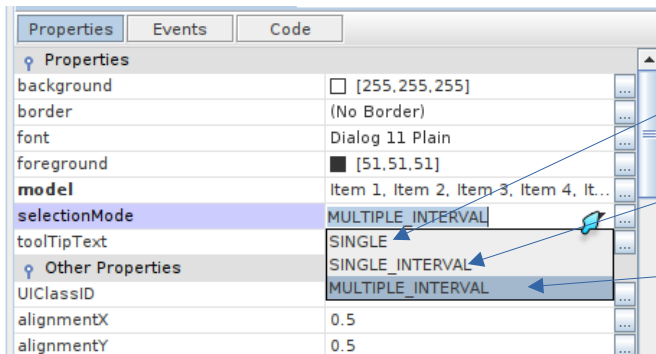
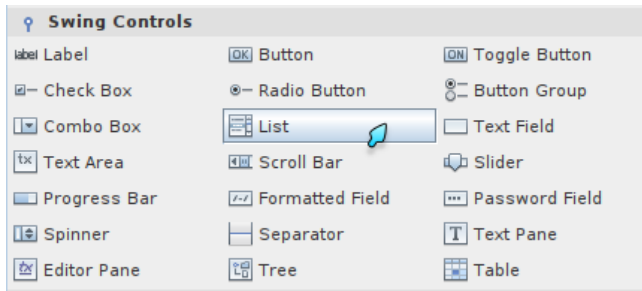


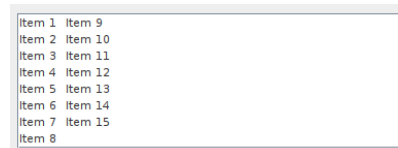
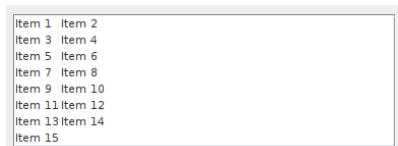
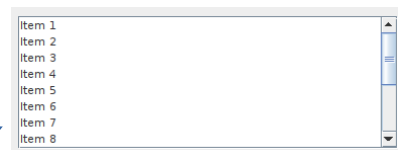
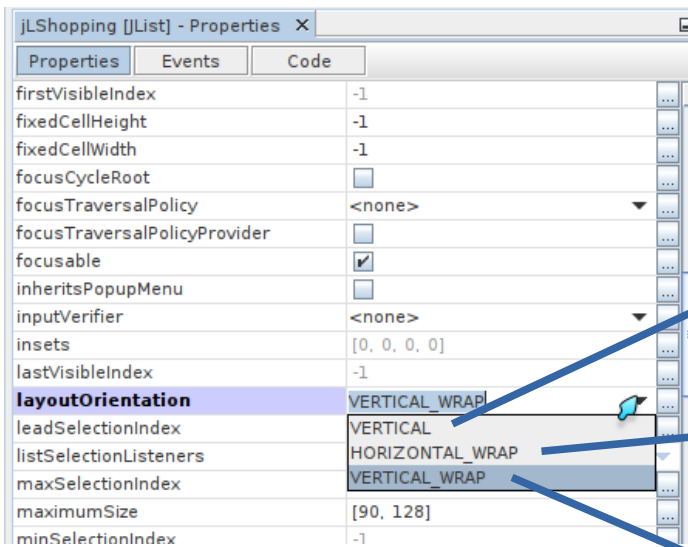
JList przedstawia użytkownikowi grupę przedmiotów wyświetlonych w jednej lub wielu kolumnach, jest to lista, w której można zaznaczyć elementy do wyboru. Jako że lista posiada wiele elementów zazwyczaj jest wkładana do Scroll Pane(pasek przewijania).



Możliwość zaznaczenia tylko jednego elementu listy

Możliwość zaznaczenia wielu elementów listy jeden pod drugim

Możliwość zaznaczenia dowolnych elementów listy



#### ZADANIE:

Stwórz w Netbeans program, który będzie posiadał trzy listy ułożone jedna obok drugiej w której elementy będą leżały jeden pod drugim (patrz rysunek na następnej stronie). Do tych list wczytaj z plików dane:

Do pierwszej **4\_day.txt**

Do drugiej **4\_month.txt**

Do trzeciej **4\_year.txt**

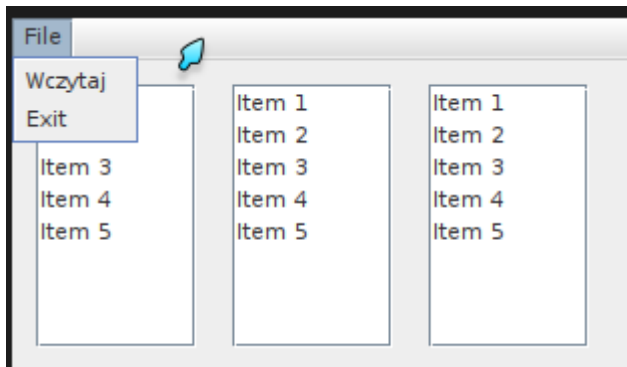
Wczytywanie danych z pliku wykonaj w osobnej klasie, którą nazwij

**ReadDataForJList**. Klasa ma posiadać tylko jedną metodę: readData(File filename), która ma zwracać **DefaultListModel**.

Program ma posiadać proste menu z polami:

- Wczytaj – po jego kliknięciu do poniższych JList zaczytują się dane z plików,
- Exit – po kliknięciu którego zamyka się aplikacja.

Przykładowy wygląd:



Jak utworzyć **DefaultListModel**?

```
DefaultListModel defaultListModel = new DefaultListModel();
defaultListModel.addElement("Janosik");
defaultListModel.addElement("Rumcajs");
jLShopping.setModel(defaultListModel);
```

Dodawanie elementów do defaultListModel

Dodanie do JList – Janosik i Rumcajs wyświetlą się w JList

Metoda zwracająca obiekt **DefaultListModel**:

```
private DefaultListModel getListModel(){
    DefaultListModel defaultListModel= new DefaultListModel();
    defaultListModel.addElement("Gerałt");
    defaultListModel.addElement("Ciri");
    defaultListModel.addElement("Yenefer");
    defaultListModel.addElement("Yarpen");
    return defaultListModel;
}
```

Użycie:

```
jList.setModel(nazwa_obiektu.getListModel());
```

Dodawanie elementów/wypełnienie listy w kodzie programu:

```
String week[] = { "Monday", "Tuesday", "Wednesday",  
                 "Thursday", "Friday", "Saturday", "Sunday"};  
jLShopping.setListData(week);
```

Tworzysz tablicę z elementami listy

Dodajesz do JList dane – tutaj tablica

Nazwa listy w programie

Dodawanie elementów można też wykonać używając poniższej instrukcji (preferowane - metoda przykładowa)

```
private void addElementToListUsingDefaultListModel(){  
    DefaultListModel defaultListModel = new DefaultListModel();  
    defaultListModel.addElement("Janosik");  
    defaultListModel.addElement("Rumcajs");  
    jLShopping.setModel(defaultListModel);  
}
```

Dodanie elementu do już istniejącej listy (metoda przykładowa):

```
private void addElementToExistingJList(){  
    DefaultListModel defaultListModel = new DefaultListModel();  
    defaultListModel.addElement("Geralt");  
    defaultListModel.addElement("Ciri");  
    defaultListModel.addElement("Yenifer");  
    jLShopping.setModel(defaultListModel);  
    defaultListModel.addElement("Yarpen");  
}
```

Elementy listy

Dodanie elementów do LISTY

Dodanie nowego elementu do już istniejącej listy

UWAGA:

**DefaultListModel** ma być zadeklarowany globalnie aby można było wykonać powyższe operacje w różnych metodach danej klasy.

#### USUWANIE:

Usunięcie elementu poprzez podanie "nazwy"

```
defaultListModel.removeElement("Geralt");
```

Usunięcie elementu poprzez podanie indeksu

```
defaultListModel.removeElementAt(2); //usunięcie 3 elementu z listy
```

Usunięcie wszystkich elementów listy:

```
defaultListModel.removeAllElements();
```