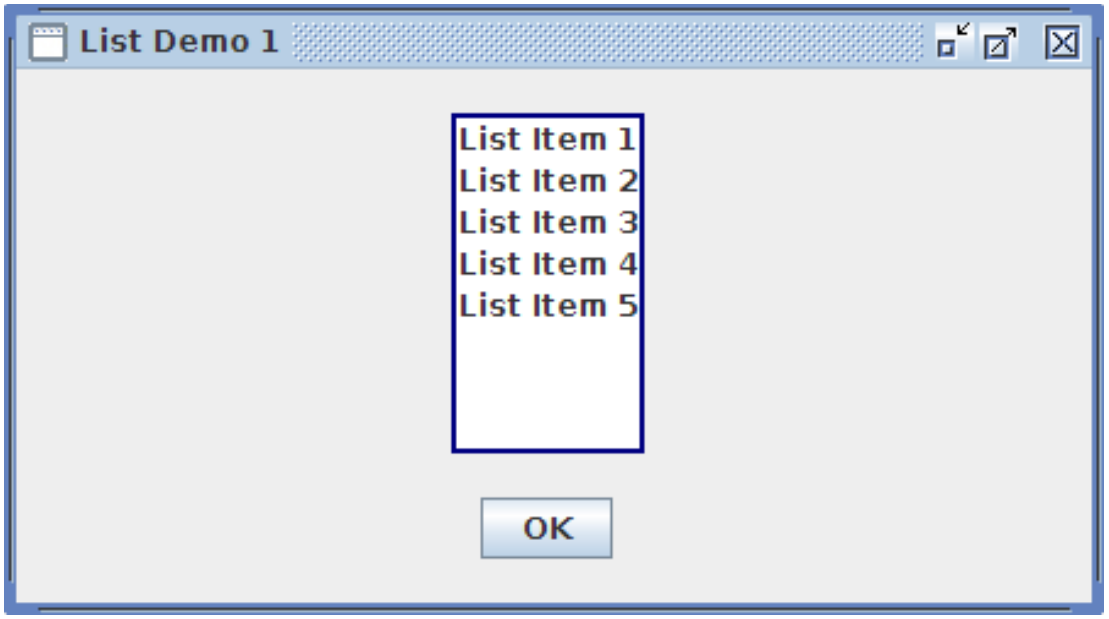
17. GUI Programming :: The *JList* Class

A GUI **list** is a component that will display a horizontal or vertical list of one or more list items:



To create a list we use the *javax.swing.JList* class. *JList* is a **generic class** which means that a *JList* can store items of any class and when we declare a *JList* object we have to specify a **type parameter** *E* which specifies the class of each list item:

```
JList<E> list = new JList<>(list-of-items-each-of-class-E);
```

17. GUI Programming :: The *JList* Class (continued)

For example:

```
String[] listItems = new String[5];
for (int i = 0; i < 5; ++i) listItems[i] = "List Item " + (i + 1);
JList<String> list = new JList<>(listItems);
```

A *JList* has a **selection mode** which must be one of these static constants declared in the *javax.swing.ListSelectionModel* class:

`ListSelectionModel.SINGLE_SELECTION`
Only one list item may be selected at a time.

`ListSelectionModel.SINGLE_INTERVAL_SELECTION`
Only one group of contiguous list items may be selected at a time.

`ListSelectionModel.MULTIPLE_INTERVAL_SELECTION`
Multiple groups of contiguous list items may be selected.

To specify the SINGLE_SELECTION mode:

```
list.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
```

17. GUI Programming :: The *JList* Class (continued)

The items in the list may be displayed in one of three layouts:

`JList.VERTICAL`

The list items are arranged vertically:

```
1
2
3
4
...
```

`JList.HORIZONTAL_WRAP`

The list items are arranged horizontally, wrapping at the right edge when necessary:

```
1   2   3
4   5   ...
```

`JList.VERTICAL_WRAP`

The list items are arranged vertically, wrapping to the next column when necessary:

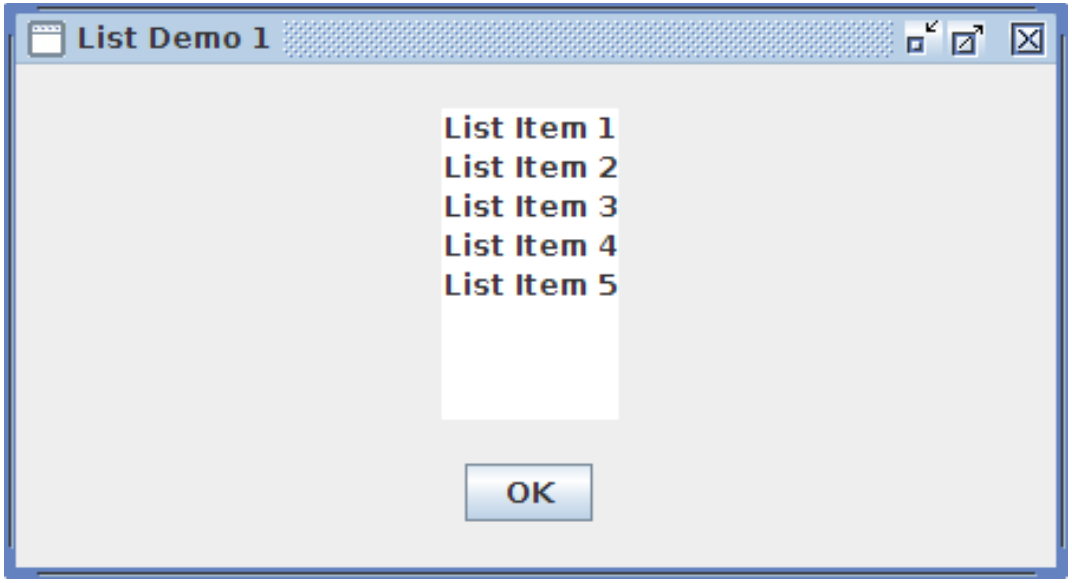
```
1   4
2   5
3   ...
```

To specify the VERTICAL layout:

```
list.setLayoutOrientation(JList.VERTICAL);
```

17. GUI Programming :: The *JList* Class (continued)

By default, the list does not have a border.



To add a border we can use the *javax.swing.BorderFactory* class to create a border. *BorderFactory* can create many types of borders (e.g., beveled, dashed line, solid line, etched). To create a solid line border we call: *BorderFactory.createLineBorder*(*java.awt.Color color*, **int** *thickness*). For example, to create a dark blue border that with thickness of 2 pixels we call:

```
BorderFactory.createLineBorder(new Color(0, 0, 128), 2));
```

Color is a class in the *java.awt* package. The *Color*(**int** *red*, **int** *green*, **int** *blue*) method accepts three arguments (each in the range 0–255) that specifies the red, green, and blue components of the desired colors. To add the border to the *JList*:

```
list.setBorder(BorderFactory.createLineBorder(new Color(0, 0, 128), 2));
```