

JDK 1.8부터 추가된 람다식
<p>1. 람다식의 도입으로 인해, 이제 자바는 객체지향언어인 동시에 함수형 언어가 되었다.</p> <p>2. 람다식은 간단히 말해서 메서드를 하나의 식으로 표현한 것이다. 메서드를 람다식으로 표현하면 메서드의 이름과 반환타입이 없게 되므로 람다식을 '익명 함수(Anonymous function)' 라고 한다.</p> <p>3. 람다식은 메서드의 매개변수로 전달되어지는 것이 가능하고, 메서드의 결과로 반환될 수 있다.</p> <p>4. 람다식은 익명함수 답게 메서드에서 이름과 반환타입을 제거하고, 매개변수 선언부와 몸통{} 사이에 '->'를 추가한다.</p> <p>형식) 반환타입 메서드이름(매개변수 선언){</p> <p style="padding-left: 40px;">몸체 문장;</p> <p style="padding-left: 20px;">} 이것을 람다식으로 바꾸면</p> <p>(매개변수 선언) -> {</p> <p style="padding-left: 40px;">몸체 문장;</p> <p style="padding-left: 20px;">}</p>

java GUI 중 java.awt 패키지의 AWT
<p>1. java.awt 의 AWT로 GUI 컴포넌트를 제공한다.</p> <p>2. AWT를 사용하기 위해서는 java.awt 패키지를 import 해야 한다.</p> <p>3. Panel 은 여러 개의 컴포넌트를 그룹별로 모을 때 주로 사용하는 일종의 컨테이너이다.</p> <p>4. TextField는 한줄 짜리 텍스트를 입력받을 수 있는 입력박스를 만들어 준다.</p> <p>5. Label은 컴포넌트가 여러개 있을 때 각 컴포넌트가 어떤 역할을 하는지 설명을 덧붙이기 위해서 사용한다.</p> <p>6. TextArea는 여러 줄의 텍스트를 입력받기 위한 입력박스를 만들어 준다.</p> <p>7. Checkbox는 하나 또는 복수개의 선택을 할수 있는 체크박스를 만들어 준다.</p> <p>8. Checkbox에 체크 박스 그룹 객체를 적용하면 단 하나만 선택 가능한 라디오 버튼을 만들어 준다.</p> <p>9. Choice는 드롭다운 리스트 즉 콤보박스를 작성할 때 사용한다.</p> <p>10. List는 여러 개의 목록을 열거해서 화면에 보여주는 리스트 박스를 만들어 준다.</p> <p>11. Canvas는 흰 도화지 같은 컴포넌트로 그림을 그리는 경우에 많이 사용한다.</p> <p>12. 메뉴는 사용자가 프로그램의 기능을 선택할 수 있도록 여러 개의 아이템 항목을 단계적으로 나타낸다. 메뉴를 구성하는 요소로 메뉴 바, 메뉴, 메뉴 아이템이 있다.</p>

자바 이벤트 처리
<p>1. 자바에서 이벤트란 컴퓨터에 일어나는 어떠한 사건을 말한다.</p> <p>2. 이러한 사건을 처리하기 위해서 자바에서는 java.awt.event패키지에 이벤트만 처리하는 이벤트 리스너 인터페이스를 별도로 정의를 하고 있다. 그리고 각 리스너 인터페이스에 이벤트를 처리하는 추상메서드를 각각 정의되어 있다. 이 리스너 인터페이스를 상속 받은 자손클래스에서 이벤트를 처리하는 모든 추상메서드를 오버라이딩을 해야한다. 즉 불필요한 추상메서드까지 오버라이딩을 해야 하기 때문에 자바에서는 추상메서드가 하</p>

나 이상인 리스너 인터페이스는 미리 추상메서드를 구현해 놓은 별도의 어댑터 추상클래스를 따로 정의를 하고 있다. 즉 이벤트를 처리하기 위해서 리스너 인터페이스를 상속 받는 경우 추상메서드가 하나 이상이면 불필요한 추상메서드까지 오버라이딩을 해야 하기 때문에 미리 구현해 놓은 어댑터 추상클래스를 익명클래스 문법으로 이벤트 처리하면 꼭 필요한 메서드만 오버라이딩을 해서 이벤트를 처리하기 때문에 매우 효과적으로 이벤트를 처리할 수 있다.

3. 이벤트 리스너 인터페이스에 정의된 이벤트 처리 추상메서드가 딱 하나만 존재하는 경우는 불필요한 추상메서드를 오버라이딩을 할 이유가 없기 때문에 별도의 어댑터 추상클래스가 따로 구현되어져 있지 않다. 이런 경우는 리스너 인터페이스를 구현한 자손 클래스에서 딱 하나 뿐인 추상메서드를 오버라이딩을 해서 이벤트를 처리하면 된다.

4. 이벤트 등록 메서드는 다음과 같이 정의된다.

```
add+이벤트 리스너 인터페이스 이름();
```

대표적으로 버튼 이벤트를 처리해주는 리스너 인터페이스 이름은 ActionListener이다.

그러므로 이벤트 등록 메서드는 다음과 같다.

```
addActionListener();
```

자바 GUI 스윙

1. 스윙은 보다 더 세련되고 다양한 형태의 gui를 제공한다.

2. 스윙을 사용하려면 스윙 기본 패키지인 javax.swing을 import해야 한다.

3. 스윙 컴포넌트는 기존의 AWT 컴포넌트 이름 앞에 접두어 영문 대문자 J만 추가하면 된다. 예를 들어 AWT의 Button에 접두어 영문 대문자 J를 붙여서 JButton이 되면 바로 스윙 버튼 컴포넌트가 된다.

4. 윈도우 탐색기같은 디렉토리 구조를 나타내기 위한 계층적인 구조를 만들고 싶다면 스윙의 JTree를 사용한다.

5. 표 테이블 형태를 자료를 다루기 위해서는 스윙의 JTable을 사용한다.