



Język JAVA

podstawy programowania

Jacek Rumiński

wykład 2, część 2

Plan wykładu:

1. Rodzaje programów w Javie
2. Tworzenie aplikacji
3. Tworzenie apletów
4. Obsługa archiwów
5. Wyjątki
6. Klasa w klasie !

Co to jest aplet (applet: application-let)?

Aplet jest formą aplikacji (komponentem) wywoływanej w ściśle określonym środowisku. Aplet nie jest wywoływany wprost (nie jest samodzielną aplikacją) przez kod klasy *.class lecz poprzez plik **HTML**, w kodzie którego zawarto odniesienie do kodu apletu *.class, np.:

```
<applet code="Jedi2.class" width="200" height="100">  
</applet>
```

Zapis ten oznacza, że w oknie o szerokości 200 (pikseli) i wysokości 100 będzie uruchomiony aplet o kodzie Jedi2.class.

Aplet jest programem graficznym – posiada bowiem graficzny obszar roboczy (o podanej szerokości i wysokości).

Własność apletu.

Ponieważ applet jest aplikacją graficzną polecenie:

```
System.out.println("Rycerz Luke ma niebieski miecz.");
```

nie spowoduje wyświetlenia tekstu w oknie apletu (czyli na stronie WWW), lecz wyświetli tekst w konsoli Javy (zwykle ukrytej). Uzyskanie napisu w polu graficznym będzie możliwe poprzez zastosowanie funkcji „rysującej” tekst.

Aplet uruchamiany jest w określonym kontenerze – przeglądarce WWW. Przeglądarka „kontroluje” cykl życia apletu.

W przypadku tradycyjnej aplikacji kontrola cyklu życia związana była z tylko jedną funkcją – main() – rozpoczęcie wykonywania instrukcji.

W przypadku apletu – sytuacja wygląda nieco inaczej.

Cykl życia apletu.

Tworząc aplet tworzymy klasę dziedziczącą z klasy *Applet*, wykorzystując podstawowe metody takie jak:

- *init()* – przeglądarka -> aplet załadowany;
- *start()* – przeglądarka -> aplet uruchom się;
- *paint()* – system graficzny -> aplet odrysuj się;
- *stop()* – przeglądarka -> aplet zatrzymaj wykonywanie działań;
- *destroy()* – przeglądarka -> aplet zwolnij zasoby - koniec.

Metody te są częścią standardowych klasy *Applet* (`java.applet.Applet`) oraz *JApplet* (`javax.swing.JApplet`, która dziedziczy po klasie `java.applet.Applet`).

Żeby utworzyć aplet **trzeba wykorzystać co najmniej jedną z trzech metod: *init()*, *start()*, *paint()*.**

Budowa apletu – podstawy dziedziczenia.

Zanim zrobimy przykłady przydatne będzie przedstawienie podstaw dotyczących dziedziczenia.

Jeśli klasa **A** dziedziczy po innej klasie **B** oznacza to, że przejmuje wszystkie jej cechy (pola) i zachowanie (metody), z wyjątkiem tych, które są specjalnie oznaczone (o tym później).

klasa A dziedziczy po B -> **JAVA:** `class A extends B{
}`

W procesie dziedziczenia można nadpisać (czyli na nowo zdefiniować działanie) dziedziczone metody (z wyjątkiem tych specjalnie oznaczonych).

Budowa apletu - dziedziczenie.

```
class Jedi{
    public String kolor_miecza;
    public String nazwa;
    public void ruchMiecza(){
        System.out.println("W LEWO!!!");
    }
}
// koniec class Jedi
class Luke extends Jedi{
    //nadpisanie metody - overriding
    public void ruchMiecza(){
        System.out.println("W PRAWO!!!");
    }
}
//koniec class Luke

(...)
Jedi rycerz = new Jedi(); // nowy obiekt klasy Jedi – wywołano domyślny konstruktor
rycerz.nazwa="Vader"; // przypisanie do pola „nazwa” wartości „Vader”
Luke rycerzLuke = new Luke(); //nowy obiekt klasy Luke– wywołano domyślny konstruktor
rycerzLuke.nazwa="Luke Skywalker"; //przypisanie wartości do odziedziczonego pola
```

Budowa apletu – podstawy dziedziczenia.

Dziedziczenie upraszcza rozwój aplikacji – przygotowujemy określoną klasę z potrzebną funkcjonalnością (metody), a później dziedziczymy po niej wprowadzając zmiany i uzupełnienia według potrzeb !

W klasie Applet (JApplet) wprowadzono szereg cech (pól) i funkcji, z których możemy (lub musimy) korzystać.

Podsumowując - stworzenie apletu to:

- 1.) utworzenie klasy, która dziedziczy po Applet (JApplet)*
- 2.) nadpisanie metod cyklu życia (co najmniej init() lub start() lub paint());*
- 3.) dopisanie własnego kodu i korzystanie z innych klas*

.

Kod programu: UkrytyJedi.java

```
import java.applet.Applet; //ponieważ używamy klasy Applet
import java.awt.Graphics; //ponieważ używamy klasy Graphics

public class UkrytyJedi extends Applet{

    public void paint(Graphics g){
        g.drawString("Rycerz Luke ma niebieski miecz.", 15,15);
    }
} // koniec public class UkrytyJedi.class extends Applet
```

Kod programu: UkrytyJedi.html

```
<html>

    <applet code="UkrytyJedi.class" width="400" height="300"> </applet>
```

Obiekt g klasy **Graphics** reprezentuje „system graficzny” danej platformy .

Uruchamianie apletu.

Na ogół aplet będzie uruchamiany poprzez wywołanie w przeglądarce WWW pliku HTML, który zawiera element (znacznik) `<applet>`.

W celach testowych (np. żeby sprawdzić czy aplet działa dobrze, przed uruchomieniem w przeglądarce, które czasem się różnią) można zastosować specjalny program z JDK o nazwie `appletviewer.exe`:

```
C:\> appletviewer UkrytyJedi.html
```

Standardowe ustawienia platformy Javy silnie ograniczają możliwości apletu. Aplet działa w „piaskownicy”, która wydziela jego możliwe działania (nie może otwierać lokalnych plików, łączyć się z komputerem innym niż ten, z którego pochodzi, itd.).

Można to zmienić – ale to temat na wykład w przyszłości...

Co dalej?

Nic nie stoi na przeszkodzie aby napisać taki kod w Javie, aby mógł być wykorzystany jako tradycyjna aplikacja lub aplet, w zależności od wywołania.

ZRÓB TO SAM

Zrób, proszę takie ćwiczenie tworząc w jednym kodzie aplikację i aplet. Dla ułatwienia dodam, że w kodzie trzeba użyć funkcję `main()` i jednocześnie dziedziczyć po klasie `Applet` nadpisując jedną z metod cyklu życia.

Plan wykładu:

1. Rodzaje programów w Javie
2. Tworzenie aplikacji
3. Tworzenie apletów
4. Obsługa archiwów (wykład 2, część 3)
5. Wyjątki
6. Klasa w klasie !