
package dethiso1;
import java.awt.Button;
import java.awt.Checkbox;
import java.awt.CheckboxGroup;
import java.awt.Choice;
import java.awt.FlowLayout;
import java.awt.Frame;
import java.awt.Label;
import java.awt.TextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;
/**
 * @author TranNguyenHuong
 */
public class DeThiSo1 {
    public static void main(String[] args) {

```

```

        form f = new form("Thông tin nhân viên");
        f.setSize(300, 400);
        f.setVisible(true);
        f.setLocation(450, 75);
        f.setResizable(false);
        f.addWindowListener(
            new WindowAdapter() {
                @Override
                public void windowClosing(WindowEvent we) {
                    System.exit(0);
                }
            }
        );
    }
}

```

Câu 2: Kết quả chạy chương trình

Xét chương trình sau:

```

1. class Msg{
2.     public Msg()
3.     { System.out.println("Hello");}
4.     public void sayGoodbye()
5.     {System.out.println("Goodbye");}
6. }
7. class TestMsg extends Msg{
8.     public TestMsg(){
9.         super();
10.        System.out.println("Hello Java");
11.    }
12. public void sayGoodbye()
13.     {System.out.println("Goodbye Java");}
14. }
15. public class Demo {
16.     public static void main(String args[])
17.     {
18.         Msg m1 = new Msg();
19.         Msg m2 = new TestMsg();
20.         m1.sayGoodbye();
21.         m2.sayGoodbye();
22.     }
23. }

```

Giải thích:

Lớp **Msg** có hàm tạo không tham số để in ra chữ Hello (dòng 3) và có hàm sayGoodbye để in ra chữ Goodbye (dòng 5)

Lớp **TestMsg** kế thừa lớp **Msg**

- Hàm tạo của **TestMsg** gọi hàm tạo của lớp **Msg** (dòng 9) và có thêm lệnh in lên màn hình chữ Hello Java (dòng 10). Khi hàm tạo này được gọi sẽ ra chữ Hello và Hello Java
- Hàm sayGoodbye cùng tên với hàm của lớp **Msg**, in ra chữ Goodbye Java (dòng 13).

Tại hàm **main()**

- Dòng 18 khai báo đối tượng m1 của lớp Msg, do đó hàm tạo của Msg được gọi → in ra màn hình chữ Hello
- Dòng 19. Khai báo đối tượng m2 và khởi tạo đối tượng của lớp TestMsg → hàm tạo của TestMsg được gọi → in ra Hello và Hello Java
- Dòng 20. Gọi hàm sayGoodbye của lớp Msg → in ra chữ Goodbye
- Dòng 21. Hợp hàm sayGoodbye của lớp TestMsg → in ra Goodbye Java

Tóm lại, kết quả in ra màn hình là

```
Hello  
Hello  
Hello Java  
Goodbye  
Goodbye Java
```

ĐỀ SỐ 02

Câu 1-02 (8 điểm) – Quản lý sinh viên

- (1 điểm) Hãy định nghĩa class Person với các thuộc tính: Họ tên, ngày sinh, địa chỉ, giới tính và các phương thức khởi tạo, hiển thị thông tin.
- (1 điểm) Định nghĩa class Student kế thừa từ Person, có thêm các thuộc tính: mã sinh viên, email, điểm tổng kết và các phương thức khởi tạo, hiển thị thông tin.
- (2 điểm) Tạo form cho phép nhập các thông tin sinh viên như định nghĩa ở trên.
- (2 điểm) Khi người dùng click nút Thêm sinh viên thì chương trình khởi tạo đối tượng sinh viên với các giá trị thuộc tính lấy từ form người dùng nhập và hiển thị thông tin lên màn hình console.
- (2 điểm) Khi người dùng click nút Lưu dữ liệu thì ghi toàn bộ thông tin vào file sinhvien.dat

Câu 2-02 (2 điểm)

Hãy cho biết kết quả (và lý do) của đoạn chương trình sau:

```
class Msg{  
    public Msg()  
    { System.out.println("Hello ");}  
    public Msg(String name)  
    { System.out.println("Hello "+name);}  
    public void sayGoodbye()  
    {System.out.println("Goodbye");}  
}  
class TestMsg extends Msg{  
    public TestMsg(String name){  
        super(name);  
        System.out.println("Hello Java");  
    }  
    public void sayGoodbye()  
    {System.out.println("Goodbye Java");}  
}
```

```

public class Main {
    public static void main(String args[]){
        Msg m1 = new Msg("Huong");
        TestMsg m2 = new TestMsg("Minh");
        Msg m3 = new TestMsg("Phuong");
        Msg m4 = new Msg();
        m1.sayGoodbye(); m2.sayGoodbye();
        m3.sayGoodbye();
    }
}

```

ĐỀ SỐ 03

Câu 1-03 (2 điểm) – Quản lý nhân viên

- (1 điểm) Hãy định nghĩa class Person với các thuộc tính: Họ tên, ngày sinh, địa chỉ, giới tính và các phương thức khởi tạo, hiển thị thông tin.
- (1 điểm) Định nghĩa class NhanVien kế thừa từ Person, có thêm các thuộc tính: phòng ban, hệ số lương, thâm niên, lương cơ bản và các phương thức khởi tạo, tính lương thực lĩnh, hiển thị thông tin cá nhân

Trong đó: $lương\ thực\ lĩnh = Lương\ cơ\ bản * hệ\ số\ (1 + thâm\ niên/100)$

- (3 điểm) Hãy đọc file nhanvien.dat gồm danh sách các nhân viên với định dạng lưu trữ như sau: mỗi nhân viên được lưu trên 1 dòng, mỗi thuộc tính phân cách bởi ký tự “\$”. Dữ liệu đọc được lưu dưới dạng một mảng các đối tượng nhân viên (arraylist)
- (3 điểm) Hiển thị danh sách trên trong một table (hiển thị trên console chỉ được 1.5 điểm).

Câu 2-02 (2 điểm)

Hãy cho biết kết quả (và lý do) của đoạn chương trình sau:

```

class Msg{
    public Msg(){}
    public Msg(String str)
    { System.out.println("Hello" + str); }
    public void sayGoodbye()
    {System.out.println("Goodbye"); }
}
class TestMsg extends Msg{
    public TestMsg(String str){
        System.out.println("Hello" + str);}
    public void sayGoodbye(){
        System.out.println("Goodbye Java");}
}
public class Demo {
    public static void main(String args[]){
        Msg m1 = new Msg("Mark");
        Msg m2 = new TestMsg("Java");
        m1.sayGoodbye();
        m2.sayGoodbye();
    }
}

```

}

}

ĐỀ SÓ 04

Câu 1-04 (8 điểm) – Quản lý sinh viên

- a. (1 điểm) Hãy định nghĩa class Person với các thuộc tính: Họ tên, ngày sinh, địa chỉ, giới tính và các phương thức khởi tạo, hiển thị thông tin.
- b. (1 điểm) Định nghĩa class Student kế thừa từ Person, có thêm các thuộc tính: mã sinh viên, email, điểm tổng kết và các phương thức khởi tạo, hiển thị thông tin.
- c. (3 điểm) Hãy đọc file sinhvien.dat gồm danh sách các Sinh viên với định dạng lưu trữ như sau: mỗi Sinh viên được lưu trên 1 dòng, mỗi thuộc tính phân cách bởi ký tự "\$". Dữ liệu đọc được lưu dưới dạng một mảng các đối tượng Sinh viên (arraylist).
- d. (3 điểm) Hiển thị danh sách trên trong một table (hiển thị trên console chỉ được 1.5 điểm).

Câu 2-04 (2 điểm)

Hãy chỉ ra lỗi sai của đoạn chương trình sau:

```
class Msg{  
    public Msg(){  
        System.out.println("Hello");  
    }  
    public void sayGoodbye(){  
        System.out.println("Goodbye");  
    }  
}  
  
class TestMsg extends Msg{  
    public void sayGoodbye(){  
        System.out.println("Goodbye Java");  
    }  
}  
  
public class Demo {  
    public static void main(String args[]){  
        Msg m1 = new Msg("OOP");  
        TestMsg m2 = new Msg(); m1.sayGoodbye();  
        m2.sayGoodbye();  
    }  
}
```

ĐỀ SÓ 05

Câu 1-05 (8 điểm) – Quản lý sinh viên

- a. (2 điểm) Hãy định nghĩa class Sinhvien với các thuộc tính: mã sinh viên, họ tên, ngày sinh, giới tính, điểm trung bình và các phương thức khởi tạo, hiển thị thông tin.
- b. (1 điểm) Định nghĩa class SinhvienHTTT kế thừa từ sinhvien, có thêm thuộc tính:

học phí và phương thức khởi tạo, hiển thị thông tin.

c. (1 điểm) Định nghĩa class SinhVienMatMa kế thừa từ SinhVien, có thêm thuộc tính: đơn vị, lương và phương thức khởi tạo, hiển thị thông tin.

d. (4 điểm) Hãy tạo các form cho phép nhập dữ liệu sinh viên, sinh viên mật mã thì lưu dữ liệu vào file svmm.dat, sinh viên an toàn thì lưu dữ liệu vào file svat.dat

Câu 2-05 (2 điểm)

Hãy chỉ ra lỗi sai của đoạn chương trình sau:

```
interface Action{
    public void drive(){}
    public void stop(){}
}
class Vehicle implements Action{
    private int id;
    public Vehicle(int id){
        this.id = id;
        System.out.println("Hello world");
    }
}
public class Demo {
    public static void main(String args[]){
        Vehicle
        m = new Vehicle ();
    }
}
```

ĐỀ SỐ 06

Câu 1-06 (8 điểm) – Từ điển

a. (2 điểm) Hãy định nghĩa class Word với các thuộc tính: id, en(từ tiếng anh), vn(nghĩa tiếng việt) và các phương thức khởi tạo, hiển thị thông tin, các phương thức getters/setters.

Định nghĩa class Dictionary cho phép

b. (1 điểm) Khởi tạo một danh sách (> 10 từ), có thể tự khởi tạo hoặc nhập dữ liệu

c. (3 điểm) Tạo một Form tìm kiếm, thực hiện tìm kiếm một từ trong từ điển (sử dụng thuật toán tìm nhị phân, tìm tuần tự được ½ số điểm). Nếu tìm thấy thì hiển thị nghĩa tiếng việt, trái lại thực hiện theo câu 2.c

d. (2 điểm) Bổ sung một từ vào từ điển nếu từ đó chưa có trong từ điển *Chú ý: các từ trong danh sách phải đúng thứ tự từ điển (theo từ tiếng anh)*

Câu 2-06 (2 điểm)

Hãy chỉ ra lỗi sai của đoạn chương trình sau:

```
interface Action{
    public void drive();
```

```

public void stop();
}
class Vehicle implements Action{
private int id;
public void Vehicle(int id){
this.id = id; System.out.println("Hello world");
}
public void drive(){ System.out.println("I'm driving");
}
}
public class Demo {
public static void main(String args[]){ Vehicle
m = new Vehicle (); m.drive();
m.stop();
}
}

```

ĐỀ SỐ 07

Câu 1-07 (8 điểm) – Thí sinh

- (2 điểm) Hãy định nghĩa class Thí sinh với các thuộc tính: họ tên, ngày sinh, địa chỉ và các phương thức khởi tạo, nhập và in thông tin
- (1 điểm) Định nghĩa class Thí sinh khống A kế thừa từ Thí sinh, có thêm các thuộc tính: điểm toán, lý, hóa và các phương thức khởi tạo, nhập, in thông tin
- (1 điểm) Định nghĩa class Thí sinh khống C kế thừa từ Thí sinh, có thêm các thuộc tính: điểm văn, sử, địa và các phương thức khởi tạo, nhập, in thông tin.
- (2 điểm) Thực hiện nhập danh sách n thí sinh của cả 2 khống A, C và in danh sách thí sinh trùng tuyển (tổng điểm > 20) ra màn hình.

Câu 2-07 (2 điểm)

Hãy cho biết kết quả (và lý do) của đoạn chương trình sau:

```

public class FileDemo{
public static void method1(File f1) {
String contents[] = f1.list();
for (int i=0; i< contents.length; i++){
System.out.println(contents[i] + "\n");
}
}
public static void main(String[] args){
method1 (new File("D:\\Documents\\utt"));
}
}

```

ĐỀ SỐ 08

Câu 1-08 (8 điểm) - Number

- a. (2 điểm) Định nghĩa class Number với các thuộc tính: value, min, max và các phương thức khởi tạo, nhập và kiểm tra nguyên tố của giá trị value.
- b. (2 điểm) Hãy tạo form cho phép nhập thông tin Number.
- c. (2 điểm) Nếu người dùng click nút bấm khởi tạo thì hãy khởi tạo đối tượng Number (có kiểm tra điều kiện nhập min < value < max, bắt các lỗi exception).
- d. (2 điểm) Nếu người dùng click vào nút bấm Lưu trữ thì lưu thông tin vào file gồm value, min, max nếu value là số nguyên tố.

Câu 3-08 (2 điểm)

Giải thích ý nghĩa và chỉ ra lỗi sai (giải thích) trong đoạn chương trình sau:

```
class MyForm extends Frame implements ActionListener
{
    public MyForm(String title){
        super(title); setSize(300, 200);
        setVisible(true);
    }
}
```

ĐỀ SỐ 09

Câu 1-09 (8 điểm)

- a. (2 điểm) Hãy định nghĩa class Book gồm các thuộc tính id, authors, title, category (nhận 1 trong các giá trị: tạp chí, KHXH, KHTN, luận văn) và các phương thức khởi tạo, các getters/setters.
- b. (2 điểm) Hãy thiết kế form cho phép nhập thông tin để thêm sách
- c. (3 điểm) Nếu người dùng click nút Thêm sách thì thực hiện khởi tạo đối tượng sách với các thuộc tính lấy từ form do người dùng nhập, đồng thời lưu dữ liệu vào file books.dat
- d. (1 điểm) Nếu người dùng click nút Nhập lại thì reset form về trạng thái ban đầu.

Câu 2-09 (2 điểm)

Giải thích ý nghĩa từ khóa super, toán tử this và Hãy chỉ ra lỗi sai của đoạn chương trình sau:

```
class ThiSinh{
    private String name;
    public ThiSinh(String name){
        this.name = name;
    }
}
class ThiSinhA extends ThiSinh{
```

```

        double toan, ly, hoa;
        public ThiSinhA(String name,double toan, double ly, double hoa){
            super(name); this.toan = toan;
            this.ly = ly; this.hoa = hoa;
        }
        public void display(){
            System.out.println("Ho ten: " + this.name); System.out.println("Tong
diem: " + (this.toan +
            this.ly + this.hoa));
        }
    }
}

```

ĐỀ SỐ 10

Câu 1-10 (8 điểm)

- (2 điểm) Hãy định nghĩa class Giangvien gồm các thuộc tính: họ tên, mã giảng viên, địa chỉ, giới tính, khoa (nhận 1 trong các giá trị: ĐVT, CNTT, HTTT, CNM, CNPM) và các phương thức khởi tạo.
- (2 điểm) Hãy thiết kế form cho phép nhập thông tin giảng viên
- (3 điểm) Nếu người dùng click nút Lưu trữ thì thực hiện khởi tạo đối tượng sách với các thuộc tính lấy từ form do người dùng nhập, đồng thời lưu dữ liệu vào file giangvien.dat
- (1 điểm) Nếu người dùng click nút Clear thì reset form về trạng thái ban đầu.

Câu 2-10 (2 điểm)

Cho đoạn chương trình sau:

```

public class Demo {
    public static void main(String args[]) {
        int value;
        try
        {
            value = Integer.parseInt(args[0]);
            System.out.println("value = " + value);
        }
        catch(NumberFormatException ex)
        {
            ex.printStackTrace();
        }
    }
}

```

Hãy giải thích ý nghĩa của try – catch trong đoạn chương trình trên. Cho biết kết quả nếu thực thi chương trình trên nếu args[0] = “10” và trường hợp args[0] = “A10”.

6.3. DANH SÁCH ĐỀ TÀI LÀM BÀI TẬP LỚN

Đề tài 1. GAME TERIS

Game được thiết kế và phát triển lần đầu tiên vào năm 1985 bởi một lập trình viên người Nga tên là Alexey Pajitnov. Kể từ đó, game này được phát triển thành nhiều phiên bản trên nhiều hệ máy khác nhau.

Trong game có 7 khối gạch có dạng hình chữ S, hình chữ Z, hình chữ T, hình chữ L, hình đường thẳng, hình chữ L ngược và hình vuông. Mỗi khối gạch được tạo nên bởi 4 hình vuông. Các khối gạch sẽ rơi từ trên xuống. Người chơi phải di chuyển và xoay các khối gạch sao cho chúng vừa khít với nhau, nếu các khối gạch lấp đầy một đường ngang thì người chơi ghi điểm. Trò chơi kết thúc khi các khối gạch chồng lên nhau quá cao.



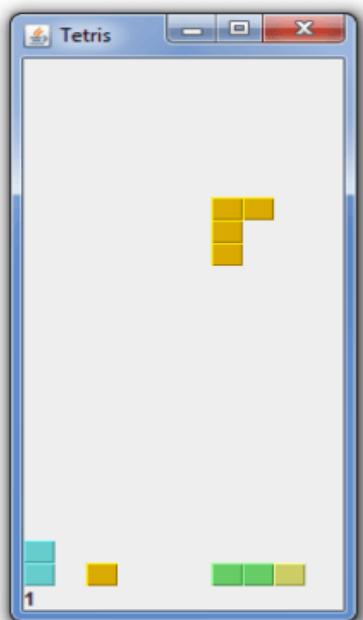
Mô tả game

Kích thước của các đối tượng trong game này được tính theo đơn vị 1 hình vuông chứ không theo pixel nữa. Kích thước của cửa sổ giờ đây là 10×22 hình vuông. Kích thước thật của các hình vuông được tính động theo kích thước cửa sổ. Tức là nếu kích thước cửa sổ là 200×400 pixel thì kích thước thật của mỗi hình vuông là $200 \times 400 / 10 \times 22$ là khoảng 20×18 pixel (nhưng chúng ta cũng không quan tâm đến con số này).

Mỗi lần chạy chương trình là game sẽ tự động chạy ngay, tuy nhiên chúng ta có thể cho tạm dừng game bằng cách bấm phím P. Nếu muốn khối gạch rơi nhanh xuống đáy ngay lập tức thì bấm phím Space. Bấm phím D sẽ làm cho khối gạch rơi xuống thêm 1 hàng, có thể dùng phím này để tăng tốc độ rơi thêm một chút. Tốc độ game không thay đổi. Điểm sẽ là số hàng đã được loại bỏ.

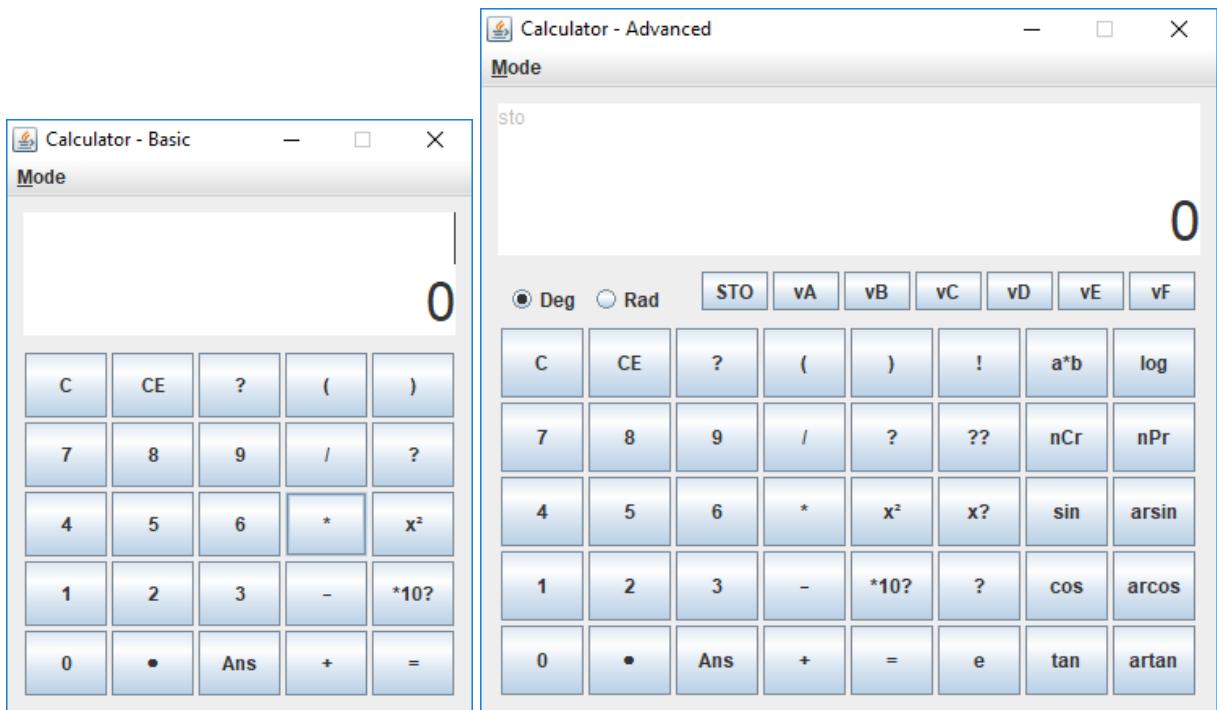
(Xem mã nguồn ở phụ lục I)

Giao diện khi chơi



Đề tài 2. CALCULATOR

- Xây dựng máy tính một chiếc máy tính điện tử để tính toán. Gồm 2 chế độ Basic (cơ bản) và Advanced (nâng cao), các phép tính như hình dưới đây



(Xem mã nguồn ở phụ lục 2)

Đề tài 3. TRÒ CHƠI FUZZLE

Xây dựng giao diện chương trình trò chơi puzzle như sau:

Puzzle			
15	14	13	12
11	10	9	8
7	6	5	4
3	2	1	

Khi người sử dụng click vào nút cùng hàng hoặc cùng cột với nút không có số thì đổi chỗ nút vừa click và nút không số cho nhau. Khi người sử dụng ấn nút không cùng hàng hay không cùng cột thì hiển thông báo "Không hợp lệ" còn khi người sử dụng đã sắp xếp các nút theo thứ tự từ 1 đến 15 thì hiện ra thông báo "Chúc mừng".

Đề tài 4. MÔ PHỎNG HÒM THƯ

Người ta muốn mô phỏng một thùng thư tự động. Thùng thư này có thể chứa được một số lượng bưu phẩm có giới hạn. Có hai loại bưu phẩm :

- Thư: giá tiền vận chuyển bình thường là 3000 VND cho một thư. Nếu là thư cần chuyển nhanh thì sẽ phải trả thêm 2000VND .
- Bưu kiện: giá thành vận chuyển là 20000 VND cho một bưu kiện, không phân biệt độ lớn hay trọng lượng.

Anh (chị) hãy viết một chương trình cho phép tính tự động giá tiền vận chuyển của tất cả các bưu phẩm chứa trong thùng thư. Chương trình phải có các chức năng sau:

1. Khởi tạo một thùng thư có chứa một số bưu phẩm (gồm cả hai loại: thư và bưu kiện)
2. Bổ bưu phẩm vào trong thùng thư

3. In ra toàn bộ các bưu phẩm có trong thùng thư với các thông tin sau : loại bưu phẩm (nếu là thư thì cần phân biệt giữa thư gửi thường và thư chuyển nhanh) và giá thành vận chuyển.

Đề tài 5. QUẢN LÝ BĂNG ĐĨA

Hãy viết một chương trình quản lý mua bán tại một cửa hàng. Hiện nay, cửa hàng này có bán ba loại mặt hàng sau: sách, CD nhạc, CD phần mềm. Mỗi loại mặt hàng phải chịu một mức thuế VAT khác nhau. Ví dụ: 1% cho sách, 2% cho CD nhạc và 3% cho CD phần mềm. Mỗi mặt hàng có các thông tin liên quan sau (ngoài thông tin về thuế) :

- Sách : tên sách, tên tác giả, giá tiền, số lượng
- CD nhạc : tên đĩa CD, tên nghệ sĩ, giá tiền, số lượng
- CD phần mềm : tên phần mềm, hệ điều hành, giá tiền, số lượng.

Chương trình sẽ phải có các chức năng sau :

1. Hiển thị hóa đơn bán hàng sau mỗi lần bán hàng

Giả sử chúng ta đã nhập vào thông tin mặt hàng đã được bán như sau :

Sách :	'Ngôn ngữ C++',	'Minh'	50000 vnd,	1	
CD nhạc :	'Chờ anh',	'My Tam'	20000 vnd,	1	
CD phần mềm :	'Word',	'Windows'	30000 vnd,	1.	

Chương trình sẽ phải hiển thị các hóa đơn tương ứng như sau (với các thông tin kèm theo tên mặt hàng, tên tác giả, tên nghệ sĩ hoặc tên hệ điều hành tùy theo loại hàng, giá tiền, mức thuế, số lượng, tổng số tiền). Ví dụ:

Sách :	'Ngôn ngữ C++',	'Minh'	50000 vnd,	1%	1	50500vnd
CD nhạc :	'Chờ anh',	'My Tam'	20000 vnd,	2%	1	20400vnd
CD phần mềm :	'Word',	'Windows'	30000 vnd,	3%	1	30900vnd.

2. Lưu trữ toàn bộ các giao dịch bán hàng trong bộ nhớ (các mặt hàng đã bán)

3. hiển thị toàn bộ danh sách các mặt hàng đã bán (tên, giá, mức thuế, số lượng, tổng số tiền) và tổng số tiền thu được sau khi bán các mặt hàng này. Ví dụ sau minh họa với các giao dịch bán hàng trong phần trước :

----- Danh sách các giao dịch bán hàng -----					
Sách :	'Ngôn ngữ C++',	50000 vnd,	1%	1	50500 vnd
CD nhạc :	'Chờ anh',	20000 vnd,	2%	1	20400 vnd
CD phần mềm :	'Word',	30000 vnd,	3%	1	30900 vnd
Số tiền tổng cộng :					101800 vnd

Đề tài 6. BÁN VÉ MÁY BAY

Hãy viết một chương trình quản lý đơn giản dành cho một văn phòng du lịch chuyên bán vé máy bay. Chương trình này được mô tả như sau:

- Mỗi khách hàng có thể mua một hay nhiều vé máy bay cho hành khách.
- Chương trình phải lưu lại các thông tin cá nhân sau của khách hàng :
 - Họ và tên
 - Số điện thoại: văn phòng, nhà riêng, di động hoặc các số khác nếu có.
 Chương trình phải được thiết kế sao cho việc thêm một kiểu số điện thoại mới được dễ dàng.
- Những thông tin liên quan tới vé máy bay:
 - Thông tin cá nhân về hành khách: họ và tên, số điện thoại (văn phòng, nhà riêng, di động hoặc các số khác nếu có),
 - Số hiệu chuyến bay,
 - Sân bay đến và đi,

- Giá vé,
- Phần trăm giảm giá: tùy theo loại hành khách. Ví dụ như, giảm 10% giá vé cho sinh viên và người cao tuổi từ 65 trở lên, giảm 25% giá vé cho trẻ em dưới 5 tuổi. Chương trình phải được thiết kế sao cho việc thêm một kiểu phần trăm giảm giá vé mới được dễ dàng.

Chương trình phải cho phép người sử dụng thực hiện được các chức năng sau:

1. Hiển thị hóa đơn sau mỗi lần bán hàng. Trên hóa đơn phải có các thông tin sau:
 - Thông tin cá nhân của khách hàng,
 - Thông tin liên quan tới vé máy bay,
 - Số tiền phải trả.
2. Lưu toàn bộ các hóa đơn bán hàng trong bộ nhớ.
3. Hiển thị toàn bộ các hóa đơn cùng với tổng số tiền.
4. Tìm kiếm số điện thoại của một hành khách theo tên và số hiệu chuyến bay của người đó.

Đề tài 7. QUẢN LÝ TUYỂN SINH VÀO TRƯỜNG ĐẠI HỌC

- Xây dựng một cơ sở dữ liệu Sinh viên, quản lý sinh viên thi vào trường Đại học công nghệ GTVT. Các thông tin quản lý bao gồm: Họ tên, ngày sinh, địa chỉ, giới tính, điểm thi 3 môn, điểm khu vực, điểm ưu tiên, điểm khuyến khích. Các ngành học, chỉ tiêu đào tạo của mỗi ngành.
- Chương trình tối thiểu có các chức năng sau: Nhập/bổ sung, sửa, xóa, hiển thị và lưu thông tin các thí sinh dự thi. Nhập điểm thi, in kết quả thi dựa vào điểm chuẩn, tìm kiếm thông tin thí sinh dự thi.

Đề tài 8. QUẢN LÝ HỘ KHẨU

- Xây dựng các lớp liên quan về nhân khẩu (hộ gia đình, thành viên), xử lý thông tin đăng ký tạm trú, tạm vắng trong một phường, xã.
- Chương trình tối thiểu có các chức năng sau: Nhập/bổ sung, sửa, xóa, hiển thị và lưu vào tệp dữ liệu thông tin các hộ gia đình, nhân khẩu. Xử lý thông tin khai báo tạm trú, tạm vắng của các nhân khẩu. Tìm kiếm thông tin về hộ gia đình, nhân khẩu theo tiêu chí nào đó (theo họ tên, năm sinh, v.v.).

Đề tài 9. QUẢN LÝ KHO HÀNG

- Công tác quản lý kho hàng quản lý theo các đối tượng: hàng hóa, nhân viên và nhập xuất hàng. Các thông tin về mỗi đối tượng này sẽ được lưu vào các tệp dữ liệu để phục vụ công tác tra cứu sau này.
- Chương trình tối thiểu có các chức năng sau: Nhập/bổ sung, sửa, xóa, hiển thị và lưu thông tin các mặt hàng trong kho, thông tin nhân viên và quản lý nhập xuất hàng hóa. Tìm kiếm thông tin hàng hóa, tìm kiếm thông tin nhân viên.

Đề tài 10. QUẢN LÝ CÁC CỬA HÀNG ĐIỆN THOẠI

- Công tác bán hàng được quản lý theo các đối tượng: hàng hóa (điện thoại), khách mua, hóa đơn mua hàng của khách. Thông tin về mỗi đối tượng quản lý được lưu vào các tệp tin trên đĩa để phục vụ các yêu cầu tra cứu về sau.
- Chương trình tối thiểu có các chức năng sau: Nhập, bổ sung, sửa, xóa, hiển thị và lưu vào các tệp dữ liệu thông tin các mặt hàng điện thoại, thông tin khách hàng và hóa đơn bán hàng. Tìm kiếm thông tin hàng, hóa đơn. In (hiển thị) hóa đơn bán hàng cho khách.

TÀI LIỆU THAM KHẢO

- [1]. Đoàn Văn Ban, Đoàn Văn Trung (2011), *Lập trình Java*, NXB Giáo dục Việt Nam.
- [2]. Phương Lan (chủ biên), Hoàng Đức Hải (2007) *Java tập 1, Java Tập 2, NXB Lao động xã hội*.
- [3]. Cay S. Horstmann, Gary Cornell (2016), Core Java, Volume I--Fundamentals (8th Edition)
- [4]. Website: Nhóm chuyên gia Tutorials Point: <http://www.tutorialspoint.com/java/>.
- [5]. Website học ngôn ngữ Java
 - <http://docs.oracle.com/javase/tutorial/java/index.html>
 - <http://docs.oracle.com/javase/8/docs/>
- [6]. Download JDK và Netbean:
 - <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- [7]. Trang web bài tập Java trực tuyến: <https://www.hackerrank.com/>
- [8]. Xếp hạng NNLT: <http://www.tiobe.com/tiobe-index/>
- [9]. Giới thiệu về Java
 - [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))
 - [https://vi.wikipedia.org/wiki/Java_\(ng%C3%B4n_ng%C3%A1ng%BB%AF_l%C1%E1%BA%ADptr%C3%ACnh\)](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%C3%A1ng%BB%AF_l%C1%E1%BA%ADptr%C3%ACnh))

PHỤ LỤC 1 – CODE TRÒ CHƠI TERIS

```
//Tetris.java
package tetris;
import java.awt.BorderLayout;
import java.util.Random;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.SwingUtilities;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.Timer;
import tetris.Tetris.Shape.Tetrominoes;
public class Tetris extends JFrame {
    private JLabel statusbar;
    public Tetris() {
        initUI();
    }
    private void initUI() {
        statusbar = new JLabel(" 0");
        add(statusbar, BorderLayout.SOUTH);
        Board board = new Board(this);
        add(board);
        board.start();
        setSize(200, 400);
        setTitle("Tetris");
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
    }
    public JLabel getStatusBar() {
        return statusbar;
    }
    public static class Shape {
        protected enum Tetrominoes { NoShape, ZShape, SShape, LineShape,
            TShape, SquareShape, LShape, MirroredLShape };
        private Tetrominoes pieceShape;
        private int coords[][];
        private int[][][] coordsTable;
        public Shape() {
            coords = new int[4][2];
            setShape(Tetrominoes.NoShape);
        }
        public void setShape(Tetrominoes shape) {
            coordsTable = new int[][][] {
                { { 0, 0 }, { 0, 0 }, { 0, 0 }, { 0, 0 } },
                { { 0,-1 }, { 0, 0 }, { -1, 0 }, { -1, 1 } },
                { { 0,-1 }, { 0, 0 }, { 1, 0 }, { 1, 1 } },
                { { 0,-1 }, { 0, 0 }, { 0, 1 }, { 0, 2 } },
                { { -1, 0 }, { 0, 0 }, { 1, 0 }, { 0, 1 } },
                { { 0, 0 }, { 1, 0 }, { 0, 1 }, { 1, 1 } },
            
```

```

    { {-1,-1}, { 0,-1}, { 0, 0}, { 0, 1} },
    { 1,-1}, { 0,-1}, { 0, 0}, { 0, 1} }
};

for (int i = 0; i < 4 ; i++) {
    for (int j = 0; j < 2; ++j) {
        coords[i][j] = coordsTable[shape.ordinal()][i][j];
    }
}
pieceShape = shape;
}

private void setX(int index, int x) { coords[index][0] = x; }
private void setY(int index, int y) { coords[index][1] = y; }
public int x(int index) { return coords[index][0]; }
public int y(int index) { return coords[index][1]; }
public Tetrominoes getShape() { return pieceShape; }
public void setRandomShape() {
    Random r = new Random();
    int x = Math.abs(r.nextInt()) % 7 + 1;
    Tetrominoes[] values = Tetrominoes.values();
    setShape(values[x]);
}
public int minX() {
    int m = coords[0][0];
    for (int i=0; i < 4; i++) {
        m = Math.min(m, coords[i][0]);
    }
    return m;
}
public int minY() {
    int m = coords[0][1];
    for (int i=0; i < 4; i++) {

        m = Math.min(m, coords[i][1]);
    }
    return m;
}
public Shape rotateLeft() {
    if (pieceShape == Tetrominoes.SquareShape)
        return this;
    Shape result = new Shape();
    result.pieceShape = pieceShape;
    for (int i = 0; i < 4; ++i) {
        result.setX(i, y(i));
        result.setY(i, -x(i));
    }
    return result;
}
public Shape rotateRight() {
    if (pieceShape == Tetrominoes.SquareShape)
        return this;
    Shape result = new Shape();
    result.pieceShape = pieceShape;

    for (int i = 0; i < 4; ++i) {
        result.setX(i, -y(i));
        result.setY(i, x(i));
    }
}

```

```

        }
        return result;
    }
}

public class Board extends JPanel
    implements ActionListener {
private final int BoardWidth = 10;
private final int BoardHeight = 22;
private Timer timer;
private boolean isFallingFinished = false;
private boolean isStarted = false;
private boolean isPaused = false;
private int numLinesRemoved = 0;
private int curX = 0;
private int curY = 0;
private JLabel statusbar;
private Shape curPiece;
private Tetrominoes[] board;
public Board(Tetris parent) {
    initBoard(parent);
}
private void initBoard(Tetris parent) {
    setFocusable(true);
    curPiece = new Shape();
    timer = new Timer(400, this);
    timer.start();
    statusbar = parent.getStatusBar();
    board = new Tetrominoes[BoardWidth * BoardHeight];
    addKeyListener(new TAdapter());
    clearBoard();
}
@Override
public void actionPerformed(ActionEvent e) {
    if (isFallingFinished) {
        isFallingFinished = false;
        newPiece();
    } else {
        oneLineDown();
    }
}
private int squareWidth() { return (int) getSize().getWidth() / BoardWidth; }
private int squareHeight() { return (int) getSize().getHeight() / BoardHeight; }
private Tetrominoes shapeAt(int x, int y) { return board[(y * BoardWidth) + x]; }

public void start() {
    if (isPaused)
        return;
    isStarted = true;
    isFallingFinished = false;
    numLinesRemoved = 0;
    clearBoard();
    newPiece();
    timer.start();
}
private void pause() {
    if (!isStarted)

```

```

        return;
    isPaused = !isPaused;
    if (isPaused) {
        timer.stop();
        statusbar.setText("paused");
    } else {
        timer.start();
        statusbar.setText(String.valueOf(numLinesRemoved));
    }
    repaint();
}
private void doDrawing(Graphics g) {
    Dimension size = getSize();
    int boardTop = (int) size.getHeight() - BoardHeight * squareHeight();
    for (int i = 0; i < BoardHeight; ++i) {
        for (int j = 0; j < BoardWidth; ++j) {
            Tetrominoes shape = shapeAt(j, BoardHeight - i - 1);
            if (shape != Tetrominoes.NoShape)
                drawSquare(g, 0 + j * squareWidth(),
                           boardTop + i * squareHeight(), shape);
        }
    }
    if (curPiece.getShape() != Tetrominoes.NoShape) {
        for (int i = 0; i < 4; ++i) {
            int x = curX + curPiece.x(i);
            int y = curY - curPiece.y(i);
            drawSquare(g, 0 + x * squareWidth(), boardTop + (BoardHeight - y - 1) * squareHeight(),
                       curPiece.getShape());
        }
    }
}

@Override
public void paintComponent(Graphics g) {
    super.paintComponent(g);
    doDrawing(g);
}
private void dropDown() {
    int newY = curY;
    while (newY > 0) {
        if (!tryMove(curPiece, curX, newY - 1))
            break;
        --newY;
    }
    pieceDropped();
}
private void oneLineDown() {
    if (!tryMove(curPiece, curX, curY - 1))
        pieceDropped();
}
private void clearBoard() {
    for (int i = 0; i < BoardHeight * BoardWidth; ++i)
        board[i] = Tetrominoes.NoShape;
}

private void pieceDropped() {
    for (int i = 0; i < 4; ++i) {

```

```

        int x = curX + curPiece.x(i);
        int y = curY - curPiece.y(i);
        board[(y * BoardWidth) + x] = curPiece.getShape();
    }
    removeFullLines();
    if (!isFallingFinished)
        newPiece();
}
private void newPiece() {
    curPiece.setRandomShape();
    curX = BoardWidth / 2 + 1;
    curY = BoardHeight - 1 + curPiece.minY();
    if (!tryMove(curPiece, curX, curY)) {
        curPiece.setShape(Tetrominoes.NoShape);
        timer.stop();
        isStarted = false;
        statusbar.setText("game over");
    }
}

private boolean tryMove(Shape newPiece, int newX, int newY) {
    for (int i = 0; i < 4; ++i) {
        int x = newX + newPiece.x(i);
        int y = newY - newPiece.y(i);
        if (x < 0 || x >= BoardWidth || y < 0 || y >= BoardHeight)
            return false;
        if (shapeAt(x, y) != Tetrominoes.NoShape)
            return false;
    }
    curPiece = newPiece;
    curX = newX;
    curY = newY;
    repaint();
    return true;
}
private void removeFullLines() {
    int numFullLines = 0;
    for (int i = BoardHeight - 1; i >= 0; --i) {
        boolean lineIsFull = true;
        for (int j = 0; j < BoardWidth; ++j) {
            if (shapeAt(j, i) == Tetrominoes.NoShape) {
                lineIsFull = false;
                break;
            }
        }
        if (lineIsFull) {
            ++numFullLines;
            for (int k = i; k < BoardHeight - 1; ++k) {
                for (int j = 0; j < BoardWidth; ++j)
                    board[(k * BoardWidth) + j] = shapeAt(j, k + 1);
            }
        }
    }
    if (numFullLines > 0) {
        numLinesRemoved += numFullLines;
        statusbar.setText(String.valueOf(numLinesRemoved));
    }
}

```

```

        isFallingFinished = true;
        curPiece.setShape(Tetrominoes.NoShape);
        repaint();
    }
}

private void drawSquare(Graphics g, int x, int y, Tetrominoes shape) {
    Color colors[] = { new Color(0, 0, 0), new Color(204, 102, 102),
        new Color(102, 204, 102), new Color(102, 102, 204),
        new Color(204, 204, 102), new Color(204, 102, 204),
        new Color(102, 204, 204), new Color(218, 170, 0)
    };

    Color color = colors[shape.ordinal()];
    g.setColor(color);
    g.fillRect(x + 1, y + 1, squareWidth() - 2, squareHeight() - 2);
    g.setColor(color.brighter());
    g.drawLine(x, y + squareHeight() - 1, x, y);
    g.drawLine(x, y, x + squareWidth() - 1, y);
    g.setColor(color.darker());
    g.drawLine(x + 1, y + squareHeight() - 1,
        x + squareWidth() - 1, y + squareHeight() - 1);
    g.drawLine(x + squareWidth() - 1, y + squareHeight() - 1,
        x + squareWidth() - 1, y + 1);
}

class TAdapter extends KeyAdapter {
    @Override
    public void keyPressed(KeyEvent e) {
        if (!isStarted || curPiece.getShape() == Tetrominoes.NoShape) {
            return;
        }
        int keycode = e.getKeyCode();
        if (keycode == 'p' || keycode == 'P') {
            pause();
            return;
        }
        if (isPaused)
            return;
        switch (keycode) {
        case KeyEvent.VK_LEFT:
            tryMove(curPiece, curX - 1, curY);
            break;
        case KeyEvent.VK_RIGHT:
            tryMove(curPiece, curX + 1, curY);
            break;
        case KeyEvent.VK_DOWN:
            tryMove(curPiece.rotateRight(), curX, curY);
            break;
        case KeyEvent.VK_UP:
            tryMove(curPiece.rotateLeft(), curX, curY);
            break;
        case KeyEvent.VK_SPACE:
            dropDown();
            break;
        case 'd':
            oneLineDown();
            break;
        }
    }
}

```

```
        case 'D':
            oneLineDown();
            break;
        }
    }
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            Tetris game = new Tetris();
            game.setVisible(true);
        }
    });
}
```

PHỤ LỤC 2 – CODE CALCULATOR

```
// Calculator.java
package calculator;
import java.util.Arrays;
import java.util.Stack;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;

import javax.swing.ButtonGroup;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JPanel;
import javax.swing.JRadioButton;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;

class CalculatorTheme extends JFrame implements ActionListener, KeyListener {

    private static final long serialVersionUID = 1L;
    private int frameWidth = 300;
    private int frameHeight = 400;
    private int xframeShow = 400;
    private int yframeShow = 100;
    private JTextField tfDisplay;
    private JLabel lbAns, lbStats, lbDOH, lbB;
    private int mode = 0;
    private JFrame frame;

    private JPanel mainPanel;
    private JPanel disPlayPanel;
    private JPanel buttonPanel;

    private JButton btnArr[];
    private JButton btnArrSub[];
    private JRadioButton radDeg, radRad;
    private JRadioButton radBin, radOct, radDec, radHex;

    private String lbButton[];
    private String mathElement[];
    private String varElement[];
    private double ans = 0;

    private Color colorDisableStats = Color.lightGray,
                colorEnnabaleStar = Color.black;
```

```

private boolean isSTO = false;
private Balan balan;

public CalculatorTheme() {
    frame = new JFrame();
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setResizable(false);
    frame.setBounds(xframeShow, yframeShow, frameWidth, frameHeight);
    frame.setJMenuBar(createMenuBar());
    resetValue(); // dat lai cac gia tri
    changeMode();
}
private void changeMode() {
    if (mode == 0) {
        frameWidth = 300;
        frameHeight = 380;
        frame.setTitle("Calculator - Basic");
    }
    if (mode == 1) {
        frameWidth = 460;
        frameHeight = 440;
        frame.setTitle("Calculator - Advanced");
    }
    if (mode == 2) {
        frameWidth = 460;
        frameHeight = 440;
        frame.setTitle("Calculator - Program");
    }
    createListLabelButton(mode);
    balan.setDegOrRad(true);
    balan.setRadix(10);

    frame.getContentPane().removeAll();
    frame.setSize(frameWidth, frameHeight);
    mainPanel = createMainPanel();
    frame.add(mainPanel);

    frame.getContentPane().validate();
    frame.setVisible(true);
    tfDisplay.requestFocus();
}

private JPanel createMainPanel() {
    JPanel mainPanel = new JPanel(new BorderLayout());
    disPlayPanel = createDisplayPanel();
    mainPanel.add(disPlayPanel, BorderLayout.NORTH);

    if (mode == 0) {
        buttonPanel = createButtonBasicPanel();
    }
    if (mode == 1) {
        buttonPanel = createButtonAdvancedPanel();
    }
    if (mode == 2) {
        buttonPanel = createButtonProgramPanel();
    }
}

```

```

        mainPanel.add(buttonPanel, BorderLayout.CENTER);
        mainPanel.setBorder(new EmptyBorder(10, 10, 10, 10));
        // mainPanel

        mainPanel.setVisible(true);
        return mainPanel;
    }

private JMenuBar createMenuBar() {
    JMenuBar mb = new JMenuBar();

    // menu mode
    JMenu mm = createMenu("Mode", KeyEvent.VK_M);

    mm.add(createMenuItem("Basic", KeyEvent.VK_B));
    mm.add(createMenuItem("Advanced", KeyEvent.VK_A));
    mm.add(createMenuItem("Program", KeyEvent.VK_P));
    mm.addSeparator();
    mm.add(createMenuItem("Exit", KeyEvent.VK_X));
    mb.add(mm);
    return mb;
}

// create Display Panel
private JPanel createDisplayPanel() {

    JPanel panel = new JPanel(new BorderLayout());

    if (mode == 1) {
        lbStats = new JLabel("sto");
        Font fontStats = lbStats.getFont().deriveFont(Font.PLAIN, 12f);
        lbStats.setFont(fontStats);
        lbStats.setForeground(colorDisableStats);
        lbStats.setBackground(Color.white);
        lbStats.setOpaque(true);
        panel.add(lbStats, BorderLayout.NORTH);
    }

    tfDisplay = new JTextField(frameWidth);
    Font fontDisplay = tfDisplay.getFont().deriveFont(Font.PLAIN, 25f);
    tfDisplay.setFont(fontDisplay);
    tfDisplay.setHorizontalAlignment(JTextField.RIGHT);
    tfDisplay.setBorder(null);
    tfDisplay.addKeyListener(this); // bac su kien khi an phim
    panel.add(tfDisplay, BorderLayout.CENTER);

    lbAns = new JLabel("0");
    Font fontAns = lbAns.getFont().deriveFont(Font.PLAIN, 35f);
    lbAns.setFont(fontAns);
    lbAns.setHorizontalAlignment(JLabel.RIGHT);
    lbAns.setBackground(Color.white);
    lbAns.setOpaque(true);

    panel.add(lbAns, BorderLayout.SOUTH);
    panel.setBorder(new EmptyBorder(0, 0, 10, 0));

    return panel;
}

```

```

}

// create button Basic Panel
private JPanel createButtonBasicPanel() {
    JPanel buttonBasicPanel = new JPanel(new GridLayout(5, 5, 3, 3));

    btnArr = addListButtonToPanel(lbButton, buttonBasicPanel);

    return buttonBasicPanel;
}

private JPanel createButtonAdvancedPanel() {

    // panel top
    JPanel panelLeft = new JPanel(new FlowLayout(FlowLayout.LEFT));
    ButtonGroup btnGroup = new ButtonGroup();
    radDeg = createRadio("Deg", true, panelLeft);
    btnGroup.add(radDeg);
    radRad = createRadio("Rad", false, panelLeft);
    btnGroup.add(radRad);

    JPanel panelRight = new JPanel(new GridLayout(1, 7, 3, 3));
    panelRight.setBorder(new EmptyBorder(0, 15, 10, 0));

    String lb[] = { "STO", "vA", "vB", "vC", "vD", "vE", "vF" };
    varElement = lb;
    btnArrSub = addListButtonToPanel(lb, panelRight);

    JPanel panel11 = new JPanel(new BorderLayout());
    panel11.add(panelLeft, BorderLayout.WEST);
    panel11.add(panelRight, BorderLayout.CENTER);

    // panel bottom
    JPanel panel12 = new JPanel(new GridLayout(5, 8, 3, 3));

    btnArr = addListButtonToPanel(lbButton, panel12);

    JPanel buttonAdvancedPanel = new JPanel(new BorderLayout());
    buttonAdvancedPanel.add(panel11, BorderLayout.NORTH);
    buttonAdvancedPanel.add(panel12, BorderLayout.CENTER);
    return buttonAdvancedPanel;
}

// create panel program
private JPanel createButtonProgramPanel() {

    // panel top
    JPanel panelLeft = new JPanel(new GridLayout(2, 2));
    // panelLeft.setPreferredSize(new Dimension(100, 50));
    ButtonGroup btnGroup = new ButtonGroup();
    radBin = createRadio("Bin", false, panelLeft);
    btnGroup.add(radBin);
    radOct = createRadio("Oct", false, panelLeft);
    btnGroup.add(radOct);
    radDec = createRadio("Dec", true, panelLeft);
    btnGroup.add(radDec);
    radHex = createRadio("Hex", false, panelLeft);
}

```

```

btnGroup.add(radHex);

JPanel panelRight = new JPanel(new BorderLayout());
lbDOH = new JLabel("0?? = 0?? = 0?");
lbDOH.setHorizontalAlignment(JLabel.RIGHT);
Font font = lbDOH.getFont().deriveFont(Font.PLAIN, 13f);
lbDOH.setFont(font);
panelRight.add(lbDOH, BorderLayout.PAGE_START);

// panel binary
String bin1 = "0000 0000 0000 0000 0000 0000 0000 0000?";
lbB = new JLabel(bin1);
lbB.setHorizontalAlignment(JLabel.RIGHT);
lbB.setFont(font);
panelRight.add(lbB, BorderLayout.PAGE_END);
panelRight.setBorder(new EmptyBorder(5, 5, 5, 5));
panelRight.setAlignmentX(RIGHT_ALIGNMENT);

JPanel panel1 = new JPanel(new BorderLayout());
panel1.add(panelLeft, BorderLayout.WEST);
panel1.add(panelRight, BorderLayout.CENTER);

// panel bottom
JPanel panel2 = new JPanel(new GridLayout(5, 8, 3, 3));

btnArr = addListButtonToPanel(lbButton, panel2);

JPanel buttonProgramPanel = new JPanel(new BorderLayout());
buttonProgramPanel.add(panel1, BorderLayout.NORTH);
buttonProgramPanel.add(panel2, BorderLayout.CENTER);
return buttonProgramPanel;
}

// ----- item of menu -----
// create menu
private JMenu createMenu(String title, int key) {
    JMenu menu = new JMenu(title);
    menu.setMnemonic(key);
    return menu;
}

// create menu item
private JMenuItem createMenuItem(String title, int key) {
    JMenuItem mi = new JMenuItem(title, key);
    mi.addActionListener(this);
    return mi;
}

// ----- item of display -----
// ----- item of buttonBasicPanel -----
// create button
private JButton createButton(String title) {
    JButton btn = new JButton(title);
    btn.addActionListener(this);
    btn.setMargin(new Insets(0, 0, 0, 0));
    return btn;
}

```

```

// create button and add to panel
private JButton createButton(String title, JPanel panel) {
    JButton btn = createButton(title);
    panel.add(btn);
    return btn;
}

// create Radio button
private JRadioButton createRadio(String title, boolean isSelect,
    JPanel panel) {
    JRadioButton rad = new JRadioButton(title);
    rad.addActionListener(this);
    rad.setSelected(isSelect);
    panel.add(rad);
    return rad;
}

// create List label Button
private void createListLabelButton(int mode) {
    if (mode == 0) {
        String s[] = { "C", "CE", "?", "(", ")", "7", "8", "9", "/", "?",
            "4", "5", "6", "*", "x2", "1", "2", "3", "-", "*10?", "0",
            "•", "Ans", "+", "=" };
        lbButton = s;
    }

    String s1[] = { "", "", "", "(", ")", "7", "8", "9", "/", "?", "4",
        "5", "6", "*", "2", "1", "2", "3", "-", "*10^", "0", ".",
        "Ans", "+", "" };
    mathElement = s1;

    return;
}
if (mode == 1) {
    String s[] = { "C", "CE", "?", "(", ")", "!", "a*b", "log", "7",
        "8", "9", "/", "?", "??", "nCr", "nPr", "4", "5", "6", "*",
        "x2", "x?", "sin", "arsin", "1", "2", "3", "-", "*10?", "?",
        "?", "cos", "arcos", "0", "•", "Ans", "+", "=", "e", "tan",
        "artan" };
    lbButton = s;

    String s1[] = { "", "", "", "(", ")", "!", "", "log ", "7", "8",
        "9", "/", "?", "??", "?", "?", "4", "5", "6", "*", "2",
        "^", "sin ", "arcsin ", "1", "2", "3", "-", "*10^", "?",
        "cos ", "arccos ", "0", ".", "Ans", "+", "", "e", "tan ",
        "arctan " };
    mathElement = s1;
    return;
}
if (mode == 2) {
    String s[] = { "C", "CE", "?", "(", ")", "!", "a*b", "Mod", "7",
        "8", "9", "/", "?", "<<", ">>", "And", "4", "5", "6", "*",
        "x2", "A", "B", "Or", "1", "2", "3", "?", "*10?", "C", "D",
        "Xor", "0", "•", "Ans", "+", "=", "E", "F", "Not" };
    lbButton = s;

    String s1[] = { "", "", "", "(", ")", "!", "", "Mod", "7", "8",

```

```

        "9", "/", "?", "?", "?", "4", "5", "6", "*", "2", "A",
        "B", "?", "1", "2", "3", "-", "*10^", "C", "D", "?", "0",
        ".", "Ans", "+", "", "E", "F", "¬" );
    mathElement = s1;
    return;
}
}

private JButton[] addListButtonToPanel(String lbArr[], JPanel panel) {
    JButton arr[] = new JButton[lbArr.length];
    for (int i = 0; i < lbArr.length; i++) {
        arr[i] = createButton(lbArr[i], panel);
    }
    return arr;
}

// ----- Action -----
// dat lai gia tri cho balan
private void resetValue() {
    balan = new Balan();
    balan.setError(false);
    if (mode == 2) {
        setRadix();
    }
    if (mode == 1) {
        setDegOrRad();
    }
}

// cho phep chen ky tu vao vi tri con tro
private void insertMathString(String str) {
    int index = tfDisplay.getCaretPosition();
    StringBuilder s = new StringBuilder(tfDisplay.getText() + ""); // copy
    s.insert(index, str); // insert text at index control
    String s1 = new String(s); // convert to string
    tfDisplay.setText(s1); // set text for jTextField
    tfDisplay.requestFocus(); // focus jTextField
    tfDisplay.setCaretPosition(index + str.length());
}

// tra ve ket qua
private void result() {
    balan.setError(false);
    ans = balan.valueMath(tfDisplay.getText());
    if (!balan.isError()) {
        balan.var[0] = ans;
        lbAns.setText(balan.numberToString(ans, balan.getRadix(),
            balan.getSizeRound()));
    } else {
        lbAns.setText("Math error!");
    }
    // hien thi ket qua trong che do Program
    setLabelProgram();
}

// hien thi ket qua trong che do Program
private void setLabelProgram() {

```

```

        if (mode == 2) {
            long num = 0;
            if (balan.isIntegerNumber(ans) && ans >= 0 && !balan.isError()) {
                num = (int) ans;
                lbDOH.setText(num + "??" =
                    + Long.toHexString(num).toUpperCase() + "?? = "
                    + Long.toOctalString(num) + "?");
            }

            String bi = Long.toBinaryString(num);
            StringBuilder s = new StringBuilder(bi);
            int i = s.length();
            while (i < 32) {
                s.insert(0, "0");
                i++;
            }
            bi = "";
            for (i = 1; i <= s.length(); i++) {
                bi += s.charAt(i - 1);
                if (i % 4 == 0 && i < s.length()) {
                    bi += " ";
                }
            }
            bi += "?";
        }

        lbB.setText(bi);

        enabaleLabelProgram(true);
    } else {
        enabaleLabelProgram(false);
    }
}

// dat che do hien thi hay khong hient thi cho Label Program
private void enabaleLabelProgram(boolean is) {
    if (mode == 2) {
        lbDOH.setEnabled(is);
        lbB.setEnabled(is);
    }
}

private void actionCE() {
    balan.setError(false);
    isSTO = false;
    if (mode == 1) {
        lbStats.setForeground(colorDisableStats);
    }
    tfDisplay.setText("");
    tfDisplay.requestFocus();
    enabaleLabelProgram(true);
    lbAns.setText("0");
    if (mode == 2) {
        lbDOH.setText("0?? = 0?? = 0?");
        lbB.setText("0000 0000 0000 0000 0000 0000 0000 0000?");
    }
}

```

```

private void actionDel() {
    int index = tfDisplay.getCaretPosition();
    StringBuilder s = new StringBuilder(tfDisplay.getText() + ""); // copy
    if (index > 0) {
        s.deleteCharAt(index - 1);
        String s1 = new String(s); // convert to string
        tfDisplay.setText(s1); // set text for jTextField
        tfDisplay.setCaretPosition(index - 1);
    }
    tfDisplay.requestFocus(); // focus jTextField
}

private void setDegOrRad() {
    if (radRad.isSelected()) {
        balan.setDegOrRad(false);
    }
    if (radDeg.isSelected()) {
        balan.setDegOrRad(true);
    }
    tfDisplay.requestFocus();
}

private void setRadix() {
    if (radBin.isSelected()) {
        balan.setRadix(2);
    }
    if (radOct.isSelected()) {
        balan.setRadix(8);
    }
    if (radDec.isSelected()) {
        balan.setRadix(10);
    }
    if (radHex.isSelected()) {
        balan.setRadix(16);
    }
    tfDisplay.requestFocus();
}

@Override
public void actionPerformed(ActionEvent evt) {
    String command = evt.getActionCommand();

    // action on menu
    if (command == "Basic") {
        mode = 0;
        changeMode();
        return;
    }
    if (command == "Advanced") {
        mode = 1;
        changeMode();
        setDegOrRad();
        return;
    }
    if (command == "Program") {
        mode = 2;
    }
}

```

```

changeMode();
setRadix();
return;
}
if (command == "Exit") {
    System.exit(0);
}

// action on radioButton
if (evt.getSource() == radRad || evt.getSource() == radDeg) {
    setDegOrRad();
    return;
}

// button dac biet
if (evt.getSource() == btnArr[0]) {
    resetValue();
    actionCE();
    return;
}
if (evt.getSource() == btnArr[1]) {
    actionCE();
    return;
}

if (evt.getSource() == btnArr[2]) {
    actionDel();
    return;
}

// neu trong che do Advanced
if (mode == 1) {
    if (evt.getSource() == btnArrSub[0]) {
        if (!isSTO) {
            isSTO = true;
            lbStats.setForeground(colorEnnableStar);
        } else {
            isSTO = false;
            lbStats.setForeground(colorDisableStats);
        }
    } else {

        for (int i = 1; i < btnArrSub.length; i++) {
            if (evt.getSource() == btnArrSub[i]) {
                if (isSTO) {
                    result();
                    tfDisplay.setCaretPosition(tfDisplay.getText()
                        .length());
                    insertMathString("?" + varElement[i]);
                    balan.var[i] = ans;
                    isSTO = false;
                } else {
                    insertMathString(varElement[i]);
                }
            }
        }
    }
    isSTO = false;
}

```

```

        lbStats.setForeground(colorDisableStats);
    }
}

if (mode == 2) {
    if (evt.getSource() == radBin || evt.getSource() == radOct
        || evt.getSource() == radDec || evt.getSource() == radHex) {
        setRadix();
        return;
    }
}

for (int i = 0; i < btnArr.length; i++) {
    if (evt.getSource() == btnArr[i] && !mathElement[i].equals("")) {
        insertMathString(mathElement[i]);
        return;
    }
}
if (command == "=") {
    result();
    return;
}
if (command == "a*b") {
    result();
    if (!balan.isError()) {
        lbAns.setText(balan.primeMulti(ans));
    }
    return;
}
}

@Override
public void keyPressed(KeyEvent ke) {
    if (ke.getKeyCode() == KeyEvent.VK_ENTER) {
        result();
    }
}

@Override
public void keyReleased(KeyEvent ke) {

}

@Override
public void keyTyped(KeyEvent ke) {

}
}

class ConvertNumber {

// chuyen chuoi he radix (he 2, 8, 10, 16) sang so double he 10
protected double stringRadixToDouble(String str, int radix) {
    str = str.toUpperCase();
    int len = str.length();
    int indexDot = -1;
    double num = -1;
}
}

```

```

String radixChar = "0123456789ABCDEF";

if (!isRadixString(str, radix)) {
    return num;
}
// tim vi tri dau cham de phan tach phan nguyen va phan thap phan
indexDot = str.indexOf(".");
if (indexDot > 0) {
    // chuoi phan nguyen
    String intString = str.substring(0, indexDot);
    // chuoi phan thap phan
    String floatString = str.substring(indexDot + 1, len);
    // chuyen phan nguyen vao num
    num = Integer.parseInt(intString, radix);
    // cong don phan thap phan
    for (int i = 0; i < floatString.length(); i++) {
        num += radixChar.indexOf(floatString.charAt(i))
            / Math.pow(radix, i + 1);
    }
} else {
    num = Integer.parseInt(str, radix);
}
return num;
}

// chuyen so he 10 sang he radix (2, 8, 16)
protected String doubleToStringRadix(double num, int radix, int countRound) {
    String str = "";
    String radixChar = "0123456789ABCDEF";
    long intNum = (long) num;
    double floatNum = (num - intNum);
    // doi so nguyen inNumber sang chuoi he radix
    String intString = Long.toString(intNum, radix).toUpperCase();
    String floatString = "";
    // doi phan thap phan
    while (floatNum > 0 && countRound > 0) {
        floatNum = (floatNum * radix);
        floatString += radixChar.charAt((int) floatNum);
        floatNum = floatNum - (int) floatNum;
        countRound--;
    }
    str = intString;
    if (floatString.length() > 0) {
        str += "." + floatString;
    }
    return str;
}

protected boolean isRadixString(String str, int radix) {
    str = str.toUpperCase();
    int len = str.length();
    int countDot = 0;
    String radixChar = "";
    if (radix == 2) {
        radixChar = "01.";
    } else if (radix == 8) {
        radixChar = "01234567.";
    }
}

```

```

} else if (radix == 16) {
    radixChar = "0123456789ABCDEF.";
} else {
    return false;
}
for (int i = 0; i < len; i++) {
    char c = str.charAt(i);
    if (c == '.') {
        countDot++;
    }
    if (radixChar.indexOf(c) < 0 || countDot > 1) {
        return false;
    }
}
return true;
}

// doi so numberDouble sang tich cua cac thua so nguyen to
protected String primeMulti(double numDouble) {
    if (numDouble >= 0 && (long) numDouble == numDouble) {
        long num = (long) numDouble;
        // powCount la so mu tach duoc
        int powCount = 0, i = 2, m = (int) Math.sqrt(num) + 1;
        String s = "";
        // duyet tu 2 den can bac 2 cua num
        while (i < m) {
            powCount = 0; // so mu cua i
            // tach num thanh i^n
            while (num > 0 && num % i == 0) {
                powCount++;
                num /= i;
            }
            // cong them phan tu neu no chia het
            if (powCount > 0) {
                if (s.length() > 0) {
                    s += "x"; // cong dau nhan
                }
                s += i; // cong so vao
                if (powCount > 1) {
                    s += myPowAxB(powCount); // cong so mu
                }
            }
            if (i == 2) {
                i++;
            } else {
                i += 2;
            }
        }
        // kiem tra num sau vong lap co la so nguyen to ko,
        // neu la so nguyen to thi cho them vao
        if (num > 1) {
            if (s.length() > 0) {
                s += "x";
            }
            s += num;
        } else if (s.length() == 0) {
            s += num;
        }
    }
}

```

```

        }
        return s;
    }
    return "-1";
}

private String myPowAxB(int number) {
    String numString = number + "";
    String numPow = "?123??????";
    String result = "";
    for (int i = 0; i < numString.length(); i++) {
        result += numPow.charAt(Integer.parseInt(numString.charAt(i) + ""));
    }
    return result;
}
class Balan {

    private boolean isError = false;
    private String varString[] = { "ans", "va", "vb", "vc", "vd", "ve", "vf" };
    private String constString[] = { "pi", "?", "e" }; // mang chuoi cac hang
    public double var[] = new double[varString.length];
    private double cons[] = { Math.PI, Math.PI, Math.E };
    private boolean isDegOrRad = true;
    private int radix = 10, sizeRound = 10;
    private ConvertNumber convertNumber = new ConvertNumber();

    public boolean isError() {
        return isError;
    }

    public void setError(boolean isError) {
        this.isError = isError;
    }

    public int getSizeRound() {
        return sizeRound;
    }

    public void setSizeRound(int sizeRound) {
        this.sizeRound = sizeRound;
    }

    public int getRadix() {
        return radix;
    }

    public void setRadix(int radix) {
        this.radix = radix;
    }

    public boolean isDegOrRad() {
        return isDegOrRad;
    }

    public void setDegOrRad(boolean isDegOrRad) {
        this.isDegOrRad = isDegOrRad;
    }
}

```

```

}

protected boolean isIntegerNumber(double num) {
    long a = (long) num;
    if (a == num) {
        return true;
    }
    return false;
}

private String myRound(double num, int size) {
    if (isIntegerNumber(num)) {
        return Long.toString((long) num);
    } else {
        int n = size - Long.toString((long) num).length();
        num = Math.round(num * Math.pow(10, n)) / Math.pow(10, n);
        if (isIntegerNumber(num)) {
            return Long.toString((long) num);
        } else {
            return Double.toString(num);
        }
    }
}

// num!
private long factorial(int num) {
    if (num >= 0) {
        long result = 1;
        for (int i = 1; i <= num; i++) {
            result *= i;
        }
        return result;
    }
    return -1;
}

// chinh hop chap b cua a=a!/(a-b)!=(a-b+1)(a-b+2)...a
private long permutation(int a, int b) {
    if (a < b) {
        return -1;
    }
    if (a >= 0 && b >= 0) {
        long result = 1;
        int c = a - b;
        for (int i = c + 1; i <= a; i++) {
            result *= i;
        }
        return result;
    }
    return -1;
}

// to hop chap b cua a
private long combination(int a, int b) {
    if (a < b) {
        return -1;
    }
}

```

```

        if (a >= 0 && b >= 0) {
            long result = 1;
            int c = a - b;
            if (c > b) {
                int temp = c;
                c = b;
                b = temp;
            }
            for (int i = b + 1; i <= a; i++) {
                result *= i;
            }
            result /= factorial(c);
            return result;
        }
        return -1;
    }

    private double convertToDeg(double num) {
        num = num * 180 / Math.PI;
        return num;
    }

    private double convertToRad(double num) {
        num = num * Math.PI / 180;
        return num;
    }

    // kiem tra chuoi s co la so khong (bien cung la so)
    protected boolean isNumber(String s) {
        if (radix != 10 && convertNumber.isRadixString(s, radix)) {
            return true;
        }
        if (isVarOrConst(s)) {
            return true;
        }
        try {
            Double.parseDouble(s);
        } catch (NumberFormatException e) {
            return false;
        }
        return true;
    }

    private boolean isNumber(char c) {
        String numberChar = ".0123456789abcdef";
        int index = numberChar.indexOf(c);
        if (radix == 10 && index >= 0 && index <= 10) {
            System.out.println(c + " is number");
            return true;
        }
        if (radix == 16 && index >= 0) {
            System.out.println(c + " is number");
            return true;
        }
        if (radix == 8 && index >= 0 && index <= 8) {
            System.out.println(c + " is number");
            return true;
        }
    }
}

```

```

        }
        if (radix == 2 && index >= 0 && index <= 2) {
            System.out.println(c + " is number");
            return true;
        }
        System.out.println(c + " isn't number");
        return false;
    }

// Chuoi sang so
private double stringToNumber(String s) {
    int index = indexVar(s);
    if (index >= 0) {
        return var[index];
    }
    index = indexConst(s);
    if (radix != 10) {
        if (convertNumber.isRadixString(s, radix)) {
            return convertNumber.stringRadixToDouble(s, radix);
        } else {
            isError = true;
            System.out.println("Error number in radix = " + radix);
        }
    }
    if (index >= 0) {
        return cons[index];
    }
    if (s.charAt(s.length() - 1) == '.') {
        isError = true;
        System.out.println("Error number have '.'");
        return -1;
    }
    try {
        return Double.parseDouble(s);
    } catch (Exception e) {
        isError = true;
        System.out.println("Error parse number");
    }
    return -1;
}

public String numberToString(double num, int radix, int size) {
    if (radix != 10) {
        return convertNumber.doubleToStringRadix(num, radix, size);
    }
    return myRound(num, size);
}

private int indexVar(String s) {
    for (int i = 0; i < varString.length; i++) {
        if (s.equals(varString[i])) {
            return i;
        }
    }
    return -1;
}

```

```

private int indexConst(String s) {
    for (int i = 0; i < constString.length; i++) {
        if (s.equals(constString[i])) {
            return i;
        }
    }
    return -1;
}

private boolean isVarOrConst(String s) {
    if (indexConst(s) >= 0 || indexVar(s) >= 0) {
        return true;
    }
    return false;
}

// kiem tra xem co phai toan tu
private boolean isOperator(String s) {
    String operator[] = { "+", "-", "*", "/", "?", "?", "ncr", "npr", "^",
        "~", "?", "sqrt", "??", "n?", "!", "%", ")",
        "(", "2", "sin",
        "cos", "tan", "arcsin", "arccos", "arctan", "log", "?", "sto",
        "mod", "and", "or", "xor", "not", "?", "?", "?", "¬", "<<",
        ">>", "?", "?" };
    Arrays.sort(operator);
    if (Arrays.binarySearch(operator, s) > -1) {
        return true;
    } else {
        return false;
    }
}

// thiet lap thu tu uu tien
private int priority(String s) {
    int p = 1;
    if (s.equals("?") || s.equals("sto")) {
        return p;
    }
    p++;
    if (s.equals("+") || s.equals("-")) {
        return p;
    }
    p++;
    if (s.equals("*") || s.equals("/")) {
        return p;
    }
    p++;
    if (s.equals("and") || s.equals("?)") || s.equals("or") || s.equals("?)")
        || s.equals("xor") || s.equals("?)") || s.equals("mod")
        || s.equals(">>") || s.equals("<<") || s.equals("?)")
        || s.equals("?)")) {
        return p;
    }
    p++;
    if (s.equals("?)") || s.equals("?)") || s.equals("ncr")
        || s.equals("npr")) {
        return p;
    }
}

```

```

        }
        p++;
        if (s.equals("not") || s.equals("¬")) {
            return p;
        }
        p++;
        if (s.equals("¬")) {
            return p;
        }
        p++;
        if (s.equals("sin") || s.equals("cos") || s.equals("tan")
            || s.equals("arcsin") || s.equals("arccos")
            || s.equals("arctan") || s.equals("log")) {
            return p;
        }
        p++;
        if (s.equals("?") || s.equals("n?") || s.equals("??") || s.equals("!")
            || s.equals("^") || s.equals("²") || s.equals("sqrt")) {
            return p;
        }
        p++;
        return 0;
    }

    // kiem tra xem co phai la phep toan 1 ngoi
    private boolean isOneMath(String c) {
        String operator[] = { "sin", "cos", "tan", "arcsin", "arccos",
            "arctan", "?", "sqrt", "(", "~", "not", "¬", "log" };
        Arrays.sort(operator);
        if (Arrays.binarySearch(operator, c) > -1) {
            return true;
        } else {
            return false;
        }
    }

    // kiem tra xem co phai phep toan dang sau
    private boolean isPostOperator(String s) {
        String postOperator[] = { "!", "²" };
        for (int i = 0; i < postOperator.length; i++) {
            if (s.equals(postOperator[i])) {
                System.out.println(s + "isPostOperator");
                return true;
            }
        }
        System.out.println(s + "no isPostOperator");
        return false;
    }

    // kiem tra xem cac ky tu lien nhau co la 1 tu khong
    private boolean isWord(char c1, char c2) {
        char word[][] = { { '?', '?' }, { 'n', '?' }, { 'p', 'i' },
            { 's', 'i', 'n' }, { 'c', 'o', 's' }, { 't', 'a', 'n' },
            { 'a', 'r', 'c', 's', 'i', 'n' },
            { 'a', 'r', 'c', 's', 'o', 's' },
            { 'a', 'r', 'c', 't', 'a', 'n' }, { 'a', 'n', 's' },
            { 's', 'q', 'r', 't' }, { 'n', 'c', 'r' }, { 'n', 'p', 'r' },

```

```

        { 's', 't', 'o' }, { 'a', 'n', 'd' }, { 'o', 'r' },
        { 'x', 'o', 'r' }, { 'n', 'o', 't' }, { 'm', 'o', 'd' },
        { '<', '<' }, { '>', '>' } };
    for (int i = 0; i < word.length; i++) {
        for (int j = 0; j < word[i].length; j++) {
            for (int k = j + 1; k < word[i].length; k++) {
                if (c1 == word[i][j] && c2 == word[i][k]) {
                    System.out.println("is word: " + c1 + " " + c2);
                    return true;
                }
            }
        }
    }
    System.out.println("is't word:" + c1 + " " + c2);
    return false;
}

// kiem tra chuoi s co la 1 tu khong
private boolean isWord(String s) {
    String word[] = { "???", "n?", "pi", "sin", "cos", "tan", "arcsin",
                      "arccos", "arctan", "sqrt", "ncr", "npr", "sto", "and", "or",
                      "xor", "not", "mod", "<<", ">>", "va", "vb", "vc", "vd", "ve",
                      "ans" };
    for (int i = 0; i < word.length; i++) {
        if (s.equals(word[i])) {
            return true;
        }
    }
    return false;
}

// chuan hoa chuoi
private String standardize(String s) {
    s = s.trim();
    s = s.replaceAll("\\s+", " ");
    return s;
}

// cat chuoi thanh cac phan tu
private String[] trimString(String s) {
    String temp[] = s.split(" ");
    return temp;
}

private String standardizeMath(String[] s) { // chuan hoa bieu thuc
    String s1 = "";
    // standardize(s);

    int open = 0, close = 0;
    for (int i = 0; i < s.length; i++) {
        if (s[i].equals("(")) {
            open++;
        } else if (s[i].equals(")")) {
            close++;
        }
    }
}

```

```

for (int i = 0; i < s.length; i++) {
    // chuyen ...)(... thanh ...)*(...)
    if (i > 0 && isOneMath(s[i])
        && (s[i - 1].equals(")") || isNumber(s[i - 1]))) {
        s1 = s1 + "* ";
    }
    // 3!2!
    if (i > 0 && isPostOperator(s[i - 1]) && isNumber(s[i])) {
        s1 = s1 + "* ";
    }
    // so duong
    if ((i == 0 || (i > 0 && !isNumber(s[i - 1])
        && !s[i - 1].equals("(")) && !isPostOperator(s[i - 1]))
        && (s[i].equals("+"))
        && (isNumber(s[i + 1]) || s[i + 1].equals("+)))) {
        continue;
    }
    // check so am
    if ((i == 0 || (i > 0 && !isNumber(s[i - 1])
        && !s[i - 1].equals("(")) && !isPostOperator(s[i - 1]))
        && (s[i].equals("-"))
        && (isNumber(s[i + 1]) || s[i + 1].equals("-)))) {
        s1 = s1 + "~ ";
    }
    } // VD hoac 6? , ...)? chuyen sang 6*? , ...) *?
    else if (i > 0
        && ((isNumber(s[i - 1]) || s[i - 1].equals("(")) &&
isVarOrConst(s[i]))) {
        s1 = s1 + "* " + s[i] + " ";
    } else {
        s1 = s1 + s[i] + " ";
    }
}
// them cac dau ")" vao cuoi neu thieu
for (int i = 0; i < (open - close); i++) {
    s1 += ")";
}
System.out.println("standardizeMath: " + s1);
return s1;
}
// xu ly bieu thuc nhap vao thanh cac phan tu
private String processInput(String sMath) {
    sMath = sMath.toLowerCase();
    sMath = standardize(sMath); // chuan hoa bieu thuc
    String s = "", temp = "";
    for (int i = 0; i < sMath.length(); i++) {
        // is'nt number
        if (!isNumber(sMath.charAt(i))
            || (i < sMath.length() - 1 && isWord(sMath.charAt(i),
                sMath.charAt(i + 1)))) {
            s += " " + temp;
            temp = "" + sMath.charAt(i);
            // is operator and isn't word
            if (isOperator(sMath.charAt(i) + "") && i < sMath.length() - 1
                && !isWord(sMath.charAt(i), sMath.charAt(i + 1))) {
                s += " " + temp;
            }
        }
    }
}

```

```

        temp = "";
    } else { // isn't operator but is word
        i++;
        while (i < sMath.length()
                && !isNumber(sMath.charAt(i))
                && (!isOperator(sMath.charAt(i) + ""))
                || (i < sMath.length() - 1 && isWord(
                    sMath.charAt(i - 1), sMath.charAt(i)))) {
            temp += sMath.charAt(i);
            i++;
            if (isWord(temp)) {
                s += " " + temp;
                temp = "";
                break;
            }
        }
        i--;
        s += " " + temp;
        temp = "";
    }
} else { // is number
    temp = temp + sMath.charAt(i);
}
}
s += " " + temp;

System.out.println("process input 1 : " + s);
s = standardize(s);
s = standardizeMath(trimString(s));
System.out.println(s);
return s;
}

private String postFix(String math) {
    String[] elementMath = trimString(math);

    String s1 = "";
    Stack<String> S = new Stack<String>();
    for (int i = 0; i < elementMath.length; i++) { // duyet cac phan tu
        if (!isOperator(elementMath[i])) // neu khong la toan tu
        {
            s1 = s1 + elementMath[i] + " "; // xuat elem vao s1
        } else { // c la toan tu
            if (elementMath[i].equals("(")) {
                S.push(elementMath[i]); // c la "(" -> day phan tu vao Stack
            } else {
                if (elementMath[i].equals(")")) { // c la ")"
                    // duyet lai cac phan tu trong Stack
                    String temp = "";
                    do {
                        temp = S.peek();
                        if (!temp.equals("(")) {
                            s1 = s1 + S.peek() + " "; // trong khi c1 != "("
                        }
                        S.pop();
                    } while (!temp.equals("("));
                } else {

```

```

        // Stack khong rong va trong khi phan tu trong Stack co
        // do uu tien >= phan tu hien tai
        while (!S.isEmpty()
            && priority(S.peek()) >= priority(elementMath[i])
                && !isOneMath(elementMath[i])) {
            s1 = s1 + S.pop() + " ";
        }
        S.push(elementMath[i]); // dua phan tu hien tai vao
        // Stack
    }
}
}
while (!S.isEmpty()) {
    s1 = s1 + S.pop() + " "; // Neu Stack con phan tu thi day het vao s1
}
System.out.println("balan: " + s1);
return s1;
}

public Double valueMath(String math) {
    math = processInput(math);
    math = postFix(math);
    String[] elementMath = trimString(math);
    Stack<Double> S = new Stack<Double>();
    double num = 0.0;
    double ans = 0.0;
    System.out.println("Element math: ");
    for (int i = 0; i < elementMath.length; i++) {
        System.out.print(elementMath[i] + "\t");
        if (!isOperator(elementMath[i])) {
            S.push(stringToNumber(elementMath[i]));
        } else { // toan tu
            if (S.isEmpty()) {
                System.out.println("Stack is empty ^^ ");
                isError = true;
                return 0.0;
            }
            double num1 = S.pop();
            String ei = elementMath[i];
            if (ei.equals("~")) {
                num = -num1;
            } else if (ei.equals("sin")) {
                if (isDegOrRad) {
                    num1 = convertToRad(num1);
                }
                num = Math.sin(num1);
            } else if (ei.equals("cos")) {
                if (isDegOrRad) {
                    num1 = convertToRad(num1);
                }
                num = Math.cos(num1);
            } else if (ei.equals("tan")) {
                if (isDegOrRad) {
                    num1 = convertToRad(num1);
                }
                num = Math.tan(num1);
            }
        }
    }
}

```

```

} else if (ei.equals("arcsin")) {
    num = Math.asin(num1);
    if (isDegOrRad) {
        num = convertToDeg(num);
    }
} else if (ei.equals("arccos")) {
    num = Math.acos(num1);
    if (isDegOrRad) {
        num = convertToDeg(num);
    }
} else if (ei.equals("arctan")) {
    num = Math.atan(num1);
    if (isDegOrRad) {
        num = convertToDeg(num);
    }
} else if (ei.equals("log")) {
    num = Math.log10(num1);
} else if (ei.equals("%")) {
    num = num1 / 100;
} else if (ei.equals("2")) {
    num = Math.pow(num1, 2);
} else if (ei.equals("?") || ei.equals("sqrt")) {
    if (num1 >= 0) {
        num = Math.sqrt(num1);
    } else {
        isError = true;
        System.out.println("Error sqrt");
        return 0.0;
    }
} else if (ei.equals("not") || ei.equals("¬") || ei.equals("!")) {
    if (isIntegerNumber(num1) && num1 >= 0) {
        if (ei.equals("not") || ei.equals("¬")) {
            num = ~((long) num1);
        } else if (ei.equals("!")) {
            num = factorial((int) num1);
        }
    }
} else if (!S.empty()) {
    double num2 = S.peek();

    if (ei.equals("?") || ei.equals("sto")) {
        if (indexVar(elementMath[i - 1]) >= 0) {
            var[indexVar(elementMath[i - 1])] = num2;
            ans = num2;
            return ans;
        } else {
            isError = true;
            System.out.println("Error sto");
            return 0.0;
        }
    } else if (ei.equals("+")) {
        num = num2 + num1;
        S.pop();
    } else if (ei.equals("-")) {
        num = num2 - num1;
        S.pop();
    } else if (ei.equals("*")) {

```

```

        num = num2 * num1;
        S.pop();
    } else if (ei.equals("/")) {
        if (num1 != 0) {
            num = num2 / num1;
        } else {
            isError = true;
            return 0.0;
        }
        S.pop();
    } else if (ei.equals("^")) {
        num = Math.pow(num2, num1);
        S.pop();
    } else if (ei.equals("??") || ei.equals("n?")) {
        num = Math.pow(num1, (double) 1 / num2);
        S.pop();
    } else if (isIntegerNumber(num1) && isIntegerNumber(num2))
{
    if (ei.equals("ncr") || ei.equals("?")) {
        num = combination((int) num2, (int) num1);
        S.pop();
    } else if (ei.equals("npr") || ei.equals("?")) {
        num = permutation((int) num2, (int) num1);
        S.pop();
    } else if (ei.equals("and") || ei.equals("?")) {
        num = (long) num2 & (long) num1;
        S.pop();
    } else if (ei.equals("or") || ei.equals("?")) {
        num = (long) num2 | (long) num1;
        S.pop();
    } else if (ei.equals("xor") || ei.equals("?")) {
        num = (long) num2 ^ (long) num1;
        S.pop();
    } else if (ei.equals("mod")) {
        num = (long) num2 % (long) num1;
        S.pop();
    } else if (ei.equals("<<") || ei.equals("?")) {
        num = (long) num2 << (long) num1;
        S.pop();
    } else if (ei.equals(">>") || ei.equals("?")) {
        num = (int) num2 >> (int) num1;
        S.pop();
    }
}
} else {
    System.out.println("Error stack empty");
    isError = true;
    return 0.0;
}
S.push(num);
}
ans = S.pop();
System.out.println("\nans = " + ans + "\t radix = " + radix);
return ans;
}

```

```
    public String primeMulti(double num) {
        return convertNumber.primeMulti(num);
    }

public class Calculator {
    public static void main(String[] args) {
        new CalculatorTheme();
    }
}
```