

Usuwanie zaznaczonego elementu z `JList` po naciśnięciu przycisku:

```
private void jBRemoveTaskActionPerformed(java.awt.event.ActionEvent evt) {  
    System.out.println("Removed task:");  
    int index = jLTask.getSelectedIndex();  
    String title = dlm.get(index).toString();  
    System.out.println("Index: "+index+"\ntitle: "+title);  
    dlm.remove(index);  
}
```

`dlm` – jest to: **DefaultListModel**, który przechowuje elementy do wyświetlania w `JList`. W powyższym przypadku jest to obiekt **String**.

Usunięcie ostatniego elementu (ostatnio dodanego):

```
dlm.removeElementAt(dlm.size()-1);
```

Różnica między: **remove** a **removeElementAt** – pierwszy zwraca usunięty obiekt (Object), a więc można przypisać do zmiennej np.:

```
String temp = (String) dlm.remove(index);
```

Dodawanie elementu do `JList`:

<div><div>Tytuł zadania:</div><div><input type="text"/></div></div> <div><div>Treść zadania:</div><div><div></div></div></div> <div><div>Nazwa</div><div><div>DODAJ ZADANIE</div></div></div>	<div>Nazwa <code>jTFTaskTitle</code></div> <div>Nazwa <code>jTATaksDescription</code></div>
---	---

Po naciśnięciu dodaj:

```
private void jBAddTaskActionPerformed(java.awt.event.ActionEvent evt) {  
    String title = jTFTaskTitle.getText();  
    dlm.addElement(title);  
}
```

Jak zapewne zauważyłeś nie użyliśmy pola: `jTATaksDescription`.

Pojawi się ono później, jak do `DefaultListModel` zapiszemy stworzony przez nas obiekt zawierający odpowiednie dane!

Przerobimy teraz nasze zadania tak aby można było zapisać dane o naszym zadaniu takie jak: tytuł i opis.

```
/**  
 * Object to store title and description of task  
 * @author adams  
 */  
public class Task {  
    private String title;  
    private String description;  
  
    public Task(String title, String description) {  
        this.title = title;  
        this.description = description;  
    }  
  
    public String getTitle() {  
        return title;  
    }  
  
    public String getDescription() {  
        return description;  
    }  
}
```

Patrz następna strona!

Teraz aby dodać nowy **Task** do **DefaultListModel** **nie wolno** zrobić jak poniżej. Sprawdź co się stanie jak wywołamy poniższy kod programu:

```
private void jBAddTaskActionPerformed(java.awt.event.ActionEvent evt) {  
    String title = jTFTaskTitle.getText();  
    String description = jTATaksDescription.getText();  
    dlm.addElement(new Task(title,description));  
}
```

Do **DefaultListModel** dodajemy tylko String z tytułem naszego zadania, ale zadanie należy też gdzieś przechować. Do obiektu **Task** należy dodać metodę **toString()**,

@Override

public String toString() {

która zwróci tylko **title** z naszego obiektu! Wtedy powyższy kod z dodawaniem **Task** do **DefaultListModel** zadziała!!!

Teraz lekko zmodyfikuj usuwanie elementów z **DefaultListModel** tak aby informacje wyświetlały się wszystkie z naszego obiektu jak poniżej:

```
Removed task:  
Index: 0      title: Zad 1  
Bardzo ciężkie zadania
```

Dodatkowo jeśli lista jest pusta masz zablokować możliwość usunięcia elementu z **jList**!

UWAGA:

Musisz zmienić sposób przechowywania danych. Już nie w **DefaultListModel** bo tam są tylko dane typu **String**.

**ArrayList<Task> listOfTasks;**