Wprowadzenie do tworzenia Graficznego Interfejsu Użytkownika (GUI)

Wstęp

1. Wymień charakterystyczne cechy graficznego sposobu prezentacji informacji oraz interakcji z użytkownikiem Jak zdefiniujesz GUI.
2. Czym jest Swing
3. Jaka jest różnica pomiędzy kontenerem, a komponentem (Container vs Component)
4. Wymień klasy reprezentujące kontenery najwyższego poziomu (Top-Level Containters) (JFrame, JDialog, JApplet)
5. Zapoznaj się ze sposobem użycia komponentów:
   http://docs.oracle.com/javase/tutorial/uiswing/components/index.html
6. Zapoznaj się z zasadami programowania przy użyciu biblioteki Swing:
   http://www3.ntu.edu.sg/home/ehchua/programming/java/j4a_gui.html
   http://docs.oracle.com/javase/tutorial/uiswing/

Wyznaczanie rozdzielczości ekranu

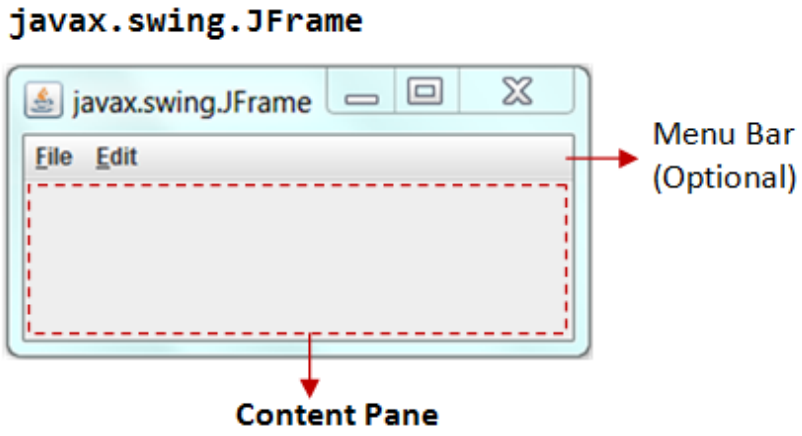
7. Wyznacz i wyświetl rozdzielczość ekranu urządzenia, na którym pracujesz. Wykorzystaj klasę Toolkit.

   ```
   Dimension d = Toolkit.getDefaultToolkit().getScreenSize();
   ```

   Odszukaj w klasie Dimension nazwy dwóch właściwości (pól) zawierających informacje o szerokości I wysokości ekranu w pikselach. Wyświetl te wartości na konsoli.

Definiowanie i wyświetlanie okna

8. Zapoznaj się z podstawowymi składowymi okna (ramki)
   http://www3.ntu.edu.sg/home/ehchua/programming/java/images/Swing_ContentPane.png



9. Obiekt klasy JFrame jest oknem z tytułem i ramką, do którego możemy dodać komponenty do wyświetlenia. Sekwencja tworzenia okna wraz z komponentami i jego wyświetlania przebiega następująco:

   // Utwórz ramkę (okno).

   JFrame okno = new JFrame("Demo");

   // Ustal działanie, gdy okno zostanie zamknięte

   okno.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

   // Utwórz komponenty i dodaj je do okna

   //...utwórz jakiś komponent

   okno.getContentPane().add(jakiskomponent, BorderLayout.CENTER);

   // Dopasuj rozmiar okna do wielkości komponentów lub ustal rozmiar w pikselach

   okno.pack() lub okno.setSize(x,y)

   // Wyświetl okno

   okno.setVisible(true);

10. Utwórz i wyświetl okno. Zwróć uwagę, iż została utworzona klasa dziedzicząca po JFrame.

```
import javax.swing.*;
public class Okno extends JFrame {
    public Okno(){
        setTitle("Moje okno");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400,300);
        setVisible(true);
    }
}
```

11. Odszukaj w klasie JFrame metodę blokującą zmianę rozmiarów okna. Czy taka metoda jest w tej klasie dostępna? Jeśli nie, sprawdź zawartość klasy, z której klasa JFrame dziedziczy. Dodaj metodę umożliwiającą zablokowanie i odblokowanie możliwości zmiany rozmiarów okna. Sprawdź działanie dodanej metody.

```
blokujZmianyRozmiaruOkna(boolean blokuj);
```

12. Dodaj konstruktor umożliwiający wyświetlenie okna w dowolnym położeniu określonym przez użytkownika. Wykorzystaj metodę setLocation. Czy znajduje się ona w klasie JFrame? Jeśli nie, odszukaj, w której klasie się ona znajduje i dowiedz się, jak ją zastosować.

```
Okno (int x, int y);
```

    Utwórz 5 okien na ekranie monitora tak, aby się wzajemnie nie zasłaniały.

13. Dodaj metodę przesuwającą okno na środek ekranu. Sprawdź działanie metody.

```
przesunNaSrodek();
```

14. Dodaj metodę przesuwającą okno do jednego, z czterech narożników ekranu (1..4). Wykorzystaj metodę setLocation.

```
zmienPolozenie(int x);
```

Kontenery i komponenty okna

15. Jaką funkcję pełni kontener, a jaką menadżer rozkładu?
16. Zapoznaj się z dostępnymi menadżerami rozkładu komponentów w kontenerze:

    https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html
17. BorderLayout stanowi domyślny menadżer rozkładu dla kontenera okna Content Pane (zobacz rysunek powyżej). Zapoznaj się z jego właściwościami. Zwróć uwagę na nazwy pięciu obszarów, w których mogą być umieszczane komponenty.

    https://docs.oracle.com/javase/tutorial/uiswing/layout/border.html

    Następnie pobierz i uruchom przykładowy program BorderLayoutDemo.java. Zwróć uwagę, jak zachowują się poszczególne elementy okna w momencie jego powiększania.
18. Poniższy przykład przedstawia, w jaki sposób można dodać komponenty do jednego z pięciu obszarów menadżera BorderLayout. Zwróć uwagę na sposób określenia obszaru, gdzie komponent jest dodawany.

```
JFrame okno = new JFrame("TopLevelDemo");
Container kontenerOkna = okno.getContentPane();
kontenerOkna.add(new JLabel("Tekst"), BorderLayout.PAGE_START);
```

19. Utwórz i wyświetl okno ze swoim imieniem i nazwiskiem w każdym z pięciu obszarów kontenera BorderLayout. Sprawdź, jak zachowują się poszczególne teksty, gdy zmieniasz rozmiar okna.
20. Jednym z powszechnie stosowanych kontenerów jest Panel (JPanel). W kontenerze tym dodawane komponenty rozmieszczane są przy wykorzystaniu menedżera rozkładu FlowLayout, tj. w kolejności ich dodawania. Zapoznaj się z jego właściwościami

    https://docs.oracle.com/javase/tutorial/uiswing/layout/flow.html

    Następnie pobierz przykładowy program FlowLayoutDemo.java i sprawdź, jak zachowują się komponenty przy zmianie rozmiarów okna.
21. Poniższa klasa zawiera definicję okna wraz komponentami. Dokonaj analizy kodu programu i odpowiedz, ile komponentów zostało umieszczonych w oknie. Następnie utwórz i wyświetl okno. Sprawdź, ile komponentów zostało wyświetlonych.

```
import javax.swing.*;
import java.awt.*;
public class OknoTestowe extends JFrame {
    public OknoTestowe() {
        setTitle("Model RGB przestrzeni barw");
```

```
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400, 200);
        JPanel kontener = new JPanel();
        kontener.add(new JButton("R-czerwony"));
        kontener.add(new JButton("G-zielony"));
        kontener.add(new JButton("B-niebieski"));
        add(kontener, BorderLayout.SOUTH);
        add(new JTextArea("Paleta barw..."), BorderLayout.CENTER);
        setVisible(true);
    }
}
```

Obsługa zdarzeń

22. Programowanie aplikacji wykorzystującej GUI związane jest nierozerwalnie z pojęciem obsługi zdarzeń. Generowane są one najczęściej przez mysz, klawiaturę, czy też poszczególne elementy interfejsu graficznego. W przypadku Javy zdarzeniem jest obiekt, który opisuje zmianę stanu swojego źródła spowodowaną np. przez naciśnięcie klawisza, kliknięcie myszką, wybór elementu z listy wyboru, zamknięcie okna. Źródłem jest obiekt, który wygenerował zdarzenie.
    Jednym z najczęściej stosowanych jest ActionListener, gdzie należy wykonać następujące operacje:
    a. Zdefiniowanie klasy implementującej interfejs ActionListener, której obiekt będzie nasłuchiwał zdarzenie
    ```
    public class Sluchacz implements ActionListener { ...
    ```
    b. Utworzenie obiektu słuchacza I przypisanie go do komponentu (np. przycisku, pola edycyjnego)
    ```
    jakisKomponent.addActionListener( new Sluchacz() );
    ```
    c. Zdefiniowanie metody actionPerformed w klasie Sluchacz, która zawierać będzie instrukcje wykonywane, gdy zdarzenie wystąpi
    ```
    public void actionPerformed(ActionEvent e) {
        //instrukcje, gdy zdarzenie wystąpi ...
    }
    ```
23. Poniższy program wyświetla komunikat na konsoli, gdy zostanie naciśnięty przycisk. Wskaż w kodzie programu operacje opisane w punkcie poprzednim. Następnie utwórz i uruchom program, aby sprawdzić jego działanie.
    ```
    class Handler implements ActionListener {
      public void actionPerformed(ActionEvent e) {
        System.out.println("Przycisk został naciśnięty!");
    ```

```
  }
}
public class GUI extends JFrame {
  public GUI() {
    JButton b = new JButton("Przycisk");
    Handler h = new Handler();
    b.addActionListener(h);
    add(b);
    setDefaultCloseOperation(DISPOSE_ON_CLOSE);
    pack();
    setVisible(true);
  }
}
```

24. Klasę definiującą obsługę zdarzeń można również umieścić wewnątrz klasy definiującej okno programu. Pozwala to na dostęp do wszystkich komponentów umieszczonych w tym oknie. Poniższy program wyświetla tekst wprowadzony przez użytkownika. Zwróć uwagę na to, w jaki sposób odczytana została zawartość pola tekstowego w metodzie obsługi zdarzeń. Następnie uruchom poniższy program i sprawdź jego działanie.

```
import java.awt.event.*; // konieczny dla obsługi zdarzeń
import javax.swing.*;
import java.awt.*;

public class InnerGUI extends JFrame {
    JTextArea t = new JTextArea("Wprowadź tekst...");
    JButton b = new JButton("Wyświetl tekst na konsoli");

    public InnerGUI() {
        Handler h = new Handler();
        b.addActionListener(h);
        add(t, BorderLayout.CENTER);
        add(b, BorderLayout.PAGE_END);
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setSize(300,200);
        setVisible(true);
```

```
    }

    private class Handler implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            System.out.println(t.getText());
        }
    }
}
```

25. Okna dialogowe stanowią mechanizm umożliwiający wyświetlanie (a także wprowadzanie) informacji poza zasadniczymi oknami aplikacji. Szczegóły dotyczące użycia okien dialogowych dostępne są pod adresem:

    https://docs.oracle.com/javase/tutorial/uiswing/components/dialog.html

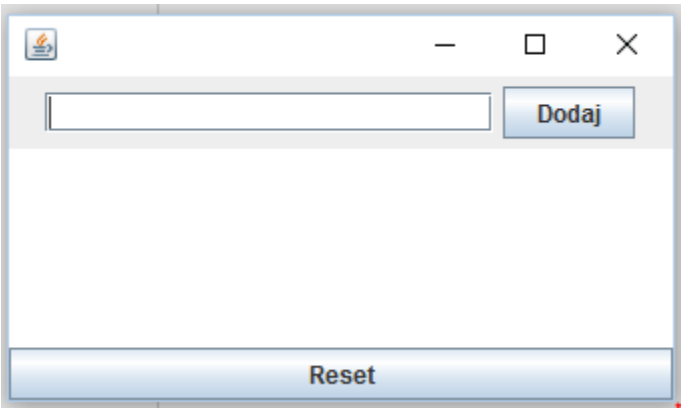
    W celu wyświetlenia tekstu w oknie dialogowym można posłużyć się instrukcją:

    ```
    JOptionPane.showMessageDialog(null, "tekst do wyświetlenia");
    ```

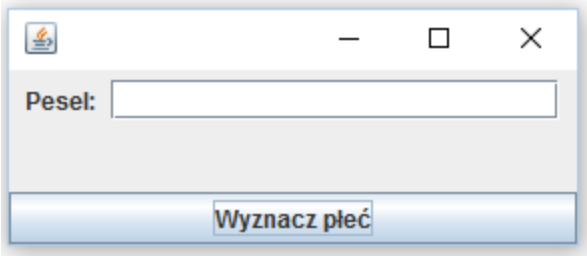
    Zmodyfikuj poprzedni program w taki sposób, aby po naciśnięciu przycisku tekst wprowadzony przez użytkownika został wyświetlony w oknie dialogowym zamiast na konsoli.

26. Zmodyfikuj poprzedni program w taki sposób, aby po naciśnięciu przycisku nastąpiło usunięcie tekstu wprowadzonego przez użytkownika. Zmień nazwę przycisku na Reset. Wykorzystaj metodę setText().

27. Zmodyfikuj poprzedni program. Dodaj pole edycyjne oraz przycisk Dodaj (obydwa komponenty umieść w kontenerze JPanel). Po naciśnięciu przycisku Dodaj program powinien dopisać wprowadzony w polu edycyjnym tekst w kolejnym wierszu okna. Dla obsługi tego przycisku utwórz kolejną klasę do obsługi zdarzenia i umieść ją wewnątrz klasy definiującej okno programu.



28. Napisz program, który na podstawie wprowadzonego numeru Pesel wyznaczy płeć. Wyznaczoną płeć wyświetl w oknie dialogowym. Szczegóły dotyczące wyznaczania płci znajdziesz pod adresem: https://pl.wikipedia.org/wiki/PESEL



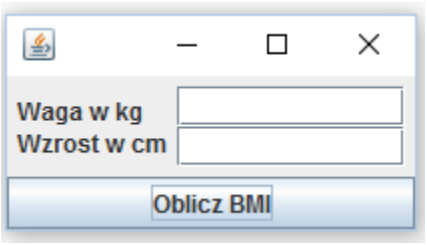
29. Napisz program do obliczania wskaźnika BMI. Szczegóły dotyczące obliczania tego wskaźnika znajdziesz pod adresem: https://pl.wikipedia.org/wiki/Wska%C5%BAnik_masy_cia%C5%82a

    W celu rozmieszczenia pól edycyjnych w oknie programu zastosuj menedżera rozkładu GridLayout. Szczegóły znajdziesz pod adresem: https://docs.oracle.com/javase/tutorial/uiswing/layout/grid.html

    Umieść etykiety pól w oddzielnym panelu, do którego zastosuj jednokolumnowy grid. Pola edycyjne umieść w oddzielnym panelu z jednokolumnowym gridem. Utworzone panele umieść w kolejnym panelu, a całość dodaj do okna. Poniżej przykładowe rozwiązanie:

```
JPanel etykiety = new JPanel(new GridLayout(0,1));
JPanel pola = new JPanel(new GridLayout(0,1));
JPanel dane = new JPanel();
// utwórz kolumnę z etykietami JLabel
etykiety.add(wagaTxt);
etykiety.add(wzrostTxt);
// utwórz kolumnę z polami JTextField
pola.add(waga);
pola.add(wzrost);
// utwórz panel z danymi i dodaj go do okna
dane.add(etykiety);
dane.add(pola);
okno.add(dane,BorderLayout.CENTER);
```

    Po naciśnięciu przycisku wyświetl obliczony wskaźnik BMI w oknie dialogowym. Jeśli wartość wskaźnika jest nieprawidłowa zastosuj ostrzegawczą ikonę.



30. Zmodyfikuj program do wyznaczania płaci na podstawie numeru Pesel tak, aby:
    a. Sprawdzana była liczba wprowadzonych znaków (dokładnie 11)

b. Sprawdzana była poprawność wprowadzonego numeru Pesel na podstawie cyfry kontrolnej