**package** swing;

**import** java.awt.FlowLayout;

**import** java.util.Vector;

**import** javax.swing.DefaultListModel;

**import** javax.swing.JFrame;

**import** javax.swing.JList;

**import** javax.swing.JScrollPane;

**import** javax.swing.ListSelectionModel;

//JFrame은 BorderLayout(동서남북)

**public** **class** JListEx **extends** JFrame {

**public** JListEx() {

**super**("JList Test"); // 부모 생성자 호출

setLayout(**new** FlowLayout());

// List가 따닥따닥 붙어서 나오는 건 setLayout을 FlowLayout으로 잡아놨기 때문.

// 현재 갖고 있는 걸 중앙을 기점으로 순서대로 붙여주게 된다.

String[] listData = { "Hong", "Gil", "Dong", "JAVA", "JSP" };

JList<String> list1 = **new** JList<String>(listData);

list1.setSelectedIndex(1);

// JList 바깥에 보여주는 작업 - 강사(부모)

// 안에 있는 데이터를 넣다 뺏다 하는 작업은 DefaultListModel - 매니저(자식)

// 데이터가 들어오는 항목은 문자열이고 (model)은 매니저

// model에 관련된 JList를 작성해서 넣는 것

// Model을 먼저 하거나~~~

// DefaultListModel<String> model = new DefaultListModel<String>();

// JList<String> list2 = new JList<String>(model);

// =

// List를 먼저 잡고 Model이 뒤따라오게 하거나~~~

JList<String> list2 = **new** JList<String>(**new** DefaultListModel<String>());

DefaultListModel<String> model = (DefaultListModel<String>) list2.getModel();

// 항상 부모는 자식에게 맞춰서 가야한다

// 부모인 ListModel은 자식인 DefaultListModel로 바꿔서 Casting한 다음에 보내줘야 한다

model.addElement("사과");

model.addElement("배");

model.addElement("딸기");

model.addElement("바나나");

list2.setSelectedIndex(1);

// Index 1번에 맞춰줬으니 "배"가 나옴

Vector<String> vListData = **new** Vector<String>();

JList<String> list3 = **new** JList<String>(vListData);

JScrollPane scroll = **new** JScrollPane(list3);

vListData.add("축구");

vListData.add("야구");

vListData.add("농구");

vListData.add("배구");

vListData.add("테니스");

vListData.add("수영");

vListData.add("육상");

vListData.add("태권도");

vListData.add("유도");

**class** Student { // 분명히 생성자라는 메소드 안에 클래스를 선언해서 쓴다?

// 로컬이너 몇번? 한번 쓰겠다

String id;

String name;

String department;

// 생성자를 잡자!

// 우클릭 - Source - Generator Constructor using field

**public** Student(String id, String name, String department) {

**super**();

**this**.id = id;

**this**.name = name;

**this**.department = department;

}

@Override

**public** String toString() { // 주소가 아닌 이름으로 나오게 하기 위해서

**return** name; // 이름만 화면에 떴으면 좋겠어

}

}

JList<Student> list4 = **new** JList<Student>(**new** DefaultListModel<Student>());

// 문자열이 아닌 student라는 클래스 객체열을 담겠다

DefaultListModel<Student> model2 = (DefaultListModel<Student>) list4.getModel();

// getmodel로 잡았으니 casting 걸어주자

list4.setSelectionMode(ListSelectionModel.***SINGLE\_SELECTION***); // 하나만 선택하고 싶어요

model2.addElement(**new** Student("100", "홍길동", "전산과"));

model2.addElement(**new** Student("200", "손오공", "건축과"));

model2.addElement(**new** Student("300", "사오정", "토목과"));

model2.addElement(**new** Student("400", "저팔계", "경영학"));

list4.setSelectedIndex(1);

add(list1);

add(list2);

add(scroll);

// add(list3); 라고 하면 scroll바 만들었어도 나오지 않는다

// 왜? list를 scroll에 갖다붙였기 때문에(JScrollPane scroll = new JScrollPane(list3);)

// Frame에 붙일 때는 list가 아닌 scroll을 붙여야 한다.

add(list4);

setBounds(900, 200, 400, 300);

setVisible(**true**);

setDefaultCloseOperation(***EXIT\_ON\_CLOSE***);

}

**public** **static** **void** main(String[] args) {

**new** JListEx();

}

}