

9장 자바GUI기초, AWT와 스윙

: GUI 프로그래밍 방법 : AWT, Swing 패키지 사용

1. AWT와 Swing 패키지

-AWT : Abstract Windowing Toolkit : 운영체제 의존, 속도 빠름, 중량 컴포넌트

-Swing : J로 시작하는 클래스, javax.swing 패키지, 경량 컴포넌트

예시 : JButton, JCheckBox, JRadioButton, JSlider, JTextField, JPasswordField, JSpinner, JTextArea, JComboBox, JList, ProgressBar, JToolTip, JScrollPane, JMenu, JDialog, JFrame, JColorChooser, JTable, JTree, JEditorPane and JTextPane, JToolBar, JTabbedPane, JSplitPane

-Swing 클래스의 특징 : J로시작, Jcomponent는 상속받는 클래스, AWT의 Container를 상속받는 몇 개의 클래스(JApplet, JDialog, JFrame 등)

2. 컨테이너와 컴포넌트

- 컨테이너 : 다른 GUI 컴포넌트를 상속할 수 있는 컴포넌트
- 컴포넌트 : 컨테이너에 포함되어야 화면에 출력될 수 있는 순수 컴포넌트
- 스윙 프레임 : 모든 스윙 컴포넌트를 담는 최상위 GUI컨테이너

3. 스윙 응용프로그램

- 스윙 응용프로그램의 종료

System.exit(0) : 언제 어디서나 무조건 종료

- 프레임 종료 버튼이 클릭되면 프레임이 보이지 않게 되지만 응용프로그램이 종료된 것이 아님

- 프레임 종료버튼이 클릭될 때 프레임을 닫고 응용 프로그램이 종료하도록 설정해야함

Frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

- main() 종료 뒤에도 프레임이 살아있는 이유 : 메인 스레드가 종료되어도 이벤트 분배 스레드가 살아있어 프레임 화면, 마우스, 키 입력을 받기 때문

4. 배치관리자 유형

- FlowLayout : 원->오, 위->아래 배치
- BorderLayout : 동서남북센터 배치
- GridLayout : 원->오, 위->아래, 칸을 나눠서 배치
- CardLayout : 오른쪽 아래 방향으로 약간 겹쳐서 배치
- 컨테이너는 생성시 디폴트 배치관리자를 설정함

Window, JWindow, Frame, JFrame, Dialog, JDialog : BorderLayout

Panel, JPanel, Applet, JApplet : FlowLayout

- 컨테이너 배치 관리자 설정 : Container.setLayout(LayoutManager lm);
- 배치 관리자가 없는 컨테이너 : null 설정 가능. 단, 절대크기, 절대위치 설정 필요

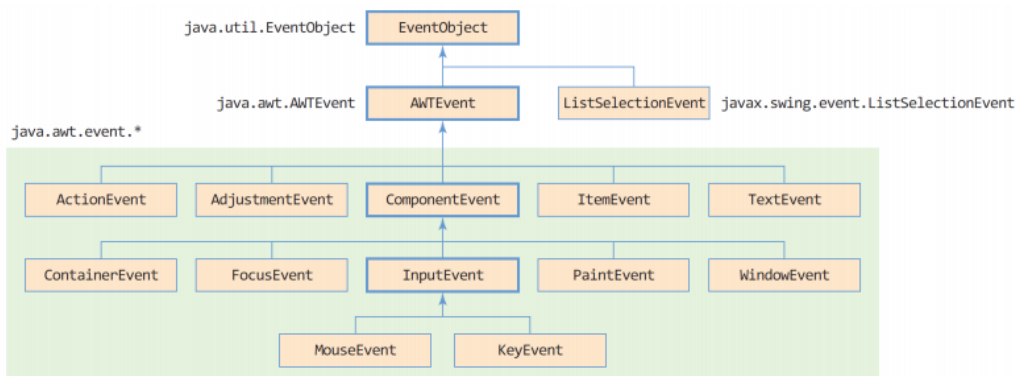
10장 자바의 이벤트 처리

1. 이벤트기반 프로그래밍

- 종류 : 사용자입력(마우스, 키보드 등), 센서 입력, 네트워크 송수신, 타 응용프로그램 메시지 등
- 이벤트 발생 -> 이벤트 객체 생성 -> 이벤트 리스너 찾기&호출 -> 이벤트 리스너 실행
- 반대개념 : 배치실행(개발자가 프로그램의 흐름 결정)

2. 이벤트 객체

- 이벤트가 발생할 때 발생한 이벤트에 대한 정보를 가진 객체
- 종류



- 포함 정보 : 이벤트 종류, 소스, 화면 좌표, 컴포넌트 내 좌표, 버튼or문자열, 마우스횟수, 키보드값, 체크박스 상태 등

- 이벤트에 따라 조금 다른 정보 포함

Ex : `ActionEvent`(액션 문자열), `MouseEvent`(마우스 정보), `ItemEvent`(아이템 체크 상태)

- 이벤트 소스 알아내기 : `Object EventObject.getSource()`
- 이벤트 객체와 이벤트 소스

이벤트 객체	이벤트 소스	이벤트가 발생하는 경우
<code>ActionEvent</code>	<code>JButton</code>	마우스나 <Enter> 키로 버튼 선택
	<code>JMenuItem</code>	메뉴 아이템 선택
	<code>TextField</code>	텍스트 입력 중 <Enter> 키 입력
<code>ItemEvent</code>	<code>CheckBox</code>	체크박스의 선택 혹은 해제
	<code>RadioButton</code>	라디오버튼의 선택 상태가 변할 때
	<code>CheckBoxMenuItem</code>	체크박스 메뉴 아이템의 선택 혹은 해제
<code>ListSelectionEvent</code>	<code>JList</code>	리스트에서 선택된 아이템이 변경될 때
<code>KeyEvent</code>	<code>Component</code>	키가 눌러지거나 눌러진 키가 떼어질 때
<code>MouseEvent</code>	<code>Component</code>	마우스 버튼이 눌러지거나 떼어질 때, 마우스 버튼이 클릭될 때, 컴포넌트 위에 마우스가 올라갈 때, 올라간 마우스가 내려올 때, 마우스가 드래그될 때, 마우스가 단순히 움직일 때
<code>FocusEvent</code>	<code>Component</code>	컴포넌트가 포커스를 받거나 잃을 때
<code>WindowEvent</code>	<code>Window</code>	Window를 상속받는 모든 컴포넌트에 대해 윈도우 활성화, 비활성화, 아이콘화, 아이콘에서 복구, 윈도우 열기, 윈도우 닫기, 윈도우 종료
<code>AdjustmentEvent</code>	<code>JScrollBar</code>	스크롤바를 움직일 때
<code>ComponentEvent</code>	<code>Component</code>	컴포넌트가 사라지거나, 나타나거나, 이동, 크기 변경 시
<code>ContainerEvent</code>	<code>Container</code>	Container에 컴포넌트의 추가 혹은 삭제

3. 이벤트 리스너

- 이벤트를 처리하는 코드, 클래스 작성, 인터페이스 제공 : 단 이벤트 전부 작성해야하기 때문에 Adapter 클래스 사용
- Adapter : 나머지 apthmeh 구현해야하는 부담을 해소하기 위한 클래스(ActionAdapter, ItemAdapter는 메소드가 하나기 때문에 Adapter X)

4. Key 이벤트와 포커스

- Key 입력시 : 키 누르는 순간(KeyPressed), 떼는 순간(KeyReleased), 누른 키를 떼는 순간(KeyTyped, Unicode키(문자키)인 경우)
- 포커스 필요 : `component.setFocusable(true); component.requestFocus();`
- 컴포넌트에 포커스 주기
- 키 판별 : `getKeyChar()`, `getKeyCode()` `getKeyText(int keyCode)`

11 장 기본적인 스윙 컴포넌트와 활용(컴포넌트기반 GUI 프로그래밍)

1. JLabel

- 텍스트 레이블 생성 : `JLabel la = new JLabel("hello");`
- 이미지 레이블 생성 : `ImageIcon image = new ImageIcon("이미지 위치"); JLabel il = new il(image);`
- 수평 정렬 값을 가진 레이블 생성 : 이미지-텍스트 순서로 나옴
`ImageIcon image = new ImageIcon("이미지 위치");`
`JLabel label = new JLabel("hello", image, SwingConstants.CENTER);`

2. JButton

- 빈, 이미지, 문자열, 문자열과 이미지 버튼 생성 가능
- 3개의 이미지를 연결하여 보통, 버튼위에 마우스가 있을 때, 누르는 동안 이미지를 다르게 출력 가능
- 수평 정렬(SwingConstants.LEFT, RIGHT, CENTER), 수직정렬(SwingConstants.TOP, BOTTOM, CENTER)

3. JCheckBox

- 선택, 비선택 가능
- 이벤트 처리 가능 : 선택시 이미지 출력 등

4. JRadioButton

- 다른 버튼이 선택되면 이전에 선택된 버튼은 자동으로 해제
- 생성과정 : 버튼 그룹 객체 생성 -> 라디오버튼 컴포 생성 -> 컴포 그룹에 삽입 -> 버튼 컨테이너 삽입

5. JTextField

- 문자열 입력 창, Enter키 입력되면 Action이벤트 발생
- 문자열 편집 불가능하게 설정 가능 : `JTextFiled.setEditable(false);`
- 입력창에 문자열 출력 가능, 폰트 지정 가능)

6. TextArea

- 여러줄을 입력할 수 있는 텍스트 입력창, JScrollPane 컴포를 삽입하면 스크롤바 지원됨
Ex : `JTextArea ta = new JTextArea(7, 20); container.add(new JScrollPane(ta));`

7. JList<E>

- 여러 개 아이템을 리스트 형식으로 보여주고 선택하는 컴포넌트
- JComboBox<E>와 기본적으로 같은 기능
- JScrollPane 에 JList<E>를 삽입하여 스크롤 가능
- 배열, Vector 로 아이템 제공 가능 : 단, 리스트가 생성되고 나서 수정해도 리스트 수정 안됨
- `setListData()` 호출하여 다시 배열을 달아줄 것!

8. JComboBox<E>

- 텍스트필드와 버튼, 드롭다운 리스트로 구성
- 아이템 선택 시, Action이벤트 발생
- 선택된 아이템 알아내기
`Int getSelectedIndex();` // 인덱스 번호 리턴, `Object getSelectedItem();` //레퍼런스 리턴

9. JSlider

- 마우스로 움직이면서 값 선택하는 컴포넌트

12장 그래픽(그래픽기반 GUI 프로그래밍)

1. Graphics 객체

- java.awt.Graphics : 컴포넌트 그리기에 필요한 도구 제공
- 사용자가 원하는 모양을 그리고자 할 때 paintComponent(Graphics g)를 오버라이딩
- 색상 선택, 문자열 출력, 도형 그리기, 도형 칠하기, 이미지 출력, 클리핑 등 + 문자열 그리기(draw String)

2. 색상 선택과 폰트 지정

- Color(int r, int g, int b), Color(int rgb)//단 32비트 중 하위 24비트만 사용, Color.YELLOW 등 사용
- g.setColor 사용 가능
- Font(String fontFace, int style(Font.ITALIC), int size)
- g.setFont 사용 가능

3. 도형 그리기와 칠하기

- 선(drawLine), 원(drawOval), 사각형, 둥근모서리 사각형, 원호, 페다각형
- 칠하기는 draw를 fill로 대치 //색은 앞줄에 미리 설정해둘 것

Ex : g.setColor(Color.RED); g.fillRect(10, 10, 50, 50)

4. 이미지 출력

- JLabel : 그리기 쉬움, 이미지가 컴포넌트이므로 이벤트 발생, 단 크기 조절 불가
- JPanel에서 Graphics : 크기조절, 일부출력 등 가능, 단 컴포넌트로 관리X
- 원본크기로 그리기, 크기조절하여 그리기, 일부분 크기 조절하여 그리기 가능
- ImageObserver 는 디코딩등으로 시간이 오래 걸릴 수 있기 때문에 이미지 그리기가 완료 되었는지 통보받을 때 사용. 보통 this/null 로 통보를 받지 않을 수 있음.

5. 클리핑

- 클리핑 영역에서만 그래픽이 보이도록 함.
- setClip(x, y, w, h)

6. 스윙의 paint 매커니즘

- 스윙 페인팅에 관여되는 JComponent 메소드

paint(Graphics g), paintComponent(Graphics g), paintBorder(Graphics g), paintChildren(Graphics g)

- 개발자가 paintComponent()를 직접 호출하면 안됨 : 자동으로 호출되기 때문
- repaint() : 지금 당장 컴포넌트를 다시 그리도록 지시, 호출되면 paintComponent 자동호출
- revalidate() : 컨테이너 배치 관리자에게 자식 컴포를 다시 배치하도록 지시하는 메소드
- 변화가 생겨서 다시 그려야할 때, 주로 revalidate와 repaint 를 세트로 호출함.

13장 스레드와 멀티태스킹

1. 멀티태스킹

- 하나의 응용프로그램이 여러 개의 작업을 동시에 처리
- 스레드(thread) : 실행 메소드가 실행될 길게 늘어뜨려졌다가 다른 스레드와 만날 때 멀티태스킹
- 스레드 : 사용자가 작성한 코드, JVM에 의해 스케줄링되어 실행되는 단위
- 멀티스레딩만 멀티태스킹이 가능. 지연시간 줄임
- 자바 스레드 : JVM에 의해 관리
- 만드는 방법 : `java.lang.Thread` 클래스 이용 / `java.lang.Runnable` 인터페이스 이용

2. Thread 클래스를 이용한 스레드 생성

- `class TimerThread extends Thread{ ... public void run(){//run 오버라이딩}}`
- 스레드 객체 생성하여 스레드 시작 : `TimerThread tth = new TimerThread(); tth.start();`
- `run()` 메소드가 종료하면 스레드 종료. 한 번 종료한 스레드는 다시 시작 불가
- 한 스레드에서 다른 스레드 종료 가능

3. Runnable 인터페이스로 스레드 만들기

- `class TimerRunnable implements Runnable{... public void run(){//run() 메소드 구현}}`
- `Thread tth = new Thread(new TimerRunnable()); tth.start();`

4. 스레드 정보

- : 이름, ID, PC, 상태, 우선순위, 그룹, 레지스터 스택
- 상태 : `new`, `runnable`, `waiting`, `timed_wating`, `block`, `terminated`
- 우선순위 : 최대10 최소1 보통5, `void setPriority(int priority)`, `int getPriority()`

5. 스레드 종료와 타 스레드 강제 종료

- 스스로 종료 : `run()` 메소드 리턴
- 강제종료 : `interrupt();`

6. 스레드 동기화 : thread synchronization

- 충돌을 방지하기 위해 공유 데이터를 배타적으로 접근하기 위한 것
- `synchronized`로 동기화 블록 지정 : 독점해야하는 부분을 표시하는 키워드, 먼저 실행한 스레드가 모니터 소유
- `wati()-notify()` 메소드로 스레드 실행 순서 제어 : `wait()`, `nofity(스레드 하나만 깨움)`, `nofityAll(스레드 전부 깨움)`, `synchronomized` 블록 내에서만 사용되어야함.)

14장 고급 스윙 컴포넌트

1. 메뉴만들기

- JMenuBar, JMenu, JMenuItem, 분리선
- 순서대로 만들어서 상위에 갖다 붙임
- Menu에 ActionListener 이용 : JMenuItem 에 리스너 설정 -> 선택되면 리스너 호출(다붙여줘야함)

2. 툴바

- JToolBar : 여러 컴포를 담을 수 있는 컨테이너, 드래그로 다이얼로그 형태로 떨어져서 존재 가능
- 툴바 고정 방법 : JToolBar.setFloatable(false)

3. 툴팁

- 톱 문구 : 마우스를 올리면 잠깐 나타났다가 사라짐
- JToolTip 클래스 이용(복잡해서 권장X)
- JComponent.setToolTipText(String msg) 권장
- 툴팁 활성화 및 시간 제어 : ToolTipManager 생성

```
ToolTipManager m = ToolTipManager.sharedInstance();  
m.setEnvalued(false) // 툴팁 비활성화  
m.setInitialDelay(1000)//마우스가 올라온 후 1000ms 후에 툴팁 출력  
m.stDismissDelay(1000)//툴팁 지속 시간 1000ms 설정
```

4. 다이얼로그

- 사용자로부터 입력을 받기위한 대화상자
- JDialog : 최상위 컨테이너(다른 컨테이너에 붙이지 않아도 됨)
- 모달다이얼로그 : 사용자 입력 독점. 닫기 전에는 다른 창에서 작업 불가능
- 모달리스 다이얼로그 : 다른 창과 독립적으로 작동
- 팝업 다이얼로그 : JOptionPane, 입력, 확인, 메시지 가능
- 파일 다이얼로그 : JFileChooser 파일 탐색기 기능, 열기저장(Open/Save) 가능
- 컬러다이얼로그 : JColorChooser : 색상팔레트를 제공하는 모달 다이얼로그

5. 탭팬(TabbedPane)

- 여러 패널을 겹치게 하여 공간을 공유하도록 지원하는 팬
- `JTabbedPane()`, `JTabbedPane(int tabPlacement)` //placement : top, Bottom, left, right
- 탭팬에 붙은 탭 개수(`int getTabCount()`) 리턴
- 현재 선택된 탭 : `int getSelectedIndex()`, `Component getSelectedComponent()`
- 탭 삭제 : `void remove(Component comp)`, `void remove(int index)`, `void removeTabAt(int index)`
- 탭 위치 변경 : `void setTabPlacement(int tabPlacement)`

6. 자바 오디오 API

- was, au, aiff, aifc 출력 가능
 - 오디오 클립 만들기 -> 오디오스트림 연결 -> 파일로부터 읽어올 스트림 객체 생성 -> 오디오 재생
- ```
Clip clip = AudioSystem.getClip(); File audioFile = new File("위치");
AudioInputStream audioStream = AudioSystem.getAudioInputStream(audioFile);
clip.open(audioStream); clip.start();
```
- Line 이벤트 : 오디오 재생 과정에서 발생, 재생시작, 재생중단, 클립단함(재생끝)