# 61FIT3JSD
# Fall 2023

**Lecture 8**

GUI programming (3)

Advanced issues

## Lecture outline

- GUI application development
- Multi-tasking GUI
- Dialog
- Scroll bar: JScrollBar

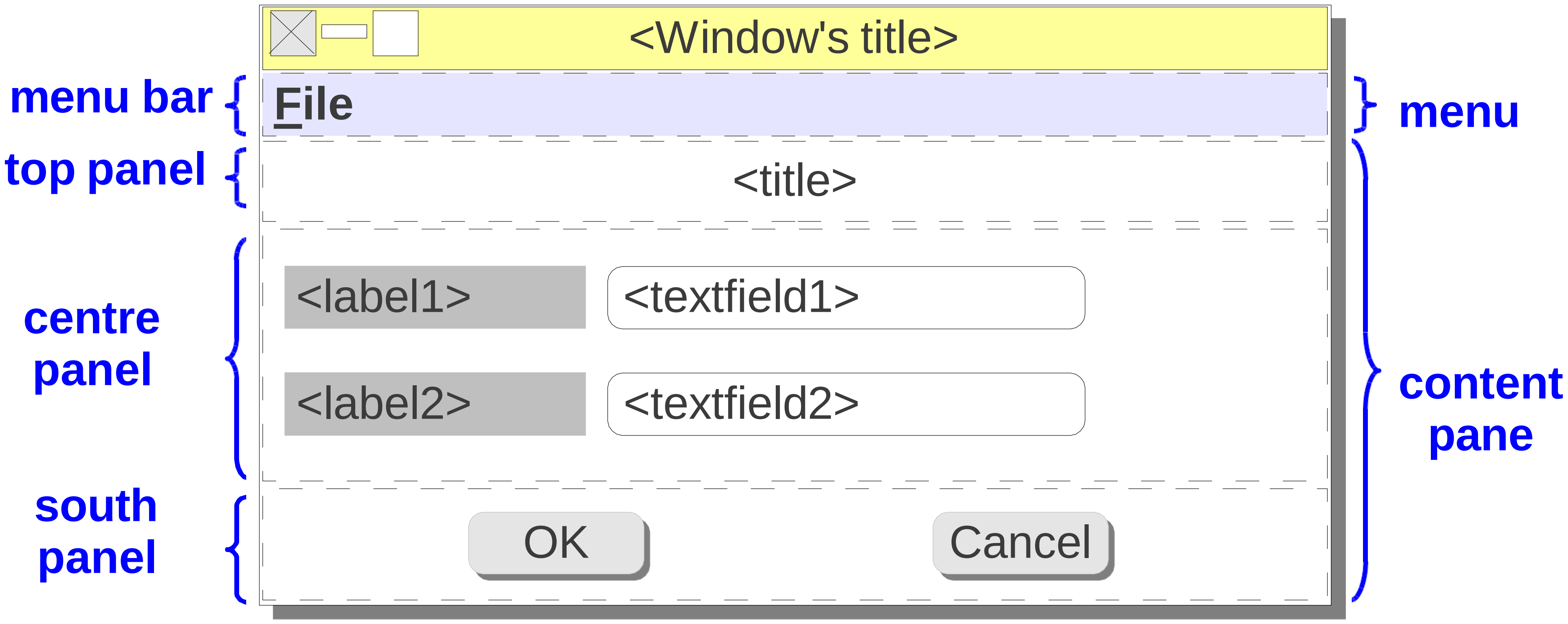
## GUI application development

- Design
- Implementation

## Design

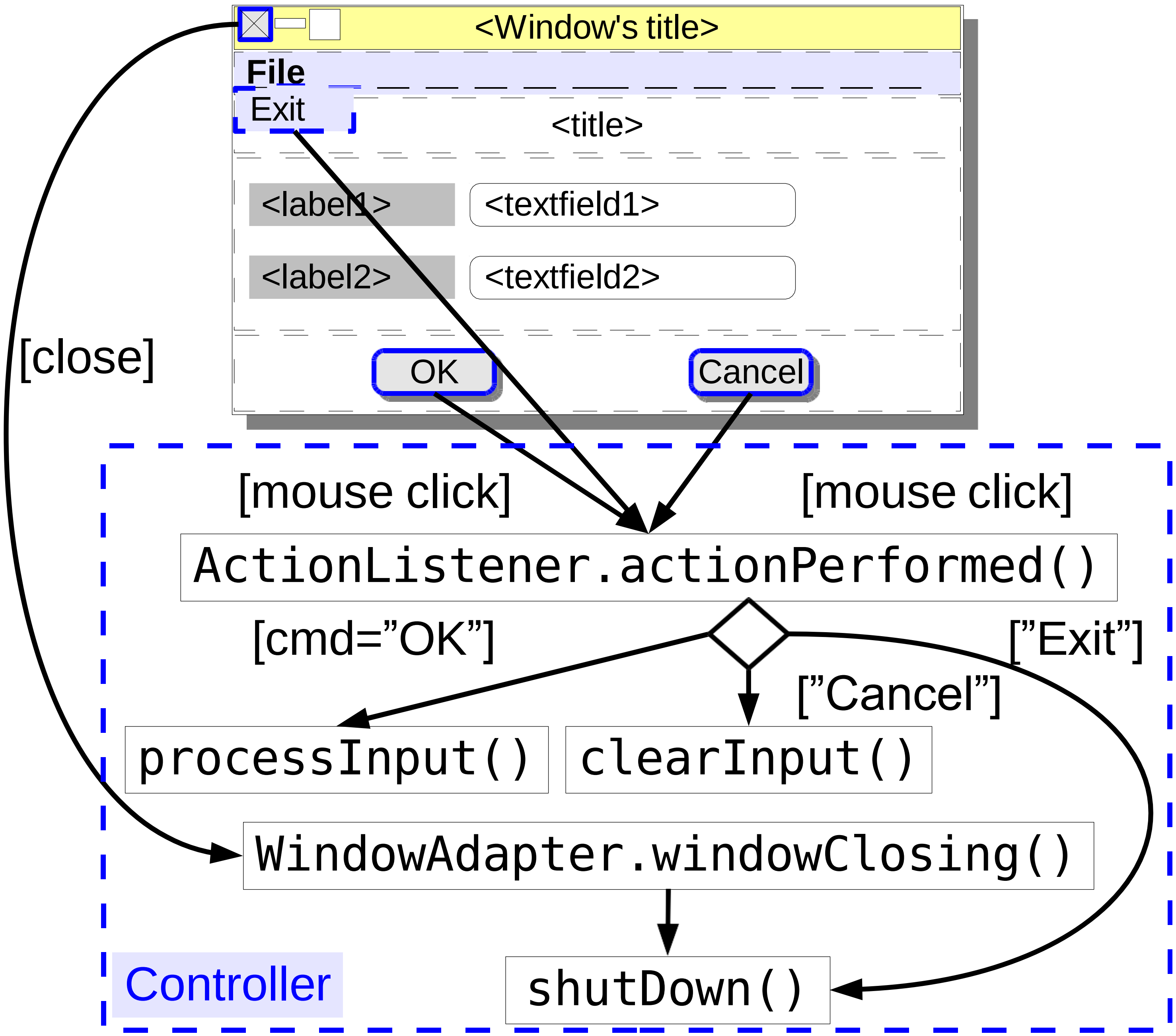
- **Model**:
  - Create domain-specific classes (e.g. Customer)
- **View**:
  - Create window and display components
- **Controller**:
  - Define event handlers (user interaction)
  - Start up: initialise view & model
  - Display the view

# View design



<Window's title>
menu bar
File
menu
top panel
<title>
centre panel
<label1>
<textfield1>
<label2>
<textfield2>
content pane
south panel
OK
Cancel

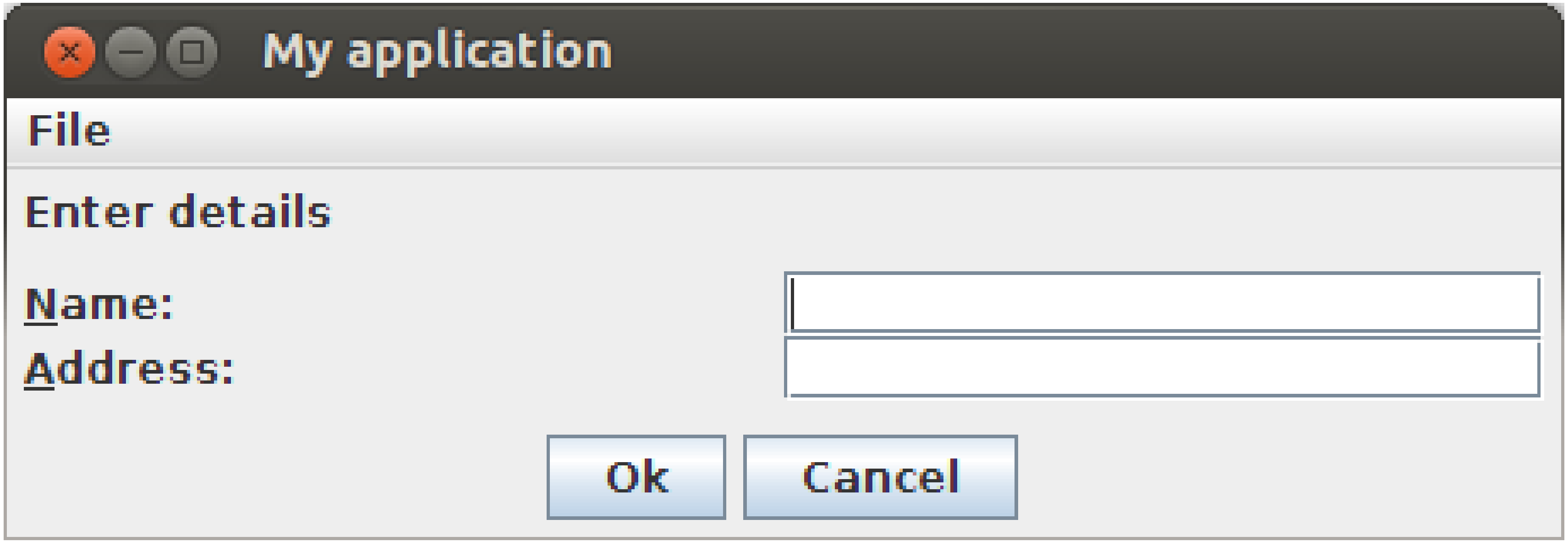
# Controller design



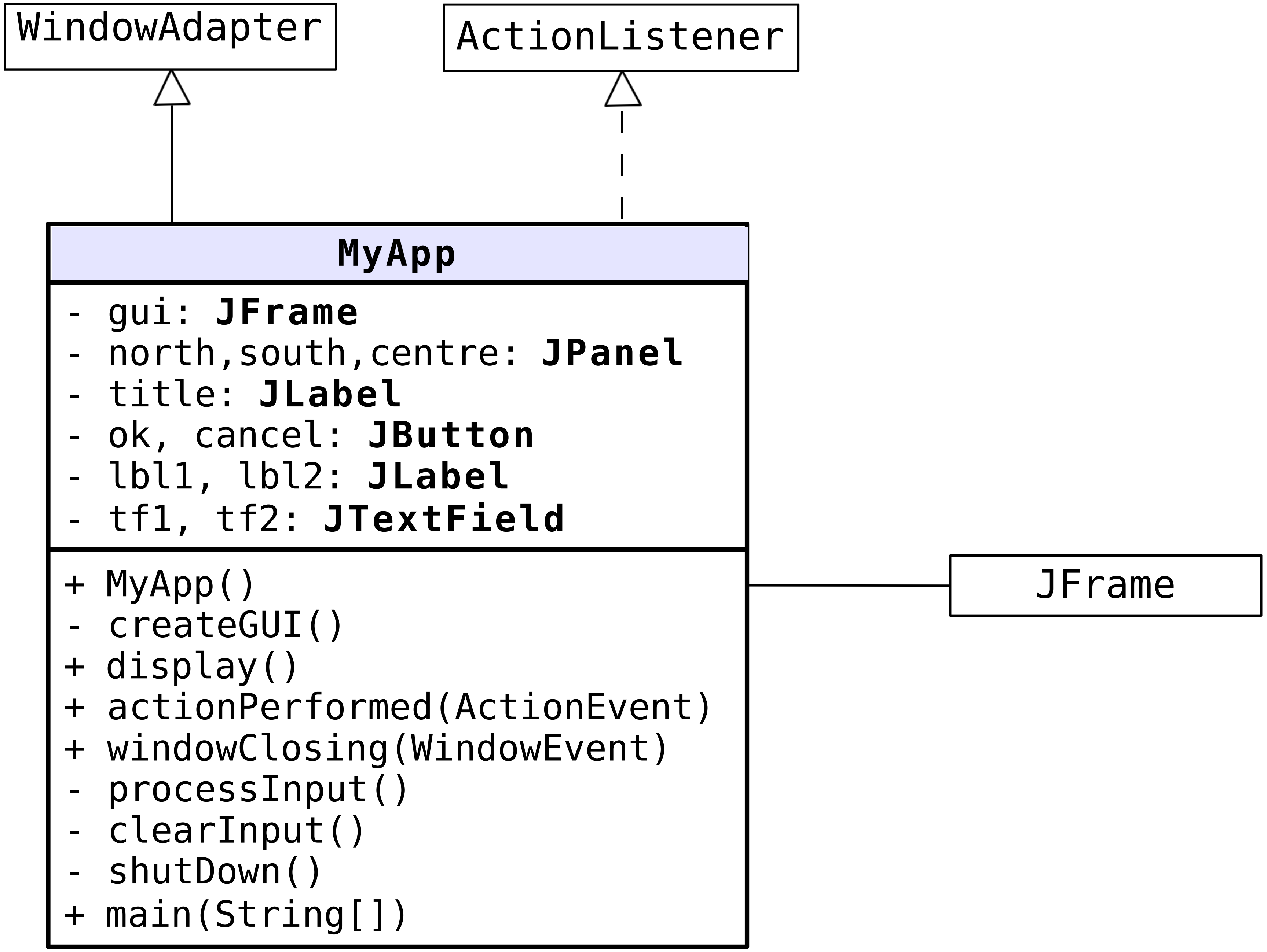
## Design #1: all-in-one

- Model, View and Controller are combined into one class
- Used for small applications:
  - **model**: primitive data values
  - **view**: simple interface
  - **controller**: simple user actions
- **Pros**: less code to write
- **Cons**: longer class → more difficult to maintain; not suitable for larger applications

## Example: MyApp



# MyApp design #1



## Implementation

- GUI development tasks:
  - set up the window: layout, menu
  - create & set up the container objects
  - add display components to the containers
  - add the containers to the window

DEMO

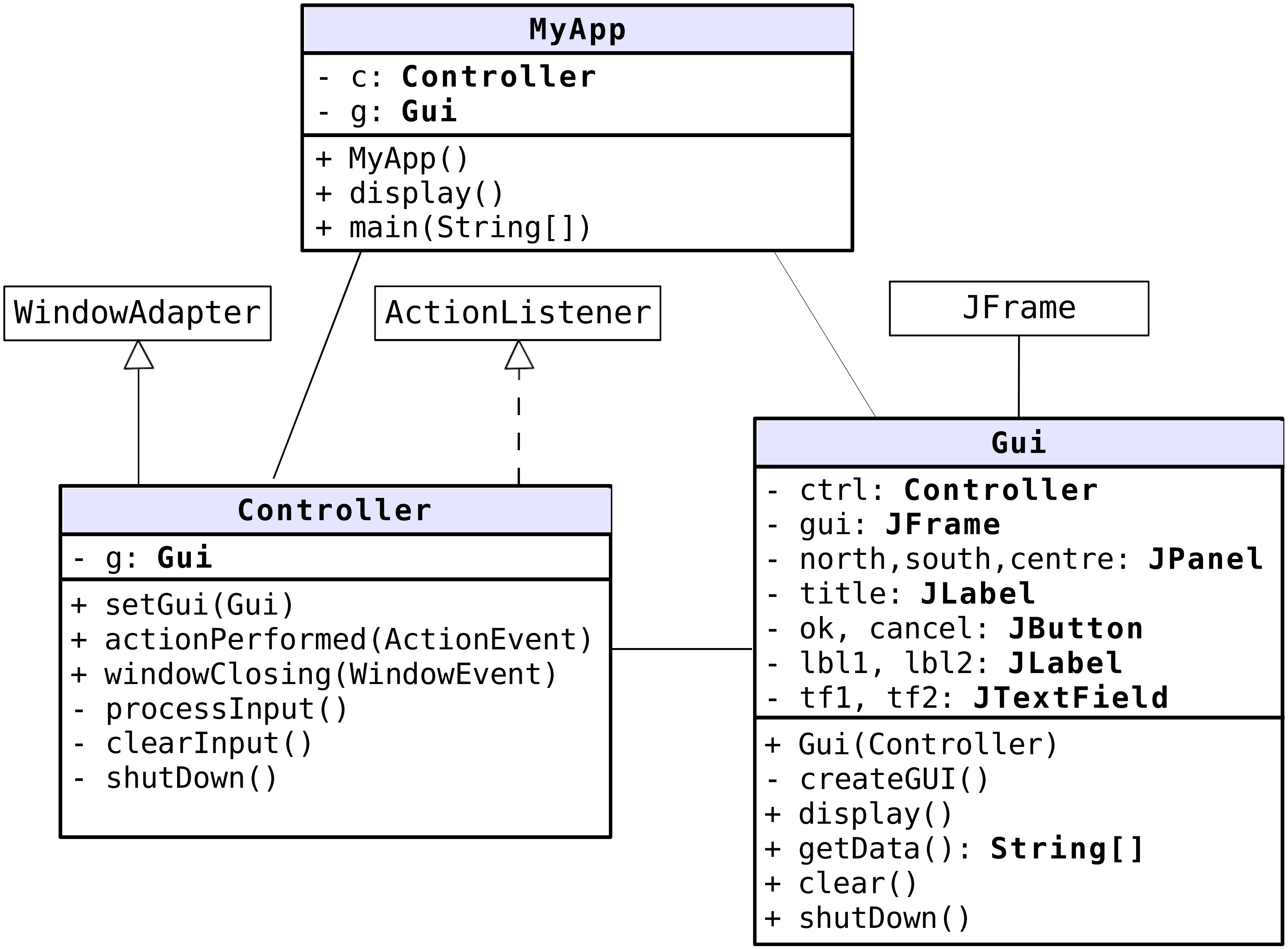
## Example: MyApp #1

```
lect08.allinone.MyApp
```

## Design #2: independent controller

- Model, View may be combined into one class
- Controller is a separate class
- Used for medium-large applications:
  - **model**: domain-specific classes (e.g. Customer, Order, etc) that <u>may not</u> require separate classes
  - **view**: simple view, specific to each domain class
  - **controller**: data handling is likely to change
- **Pros**: easier to maintain (e.g. when data handling logics or view specifications are changed)
- **Cons**: more complex to design and code

# Example: MyApp design #2



DEMO

## MyApp #2

```
lect08.independent.MyApp
```

## Multi-tasking GUI

- A multi-tasking GUI application can handle multiple events at the same time
- Examples:
  - store program data to a database
  - view a report
  - print data

## Multi-tasking in Swing

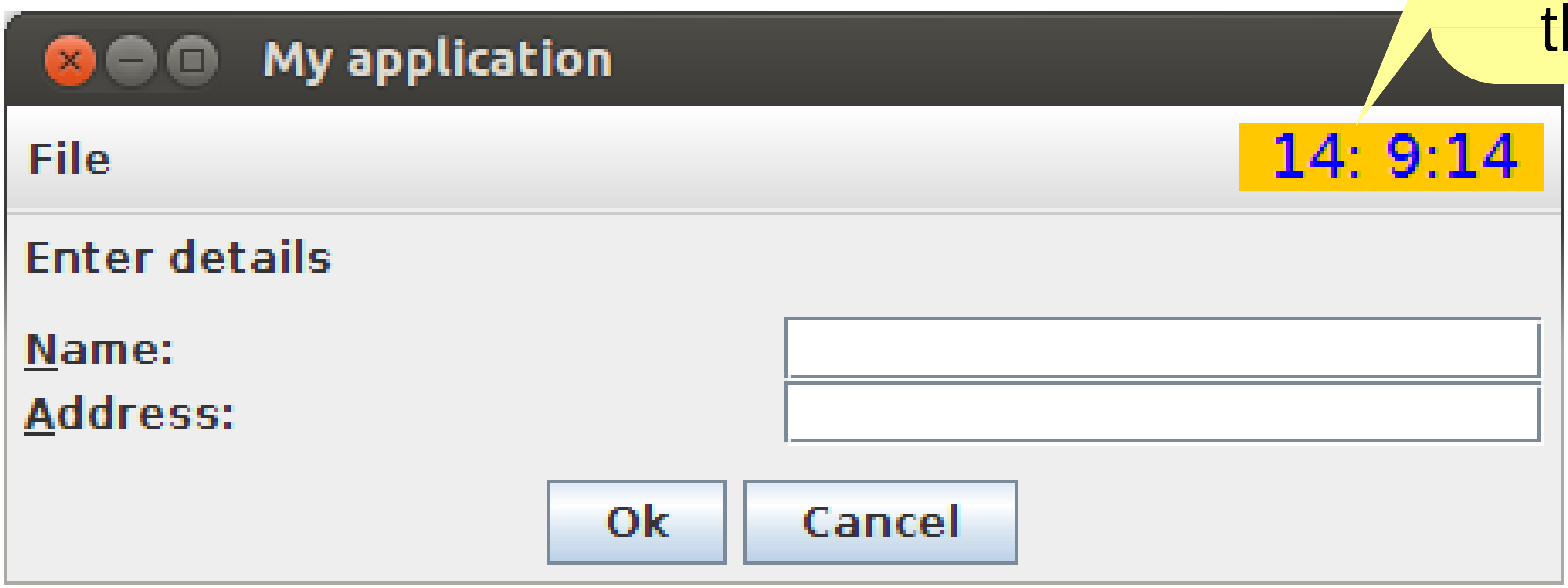
- Wrap the task in a `Runnable` object
- Start the task object using a `Thread` object
- Task thread is run concurrently with the GUI's thread:
    - user interaction is not blocked

DEMO

# Multi tasking

```
lect08.multitask.MyApp
```

Timer task running on a separate thread



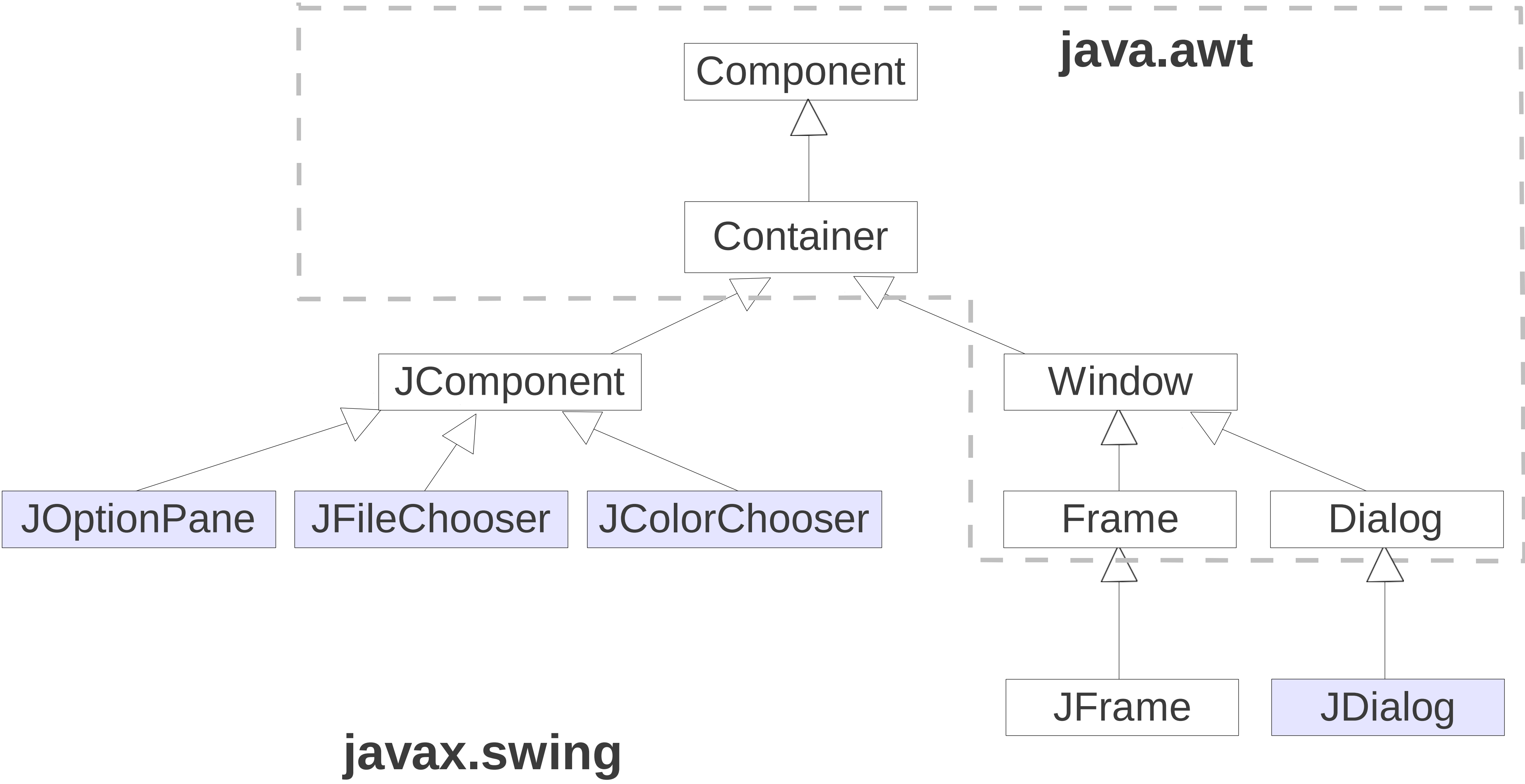
## Dialog

- Separate sub-window that:
  - displays temporary notice or
  - obtains basic, context-dependent input
- Examples:
  - program message (informational, error)
  - progress status
  - browse a file or choose a colour
- Attached to a window (its parent)
- Can be modal or non-modal

## Swing dialogs

- `JOptionPane`: simple, standard dialog
- `JFileChooser`: browse a file
- `JColorChooser`: choose a color
- `JDialog`: custom dialog

# Dialog component hierarchy



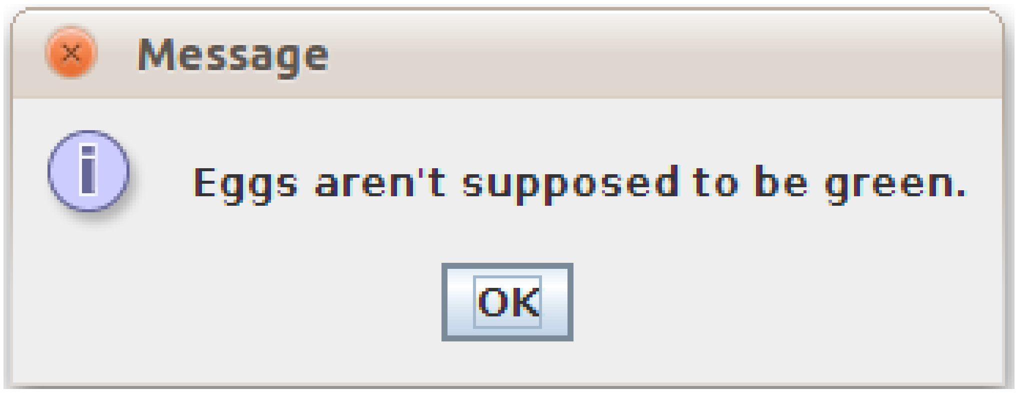
## Class `JOptionPane`

- A container that uses `JDialog` as the window
- Creates modal dialogs
- Customisable features:
  - title
  - message or a collection of components
  - icons
  - buttons
  - button texts

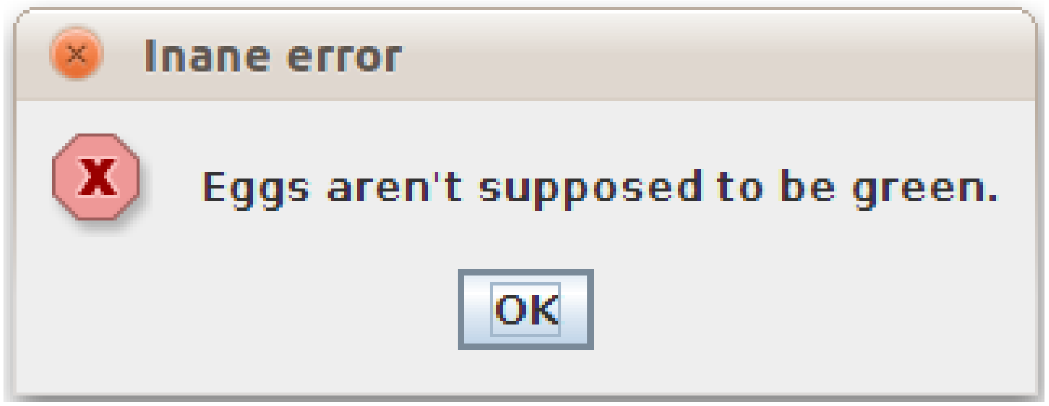
## Types of dialog

- Message dialog:
  - one-button dialog
- Option dialog:
  - like a message but has a variety of buttons
- Input dialog:
  - to obtain a text input

# Example (1)

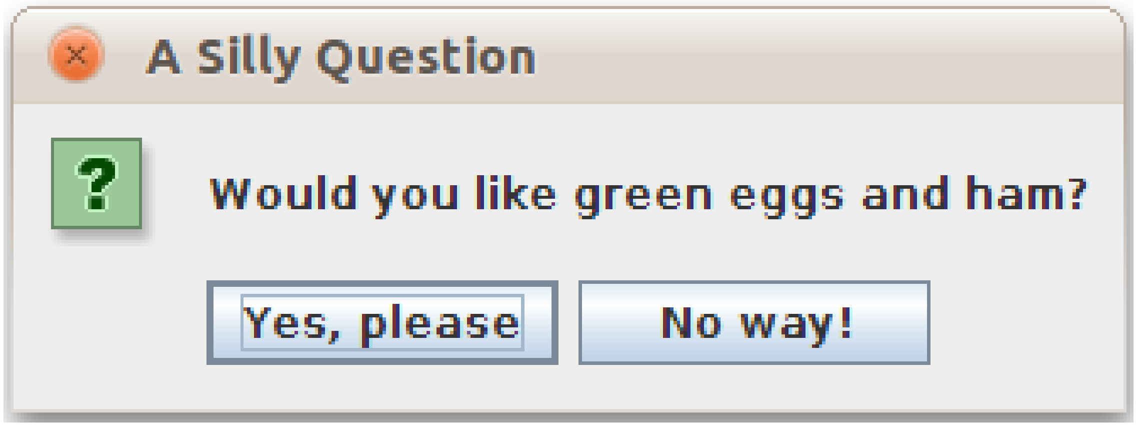

Message

Eggs aren't supposed to be green.


Inane error

Eggs aren't supposed to be green.

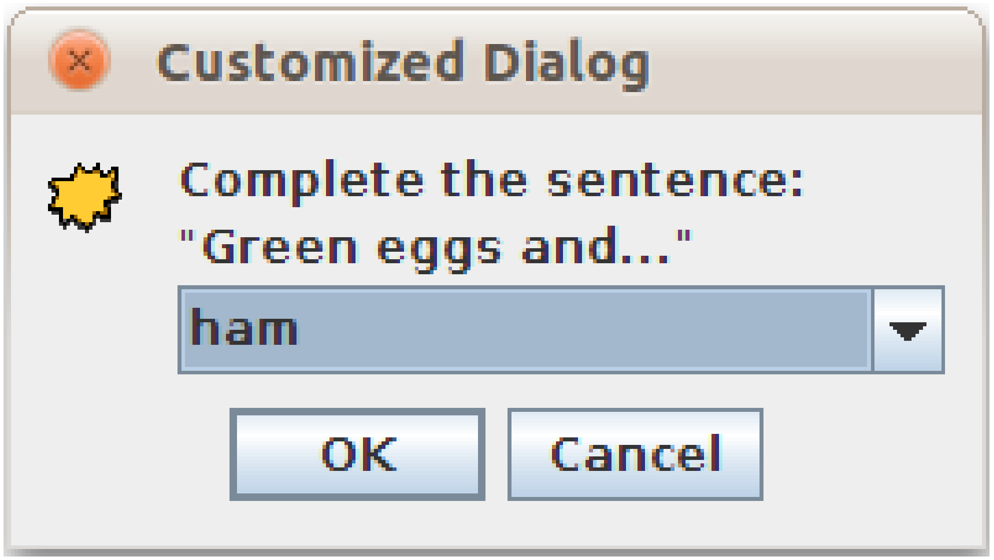
message dialogs


A Silly Question

Would you like green eggs and ham?

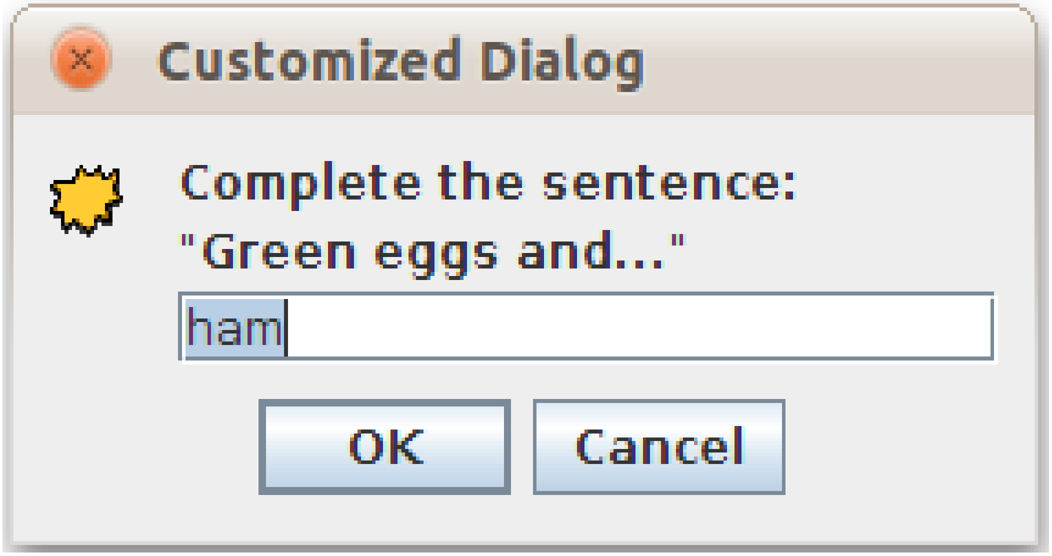
option dialog

## Example (2)


Customized Dialog

Complete the sentence:
"Green eggs and..."

ham


Customized Dialog

Complete the sentence:
"Green eggs and..."

ham

input dialogs

## Methods to create dialogs

- `showMessageDialog`
- `showOptionDialog`
- `showInputDialog`

## showMessageDialog

- `parentComponent`: the parent window (frame)
- `mesg`: the message to show
- `title`: the dialog title
- `messageType`:
  - INFORMATION_MESSAGE
  - ERROR_MESSAGE
  - WARNING_MESSAGE
  - PLAIN_MESSAGE

# showOptionDialog

- `parentComponent`
- `mesg`
- `title`
- `optionType`: a combination of Yes/No/Cancel
- `messageType`
- `icon`: an Icon object
- `options` (optional): list of button texts (matches with optionType)
- `initialValue`: initial (selected) button

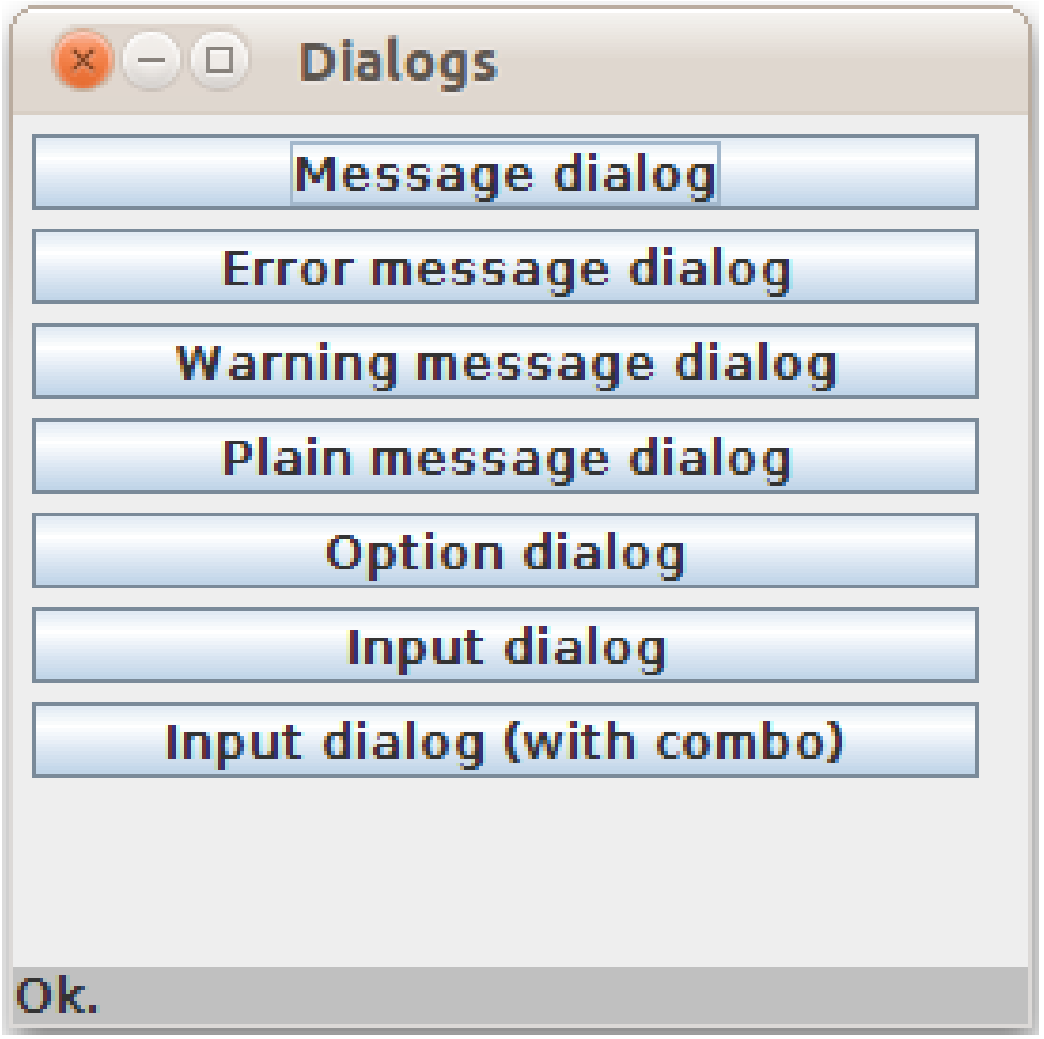
## showInputDialog

- `parentComponent`
- `mesg`
- `title`
- `messageType`
- `icon`: an Icon object
- `options` (optional): list of allowed values to select
- `initialValue`: initially (selected) value

DEMO

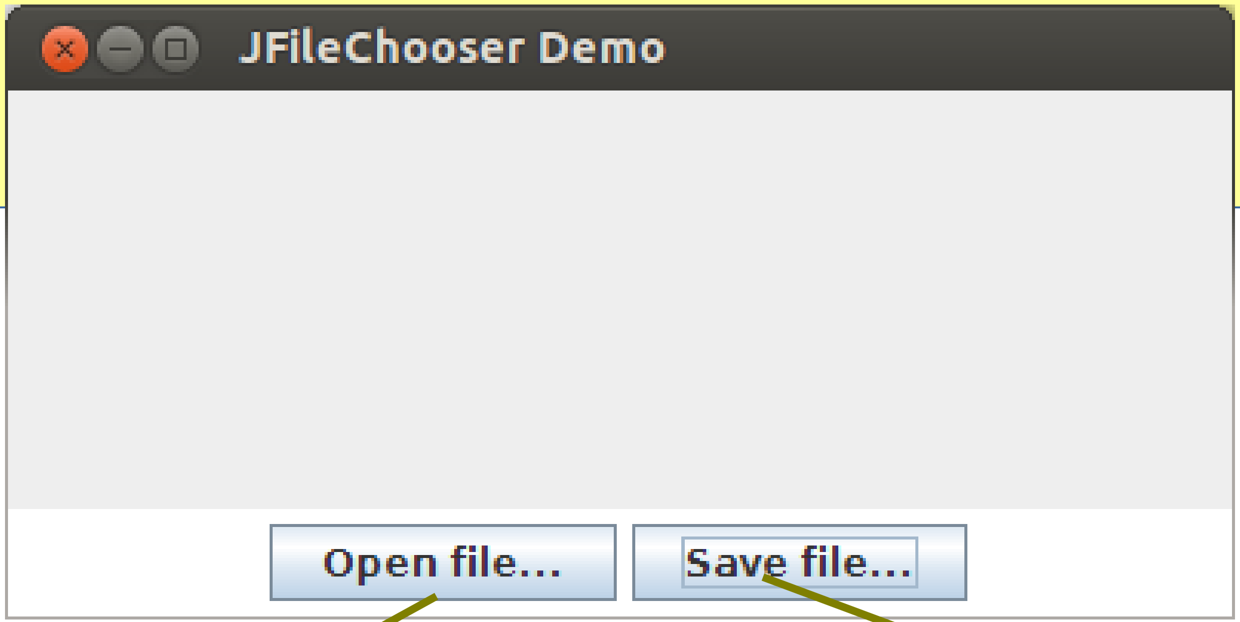
# JOptionPane

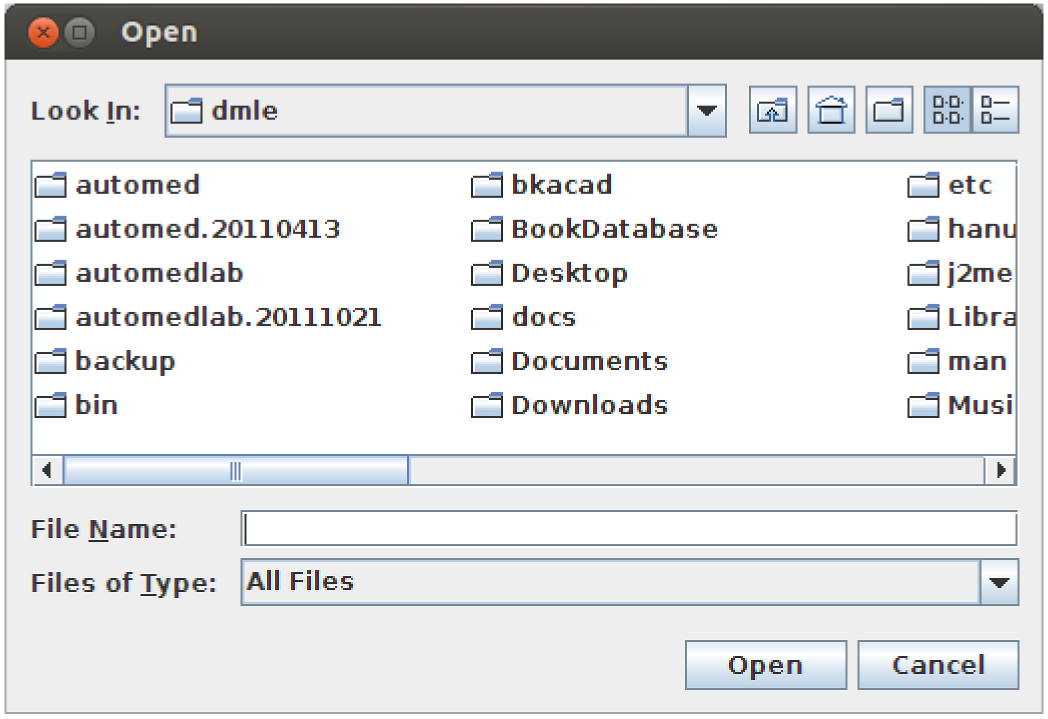
lect08.dialogs.SimpleDialogDemo

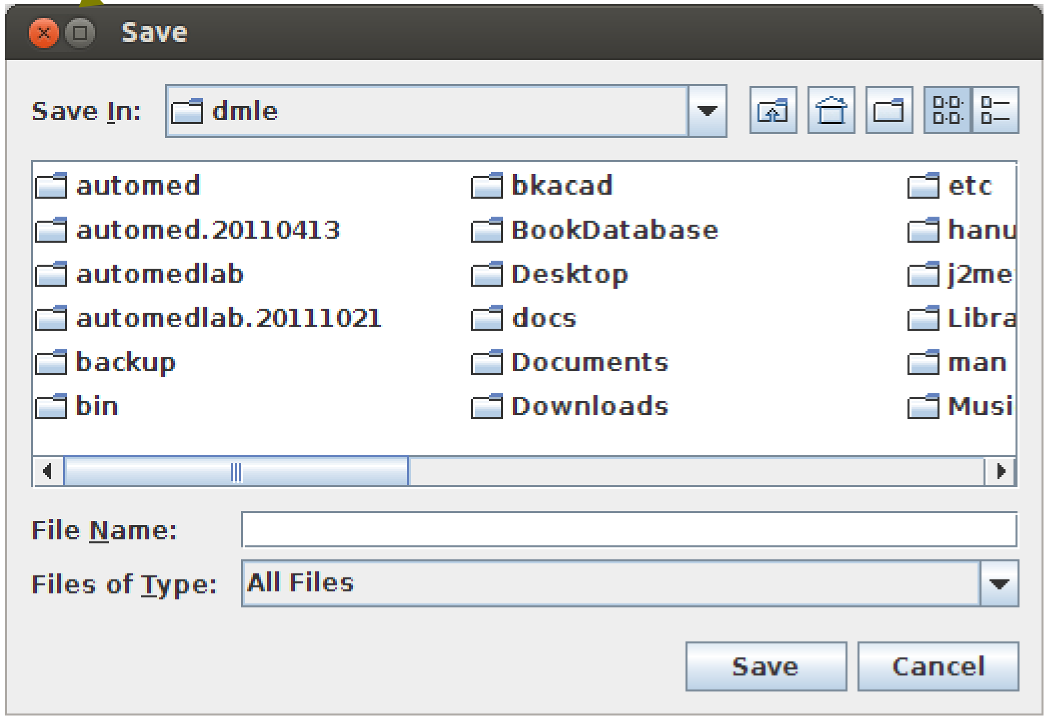


# JFileChooser

lect08.dialogs.
JFileChooserDemo



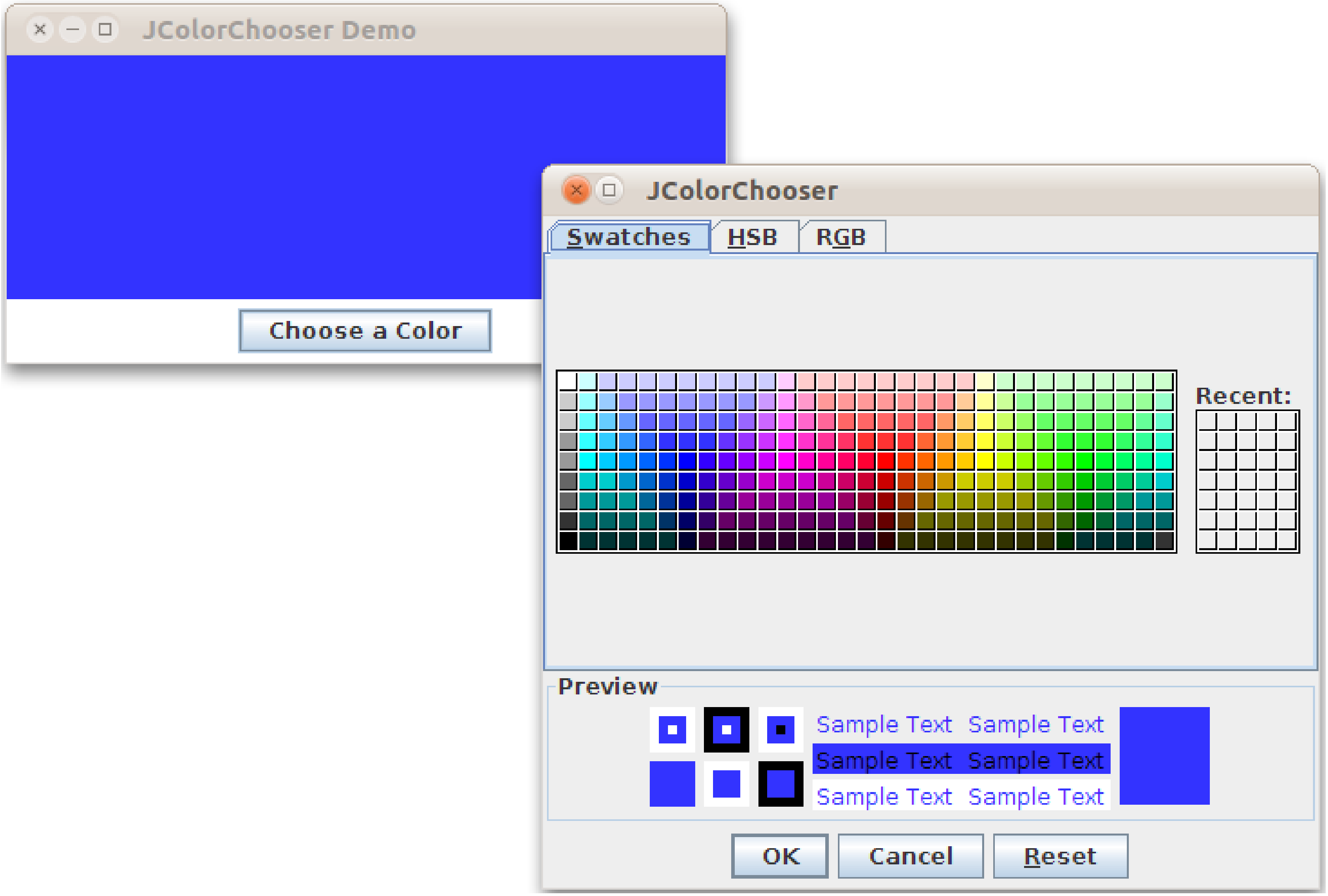




DEMO

# JColorChooser

```
lect08.dialogs.JColorChooserDemo
```



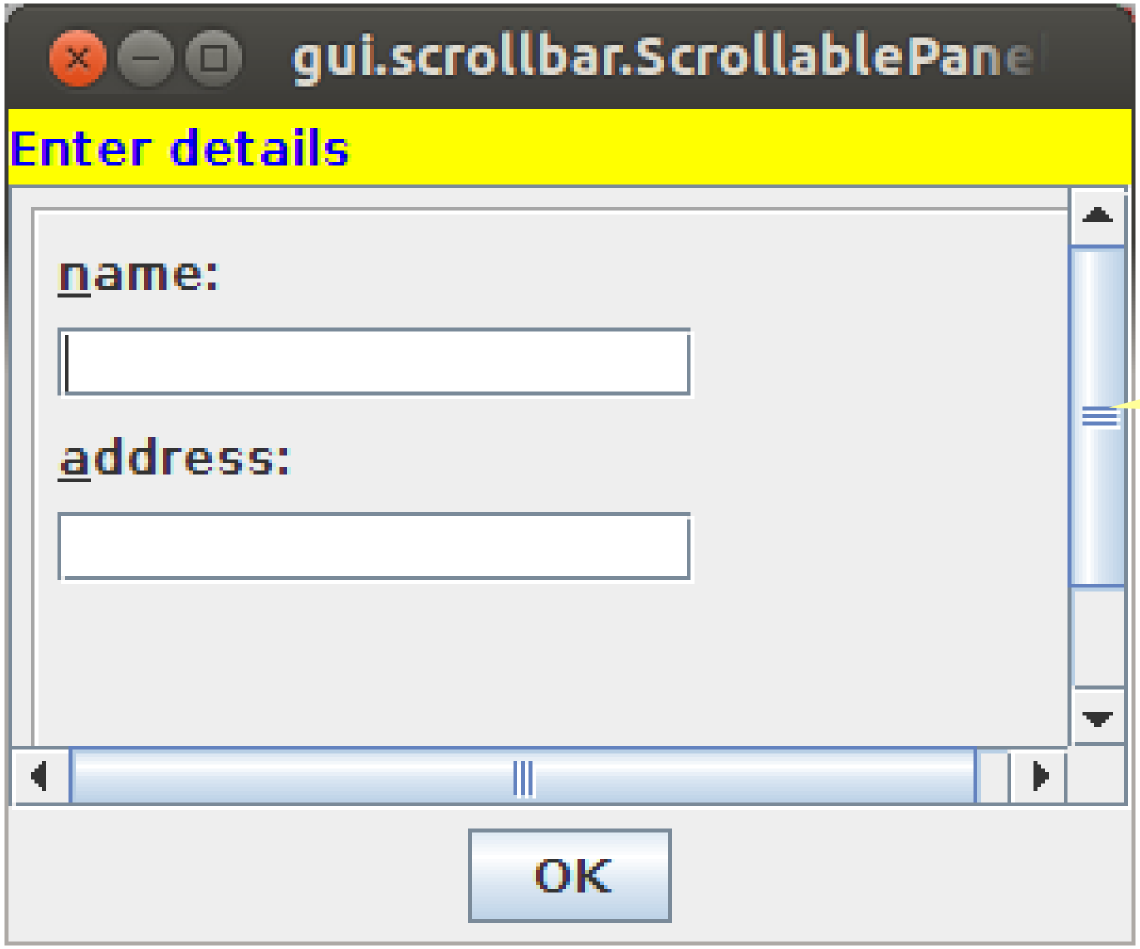
## Scroll bar

- Class: `JScrollPane`
- Represents a fixed, sliding view of a display component
- Create a `JScrollPane` object using the component as input
- Add the `JScrollPane` object to the window
- Examples:
  - scrollable panel
  - scrollable text field
  - scrollable table (later)

DEMO

# Scrollable panel

```
lect08.scrollbar.ScrollablePanelDemo
```



Scroll bar for a panel