

## 6부 AWT

### - 25장 AWT를 이용한 GUI 작성

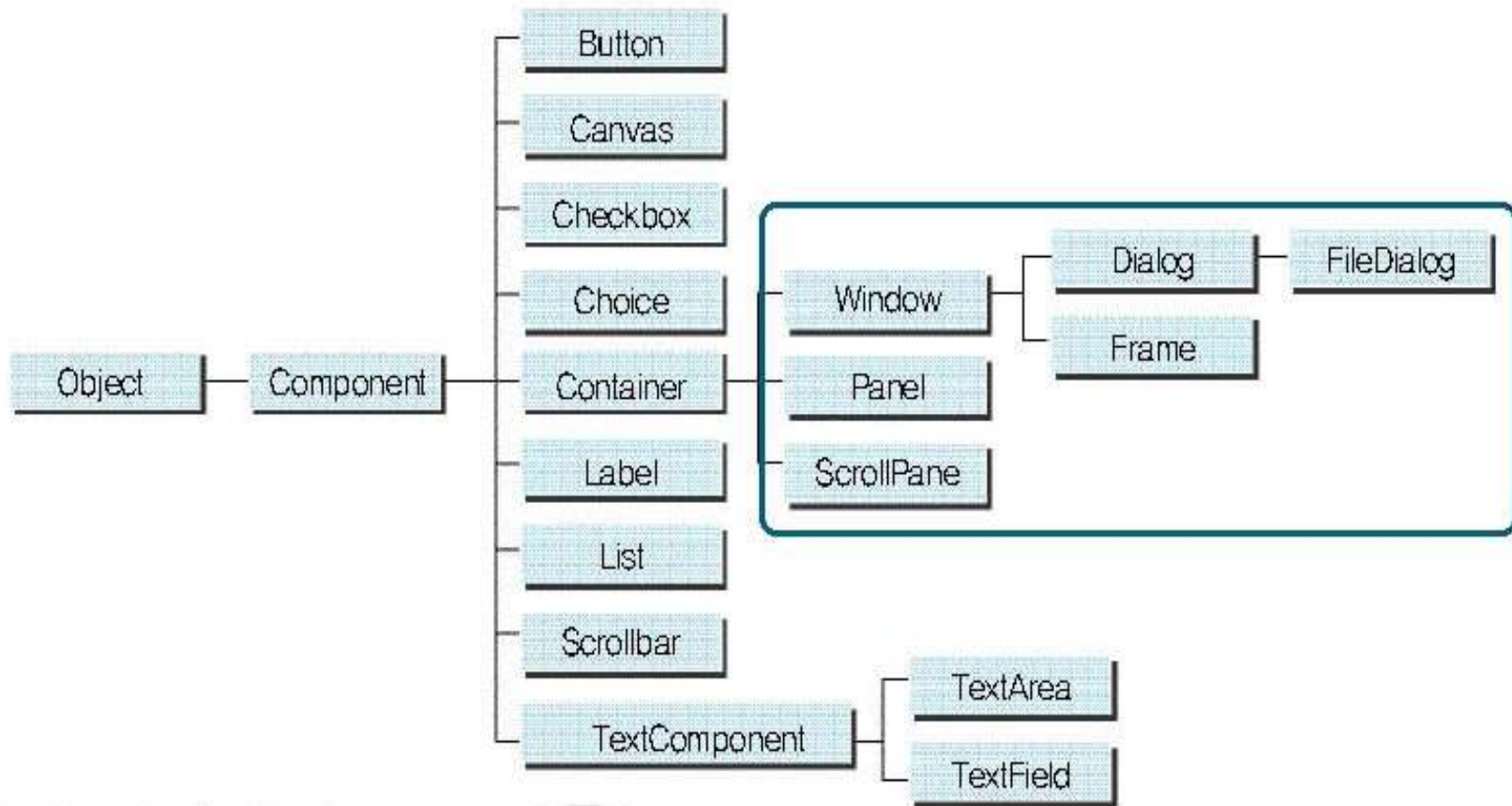
최 문 환



# 25장 AWT를 이용한 GUI 작성

1. AWT
2. Container 클래스
3. 프레임 창 닫기
4. 배치관리자
5. Panel 컨테이너

# 1. AWT



# Component 클래스에 속하는 중요한 메서드

생성자	
Component ( )	
메서드	
public void add(PopupMenu pop)	컴포넌트에 팝업 메뉴를 추가합니다.
public Color getBackground()	컴포넌트 바탕색을 반환합니다.
public Color getForeground()	컴포넌트 전경색을 반환합니다.
public Graphics getGraphics()	컴포넌트 Graphic 객체를 반환합니다.
public String getName()	컴포넌트 이름을 반환합니다.
public Container getParent()	부모 컨테이너를 반환합니다.
public Dimension getSize()	컴포넌트 크기를 반환합니다.

# Component 클래스에 속하는 중요한 메서드

<code>public boolean isEnabled()</code>	컴포넌트 활성화 여부를 반환합니다.
<code>public boolean isVisible()</code>	컴포넌트가 화면에 보이는지 알아냅니다.
<code>public void remove(MenuComponent popup)</code>	팝업 메뉴를 제거합니다.
<code>public void repaint()</code>	컴포넌트를 다시 그립니다.
<code>public setBackground(Color color)</code>	컴포넌트의 바탕색을 지정합니다.
<code>public setEnabled(boolean b)</code>	컴포넌트의 활성화 여부를 지정합니다.
<code>public setFont(Font f)</code>	컴포넌트의 폰트를 설정합니다.
<code>public setLocation(int x, int y)</code>	컴포넌트의 위치를 설정합니다.
<code>public setSize(int w, int h)</code>	컴포넌트의 크기를 설정합니다.
<code>public setBounds(int x, int y, int w, int h)</code>	컴포넌트의 위치와 크기를 설정합니다.
<code>public setVisible(boolean b)</code>	컴포넌트를 화면에 보이도록 합니다.

## ▪ 2. Container 클래스

생성자	
public Frame ( )	기본 생성자입니다.
public Frame (String title)	타이틀바의 문자열을 지정합니다.
메서드	
public String getTitle()	타이틀바의 문자열을 가져옵니다.
public boolean isResizable()	창의 크기 변경 여부를 알아냅니다.
public void setMenuBar(MenuBar mb)	프레임에 메뉴를 등록합니다.
public void setResizable(boolean b)	창의 크기 변경 여부를 지정합니다.
public void setTitle(String title)	타이틀바의 문자열을 지정합니다.

# <예제> 프레임 객체 생성하기

```
001:import java.awt.*;
002:public class FrameTest01 {
003:  public static void main(String[] args) {
004:    Frame f = new Frame("Frame Demo");
005:    f.setSize(300,200);
006:    f.setVisible(true);
007:  }
008:}
```

### 3. 프레임 창 닫기

[닫기] 버튼이 눌렸을 때 창이 닫히도록 하는 것마저도 프로그래머가 직접 구현해야 합니다.

<예제>-프레임 창 닫기 위한 이벤트 처리하기-[파일 이름 : FrameTest02.java]

```
001:import java.awt.*;
002:import java.awt.event.*;
003://프레임의 상속을 받아 FrameEvent 클래스를 새로 설계한다.
004:class FrameEvent extends Frame {
005: public FrameEvent( ) {    //생성자
006:     super("윈도우 이벤트");    //상위 클래스인 Frame 클래스의 생성자 호출
007:     setSize(300,200);        //윈도 크기를 지정
008:     setVisible(true);        //윈도가 화면에 나타나도록 한다.
009:
```



### 3. 프레임 창 닫기

<예제>-프레임 창 닫기 위한 이벤트 처리하기-[파일 이름 : FrameTest02.java]

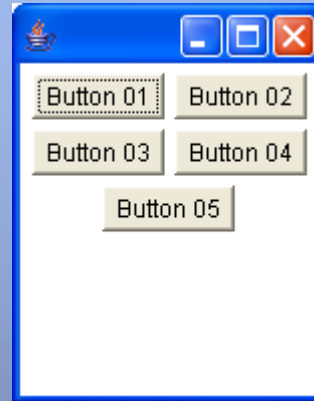
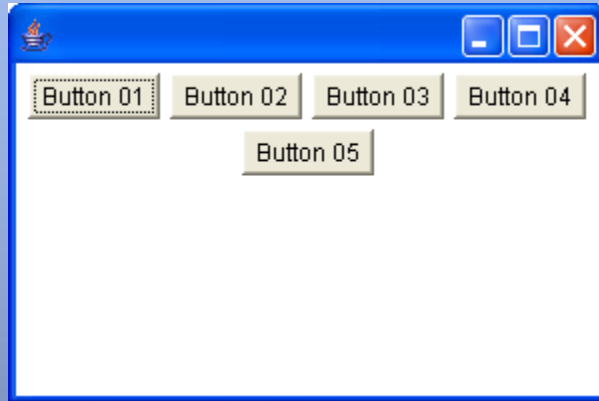
```
010: //프레임에 이벤트를 처리하기 위한 리스너 객체 생성과 등록
011: addWindowListener(new WindowAdapter() {
012:     //[닫기]버튼이 눌렸을 때 호출되는 메서드 오버라이딩
013:     public void windowClosing(WindowEvent e) {
014:         dispose(); //프레임 창이 닫히도록 한다.
015:         System.exit(0); //프로세스를 강제 종료함, 윈도우를 종료하기 위한 코
딩
016:     } //windowClosing 메서드 끝
017: } //클래스 정의 끝
018: ); //addWindowListener 메서드 끝
019: } //생성자 끝
020: } //클래스 정의 끝
021:
022: public class FrameTest02 {
023:     public static void main(String[] args) {
024:         new FrameEvent( ); //FrameEvent 클래스로 객체 생성
025:     }
026: }
```

## 4.1 컨테이너에게 배치관리자 지정하는 절차

```
FlowLayout layout = new FlowLayout( );  
Frame f = new Frame();  
f.setLayout(layout);  
Button bt01= Button01();  
f.add(bt01);
```



## 4.2 플로우 레이아웃



<예제>-프레임의 배치관리자로 플로우 레이아웃 설정하기

```
001:import java.awt.*;
002:import java.awt.event.*;
003:class FrameEx extends Frame {
004: public FrameEx( ) {
005:     //플로우 레이아웃(FlowLayout)을 생성하여
006:     //setLayout 메서드로 프레임(FrameEx 클래스)의 배치관리자로 설정
007:     setLayout(new FlowLayout());
008:
```

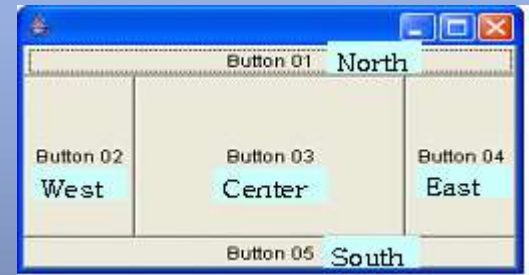
# <예제>-프레임의 배치관리자로 플로우 레이아웃 설정하기

```
009: //버튼(컴포넌트)을 프레임(FrameEx 클래스)에 추가하면
010: //플로우 레이아웃 배치관리자에 의해서 물 흐르듯이 배치된다.
011: add(new Button("Button 01"));
012: add(new Button("Button 02"));
013: add(new Button("Button 03"));
014: add(new Button("Button 04"));
015: add(new Button("Button 05"));
016:
017: setSize(300,200);
018: setVisible(true);
019:
020: addWindowListener(new WindowAdapter( ){
021:     public void windowClosing(WindowEvent e) {
022:         dispose();
023:         System.exit(0);
024:     } //windowClosing 메서드 끝
025: } //클래스 정의 끝
026: ); //addWindowListener 메서드 끝
027: }
028: }
029:
030: public class FrameTest03{
031:     public static void main(String[] args) {
032:         new FrameEx( );
033:     }
034: }
```

## 4.3 보더 레이아웃

<예제>-프레임의 배치관리자로 보더 레이아웃 설정하기

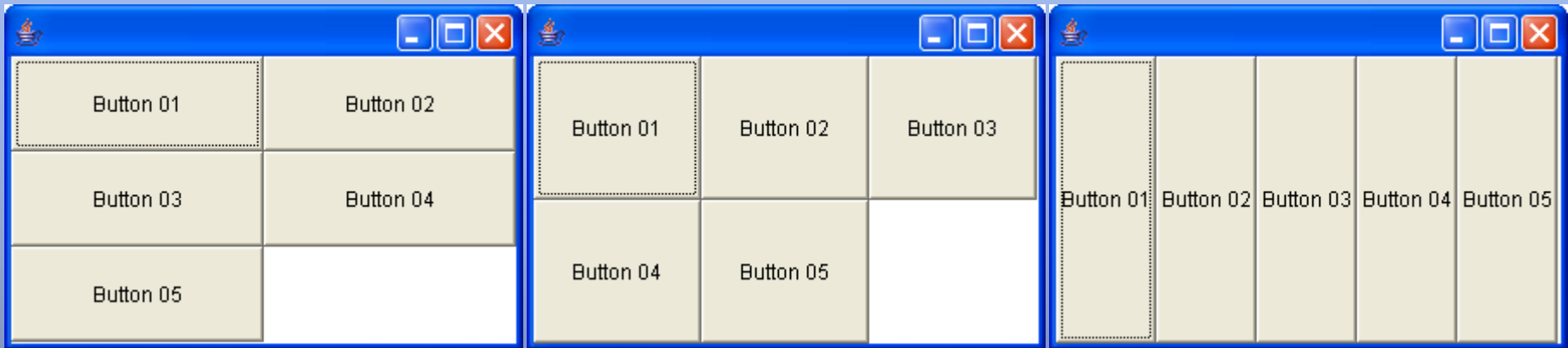
```
001:import java.awt.*;
002:import java.awt.event.*;
003:class FrameEx extends Frame {
004:  public FrameEx( ) {
005:    //보더 레이아웃(BorderLayout)을 배치관리자로 설정
006:    setLayout(new BorderLayout());
007:    //컴포넌트를 컨테이너에 추가할 때 레이아웃의 위치를 지정
008:    add(new Button("Button 01"), "North");
009:    add(new Button("Button 02"), "West");
010:    add(new Button("Button 03"), "Center");
011:    add(new Button("Button 04"), "East");
012:    add(new Button("Button 05"), "South");
```



## 4.3 보더 레이아웃

```
013:
014:  setSize(300,200);
015:  setVisible(true);
016:
017:  addWindowListener(new WindowAdapter() {
018:      public void windowClosing(WindowEvent e) {
019:          dispose();
020:          System.exit(0);
021:      } //windowClosing 메서드 끝
022:  } //클래스 정의 끝
023: ); //addWindowListener 메서드 끝
024: }
025:}
029:
030:public class FrameTest04{
031:  public static void main(String[] args) {
032:      new FrameEx( );
033:  }
034:}
```

## 4.4 그리드 레이아웃



<예제>-프레임의 배치관리자로 그리드 레이아웃 설정하기

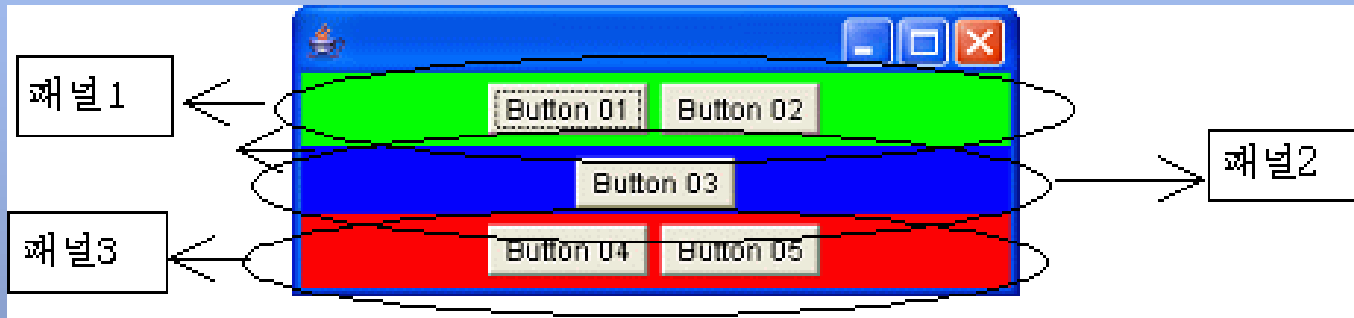
```
001:import java.awt.*;
002:import java.awt.event.*;
003:class FrameEx extends Frame {
004: public FrameEx( ) {
005:     //그리드 레이아웃(GridLayout)을 배치관리자로 설정
006:     setLayout(new GridLayout(3, 2));
007:     //setLayout(new GridLayout(2, 3));
008:     //setLayout(new GridLayout(1, 5));
009:
```

## 4.4 그리드 레이아웃

```
010: //버튼(컴포넌트)을 프레임(FrameEx 클래스)에 추가하면
011: //그리드 레이아웃에 명시한 행, 열에 의해 배치 형태가 결정됨
012: add(new Button("Button 01"));
013: add(new Button("Button 02"));
014: add(new Button("Button 03"));
015: add(new Button("Button 04"));
016: add(new Button("Button 05"));
017:
018: setSize(300,200);
019: setVisible(true);
020:
021: addWindowListener(new WindowAdapter() {
022:     public void windowClosing(WindowEvent e) {
023:         dispose();
024:         System.exit(0);
025:     } //windowClosing 메서드 끝
026: } //클래스 정의 끝
027: ); //addWindowListener 메서드 끝
028: }
029: }
030:
031: public class FrameTest05{
032:     public static void main(String[] args) {
No.16 033:         new FrameEx( );
034:     }
035: }
```



## 5. Panel 컨테이너



```
001:import java.awt.*;
002:import java.awt.event.*;
003:class FrameEx extends Frame {
004:    //1. 패널 레퍼런스 변수 선언
005:    Panel pan01, pan02, pan03;
006:    public FrameEvent06( ) {
007:        //2. 패널 객체 생성
008:        pan01 = new Panel();
009:        pan02 = new Panel();
010:        pan03 = new Panel();
011:
012:        //3. 각 패널의 구분을 위해서 패널에 서로 다른 배경색을 지정하기
013:        pan01.setBackground(Color.green);
014:        pan02.setBackground(Color.blue);
015:        pan03.setBackground(Color.red);
```

# <예제> 패널을 이용하기

```
017: //4. 세 개의 패널을 프레임의 북쪽, 가운데, 남쪽에 배치
018: add(BorderLayout.NORTH, pan01);
019: add(BorderLayout.CENTER, pan02);
020: add(BorderLayout.SOUTH, pan03);
021:
022: //5. 세 개의 패널에 버튼을 배치한다.
023: pan01.add(new Button("Button 01"));
024: pan01.add(new Button("Button 02"));
025:
026: pan02.add(new Button("Button 03"));
027:
028: pan03.add(new Button("Button 04"));
029: pan03.add(new Button("Button 05"));
030:
031: setSize(300, 130);
032: setVisible(true);
033:
034: addWindowListener(new WindowAdapter() {
035:     public void windowClosing(WindowEvent e) {
036:         dispose();
037:         System.exit(0);
038:     } //windowClosing 메서드 끝
039: } //클래스 정의 끝
040: ); //addWindowListener 메서드 끝
041: }
042: }
```

# <예제> 패널을 이용하기

```
044:public class FrameTest06{  
045:  public static void main(String[] args) {  
046:    new FrameEx( );  
047:  }  
048:}
```