

## Wykład VI

### Java - środowisko GUI Stosowanie kontroltek biblioteki AWT

Biblioteka AWT (Abstract Window Toolkit) zawiera szereg dodatkowych narzędzi do tworzenia GUI są to standardowe kontrolki oraz menedżer układów graficznych. Elementem niezbędnym przy tworzeniu interfejsu użytkownika jest poznanie obsługi zdarzeń związanych z kontrolkami.

**Kontrolki** są komponentami, które umożliwiają użytkownikowi interakcję z aplikacją na wiele różnych sposobów — jedną z najczęściej stosowanych kontroltek jest przycisk, pole tekstowe, lista, pola przycisków wyboru i opcji.

**Menedżer układu graficznego** automatycznie pozycjonuje komponenty w ramach kontenera. Oznacza to, że wygląd okna zależy od kombinacji umieszczonych na nim kontroltek oraz użytego menedżera układu, który odpowiada za ich właściwe pozycjonowanie.

Poznamy następujące typy kontroltek:

etykiety,

- przyciski,
- pola wyboru,
- listy rozwijane,
- listy,
- paski przewijania,
- pola tekstowe.

Wszystkie wymienione powyżej kontrolki są podklasami klasy `Component`. Choć powyższy zbiór kontroltek nie jest szczególnie bogaty, to jednak z powodzeniem wystarcza do tworzenia prostych aplikacji. Znacznie bogatszy zestaw kontroltek jest dostępny w pakietach `Swing`, jak i `JavaFX`.

#### Dodawanie i usuwanie kontroltek.

Dodanie kontrolki do istniejącego okna wymaga następujących czynności:

- utworzyć egzemplarz docelowej kontrolki
- dodać ją do okna za pomocą zdefiniowanej przez klasę `Container` metody `add()`.

Metoda `add()` występuje w wielu wersjach. W pierwszej części tego rozdziału będziemy wykorzystywać następującą jej postać:

`Component add(Component compRef)`

Parametr `compRef` jest w tym przypadku egzemplarzem kontrolki, którą chcemy dodać do okna.

Metoda zwraca referencję do tego obiektu. Kiedy już kontrolka zostanie dodana do okna, automatycznie będzie widoczna w momencie wyświetlenia okna macierzystego. Jej położenie będzie zależne od przyjętego

## Usuwanie kontrolerek

W niektórych sytuacjach konieczne jest usunięcie kontrolki z okna, kiedy uznamy, że nie jest ona potrzebna. W tym celu wystarczy wywołać zdefiniowaną przez klasę Container metodę `remove()`. Ogólna postać metody `remove()` jest następująca:

```
void remove(Component compRef)
```

Tym razem parametr *compRef* jest kontrolką, którą chcemy usunąć. Wywołując metodę `removeAll()`, możemy w jednym kroku usunąć wszystkie kontrolki.

## Etykiety

Prezentacja tekstu lub opisu innych obiektów jest realizowana przez obiekt typu etykieta. **Etykieta** jest obiektem typu `Label` i zawiera ciąg znaków wyświetlany w oknie apletu lub aplikacji. Etykiety są **kontrolkami pasywnymi**, które nie obsługują żadnych interakcji z użytkownikiem.

Klasa `Label` definiuje następujące konstruktory:

```
Label() throws HeadlessException
```

```
Label(String napis) throws HeadlessException
```

```
Label(String napis, int wyrownanie) throws HeadlessException
```

Pierwsza wersja konstruktora tworzy pustą etykietę. Druga wersja tworzy etykietę zawierającą ciąg znaków przekazany za pośrednictwem parametru *napis*. Wyświetlany tekst jest wyrównywany do lewej krawędzi kontrolki etykiety. Trzecia wersja konstruktora tworzy etykietę zawierającą ciąg znaków przekazany za pośrednictwem parametru *napis*, który jest wyrównywany zgodnie z wartością przekazaną w argumencie *wyrownanie*. Wartość parametru *wyrownanie* musi być równa jednej z następujących stałych: `Label.LEFT`, `Label.RIGHT` lub `Label.CENTER`.

Wyłączenie menedżera rozłożenia elementów w oknie programu

Przykład kodu źródłowego z zastosowaniem obiektu etykiety

```
import java.awt.*;
import java.awt.event.*;
//Demonstruje użycie etykiet w aplikacji okienkowej

public class Okno_kontrolki extends Frame implements ActionListener,
WindowListener
{
    //Definiuje i inicjalizuje etykiety
    Label pierwsza = new Label("Jeden");
    Label druga = new Label("Dwa", Label.RIGHT);
    Label trzecia = new Label("Trzy", Label.CENTER);

    public Okno_kontrolki ()
    {
        //włączony nasłuchiwanie dla interfejsu okna
        this.addWindowListener(this);
        //dodaje etykiety do okna
        add(pierwsza);
        add(druga);
        add(trzecia);
        //Wyłączenie menedżera rozłożenia elementów w oknie programu
        this.setLayout(null);
        pierwsza.setBounds(100,20,80,40);
        druga.setBounds(100,60,80,40);
        trzecia.setBounds(100,100,80,40);
    }

    public static void main(String[] args) {
        Okno_kontrolki okienko = new Okno_kontrolki();
        okienko.setSize(new Dimension(600, 800));
        //okienko.setSize(300, 200);
        okienko.setTitle("Aplikacja wykorzystująca pakiet AWT");
        okienko.setVisible(true);
    }
}
```

```

    }
    @Override
    public void windowOpened(WindowEvent e) {
    }
    @Override
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
    @Override
    public void windowClosed(WindowEvent e) {
    }
    @Override
    public void windowIconified(WindowEvent e) {
    }
    @Override
    public void windowDeiconified(WindowEvent e) {
    }
    @Override
    public void windowActivated(WindowEvent e) {
    }
    @Override
    public void windowDeactivated(WindowEvent e) {
    }
    @Override
    public void actionPerformed(ActionEvent e) {
    }
}

```

### Pole tekstowe kontrolka typu TextField

Klasa `TextField` implementuje jednowierszowe pole tekstowe nazywane często **kontrolką edycji**. Pola tekstowe umożliwiają użytkownikom wpisywanie tekstu i jego edycję za pomocą klawiszy strzałek, kombinacji klawiszy odpowiadających operacjom wycinania i wklejania oraz z wykorzystaniem techniki zaznaczania tekstu kursorem myszy. `TextField` jest podklasą klasy `TextComponent` i definiuje następujące konstruktory:

`TextField()` throws `HeadlessException`

`TextField(int Liczba_znakow)` throws `HeadlessException`

`TextField(String napis)` throws `HeadlessException`

`TextField(String napis, int Liczba_znakow)` throws `HeadlessException`

Pierwsza wersja konstruktora tworzy domyślne pole tekstowe. Druga wersja tworzy pole tekstowe, którego szerokość jest równa liczbie *Liczba\_znakow*. Trzecia wersja konstruktora inicjalizuje nowe pole tekstowe tekstem reprezentowanym przez parametr *napis*. Wersja czwarta jednocześnie inicjalizuje i ustawia szerokość tworzonego pola tekstowego.

Pobranie łańcucha znaków z pola tekstowego wymaga zastosowania metody `getText()`.

Wprowadzenie łańcucha znaków do pola tekstowego wymaga zastosowania metody `setText()`. Składnia wymienionych metod przedstawiono poniżej:

`String getText()`

`void setText(String str)`

Parametr *str* reprezentuje w tym przypadku nowy tekst umieszczany w polu tekstowym. Użytkownik może zaznaczyć wybrany przez siebie fragment tekstu wyświetlanego i edytowanego w polu tekstowym. Można to zrobić także z poziomu kodu programu — wystarczy wywołać metodę `select()`. Nasz program może uzyskać aktualnie zaznaczony fragment tekstu za pomocą metody `getSelectedText()`.

Składnia metod jest następująca:

String getSelectedText()

void select(int *startIndex*, int *endIndex*)

Metoda getSelectedText() zwraca zaznaczoną część tekstu, natomiast metoda select() powoduje zaznaczenie znaków w polu tekstowym od indeksu *startIndex* do indeksu *endIndex*-1.

Przykład kodu źródłowego z etykietami i polami tekstowymi

```
import java.awt.*;
import java.awt.event.*;
//Demonstruje użycie etykiet w aplikacji okienkowej

public class Okno_kontrolki extends Frame implements ActionListener,
WindowListener
{
    //Definiuje i inicjalizuje etykiety
    Label pierwsza = new Label("Jeden");
    Label druga = new Label("Dwa", Label.RIGHT);
    Label trzecia = new Label("Trzy", Label.CENTER);
    //Definiuje i inicjalizuje pola tekstowe
    TextField pierwsze_pole = new TextField("Tekst domyślny" );
    TextField drugie_pole = new TextField(80);
    TextField trzecie_pole = new TextField ("Tekst domyślny z liczbą znaków",80);

    public Okno_kontrolki ()
    {
        // włączone nasłuchiwanie dla interfejsu okna
        this.addWindowListener(this);
        // dodaje etykiety do okna
        add (pierwsza);
        add(druga );
        add(trzecia );
        // dodaje pola tekstowe do okna
        add (pierwsze_pole);
        add(drugie_pole);
        add(trzecie_pole );
        //Wyłączenie menedżera rozłożenia elementów w oknie programu
        this.setLayout(null);
        pierwsza.setBounds(100,40,80,20);
        druga.setBounds(100,80,80,20);
        trzecia.setBounds(100,120,80,20);
        pierwsze_pole.setBounds(200,40,180,20);
        drugie_pole.setBounds(200, 80, 180, 20);
        trzecie_pole.setBounds(200,120, 180, 20);
    }

    public static void main(String[] args) {
        Okno_kontrolki okienko = new Okno_kontrolki();
        okienko.setSize(new Dimension(600, 800));
        //okienko.setSize(300, 200);
        okienko.setTitle("Aplikacja wykorzystująca pakiet AWT");
        okienko.setVisible(true);
    }

    @Override
    public void windowOpened(WindowEvent e) {
    }

    @Override
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }

    @Override
    public void windowClosed(WindowEvent e) {
    }
}
```

```

@Override
public void windowIconified(WindowEvent e) {

}
@Override
public void windowDeiconified(WindowEvent e) {

}
@Override
public void windowActivated(WindowEvent e) {

}
@Override
public void windowDeactivated(WindowEvent e) {

}
@Override
public void actionPerformed(ActionEvent e) {

}
}

```

### Przyciski kontrolka typu Button

Jedną z najczęściej wykorzystywanych kontrolki jest przycisk. **Przycisk** jest komponentem zawierającym etykietę i generującym zdarzenie w momencie naciśnięcia przez użytkownika. Wszystkie przyciski są obiektami klasy Button. Klasa ta definiuje następującą parę konstruktorów:

Button() throws HeadlessException

Button(String *napis*) throws HeadlessException

Pierwsza wersja konstruktora tworzy pusty przycisk. Druga wersja tworzy przycisk z etykietą przekazaną za pośrednictwem parametru *napis*.

Po utworzeniu przycisku możemy w dowolnym momencie ustawić lub zmienić jego etykietę za pomocą metody setLabel(). Metoda getLabel() służy do odczytywania bieżącej etykiety przycisku.

Składnia obu metod przedstawiono poniżej:

void setLabel(String *napis*)

String getLabel()

Parametr *napis* metody setLabel() zawiera nową etykietę modyfikowanego przycisku.

### Obsługa zdarzeń przycisków

Za każdym razem, gdy użytkownik naciska przycisk, generowane jest zdarzenie typu ActionEvent. Obiekt zdarzenia jest przekazywany wszystkim obiektom nasłuchującym, które zarejestrowały swoje zainteresowanie sygnałami o tego typu zdarzeniach generowanych przez dany komponent. Każdy z tych obiektów nasłuchujących musi implementować interfejs ActionListener. Interfejs ten definiuje metodę actionPerformed(), która jest wywoływana w odpowiedzi na wystąpienie zdarzenia typu ActionEvent. Reprezentujący to zdarzenie obiekt klasy ActionEvent jest przekazywany do metody actionPerformed() w formie jej argumentu. Obiekt ten zawiera zarówno referencję do przycisku, który wygenerował dane zdarzenie, jak i referencję do tzw. **łańcucha polecenia**. Łańcuch polecenia domyślnie zawiera etykietę przycisku. Do identyfikacji przycisku można użyć zarówno referencji do jego obiektu, jak i łańcucha polecenia.

Przykład kodu źródłowego:

```

import java.awt.*;
import java.awt.event.*;
//Demonstruje użycie etykiet w aplikacji okienkowej

public class Okno_kontrolki extends Frame implements ActionListener,

```



```

WindowListener
{
    //Definiuje i inicjalizuje etykiety
    Label pierwsza = new Label("Jeden");
    Label druga = new Label("Dwa", Label.RIGHT);
    Label trzecia = new Label("Trzy", Label.CENTER);
    //Definiuje i inicjalizuje pola tekstowe
    TextField pierwsze_pole = new TextField("Tekst domyślny" );
    TextField drugie_pole = new TextField(80);
    TextField trzecie_pole = new TextField ("Tekst domyślny z liczbą znaków",80);
    //Definiuje i inicjalizuje przyciski
    Button oblicz = new Button ("Oblicz");
    Button koniec = new Button ("Koniec");

    public Okno_kontrolki ()
    {
        // włącza nasłuchiwanie dla interfejsu okna
        this.addWindowListener(this);
        // dodaje etykiety do okna
        add (pierwsza);
        add(druga );
        add(trzecia );
        // dodaje pola tekstowe do okna
        add (pierwsze_pole);
        add(drugie_pole);
        add(trzecie_pole );
        // dodaje przyciski do okna
        add (oblicz);
        add(koniec);
        //Wyłączenie menedżera rozłożenia elementów w oknie programu
        this.setLayout(null);
        // położenie etykiet w oknie
        pierwsza.setBounds(100,40,100,20);
        druga.setBounds(100,80,100,20);
        trzecia.setBounds(100,120,100,20);
        // położenie pola tekstowego do okna
        pierwsze_pole.setBounds(200,40,180,20);
        drugie_pole.setBounds(200, 80, 180, 20);
        trzecie_pole.setBounds(200,120, 180, 20);
        // położenie przycisków w oknie
        oblicz.setBounds(200,200,100,60);
        koniec.setBounds(400, 200, 100, 60);

        oblicz.addActionListener(this);
        koniec.addActionListener(this);
    }

    public static void main(String[] args) {
        Okno_kontrolki okienko = new Okno_kontrolki();
        okienko.setSize(new Dimension(600, 800));
        //okienko.setSize(300, 200);
        okienko.setTitle("Aplikacja wykorzystująca pakiet AWT");
        okienko.setVisible(true);
    }
    @Override
    public void windowOpened(WindowEvent e) {
    }
    @Override
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
    @Override
    public void windowClosed(WindowEvent e) {
    }
    @Override

```



```

public void windowIconified(WindowEvent e) {
}
@Override
public void windowDeiconified(WindowEvent e) {
}
@Override
public void windowActivated(WindowEvent e) {
}
@Override
public void windowDeactivated(WindowEvent e) {
}
@Override
public void actionPerformed(ActionEvent e) {
    String str = e.getActionCommand();

    if((Button)e.getSource() == oblicz )
    { //Zmiana tekstu w polu tekstowym za pomocą metody set
      pierwsze_pole.setText("Naciśnięty przycisk Oblicz");
    }
    if(str.equals("Koniec"))
    { //Zmiana tekstu w etykiecie za pomocą metody set
      pierwsza.setText("Kliknięto Koniec");
    }
}
}

```

## Stosowanie pól wyboru

**Pole wyboru** jest kontrolką umożliwiającą zaznaczenie pola ma dwa stany włączony i wyłączony. Każde pole wyboru jest powiązane z etykietą opisującą reprezentowaną przez to pole opcję. Zmiana stanu pola wyboru wymaga wskazania kursorem myszy i kliknięcia. Pola wyboru mogą być stosowane samodzielnie lub występować w odpowiednio powiązanych grupach. Każde pole wyboru jest obiektem klasy `Checkbox`. Klasa `Checkbox` udostępnia następujące pięć konstruktorów:

`Checkbox()` throws `HeadlessException`

`Checkbox(String opis)` throws `HeadlessException`

`Checkbox(String opis, boolean on)` throws `HeadlessException`

`Checkbox(String opis, boolean on, CheckboxGroup grupa)` throws `HeadlessException`

`Checkbox(String opis, CheckboxGroup grupa, boolean on)` throws `HeadlessException`

Pierwsza wersja konstruktora tworzy pole wyboru, którego etykieta jest początkowo pusta. Nowo utworzone pole wyboru nie jest zaznaczone. Druga wersja tworzy pole wyboru z etykietą reprezentowaną przez parametr *opis*. Także w tym przypadku nowe pole wyboru nie będzie zaznaczone. Trzecia postać umożliwia ustawienie początkowego stanu pola wyboru. Jeśli w argumencie *on* prześlemy wartość `true`, nowo utworzone pole wyboru będzie początkowo zaznaczone; w przeciwnym przypadku nowe pole nie będzie zawierało znaku zaznaczenia. Czwarta i piąta wersja konstruktora tworzy pole wyboru, którego etykieta jest reprezentowana przez parametr *opis* i którego grupę określa wartość parametru *grupa*. Jeśli nowe pole wyboru nie ma być częścią żadnej grupy, parametr *cbGroup* powinien mieć wartość `null`.

Aby uzyskać bieżący stan pola wyboru, należy wywołać metodę `getState()`. Jeśli natomiast chcemy ustawić (zmienić) stan interesującego nas pola wyboru, powinniśmy wywołać metodę `setState()`.

Bieżącą etykietę powiązaną z polem wyboru możemy odczytać, wywołując metodę `getLabel()`. Aby tę etykietę zmienić, powinniśmy użyć metody `setLabel()`.

```

import java.awt.*;
import java.awt.event.*;
//Demonstruje użycie etykiet w aplikacji okienkowej

public class Okno_kontrolki extends Frame implements ActionListener,
WindowListener
{
    //Definiuje i inicjalizuje etykiety
    Label pierwsza = new Label("Jeden");
    Label druga = new Label("Dwa", Label.RIGHT);
    Label trzecia = new Label("Trzy", Label.CENTER);
    //Definiuje i inicjalizuje pola tekstowe
    TextField pierwsze_pole = new TextField("Tekst domyślny" );
    TextField drugie_pole = new TextField(80);
    TextField trzecie_pole = new TextField ("Tekst domyślny z
liczbą znaków",80);
    //Definiuje i inicjalizuje przyciski
    Button oblicz = new Button ("Oblicz");
    Button koniec = new Button ("Koniec");
    //Definiuje i inicjalizuje przyciski wyboru
    Checkbox windows = new Checkbox("Windows", null, true);
    Checkbox android = new Checkbox("Andorid");
    Checkbox linux = new Checkbox("Linux");
    Checkbox mac = new Checkbox("MacOS");

    public Okno_kontrolki ()
    {
        // włączony nasłuchiwanie dla interfejsu okna
        this.addWindowListener(this);
        // dodaje etykiety do okna
        add (pierwsza);
        add(druga );
        add(trzecia );
        // dodaje pola tekstowe do okna
        add (pierwsze_pole);
        add(drugie_pole);
        add(trzecie_pole );
        // dodaje przyciski do okna
        add (oblicz);
        add(koniec);
        // dodaje przyciski wyboru do okna
        add(windows);
        add(android);
        add(linux);
        add(mac);
        //Wyłączenie menedżera rozłożenia elementów w oknie programu
        this.setLayout(null);
        // położenie etykiet w oknie
        pierwsza.setBounds(100,40,100,20);
        druga.setBounds(100,80,100,20);
        trzecia.setBounds(100,120,100,20);
        // położenie pola tekstowego do okna
        pierwsze_pole.setBounds(200,40,180,20);
        drugie_pole.setBounds(200, 80, 180, 20);
        trzecie_pole.setBounds(200,120, 180, 20);
        // położenie przycisków w oknie
        oblicz.setBounds(200,200,100,60);
        koniec.setBounds(400, 200, 100, 60);
        // położenie przycisków wyboru w oknie
        windows.setBounds(200,300,150,40);
        android.setBounds(200, 340, 150, 40);
        linux.setBounds(200,380,150,40);
        mac.setBounds(200, 420, 150, 40);
    }
}

```

+



```

        oblicz.addActionListener(this);
        koniec.addActionListener(this);
    }
    public static void main(String[] args) {
        Okno_kontrolki okienko = new Okno_kontrolki();
        okienko.setSize(new Dimension(600, 800));
        //okienko.setSize(300, 200);
        okienko.setTitle("Aplikacja wykorzystująca pakiet AWT");
        okienko.setVisible(true);
    }
    @Override
    public void windowOpened(WindowEvent e) {
    }
    @Override
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
    @Override
    public void windowClosed(WindowEvent e) {
    }
    @Override
    public void windowIconified(WindowEvent e) {
    }
    @Override
    public void windowDeiconified(WindowEvent e) {
    }
    @Override
    public void windowActivated(WindowEvent e) {
    }
    @Override
    public void windowDeactivated(WindowEvent e) {
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        String str = e.getActionCommand();

        if((Button)e.getSource() == oblicz )
        { //Zmiana tekstu w polu tekstowym za pomocą metody set
          pierwsze_pole.setText("Naciśnięty przycisk Oblicz");
          if (windows.getState()) windows.setLabel("Windows zaznaczone");
          if (android.getState()) android.setLabel("Android zaznaczone");
          if (linux.getState()) linux.setLabel("Linux zaznaczone");
          if (mac.getState()) mac.setLabel("MacOS zaznaczone");
        }
        if(str.equals("Koniec"))
        { //Zmiana tekstu w etykiecie za pomocą metody set
          pierwsza.setText("Kliknięto Koniec");
        }
    }
}

```



## Obsługa zdarzeń pól wyboru

Zdarzenia związane z obsługą przycisków wyboru są powiązane ze zdarzeniem typu `ItemEvent`. Obiekt tego zdarzenia jest przekazywany wszystkim obiektom nasłuchującym, które zarejestrowały swoje zainteresowanie sygnałami o tego typu zdarzeniach generowanych przez dany komponent. Każdy z tych obiektów nasłuchujących musi implementować interfejs `ItemListener`. Interfejs ten definiuje metodę `itemStateChanged()`, która otrzymuje reprezentujący zdarzenie obiekt klasy `ItemEvent` w formie swojego argumentu. Przekazywany w ten sposób obiekt zawiera niezbędne informacje o zdarzeniu (np. o tym, czy zdarzenie wiąże się z zaznaczeniem, czy usunięciem zaznaczenia danego pola wyboru).

Przykładowy kod źródłowy:

```
import java.awt.*;
import java.awt.event.*;
//Demonstruje użycie etykiet w aplikacji okienkowej

public class Okno_kontrolki extends Frame implements ActionListener,
WindowListener, ItemListener
{
    //Definiuje i inicjalizuje etykiety
    Label pierwsza = new Label("Jeden");
    Label druga = new Label("Dwa", Label.RIGHT);
    Label trzecia = new Label("Trzy", Label.CENTER);
    //Definiuje i inicjalizuje pola tekstowe
    TextField pierwsze_pole = new TextField("Tekst domyślny" );
    TextField drugie_pole = new TextField(80);
    TextField trzecie_pole = new TextField ("Tekst domyślny z liczbą znaków",80);
    //Definiuje i inicjalizuje przyciski
    Button oblicz = new Button ("Oblicz");
    Button koniec = new Button ("Koniec");
    //Definiuje i inicjalizuje przyciski wyboru
    Checkbox windows = new Checkbox("Windows");
    Checkbox android = new Checkbox("Andorid");
    Checkbox linux = new Checkbox("Linux");
    Checkbox mac = new Checkbox("MacOS");

    public Okno_kontrolki ()
    {
        // włączone nasłuchiwanie dla interfejsu okna
        this.addWindowListener(this);
        // dodaje etykiety do okna
        add(pierwsza);
        add(druga );
        add(trzecia );
        // dodaje pola tekstowe do okna
        add(pierwsze_pole);
        add(drugie_pole);
        add(trzecie_pole );
        // dodaje przyciski do okna
        add(oblicz);
        add(koniec);
        // dodaje przyciski wyboru do okna
        add(windows);
        add(android);
        add(linux);
        add(mac);
        //Wyłączenie menedżera rozłożenia elementów w oknie programu
        this.setLayout(null);
        // położenie etykiet w oknie
        pierwsza.setBounds(100,40,100,20);
        druga.setBounds(100,80,100,20);
        trzecia.setBounds(100,120,100,20);
        // położenie pola tekstowego do okna
```

```

    pierwsze_pole.setBounds(200,40,180,20);
    drugie_pole.setBounds(200, 80, 180, 20);
    trzecie_pole.setBounds(200,120, 180, 20);
    // położenie przycisków w oknie
    oblicz.setBounds(200,200,100,60);
    koniec.setBounds(400, 200, 100, 60);
    // położenie przycisków wyboru w oknie
    windows.setBounds(200,300,150,40);
    android.setBounds(200, 340, 150, 40);
    linux.setBounds(200,380,150,40);
    mac.setBounds(200, 420, 150, 40);
    // włączone nasłuchiwanie dla dodanych przycisków
    oblicz.addActionListener(this);
    koniec.addActionListener(this);
    // włączone nasłuchiwanie dla dodanych przycisków wyboru
    windows.addItemListener(this);
    android.addItemListener(this);
    linux.addItemListener(this);
    mac.addItemListener(this);
}

public static void main(String[] args) {
    Okno_kontrolki okienko = new Okno_kontrolki();
    okienko.setSize(new Dimension(600, 800));
    //okienko.setSize(300, 200);
    okienko.setTitle("Aplikacja wykorzystująca pakiet AWT");
    okienko.setVisible(true);
}

@Override
public void windowOpened(WindowEvent e) {
}

@Override
public void windowClosing(WindowEvent e) {
    System.exit(0);
}

@Override
public void windowClosed(WindowEvent e) {
}

@Override
public void windowIconified(WindowEvent e) {
}

@Override
public void windowDeiconified(WindowEvent e) {
}

@Override
public void windowActivated(WindowEvent e) {
}

@Override
public void windowDeactivated(WindowEvent e) {
}

@Override
public void actionPerformed(ActionEvent e) {
    String str = e.getActionCommand();
    if((Button)e.getSource() == oblicz )
    { //Zmiana tekstu w polu tekstowym za pomocą metody set
        pierwsze_pole.setText("Naciśnięty przycisk Oblicz");
    }
    if(str.equals("Koniec"))
    { //Zmiana tekstu w etykiecie za pomocą metody set
        pierwsza.setText("Kliknięto Koniec");
    }
}

@Override

```

```

public void itemStateChanged(ItemEvent e) {
    if (windows.getState()) windows.setLabel("Windows zaznaczone");
    else windows.setLabel("Windows");
    if (android.getState()) android.setLabel("Android zaznaczone");
    else android.setLabel("Android ");
    if (linux.getState()) linux.setLabel("Linux zaznaczone");
    else linux.setLabel("Linux");
    if (mac.getState()) mac.setLabel("MacOS zaznaczone");
    else mac.setLabel("MacOS");
}
}

```

Inną formą obsługi zdarzenia jest sprawdzanie który z obiektów Checkbox wygenerował zdarzenie. Należy dodać nasłuchiwanie dla elementów Checkbox oraz sprawdzać w metodzie itemStateChanged(ItemEvent e) jaki obiekt jest źródłem zdarzenia

Przykład kodu źródłowego

```

import java.awt.*;
import java.awt.event.*;
//Demonstruje użycie etykiet w aplikacji okienkowej

public class Okno_kontrolki extends Frame implements ActionListener,
WindowListener, ItemListener
{
    //Definiuje i inicjalizuje etykiety
    Label pierwsza = new Label("Jeden");
    Label druga = new Label("Dwa", Label.RIGHT);
    Label trzecia = new Label("Trzy", Label.CENTER);
    //Definiuje i inicjalizuje pola tekstowe
    TextField pierwsze_pole = new TextField("Tekst domyślny" );
    TextField drugie_pole = new TextField(80);
    TextField trzecie_pole = new TextField ("Tekst domyślny z liczbą znaków",80);
    //Definiuje i inicjalizuje przyciski
    Button oblicz = new Button ("Oblicz");
    Button koniec = new Button ("Koniec");
    //Definiuje i inicjalizuje przyciski wyboru
    Checkbox windows = new Checkbox("Windows");
    Checkbox android = new Checkbox("Andorid");
    Checkbox linux = new Checkbox("Linux");
    Checkbox mac = new Checkbox("MacOS");

    public Okno_kontrolki ()
    {
        //włączony nasłuchiwanie dla interfejsu okna
        this.addWindowListener(this);
        //dodaje etykiety do okna
        add(pierwsza);
        add(druga );
        add(trzecia );
        //dodaje pola tekstowe do okna
        add(pierwsze_pole);
        add(drugie_pole);
        add(trzecie_pole );
        //dodaje przyciski do okna
        add(oblicz);
        add(koniec);
        //dodaje przyciski wyboru do okna
        add(windows);
        add(android);
        add(linux);
        add(mac);
        //Wyłączenie menedżera rozłożenia elementów w oknie programu
        this.setLayout(null);
        //położenie etykiet w oknie

```

```

pierwsza.setBounds(100,40,100,20);
druga.setBounds(100,80,100,20);
trzecia.setBounds(100,120,100,20);
// położenie pola tekstowego do okna
pierwsze_pole.setBounds(200,40,180,20);
drugie_pole.setBounds(200, 80, 180, 20);
trzecie_pole.setBounds(200,120, 180, 20);
// położenie przycisków w oknie
oblicz.setBounds(200,200,100,60);
koniec.setBounds(400, 200, 100, 60);
// położenie przycisków wyboru w oknie
windows.setBounds(200,300,150,40);
android.setBounds(200, 340, 150, 40);
linux.setBounds(200,380,150,40);
mac.setBounds(200, 420, 150, 40);
// włączone nasłuchiwanie dla dodanych przycisków
oblicz.addActionListener(this);
koniec.addActionListener(this);
// włączone nasłuchiwanie dla dodanych przycisków wyboru
windows.addItemListener(this);
android.addItemListener(this);
linux.addItemListener(this);
mac.addItemListener(this);
}
public static void main(String[] args) {
    Okno_kontrolki okienko = new Okno_kontrolki();
    okienko.setSize(new Dimension(600, 800));
    //okienko.setSize(300, 200);
    okienko.setTitle("Aplikacja wykorzystująca pakiet AWT");
    okienko.setVisible(true);
}
@Override
public void windowOpened(WindowEvent e) {
}
@Override
public void windowClosing(WindowEvent e) {
    System.exit(0);
}
@Override
public void windowClosed(WindowEvent e) {
}
@Override
public void windowIconified(WindowEvent e) {
}
@Override
public void windowDeiconified(WindowEvent e) {
}
@Override
public void windowActivated(WindowEvent e) {
}
@Override
public void windowDeactivated(WindowEvent e) {
}
@Override
public void actionPerformed(ActionEvent e) {
    String str = e.getActionCommand();

    if((Button)e.getSource() == oblicz )
    { //Zmiana tekstu w polu tekstowym za pomoca metody set
      pierwsze_pole.setText("Naciśnięty przycisk Oblicz");
    }
}

```

```

    }
    if(str.equals("Koniec"))
    { //Zmiana tekstu w etykiecie za pomocą metody set
        pierwsza.setText("Kliknięto Koniec");
    }
}
@Override
public void itemStateChanged(ItemEvent e) {

    if (windows.getState()) windows.setLabel("Windows zaznaczone");
    else windows.setLabel("Windows");
    if (android.getState()) android.setLabel("Android zaznaczone");
    else android.setLabel("Android ");
    if (linux.getState()) linux.setLabel("Linux zaznaczone");
    else linux.setLabel("Linux");
    if (mac.getState()) mac.setLabel("MacOS zaznaczone");
    else mac.setLabel("MacOS");

    if ((Checkbox) e.getSource() == windows) pierwsze_pole.setText("Kliknięty przycisk Windows");
    if ((Checkbox) e.getSource() == android) pierwsze_pole.setText("Kliknięty przycisk Android");
    if ((Checkbox) e.getSource() == linux) pierwsze_pole.setText("Kliknięty przycisk Linux");
    if ((Checkbox) e.getSource() == mac) pierwsze_pole.setText("Kliknięty przycisk MacOS");
}

}

```

### Przyciski opcji grupowanie - CheckboxGroup

W pakiecie AWT można zmienić funkcjonalność elementów Checkbox poprzez ustawienie wzajemnie wykluczających się pól wyboru, w których jednocześnie może być zaznaczone jedno i tylko jedno pole wyboru z danej grupy. Pola wyboru należące do takiej grupy są często nazywane **przyciskami opcji**.

Utworzenie **przyciskami opcji** wymaga wcześniejszego zdefiniowania grupy, do której te pola będą należały, oraz wprost wskazać tę grupę podczas konstruowania odpowiednich pól wyboru. Grupy pól wyboru są obiektami klasy CheckboxGroup. Klasa ta definiuje tylko jeden konstruktor, który tworzy grupę pustą.

Wywołując metodę `getSelectedCheckbox()`, możemy w prosty sposób określić, które z pól wyboru danej grupy jest aktualnie zaznaczone. Mamy także możliwość ustawiania jednego z tych pól wyboru za pomocą metody `setSelectedCheckbox()`.

Przykład kodu źródłowego:

```

import java.awt.*;
import java.awt.event.*;
//Demonstruje użycie etykiet w aplikacji okienkowej

public class Okno_kontrolki extends Frame implements ActionListener,
WindowListener, ItemListener
{
    //Definiuje i inicjalizuje etykiety
    Label pierwsza = new Label("Jeden");
    Label druga = new Label("Dwa", Label.RIGHT);
    Label trzecia = new Label("Trzy", Label.CENTER);
    //Definiuje i inicjalizuje pola tekstowe
    TextField pierwsze_pole = new TextField("Tekst domyślny" );
    TextField drugie_pole = new TextField(80);
    TextField trzecie_pole = new TextField ("Tekst domyślny z liczbą znaków",80);
    //Definiuje i inicjalizuje przyciski
    Button oblicz = new Button ("Oblicz");
    Button koniec = new Button ("Koniec");
    //Definiuje i inicjalizuje grupę przycisków opcji
    CheckboxGroup grupa = new CheckboxGroup();
    //Definiuje i inicjalizuje przyciski wyboru
    Checkbox windows = new Checkbox("Windows", grupa, true);

```



```

        Checkbox android = new Checkbox("Andorid", grupa, false);
        Checkbox linux = new Checkbox("Linux", grupa, false);
        Checkbox mac = new Checkbox("MacOS", grupa, false);
public Okno_kontrolki ()
{
    // włączony nasłuchiwanie dla interfejsu okna
    this.addWindowListener(this);
    // dodaje etykiety do okna
    add (pierwsza);
    add(druga );
    add(trzecia );
    // dodaje pola tekstowe do okna
    add (pierwsze_pole);
    add(drugie_pole);
    add(trzecie_pole );
    // dodaje przyciski do okna
    add (oblicz);
    add(koniec);
    // dodaje przyciski wyboru do okna
    add(windows);
    add(android);
    add(linux);
    add(mac);
    //Wyłączenie menedżera rozłożenia elementów w oknie programu
    this.setLayout(null);
    // położenie etykiet w oknie
    pierwsza.setBounds(100,40,100,20);
    druga.setBounds(100,80,100,20);
    trzecia.setBounds(100,120,100,20);
    // położenie pola tekstowego do okna
    pierwsze_pole.setBounds(200,40,180,20);
    drugie_pole.setBounds(200, 80, 180, 20);
    trzecie_pole.setBounds(200,120, 180, 20);
    // położenie przycisków w oknie
    oblicz.setBounds(200,200,100,60);
    koniec.setBounds(400, 200, 100, 60);
    // położenie przycisków wyboru w oknie
    windows.setBounds(200,300,150,40);
    android.setBounds(200, 340, 150, 40);
    linux.setBounds(200,380,150,40);
    mac.setBounds(200, 420, 150, 40);
    // włączone nasłuchiwanie dla dodanych przycisków
    oblicz.addActionListener(this);
    koniec.addActionListener(this);
    // włączone nasłuchiwanie dla dodanych przycisków wyboru
    windows.addItemListener(this);
    android.addItemListener(this);
    linux.addItemListener(this);
    mac.addItemListener(this);
}
public static void main(String[] args) {
    Okno_kontrolki okienko = new Okno_kontrolki();
    okienko.setSize(new Dimension(600, 800));
    //okienko.setSize(300, 200);
    okienko.setTitle("Aplikacja wykorzystująca pakiet AWT");
    okienko.setVisible(true);
}
@Override
public void windowOpened(WindowEvent e) {
}
@Override
public void windowClosing(WindowEvent e) {
    System.exit(0);
}
}

```



```

@Override
public void windowClosed(WindowEvent e) {
}
@Override
public void windowIconified(WindowEvent e) {
}
@Override
public void windowDeiconified(WindowEvent e) {
}
@Override
public void windowActivated(WindowEvent e) {
}
@Override
public void windowDeactivated(WindowEvent e) {
}
@Override
public void actionPerformed(ActionEvent e) {
    String str = e.getActionCommand();

    if((Button)e.getSource() == oblicz )
    { //Zmiana tekstu w polu tekstowym za pomocą metody set
      pierwsze_pole.setText("Naciśnięty przycisk Oblicz");
    }
    if(str.equals("Koniec"))
    { //Zmiana tekstu w etykiecie za pomocą metody set
      pierwsza.setText("Kliknięto Koniec");
    }
}
@Override
public void itemStateChanged(ItemEvent e) {

//Podaje etykietę wybranego elementu Checkbox z grupy
drugie_pole.setText(grupa.getSelectedCheckbox().getLabel());

    if (windows.getState()) windows.setLabel("Windows zaznaczone");
    else windows.setLabel("Windows");
    if (android.getState()) android.setLabel("Android zaznaczone");
    else android.setLabel("Android ");
    if (linux.getState()) linux.setLabel("Linux zaznaczone");
    else linux.setLabel("Linux");
    if (mac.getState()) mac.setLabel("MacOS zaznaczone");
    else mac.setLabel("MacOS");
    if ((Checkbox) e.getSource() == windows) pierwsze_pole.setText("Kliknięty przycisk Windows");
    if ((Checkbox) e.getSource() == android) pierwsze_pole.setText("Kliknięty przycisk Android");
    if ((Checkbox) e.getSource() == linux) pierwsze_pole.setText("Kliknięty przycisk Linux");
    if ((Checkbox) e.getSource() == mac) pierwsze_pole.setText("Kliknięty przycisk MacOS");
}
}

```

