

# Swing Application Framework czyli tam i z powrotem

# Informacje o mnie

- **Imię:** Piotr
- **Nazwisko:** Kochański
- **Praca:** programista Java, PHP + kilka egzotycznych wynalazków
- **Miejsce pracy:** firma Erudis (prawdopodobnie najlepsza firma konsultingowa w okolicy :) )
- **Kontakt:** p.kochanski@erudis.pl
- **Więcej:** [www.xoft.pl](http://www.xoft.pl)

# Agenda

- Wstęp
- Co jest trudnego w tworzeniu GUI?
- Swing *brute force*
- Swing Application Framework (JSR-296)
  - co nam daje...
  - ...i czego tam nie ma
- Demonstracje

# Początki...

- 1995 - Java 1.0
- W zamierzeniach głównym przeznaczeniem Javy miały być... **aplety**, czyli aplikacje GUI uruchamiane z poziomu przeglądarki
- Plan nie do końca wypalił...
  - Tworzenie GUI w Javie nie należało do przyjemności
  - Konkurencja:
    - dawniej: DHTML, Flash
    - teraz: Flex, AJAX, Silverlight, JavaFX

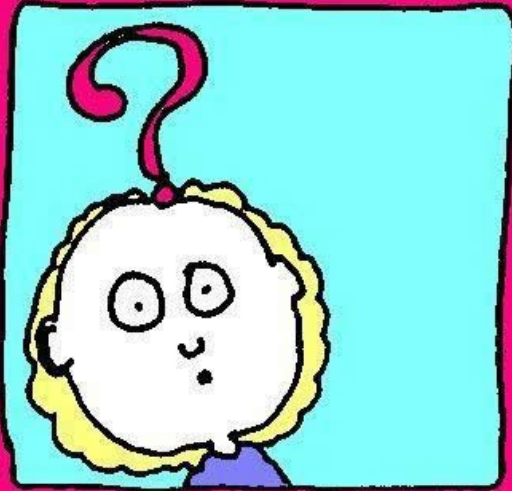


# teraz...

- SUN postanowił wrócić do źródeł
  - Matisse (NetBeans GUI Builder)
  - poprawa wydajności (Java 6.0)
  - Swing Application Framework (Java 7.0)



# Dlaczego tworzenie GUI jest trudne?



**Problem  
Kids?  
We've got  
answers**

- Przecież mamy narzędzia do wizualnego tworzenia GUI
  - Visual Studio
  - NetBeans, wtyczki do Eclipse'a, itp.
- Co więc sprawia problemy?
  - cykl życia aplikacji
  - wielowątkowość
  - zasoby/i18n
  - obsługa zdarzeń



# SAF = nie odstraszać



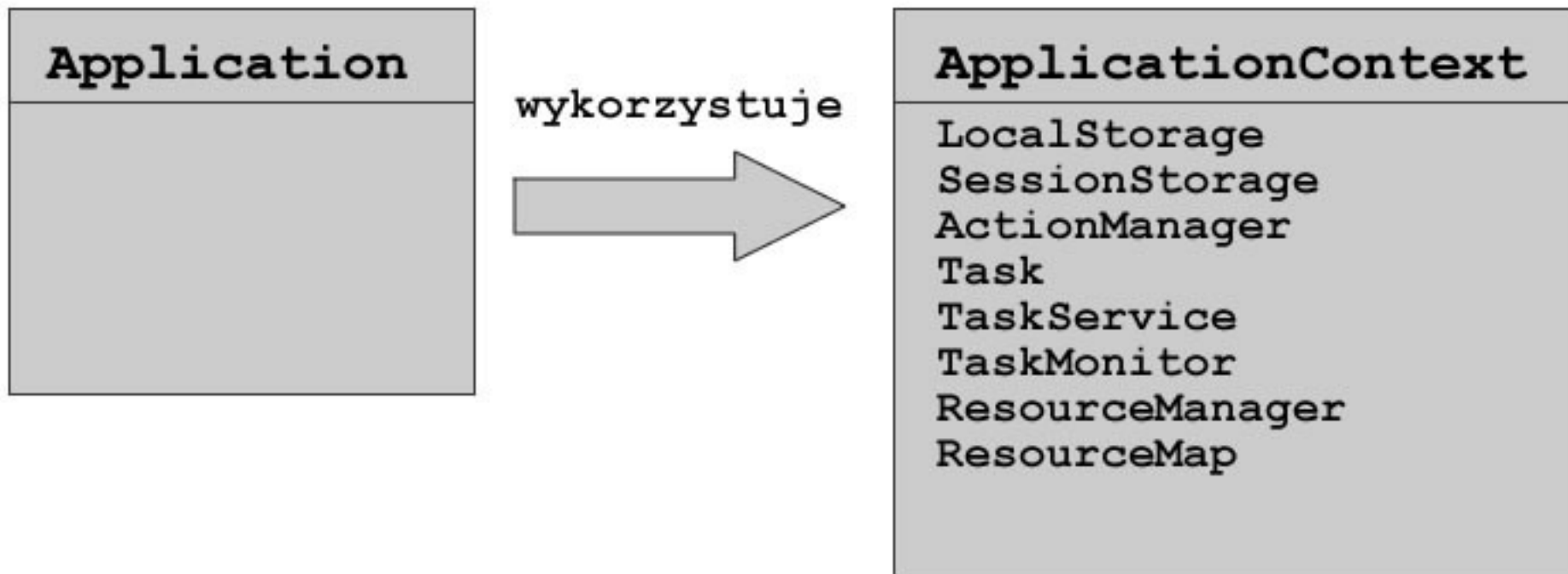
# SAF - co dostajemy?

- Minimalistyczne podejście, **bez:** języka skryptowego, znaczników GUI, IoC, XML-a, itp.
- Lepsze wsparcie ze strony narzędzi
- Standardowy sposób tworzenia GUI, programista nie jest zostawiony „sam sobie”

```
// Wsparcie narzędzi, podejście klasyczne
public class AplikacjaGUI{
    public static void main(String[] args){
        // Powodzenia!!!
    }
}
```

# Architektura SAF

- Dwie klasy do obsługi aplikacji (!!!) + dodatkowe, opcjonalne usługi



# Cykl życia: klasa `Application`

- `launch()` wywołuje `startup()` w wątku `Event Dispatching Thread`
- `initialize()` inicjalizacja tego, co potrzeba przed utworzeniem GUI
- `startup()` każda aplikacja przestania tę metodę, tutaj tworzony jest interfejs użytkownika
- `ready()` wszystko to, co trzeba zrobić zaraz po utworzeniu UI
- `exit()` jeżeli nie zablokujemy operacji wyjścia z aplikacji, to jest ona zamykana
- `shutdown()` sprzątanie po sobie

# Zasoby



Arial-24

Myra

2-Bc



Lucida Grande Italic



# Zasoby

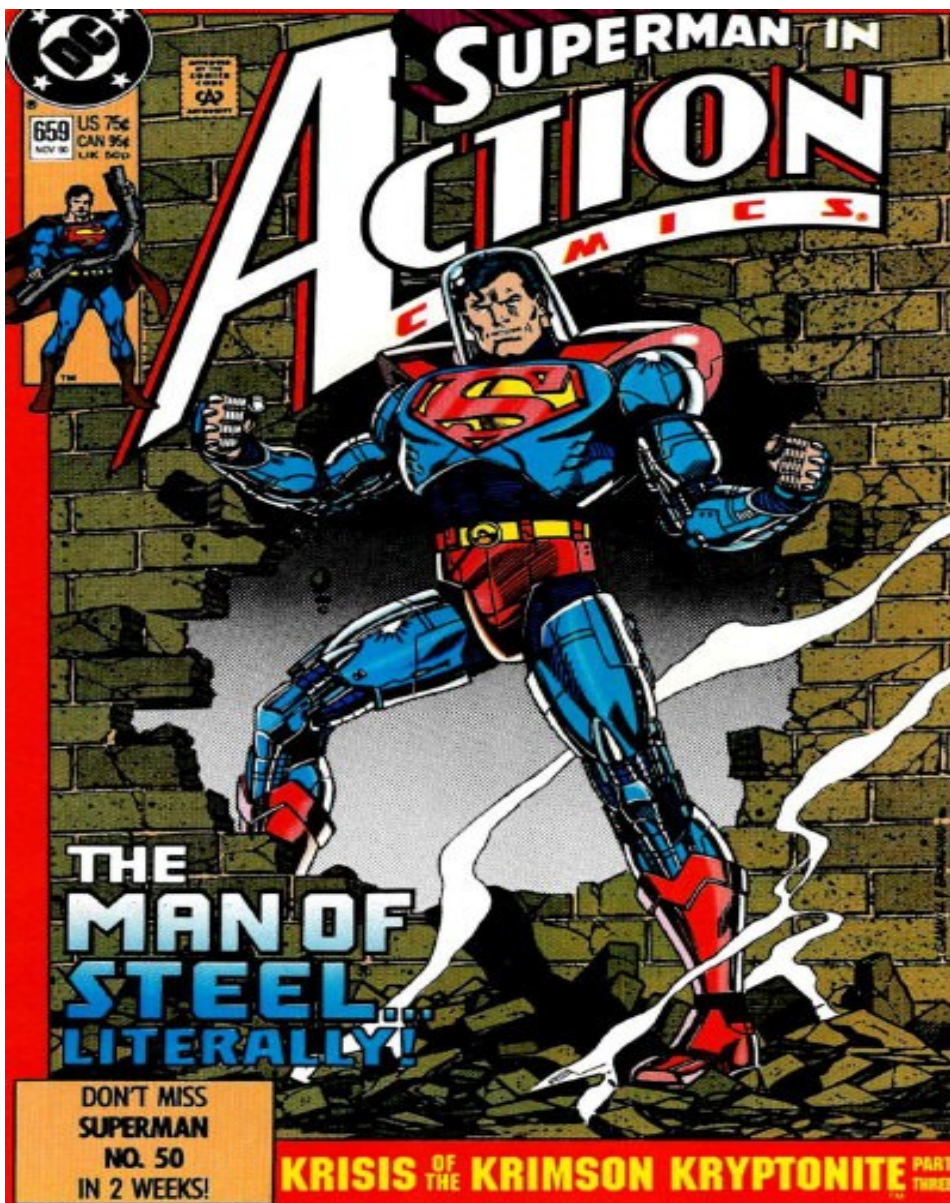
- Tekst umieszczony w etykietach, na przyciskach, w menu, ikony, typ czcionki, użyte kolory, pliki i18n, pomoc dla użytkownika...
- SAF uspoźnia obsługę zarządzania zasobami
  - globalne zasoby dla całej aplikacji
  - zasoby dla konkretnych klas
- Możliwość „wstrzykiwania zasobów”
- Konwersja (patrz dalej)
- Demonstracja

# Zasoby - konwersja

```
# resources/App.properties  
font = Arial-BOLD-22  
color = #BB0000  
icon = next.png
```

```
public class App extends SingleFrameApplication {  
    protected void startup() {  
        ApplicationContext ctx = getContext();  
        ResourceMap rsc = ctx.getResourceMap();  
        Color color = rsc.getColor("color");  
        Font font = rsc.getFont("font");  
        Icon icon = rsc.getIcon("icon");  
        label.setForeground(color);  
        label.setFont(font);  
        label.setIcon(icon);  
        //...  
    }  
}
```

# Obsługa zdarzeń: akcje



- SAF pozwala definiować akcje bardzo prosto:
  - bez interfejsów
  - `@Action`
- Akcje uruchamiane w tle
- Generator NetBeans
- Demonstracja

# Akcje: możliwości

- Łatwe przypisywanie jednej akcji do wielu kontrolek (przycisk, menu, *toolbar*)
- Blokowanie elementów UI przy akcjach uruchamianych w tle
- Możliwość monitorowania zadań uruchomionych w tle (**TaskMonitor**)
- Precyzyjna kontrola i synchronizacja zadań zadań uruchomionych w tle (**TaskService**)
- SAF jest przyjazną w użyciu nakładką na `java.util.concurrent`

# Stan sesji

- Aplikacja pamięta swoją ostatnią konfigurację:
  - rozmiar okna
  - położenie okna



# Podsumowanie

- Standardowy sposób oprogramowywania typowych zachowań UI
- Uproszczona obsługa zdarzeń, zarządzanie zasobami
- Integracja z Java 7 (SAF może się do tego czasu jeszcze zmienić!)
- Silne wsparcie ze strony NetBeans, ale można SAF używać z innymi IDE
- Przykładowa aplikacja + prezentacja:  
[www.xoft.pl](http://www.xoft.pl)