

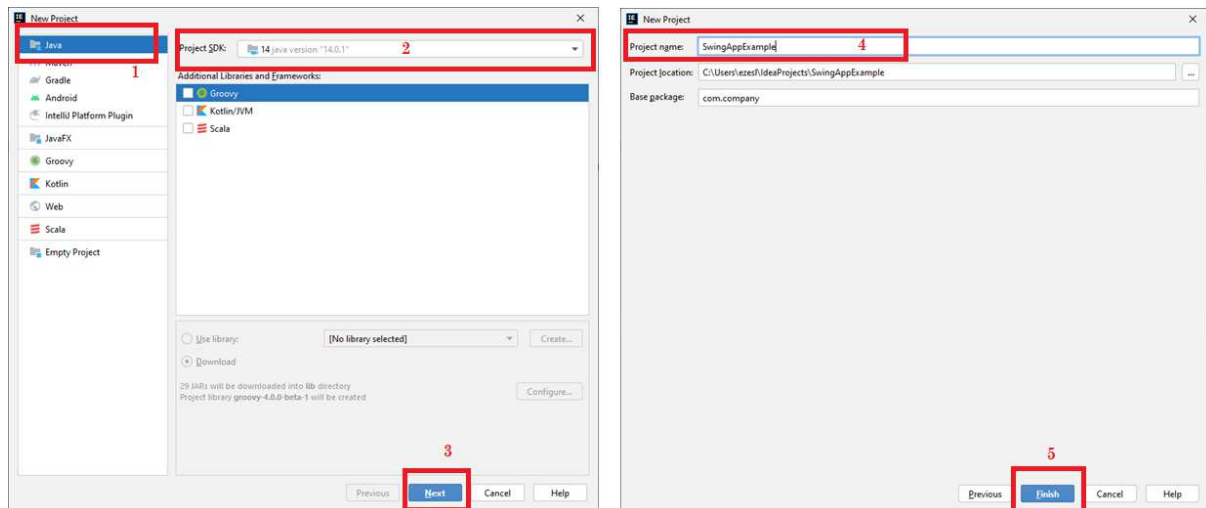
## PROGRAMOWANIE OBIEKTOWE

### GUI JAVA SWING LABORATORIUM

#### JPANEL, LAYOUT MANAGER, JBUTTON, JLABEL, JTEXTFIELD

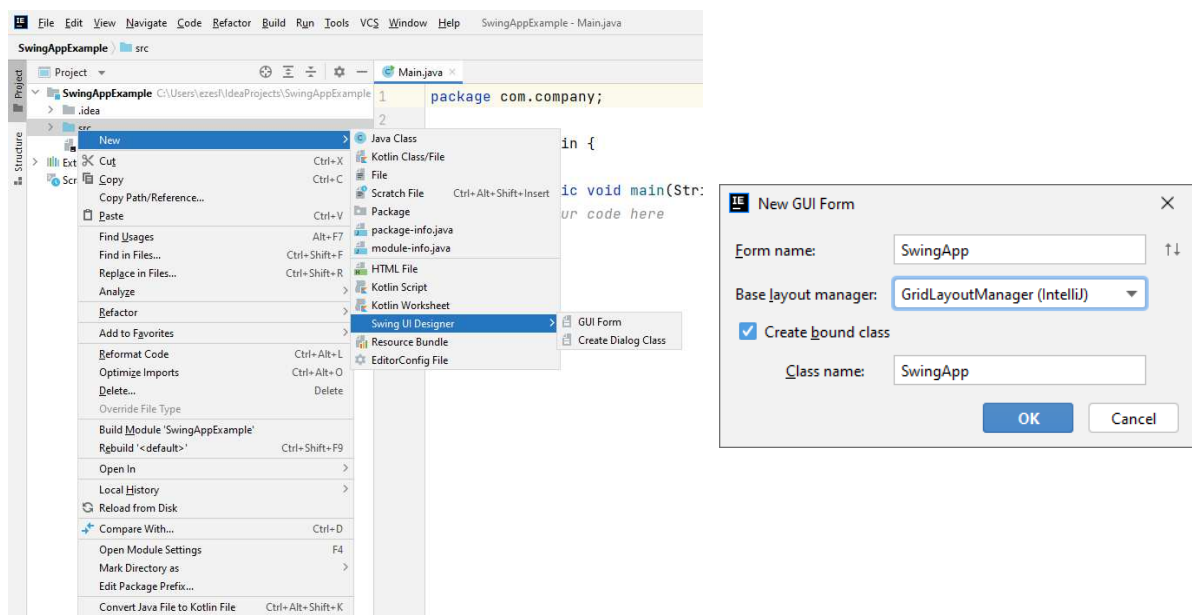
#### TWORZENIE PROJEKTU

File->New Project -> Wybieramy utworzenie projektu Java (1) oraz wersję SDK (2), przechodzimy dalej (3) i w kolejnym oknie wpisujemy nazwę projektu (4) i go tworzymy (5).



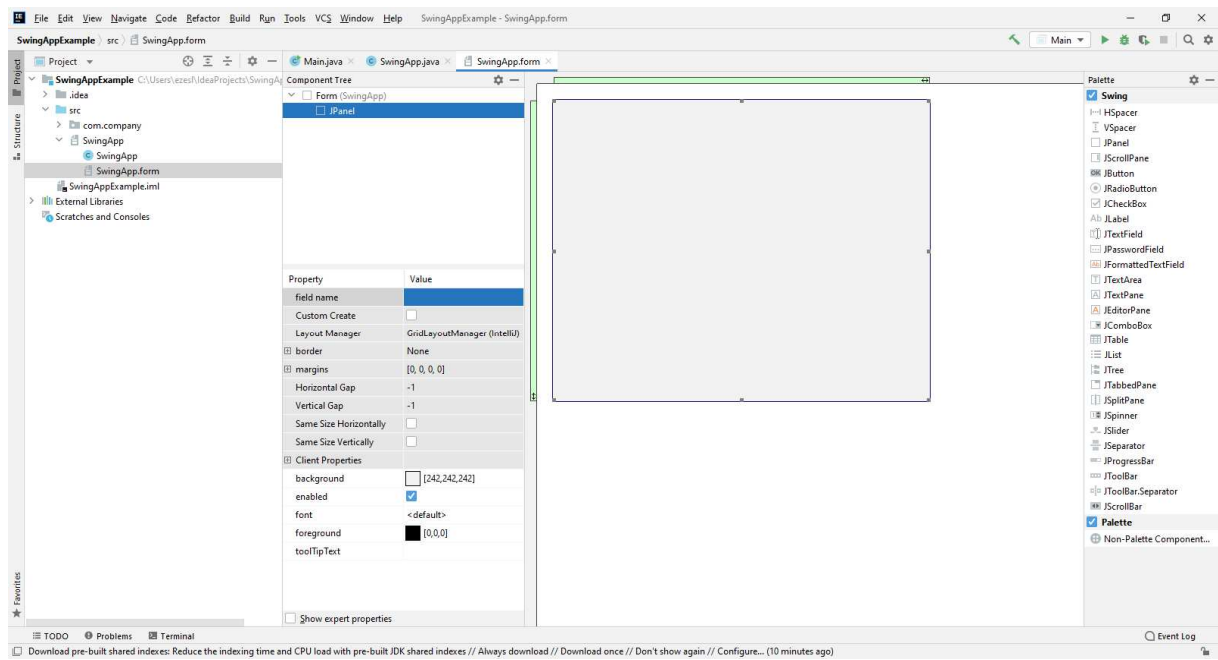
Rysunek 1. Tworzenie projektu

W kolejnym kroku należy utworzyć Swing UI Designer GUI Form. W kolejnym okienku wpisujemy nazwę oraz możemy zdefiniować base layout.



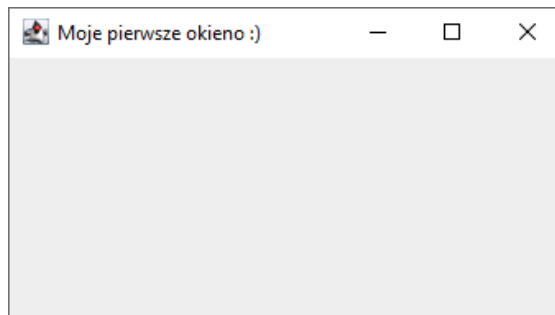
Rysunek 2. Tworzenie GUI Form

Widok po utworzeniu GUI:



Rysunek 3. Widok utworzonego GUI

## JFRAME



Rysunek 4. Widok okna GUI

```
public class Example01 extends JFrame {
    private JPanel JPanell;

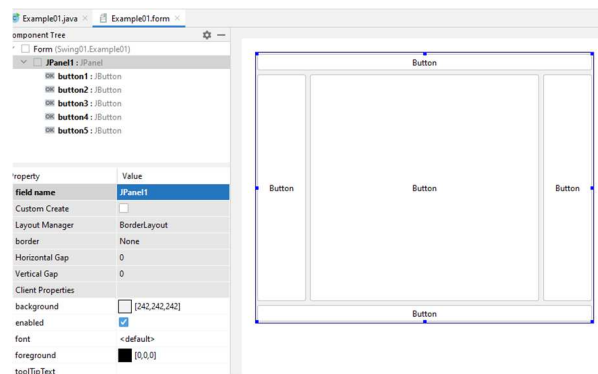
    public static void main(String[] args) {
        Example01 SwingExample = new Example01();
        SwingExample.setVisible(true); /**wyświetlanie ramki */**
    }

    //konstruktor
    public Example01() {
        //I sposób poprzez dziedziczenie z JFrame
        super("Moje pierwsze okno :) ");
        this.setContentPane(this.JPanell); // wyświetlonei na ekranie
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //metoda pozwalająca na
        zamknięcie okna
        //this.pack();
        /*upakowanie okna zgodnie z preferowanymi rozmiarami komponentów,
        które są zawarte w Frame, rozmiar będzie się doapospowywał do rozmiaru */
    }
}
```

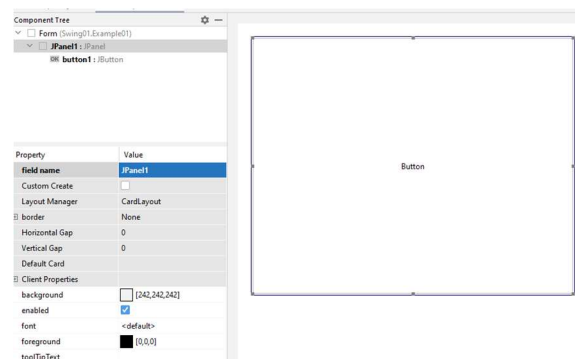
## LAYOUT MANAGER

W Javie możemy umieszczać komponenty w następującym układzie:

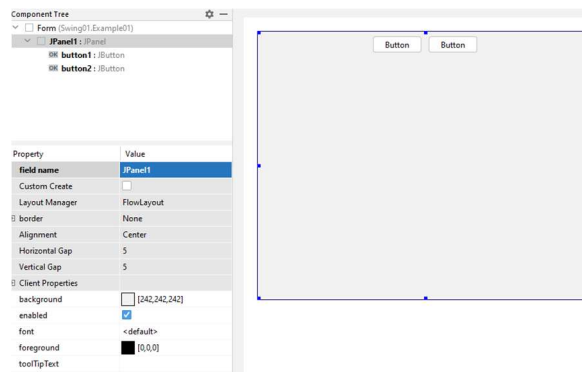
### a) BorderLayout



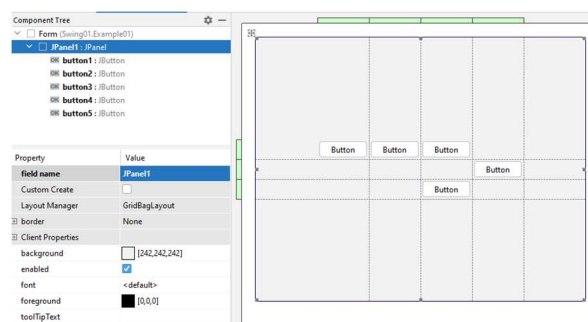
### b) CardLayout



### c) FlowLayout

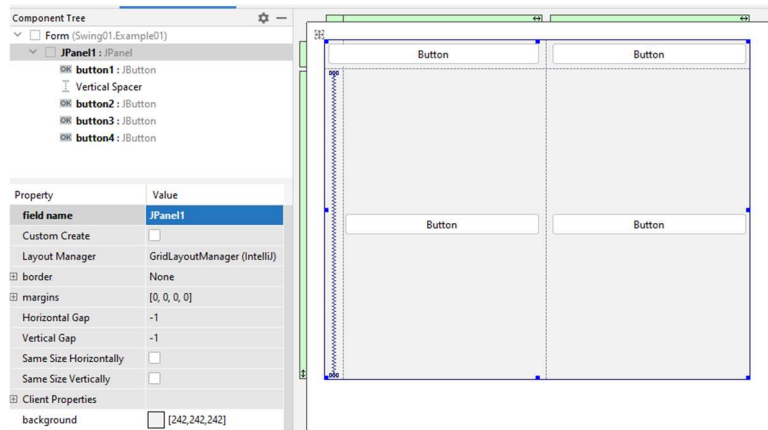


### d) GridBagLayout



### e) GridLayoutManager IntelliJ

Pozwala na umieszczanie komponentów w siatce kolumn i wierszy, a ich rozmiar może być różny.



### JBUTTON

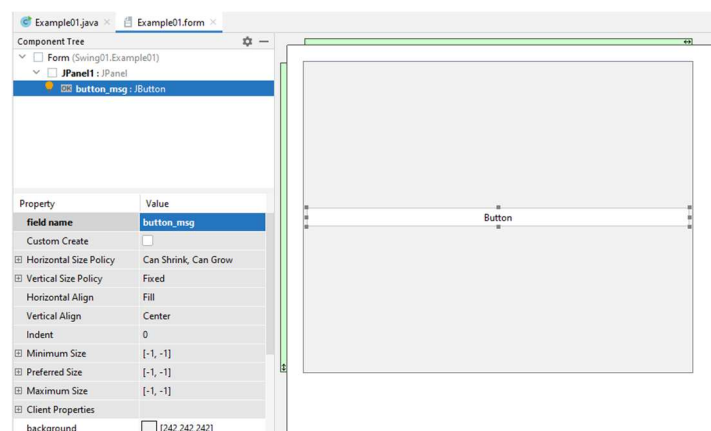
`public class JButton extends AbstractButton implements Accessible`

#### Commonly used Constructors:

Constructor	Description
<b>JButton()</b>	It creates a button with no text and icon.
<b>JButton(String s)</b>	It creates a button with the specified text.
<b>JButton(Icon i)</b>	It creates a button with the specified icon object.

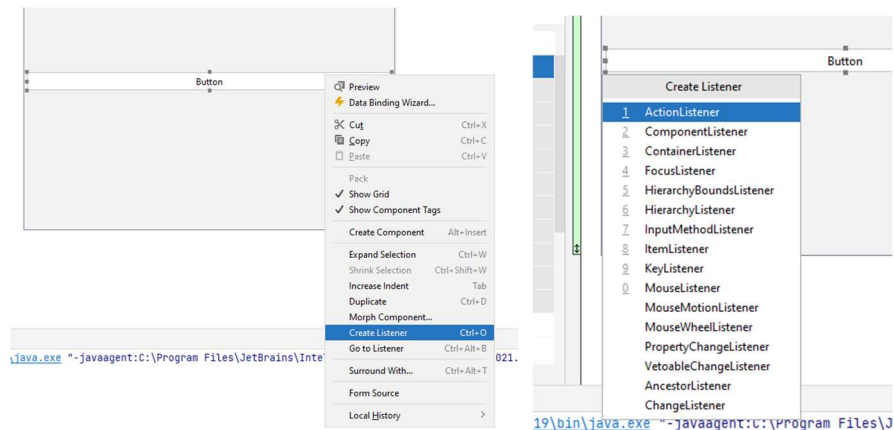
#### Commonly used Methods of AbstractButton class:

Methods	Description
<b>void setText(String s)</b>	It is used to set specified text on button
<b>String getText()</b>	It is used to return the text of the button.
<b>void setEnabled(boolean b)</b>	It is used to enable or disable the button.
<b>void setIcon(Icon b)</b>	It is used to set the specified Icon on the button.
<b>Icon getIcon()</b>	It is used to get the Icon of the button.
<b>void setMnemonic(int a)</b>	It is used to set the mnemonic on the button.
<b>void addActionListener(ActionListener a)</b>	It is used to add the action listener to this object.



**Przykład 1.** Należy utworzyć dwa przyciski, które pozwolą na wyjście z programu oraz wyświetlenie informacji.

Po dodaniu przycisku należy użyć Create Listener Ctrl+O, a następnie wybrać ActionListener, a następnie utworzyć automatycznie metodę `actionPerformed`



Kody metod:

```
button_msg.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Hello");
    }
});
buttonClose.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        dispose(); //metoda wyjścia
    }
});
buttonDate.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        //System.out.println(new Date());
        JOptionPane.showMessageDialog(null, new Date());
    }
});
```

## JLabel

```
public class JLabel extends JComponent implements SwingConstants,
Accessible
```

### Commonly used Constructors:

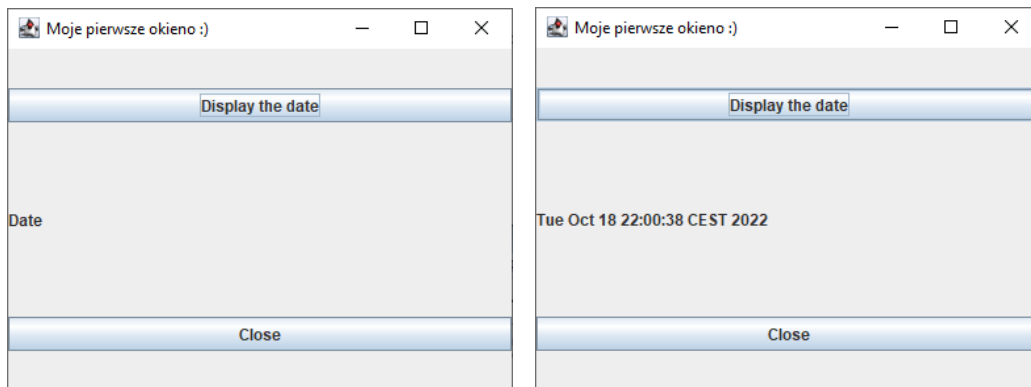
Constructor	Description
<b>JLabel()</b>	Creates a JLabel instance with no image and with an empty string for the title.
<b>JLabel(String s)</b>	Creates a JLabel instance with the specified text.
<b>JLabel(Icon i)</b>	Creates a JLabel instance with the specified image.
<b>JLabel(String s, Icon i, int horizontalAlignment)</b>	Creates a JLabel instance with the specified text, image, and horizontal alignment.

### Commonly used Methods:

Methods	Description
<b>String getText()</b>	t returns the text string that a label displays.
<b>void setText(String text)</b>	It defines the single line of text this component will display.
<b>void setHorizontalAlignment(int alignment)</b>	It sets the alignment of the label's contents along the X axis.
<b>Icon getIcon()</b>	It returns the graphic image that the label displays.
<b>int getHorizontalAlignment()</b>	It returns the alignment of the label's contents along the X axis.

**Przykład 2.** Należy utworzyć GUI zgodnie z poniższym rysunkiem oraz zaimplantować następujące zdarzenia:

- Przycisk Close zamyka okno.
- Display the date – wyświetla aktualną date w JLabel.



## JTextField

`public class` JTextField `extends` JTextComponent `implements` SwingConstants

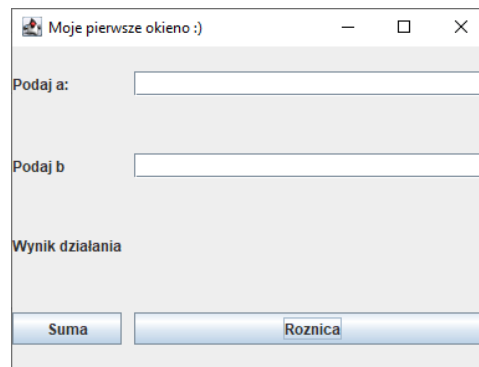
### Commonly used Constructors:

Constructor	Description
<b>JTextField()</b>	Creates a new TextField
<b>JTextField(String text)</b>	Creates a new TextField initialized with the specified text.
<b>JTextField(String text, int columns)</b>	Creates a new TextField initialized with the specified text and columns.
<b>JTextField(int columns)</b>	Creates a new empty TextField with the specified number of columns.

### Commonly used Methods:

Methods	Description
<b>void addActionListener(ActionListener l)</b>	It is used to add the specified action listener to receive action events from this textfield.
<b>Action getAction()</b>	It returns the currently set Action for this ActionEvent source, or null if no Action is set.
<b>void setFont(Font f)</b>	It is used to set the current font.
<b>void removeActionListener(ActionListener l)</b>	It is used to remove the specified action listener so that it no longer receives action events from this textfield.

**Przykład 3.** Po podaniu a i b i przyciśnięci przycisku suma lub różnica pojawi się wynik działania, close Zamyka okno.



The screenshot shows a Java Swing window titled "Moje pierwsze okno :)" with standard window controls (minimize, maximize, close). The window contains a light gray background. It has two input fields: the first is labeled "Podaj a:" and the second is labeled "Podaj b:". Below these is a label "Wynik działania". At the bottom, there are two buttons: "Suma" and "Roznica".

**Zadanie 1.** Należy zaproponować aplikację GUI do konwersji temperatury Celsjusza na Farenheit. Aplikacja powinna posiadać następujące funkcjonalności:

- Wpisywanie temp w stopniach Celsjusza.
- Pole w którym po przyciśnięciu przycisku konwertuj pojawi się wynik.

**Zadanie 2.** Wykorzystując poznane komponenty należy zaprojektować aplikacje/klika, która/e zaprezentuje/ą użycie dostępnych metod dla omawianych komponentów.