FPT Education

FPT POLYTECHNIC

LẬP TRÌNH JAVA 2

BÀI 3: JMENU, ĐỌC/GHI FILE

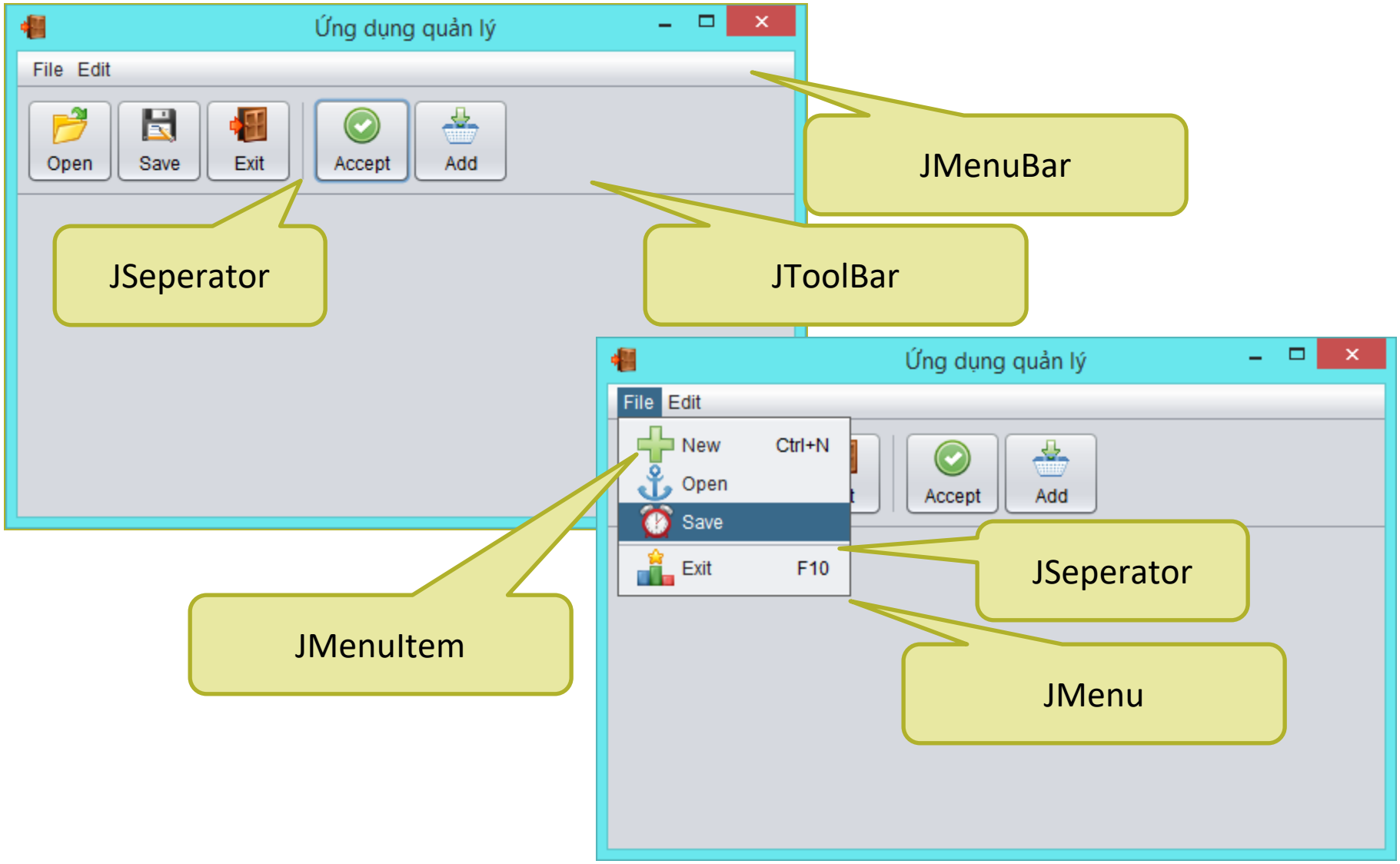
PHẦN 1

www.poly.edu.vn

Mục tiêu

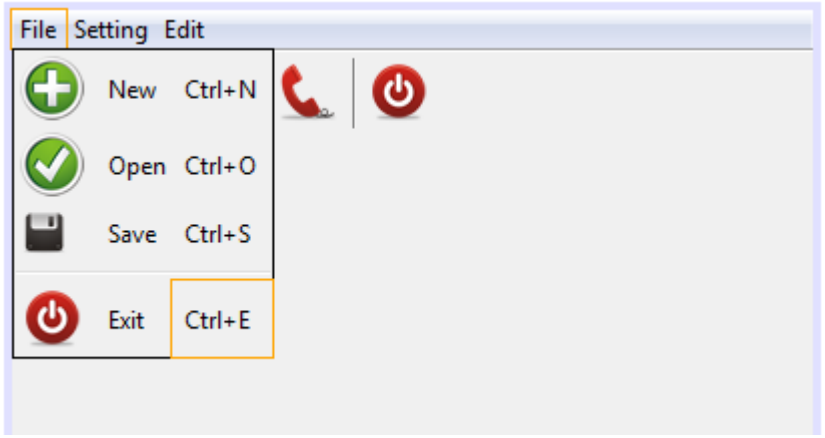
- Kết thúc bài học này bạn có khả năng
  - JMenu
  - JMenuBar
  - JMenuItem
  - JPopupMenu
  - JToolBar
  - Read/write file

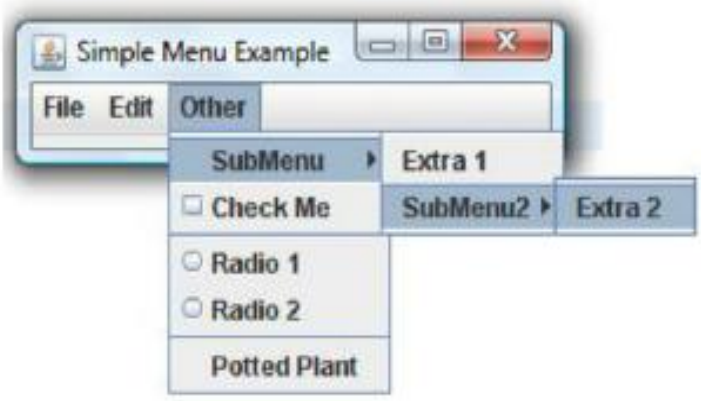
MENU...


Ứng dụng quản lý
File Edit
Open
Save
Exit
Accept
Add
JMenuBar
JSeperator
JToolBar
Ứng dụng quản lý
File Edit
New Ctrl+N
Open
Save
Exit F10
Accept
Add
JMenuItem
JSeperator
JMenu

JMenu

- JMenu có thể được dùng giống như một layout, để quản lý các Component
- Chỉ được phép có 1 lựa chọn tại 1 thời điểm
- Icon có thể dùng để thay thế cho các menu items
- Hầu hết các component chuẩn đều có thể là Menu Item (radio button...)
- Có thể gán phím tắt cho các Menu Item





JMenu

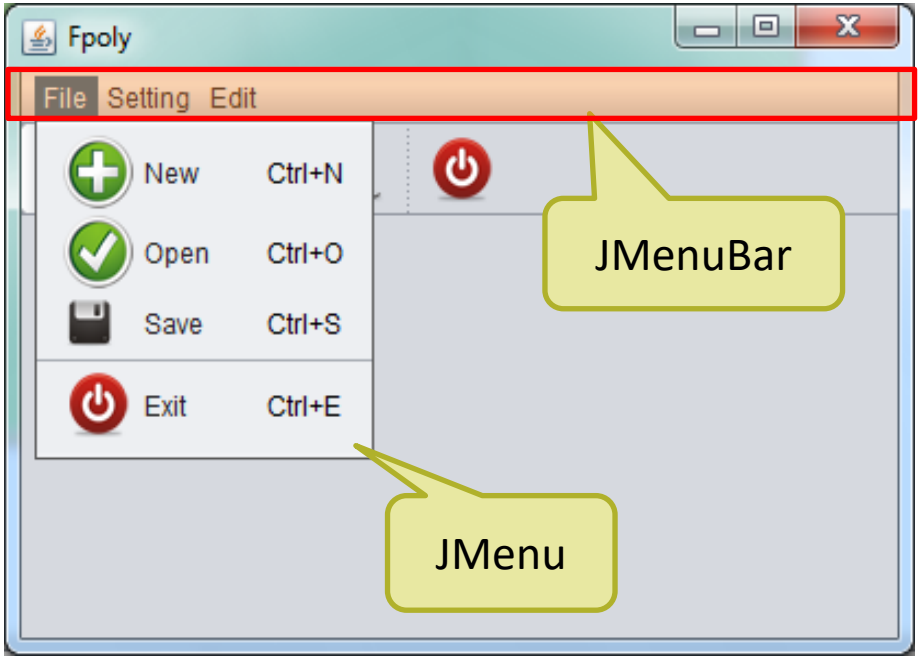
- Khởi tạo
  - JMenu()
  - Khởi tạo một menu mới không có tiêu đề
  - JMenu(Action a)
  - Khởi tạo một menu mới với các thuộc tính lấy từ Action a.
  - JMenu(String s)
  - Khởi tạo một menu mới với tiêu đề là s
  - JMenu(String s, boolean b)
  - Khởi tạo một menu mới với tiêu đề là s và qui định là một menu tách rời hay không.
- Events: ChangeEvent
  - void addChangel_istener(Changel_istener listener)
  - void removeChangeListener(ChangeListener listener)

Các phương thức

JMenuItem	add(Action a)
Component	add(Component c)
Component	add(Component c, int index)
JMenuItem	add(JMenuItem menuItem)
JMenuItem	add(String s)
void	addMenuListener(MenuListener l)
void	addSeparator()
void	applyComponentOrientation(ComponentOrientation o)
PropertyChangeListener	createActionChangeListener(JMenuItem b)
JMenuItem	createActionComponent(Action a)
JMenu.WinListener	createWinListener(JPopupMenu p)
void	doClick(int pressTime)
void	fireMenuCanceled()
void	fireMenuDeselected()
…	…

JMenuBar

- Dùng để tạo ra 1 MenuBar theo chiều ngang của component với 0, 1 hoặc nhiều phần tử gắn lên đó
- Dùng phương thức add để thêm vào các JMenu trên JMenuBar
- JMenubar sẽ hiển thị các JMenu theo thứ tự từ trái sang phải.
- Khởi tạo
  - JMenuBar()
- Sự kiện
  - ActionListener



Các phương thức

JMenu	add(JMenu c)
void	addNotify()
AccessibleContext	getAccessibleContext()
Component	getComponent()
int	getComponentIndex(Component c)
JMenu	getHelpMenu()
Insets	getMargin()
JMenu	getMenu(int index)
int	getMenuCount()
SingleSelectionModel	getSelectionModel()
MenuElement[]	getSubElements()
MenuBarUI	getUI()
String	getUIClassID()
boolean	isBorderPainted()
boolean	isSelected()

JMenuBar

- Có thể gắn 1 menu bar lên frame theo 1 trong 2 cách sau:
- Sử dụng setJMenuBar()
  - JFrame frame = new JFrame;
  - JMenuBar menuBar = new JMenuBar();
    frame.setJMenuBar(menuBar);
- Dùng layout để định vị
  - menuBar.setBorder(new BevelBorder(BevelBorder.RAISED));
    frame.getContentPane( ).add(menuBar, BorderLayout.SOUTH);

JMenuItem

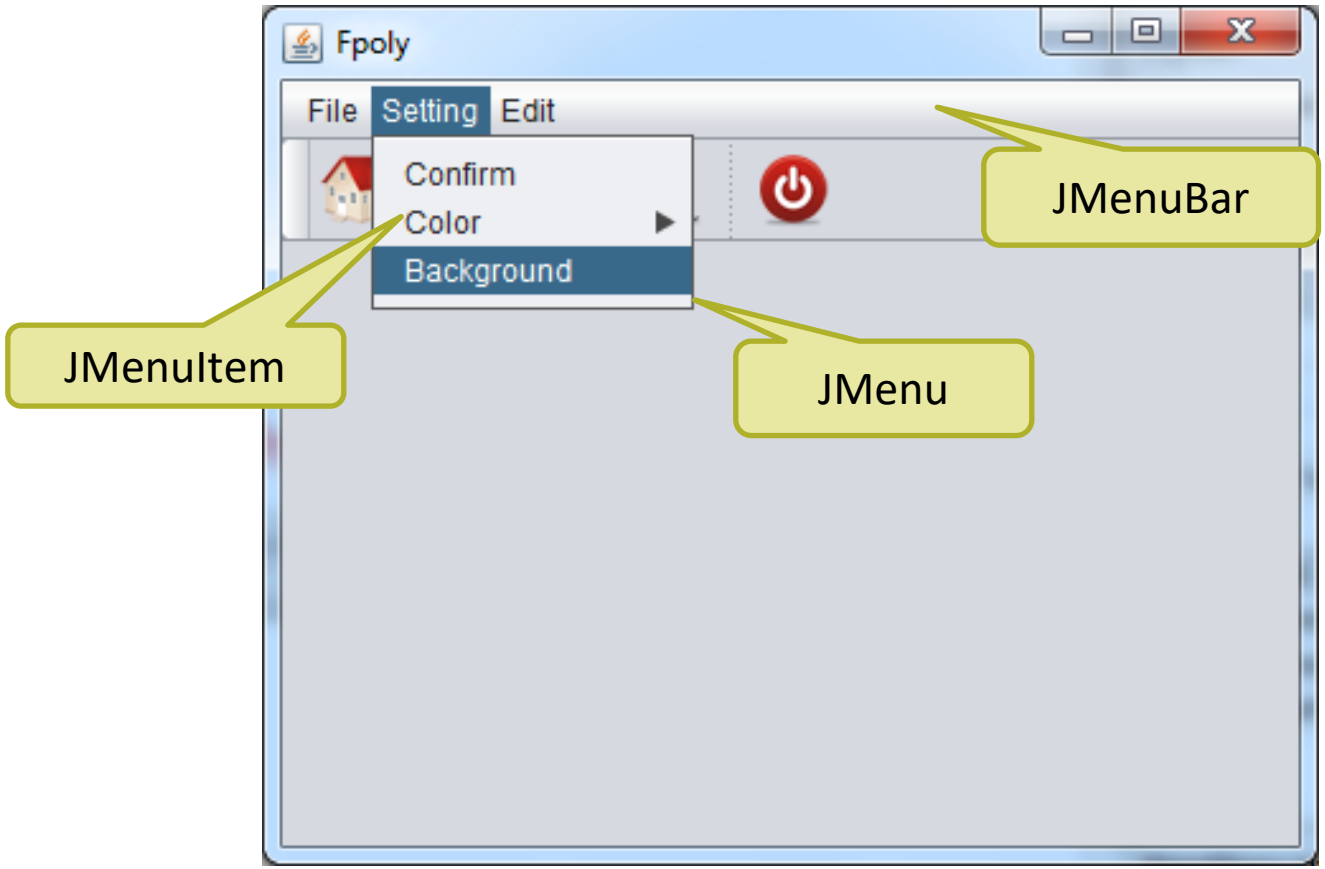
- JMenuItem là 1 loại nút đặc biệt (xử lý mouseListenner)
- Khởi tạo
  - JMenuItem()
    - Khởi tạo một JMenuItem không có tiêu đề và icon.
  - JMenuItem(Action a)
    - Khởi tạo một menu item với thuộc tính là a.
  - JMenuItem(Icon icon)
    - Khởi tạo một JMenuItem với icon.
  - JMenuItem(String text)
    - Khởi tạo một JMenuItem với tiêu đề là text.
  - JMenuItem(String text, Icon icon)
    - Khởi tạo một JMenuItem với tiêu đề và icon.
  - JMenuItem(String text, int mnemonic)
    - Khởi tạo một JMenuItem với tiêu đề và phím tắt

Các phương thức

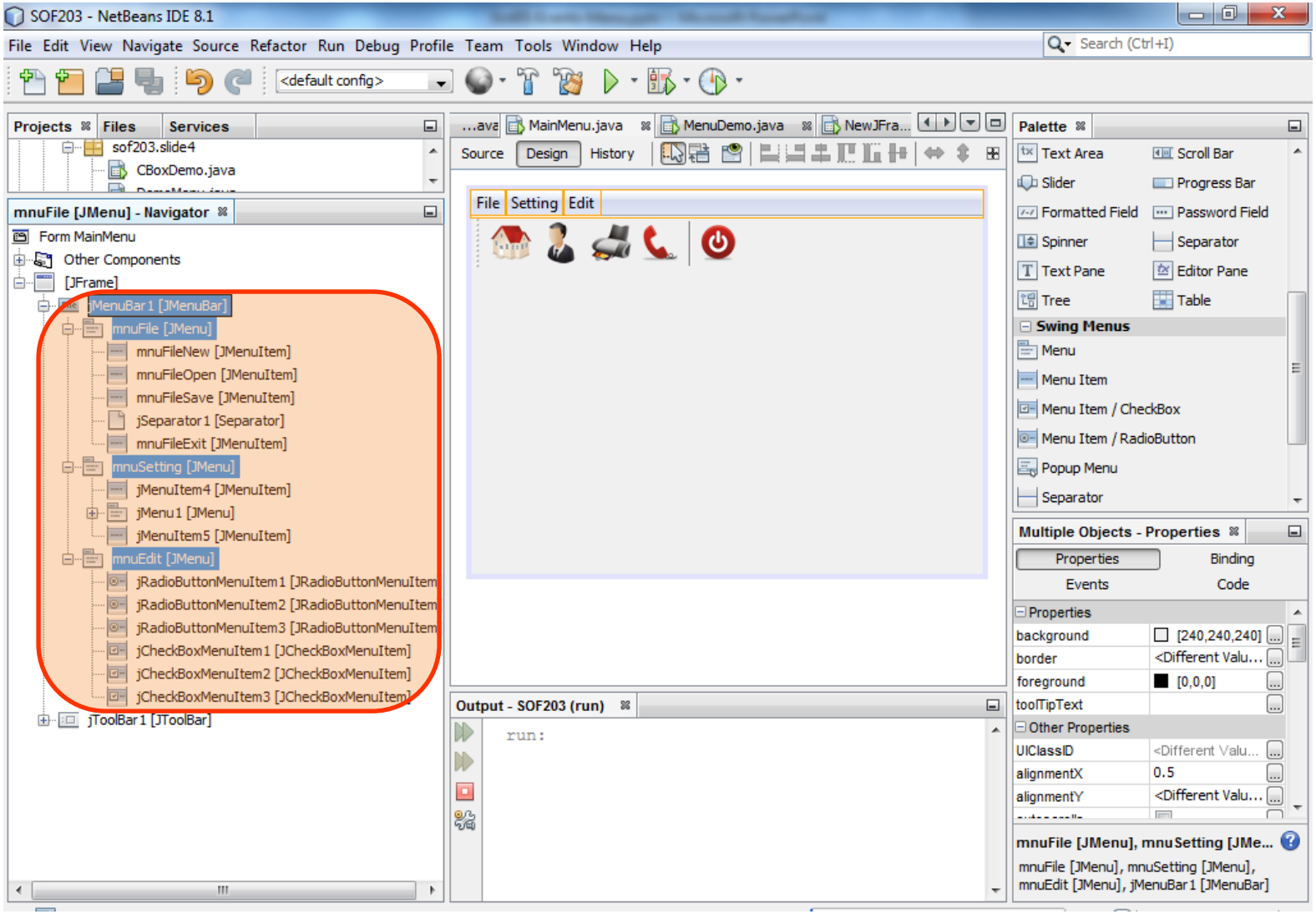
void	actionPropertyChanged(Action action, String propertyName)
void	addMenuDragMouseListener(MenuDragMouseListener l)
void	addMenuKeyListener(MenuKeyListener l)
void	configurePropertiesFromAction(Action a)
void	fireMenuDragMouseDragged(MenuDragMouseEvent event)
void	fireMenuDragMouseEntered(MenuDragMouseEvent event)
void	fireMenuDragMouseExited(MenuDragMouseEvent event)
void	fireMenuDragMouseReleased(MenuDragMouseEvent event)
. . .	. . .

JMenuItem

- Các JMenuItem nằm trong JMenu, JMenuBar chứa các JMenu



JMenuItem


SOF203 - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Projects
Files
Services
sof203.slide4
CBoxDemo.java
mnuFile [JMenu] - Navigator
Form MainMenu
Other Components
[JFrame]
jMenuBar1 [JMenuBar]
mnuFile [JMenu]
mnuFileNew [JMenuItem]
mnuFileOpen [JMenuItem]
mnuFileSave [JMenuItem]
jSeparator1 [Separator]
mnuFileExit [JMenuItem]
mnuSetting [JMenu]
jMenuItem4 [JMenuItem]
jMenu1 [JMenu]
jMenuItem5 [JMenuItem]
mnuEdit [JMenu]
jRadioButtonMenuItem1 [JRadioButtonMenuItem]
jRadioButtonMenuItem2 [JRadioButtonMenuItem]
jRadioButtonMenuItem3 [JRadioButtonMenuItem]
jCheckBoxMenuItem1 [JCheckBoxMenuItem]
jCheckBoxMenuItem2 [JCheckBoxMenuItem]
jCheckBoxMenuItem3 [JCheckBoxMenuItem]
jToolBar1 [JToolBar]
MainMenu.java
MenuDemo.java
NewJFra...
Source
Design
History
File
Setting
Edit
Output - SOF203 (run)
run:
Palette
Text Area
Scroll Bar
Slider
Progress Bar
Formatted Field
Password Field
Spinner
Separator
Text Pane
Editor Pane
Tree
Table
Swing Menus
Menu
Menu Item
Menu Item / CheckBox
Menu Item / RadioButton
Popup Menu
Separator
Multiple Objects - Properties
Properties
Binding
Events
Code
background
[240,240,240]
border
<Different Valu...
foreground
[0,0,0]
toolTipText
Other Properties
UIClassID
alignmentX
0.5
alignmentY
mnuFile [JMenu], mnuSetting [JMe...
mnuFile [JMenu], mnuSetting [JMenu],
mnuEdit [JMenu], jMenuBar1 [JMenuBar]

JPopupMenu

- Là một loại menu đặc biệt mà không cần phải gắn vào menu bar
- Nó có thể được hiển thị ở bất cứ vị trí nào trên thành phần chứa.
- Bạn có thể thêm, chèn một JMenuItem, 1 component hay 1 Action tùy ý vào popup menu này với phương thức add() và insert()
- JPopupMenu sẽ gán cho mỗi menu item 1 số thứ tự rồi gắn chúng vào popup menu theo layout mà nó có
- Bạn cũng có thể thêm 1 separator vào popup menu với phương thức addSeparator()

JPopupMenu

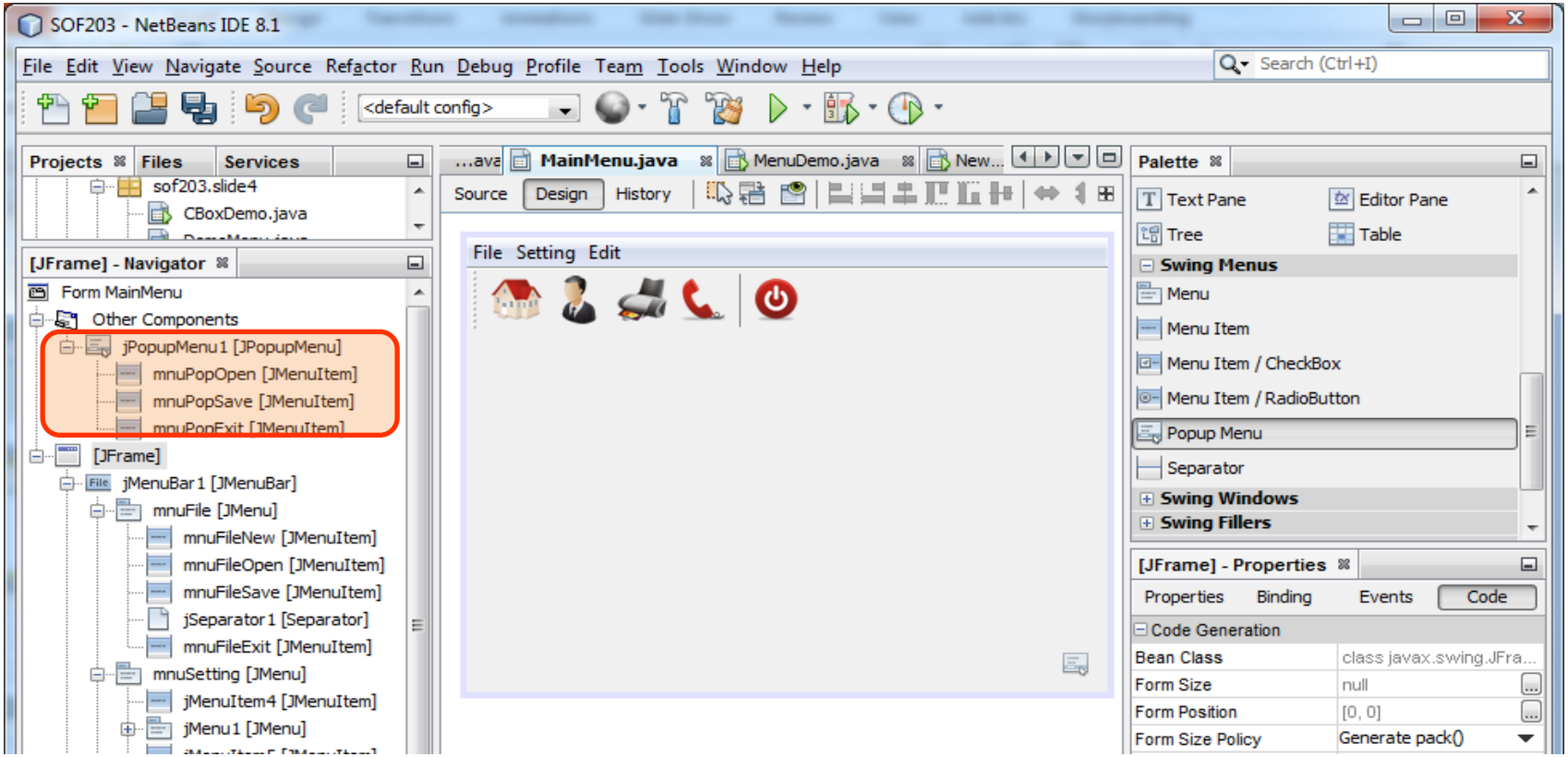
- Khởi tạo
  - public JPopupMenu()
  - public JPopupMenu(String title)
  - public JMenuItem add(JMenuItem menuItem)
  - public Component add(Component c)
  - public JMenuItem add(Action a)
  - public JMenuItem insert(Action a, int index)
  - public Component insert(Component comp, int index)
  - public void addSeparator()

JPopupMenu

- Để hiển thị popup menu dùng phương thức show()

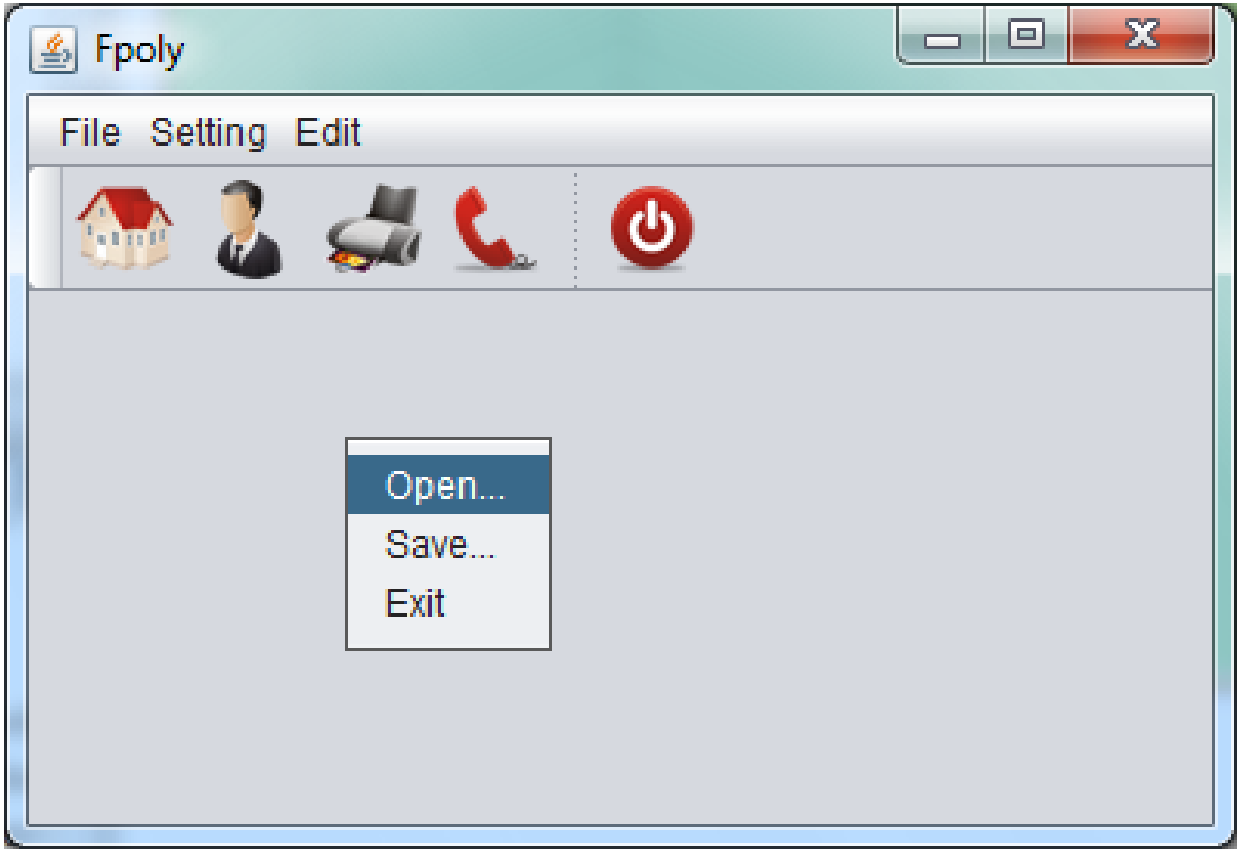
```
public void processMouseEvent(MouseEvent e) {
        if (e.isPopupTrigger()) {
                popup.show(this, e.getX(), e.getY());
        } else {
                super.processMouseEvent(e);
        }
```

Ví dụ JPopupMenu




```
private void formMouseReleased(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    if(evt.isPopupTrigger()) {
        jPopupMenu1.show(this, evt.getX(), evt.getY());
    }
}
```

Ví dụ JPopupMenu



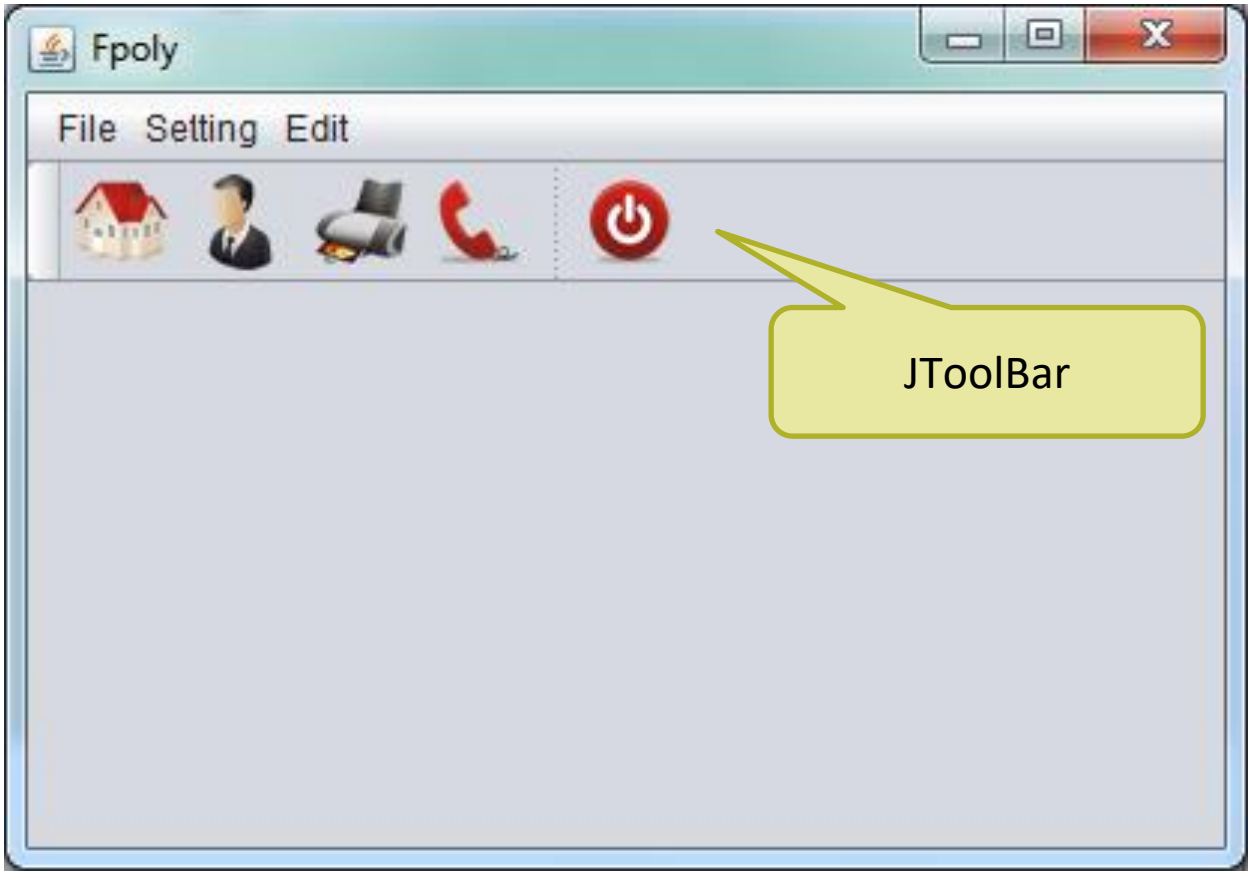
JToolBar

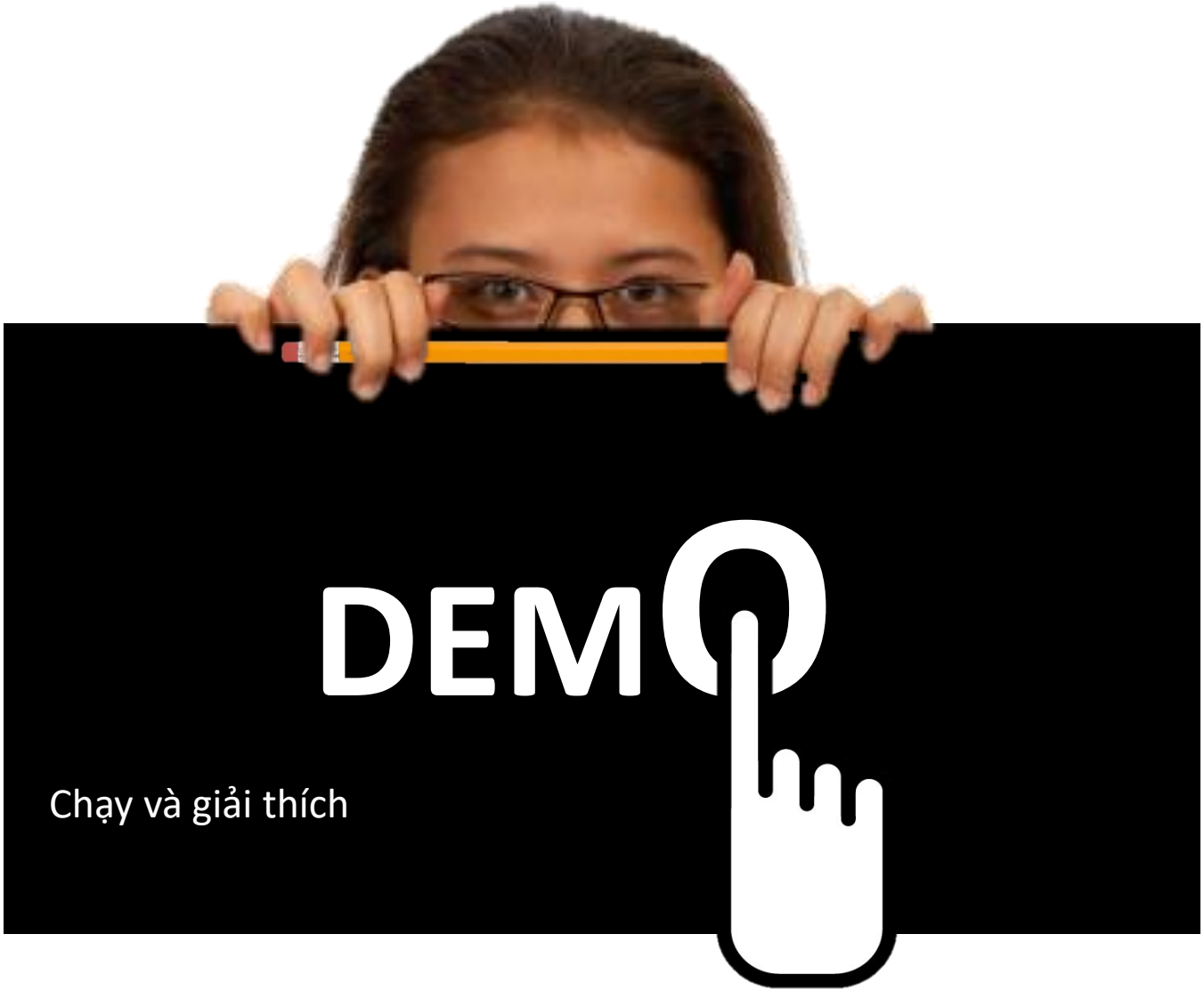
- Là lớp chứa cho nhiều thành phần khác
- Khi 1 component được gắn vào tool bar nó sẽ được định vị từ trái sang phải theo chỉ mục của nó
- Khởi tạo
  - JToolBar()
    - Khởi tạo một tool bar; Chiều mặc định nằm ngang.
  - JToolBar(int orientation)
    - Khởi tạo một tool bar với chiều ngang/dọc.
  - JToolBar(String name)
    - Khởi tạo một tool bar với tên gọi name.
  - JToolBar(String name, int orientation)
- Sự kiện
  - Dùng Sự kiện PropertyChangeEvent khi có 1 sự thay đổi giá trị của các thuộc tính

Các phương thức

JButton	add(Action a)
protected void	addImpl(Component comp, Object constraints, int index)
void	addSeparator()
void	addSeparator(Dimension size)
protected PropertyChangeListener	createActionChangeListener(JButton b)
protected JButton	createActionComponent(Action a)
AccessibleContext	getAccessibleContext()
Component	getComponentAtIndex(int i)
int	getComponentIndex(Component c)
Insets	getMargin()
int	getOrientation()
ToolBarUI	getUI()
String	getUIClassID()
boolean	isBorderPainted()
boolean	isFloatable()
. . .	. . .

JTOOLBAR




DEMO
Chạy và giải thích

FPT Education

FPT POLYTECHNIC

LẬP TRÌNH JAVA 2

BÀI 3: JMENU, ĐỌC/GHI FILE

PHẦN 2

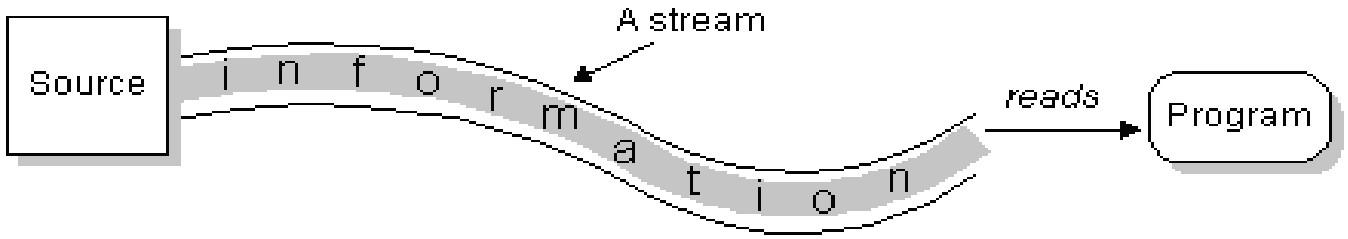
www.poly.edu.vn

GIỚI THIỆU LUỒNG DỮ LIỆU

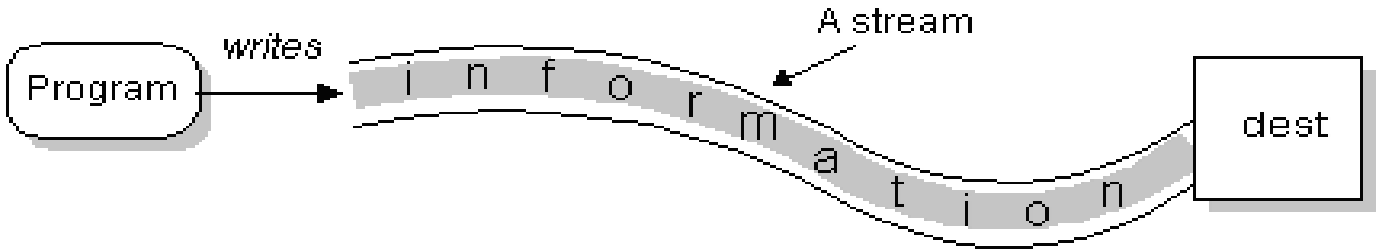
- Các hoạt động nhập/xuất dữ liệu (nhập dữ liệu từ bàn phím, đọc dữ liệu từ file, ghi dữ liệu màn hình, ghi ra file, ghi ra đĩa, ghi ra máy in…) đều được gọi là luồng (stream).
- Tất cả các luồng đều có **chung một nguyên tắc** hoạt động ngay cả khi chúng được gắn kết với các thiết bị vật lý khác nhau.

GIỚI THIỆU LUỒNG DỮ LIỆU

- Luồng vào là luồng cho phép chương trình đọc dữ liệu từ một nguồn nào đó: bán phím, file, máy scan…



- Luồng ra là luồng cho phép chương trình ghi dữ liệu lên nó để chuyển đến đích nào đó: màn hình, file, máy in…

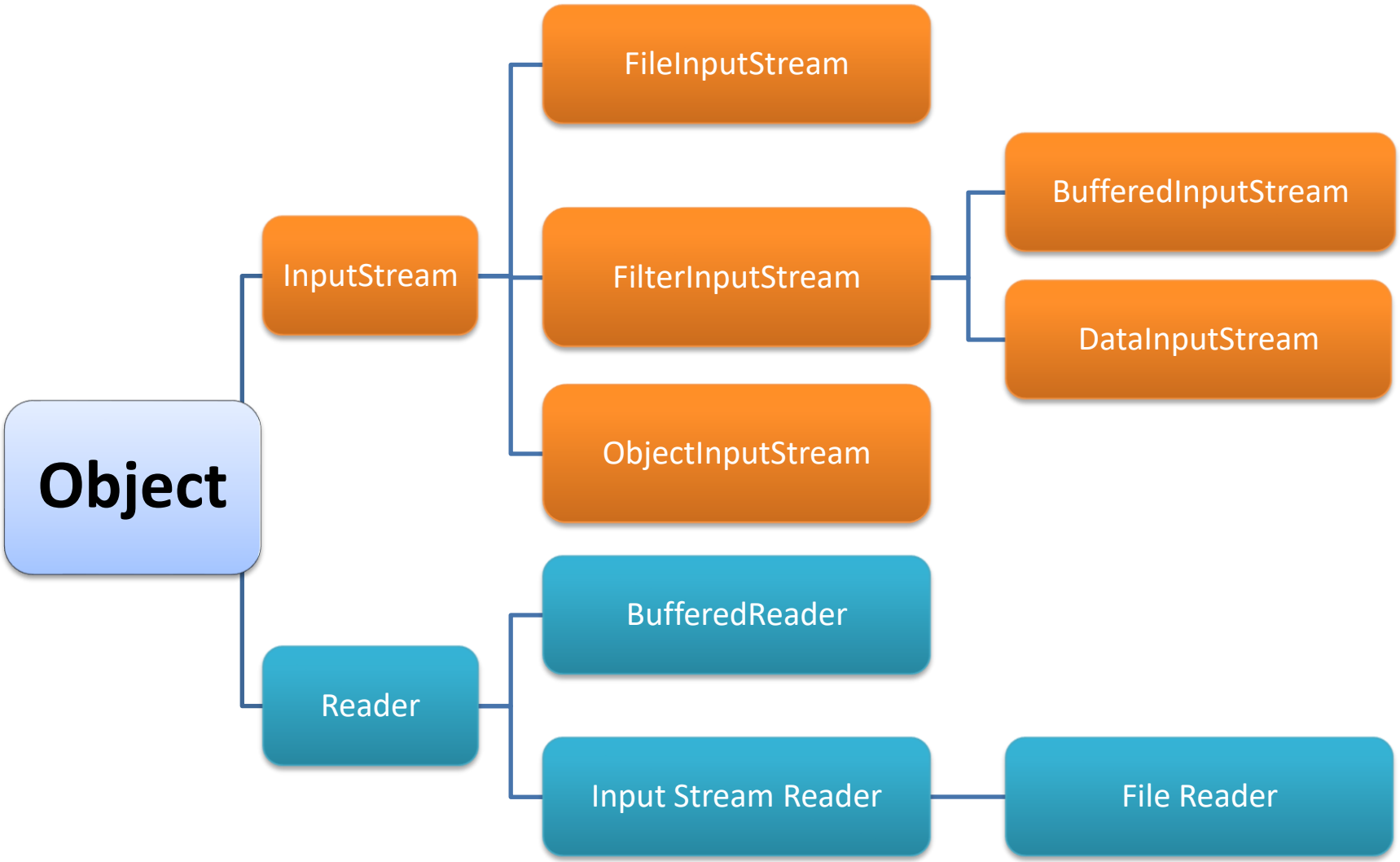


CÁC LOẠI LUỒNG DỮ LIỆU

- Có 2 kiểu luồng trong Java:
  - Luồng byte (luồng nhị phân)
  - Luồng character (luồng văn bản)
- Luồng byte
  - Hỗ trợ việc xuất nhập dữ liệu theo byte,
  - Thường được dùng khi đọc ghi dữ liệu nhị phân.
- Luồng character
  - Luồng character được thiết kế hỗ trợ việc xuất nhập dữ liệu kiểu ký tự

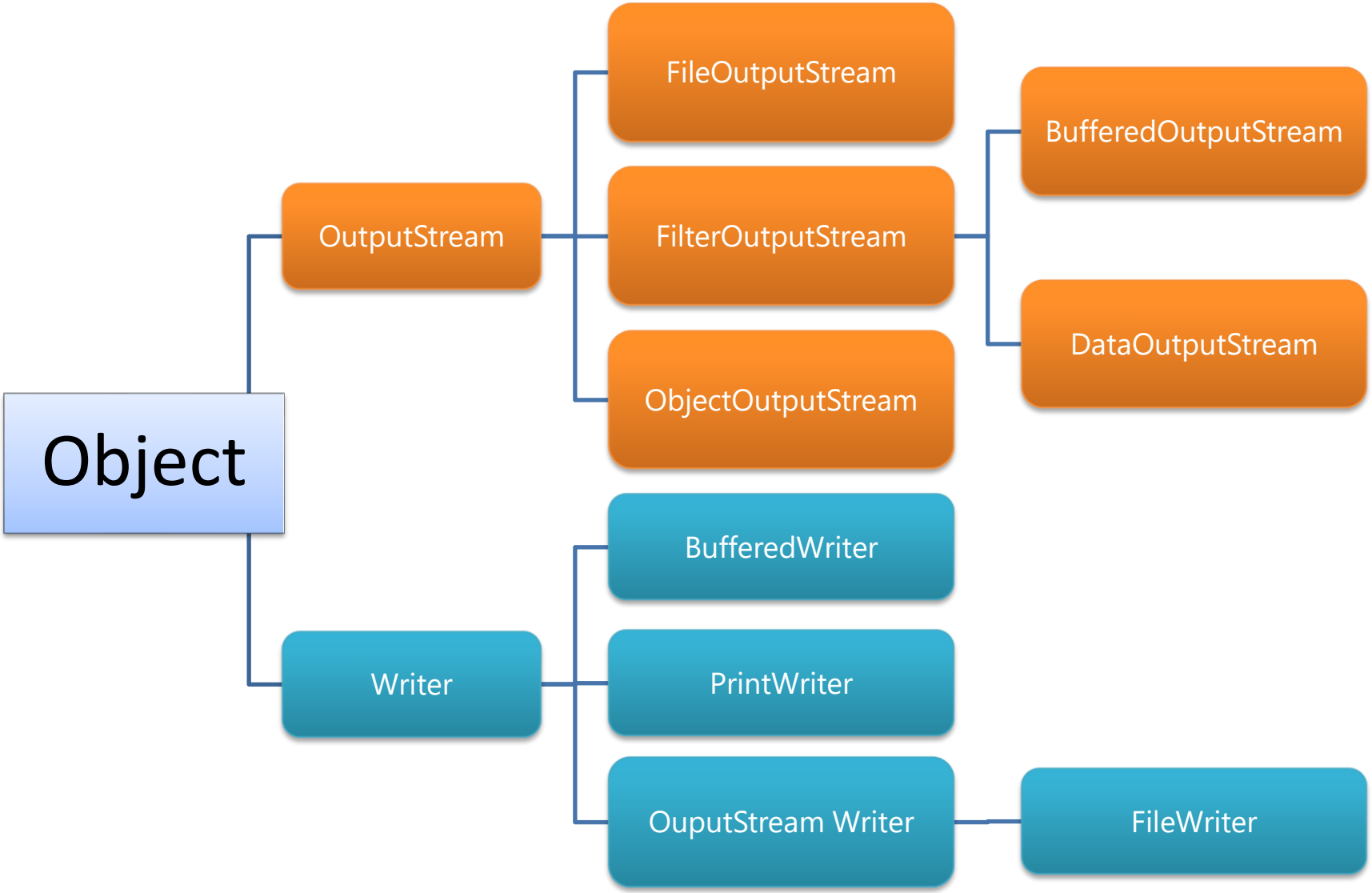
FPT Education
FPT POLYTECHNIC

KIẾN TRÚC PHÂN CẤP CÁC LOẠI LUỒNG VÀO



FPT Education
FPT POLYTECHNIC

KIẾN TRÚC PHÂN CẤP CÁC LOẠI LUỒNG RA



XỬ LÝ LUỒNG BYTE

- Sử dụng luồng mỗi byte để nhập xuất dữ liệu nhị phân
- Tất cả các luồng byte được kế thừa từ 2 class:
  - InputStream
  - OutputStream
- Có nhiều class luồng byte
  - File Input Stream
  - File Output Stream
- Chúng khác nhau về cách thức khởi tạo nhưng cách thức hoạt động là giống nhau.

FileInputStream/FileOutputStream

- Cập luồng này được sử dụng để làm việc với file nhị phân
  - Sử dụng FileInputStream để đọc dữ liệu từ file nhị phân
  - Sử dụng FileOutputStream để ghi dữ liệu vào file nhị phân

FileOutputStream

- FileOutputStream là luồng ra được sử dụng để ghi dữ liệu ra file nhị phân.

```
import java.io.FileOutputStream;
import java.io.IOException;

public class Example1 {

    public static void main(String[] args) throws IOException {
        FileOutputStream fos = new FileOutputStream("file1.dat");
        String text = "The quick brown fox jumped over the lazy dog";
        byte[] textAsBytes = text.getBytes();
        fos.write(textAsBytes);
    }
}
```

FileInputStream

- FileInputStream là luồng dữ liệu vào từ file nhị phân

```
public class Example2 {

    public static void main(String[] args) throws IOException {
        FileInputStream fis = new FileInputStream("file1.dat");
        int c;
        while ((c = fis.read()) != -1) {
            System.out.print((char) c);
        }
        fis.close();
    }
}
```

DataInputStream/DataOutputStream

- 2 luồng này giúp chúng ta đọc/ghi dữ liệu nguyên thủy
- Sử dụng read**<Type>**() để đọc dữ liệu nguyên thủy từ DataInputStream
  - read**Int**()
  - read**Double**()…
- Sử dụng write**<Type>**(Type) để ghi dữ liệu nguyên thủy lên DataOutputStream
  - write**Boolen**(boolean)
  - write**Int**(int)…

DataOutputStream

```
public class DataStreamOutput {
    public static void main(String[] args) throws IOException {
        FileOutputStream fos = new FileOutputStream("filedata.dat");
        DataOutputStream dos = new DataOutputStream(fos);
        final int NUMBER = 5;
        dos.writeInt(NUMBER);
        for (int i = 0; i <= NUMBER; i++) {
            dos.writeInt(i);
        }
        dos.writeUTF("Hello !");
        dos.writeDouble(100.75);
        dos.flush();
        dos.close();
    }
}
```

DataInputStream

```
public class DataStreamInput {
    public static void main(String[] args) throws IOException {
        FileInputStream fis = new FileInputStream("filedata.dat");
        DataInputStream dis = new DataInputStream(fis);
        int items = dis.readInt();
        for (int i = 0; i <= items; i++) {
            int n = dis.readInt();
            System.out.print(n + " ");
        }
        System.out.println(dis.readUTF());
        System.out.println(dis.readDouble());
        dis.close();
    }
}
```

ObjectInputStream/Object OutputStream

- Cập luồng này giúp chúng ta đọc/ghi đối tượng
- Sử dụng readObject() để đọc đối tượng từ DataInputStream
- Sử dụng writeObject(Serializable) để ghi đối tượng lên DataOutputStream
- Chú ý:
  - Chỉ các đối tượng được tạo từ các lớp có thực thi theo interface Serializable mới có thể đọc ghi được.

LỚP THỰC THI INTERFACE 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;
    }
    public String toString() {
        return (id+" "+desc + " " + price + " " + quantity);
    }
}
```

OBJECTOUTPUTSTREAM

```
public class ObjectExampleWrite {
    public static void main(String[] args) throws
            IOException, ClassNotFoundException {
        FileOutputStream fos = new FileOutputStream("fileobject.dat");
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        Stock[] stocks = {new Stock(1001, "CD ROM", 100.00, 20),
            new Stock(1002, "DRAM", 75.00, 30),
            new Stock(1003, "P4 Processor", 300.00, 100),
            new Stock(1004, "Canon Jet", 80.00, 10),
            new Stock(1005, "HP Scanner", 75.00, 90)};
        //Ghi mang doi tuong vao file 'fileobject.dat'
        oos.writeObject(stocks);
        oos.close();
    }
}
```

OBJECTINPUTSTREAM

```
public static void main(String[] args) {
    FileInputStream fis = null;
    ObjectInputStream ois = null;
    try {
        fis = new FileInputStream("fileobject.dat");
        ois = new ObjectInputStream(fis);
        Stock[] stocks1 = (Stock[]) ois.readObject();
        System.out.println("Doc tu file: ");
        for (Stock s : stocks1) {
            System.out.println(s);
        }
        ois.close();fis.close();

    } catch (Exception e) {
        System.out.println("Co loi: " + e);
    }
}
```

TỔNG KẾT

- JMenu
- JMenuBar
- JMenuItem
- JPopupMenu
- JToolBar
- Giải thích các loại luồng dữ liệu
- Nhập xuất các luồng byte


TIN HỌC VĂN PHÒNG QUỐC TẾ THEO CHUẨN ICDL
Syllabus 5 - Office 2007
LUẬT KINH TẾ
LUẬT KINH TẾ
KỸ THUẬT SOẠN THẢO VÀ TRÌNH BÀY VĂN BẢN
TIN HỌC VĂN PHÒNG QUỐC TẾ THEO CHUẨN ICDL
Syllabus 5 - Office 2007
Cảm ơn