

Instytut Mikroelektroniki i Optoelektroniki

Zakład Fotoniki Obrazowej i Mikrofalowej



na kierunku Elektronika

w specjalności Elektronika i inżynieria komputerowa

Program do wygodnego budowania graficznego interfejsu użytkownika z wykorzystaniem biblioteki JavaFX

Jan Jakubik

nr albumu 268990

promotor

dr inż. Piotr Witoński

Warszawa 2018

Spis treści

[1 Wstęp 3](#_Toc513753616)

[1.1 Cel pracy 4](#_Toc513753617)

[2 Graficzny interfejs użytkownika 4](#_Toc513753618)

[2.1 Rys historyczny graficznych interfejsów użytkownika 4](#_Toc513753619)

[2.2 Współczesne biblioteki do budowania graficznego interfejsu użytkownika 4](#_Toc513753620)

[3 Język Java 4](#_Toc513753621)

[3.1 Rys historyczny języka 4](#_Toc513753622)

[3.2 Skrócony opis języka 4](#_Toc513753623)

[3.3 Język Java obecnie 4](#_Toc513753624)

[4 Biblioteka JavaFX 4](#_Toc513753625)

[4.1 Opis biblioteki 4](#_Toc513753626)

[4.2 Opis kontrolek dostępnych w bibliotece 4](#_Toc513753627)

[5 JavaFX jako widok we wzorcu projektowym Model-Widok-Kontroler 4](#_Toc513753628)

[5.1 Opis wzorca Model-Widok-Kontroler (MVC) 4](#_Toc513753629)

[5.2 Język XML stosowany do opisu graficznego interfejsu użytkownika} 4](#_Toc513753630)

[5.3 Opis języka XML 4](#_Toc513753631)

[5.4 Tworzenie interfejsu za pomocą dokumentów FXML 4](#_Toc513753632)

[5.5 Język CSS stosowany do dostosowania aplikacji od strony graficznej 4](#_Toc513753633)

[5.6 Inne rozwiązania łączące język Java z XML 4](#_Toc513753634)

[6 Rozwiązania istniejące na rynku 4](#_Toc513753635)

[6.1 Rozwiązania do biblioteki JavaFX 4](#_Toc513753636)

[6.2 Inne rozwiązania do języka Java 4](#_Toc513753637)

[6.3 Popularne rozwiązania do innych języków 4](#_Toc513753638)

[7 Aplikacja wykonana w ramach pracy inżynierskiej 4](#_Toc513753639)

[7.1 Założenia projektowe 4](#_Toc513753640)

[7.2 Schemat działania aplikacji 4](#_Toc513753641)

[7.3 Opis wybranych rozwiązań 4](#_Toc513753642)

[8 Instrukcja użytkowania aplikacji 4](#_Toc513753643)

[8.1 Tworzenie nowego projektu 4](#_Toc513753644)

[8.2 4](#_Toc513753645)

[8.3 Dodawanie kontrolek do okna 4](#_Toc513753646)

[8.4 Edycja istniejących kontrolek 5](#_Toc513753647)

[8.5 Konfiguracja programu 5](#_Toc513753648)

[9 Podsumowanie/Wnioski 5](#_Toc513753649)

# Wstęp

Do napisania ba koniec

## Cel pracy

Celem pracy jest zbudowanie narzędzia które umożliwi wygodne i efektywne tworzenie graficznego interfejsu użytkownika. W większości języków programowania ogólnego przeznaczenie tworzenie elementów graficznych wiąże się z trudnym i żmudnym procesem wymagającym od programisty znacznej wiedzy teoretycznej, a także wyspecjalizowanych bibliotek programistycznych których obsługa jest trudna i nie intuicyjna. Głównym założeniem tworzonego oprogramowania jest opracowanie narzędzia które będzie spełniało cechy:

* Będzie proste i intuicyjne w użyciu
* Będzie umożliwiało tworzenie graficznego interfejsu niezależnie od platformy
* Będzie wykorzystywało nowoczesne rozwiązania istniejące na rynku

# Graficzny interfejs użytkownika

W obecnych czasach urządzenia elektroniczne z niszowych urządzeń przemysłowo naukowych stały się częścią otaczającej nas rzeczywistości. Liczby sprzedanej elektroniki użytkowej każdego roku biją kolejne rekordy, a komputery osobiste stały się powszechnym narzędziem pracy. Większość ludzi posiada telefony komórkowe zwane obecnie smartfonami które swoją mocą obliczeniową i spektrum funkcjonalności znacznie przewyższają komputery które niegdyś zarządzały na przykład lotami wahadłowców. Od powstania tego typu urządzeń problemem naukowców i inżynierów było stworzenie połączenia komunikacyjnego między człowiekiem a maszyną tak aby było one zrozumiałe dla każdej ze stron. Ze względu na bardzo dużą rozległość omawianego tematu chciałbym skupić się w kolejnych rozdziałach na komunikacji między człowiekiem a komputerem osobistym, z pominięciem takich tematów jak interfejsy komunikacji internetu rzeczy lub inteligentnych samochodów gdyż nie jest to związane bezpośrednio z tematem pracy. W pracy założę standardową fizyczną obsługę komputerów poprzez obecnie dominujący sposób obsługi czyli przez zestaw: Mysz + Klawiatura lub poprzez ekran dotykowy i nie będę więcej tego tematu poruszał. W pracy skupię się na graficznych interfejsach użytkownika przeznaczonych na komputery które są codziennym narzędziem pracy.

Komunikacja z pierwszymi komputerami odbywała się najczęściej poprzez różne terminale znakowe, jednak ze względów biznesowych postanowiono uprościć sterowanie komputerami poprzez stworzenie graficznych interfejsów użytkownika, których użytkowanie stało się bardziej intuicyjne i zrozumiałe dla konsumenta.

Współczesne programy składają się z kontrolek, zwanych również widżetami z angielskiego słowa „Widget”. Kontrolki te odpowiadają za komunikacje i interakcję między użytkownikiem a komputerem. Służą zarówno do wyświetlania treści np. obrazki, pola tekstowe, tabele danych, czy różne wykresy, jak i do sterowania przez takie kontrolki jak: przyciski, rozwijane listy, czy przełączniki.

## Rys historyczny graficznych interfejsów użytkownika

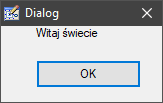
Za historycznie pierwszy graficzny interfejs użytkownika uznaje się oprogramowanie firmy Xerox stworzone w laboratorium PARC. Był to historycznie pierwszy graficzny system operacyjny w którym wyświetlanie komponentów graficznych polegało na odpowiednim wyświetlaniu bitmap. W tym oprogramowaniu pojawiły się takie elementy jak okna "pop-up" czyli wyskakujące a także listy rozwijane



Rysunek 1. Komputer Xerox Star z graficznym interfejsem użytkownika

Z czasem zaczęto dostrzegać rozwijanie graficznych interfejsów użytkownika w celach już czysto komercyjnych. Pojawił się system wyświetlania treści który polega na tym iż programy są wyświetlane w ramkach na ekranie. Rozwiązanie to zostało opatentowane przez firmę Apple 1 w systemie operacyjnym Mac OS. Taki sposób prezentacji treści został przyjęty przez wszystkie inne liczące się firmy informatyczne. Aplikacje były wyświetlane w oknach którymi zarządzał system operacyjny.

Większość komputerowych systemów operacyjnych dostarczała biblioteki programistyczne służące do obsługi elementów graficznych tak aby były one spójne z interfejsem graficznym całego systemu operacyjnego. Wiąże się to z nie przenośnością takiego oprogramowania na inne platformy. Przykładem takiej biblioteki jest biblioteka WinApi:



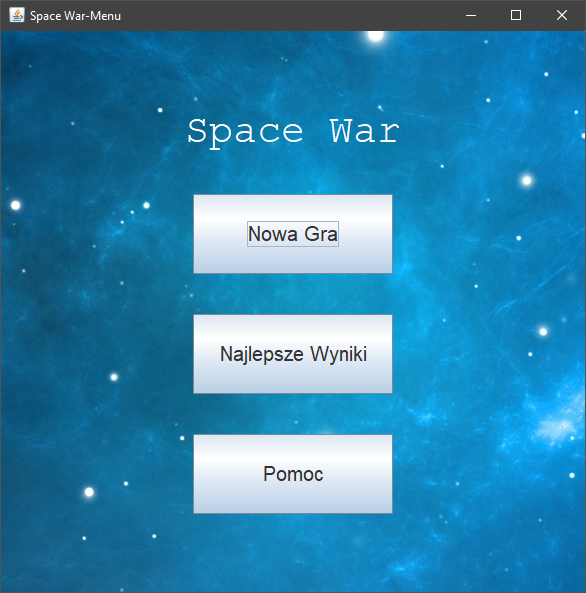
Rysunek 2. Prosty dialog WinApi wygenerowany przy pomocy programu ResEdit

Z czasem w językach programowania pojawiły się biblioteki które umożliwiały tworzenie spójnego interfejsu graficznego na różnych platformach gdzie ten sam kod wystarczyło skompilować różnymi kompilatorami. Pojawiły się także maszyny wirtualne które obsługiwały ten sam skompilowany kod maszynowy na różnych platformach. Jednak ze względu na mnogość platform i problemy patentowe żadnemu producentowi nie udało się stworzyć takiego rozwiązania które działało by na każdej platformie. Współczesną próbą wdrożenia takiego rozwiązania jest dynamiczny rozwój języków webowych a także zatwierdzenie standardu HTML 5. Oprogramowanie działające w przeglądarce jest obecnie najbardziej uniwersalnym oprogramowaniem jeśli chodzi o możliwości uruchomienia go na różnych systemach operacyjnych, jednak ze względu na ograniczenia wynikające z tego iż przeglądarki ze względów bezpieczeństwa działają w środowisku odizolowanym, programy przeglądarkowe nie nadają się do wszystkich zastosowań. Programy przeglądarkowe ze względu na to iż są interpretowane, są z reguły mniej wydajne od kompilowanych odpowiedników.

## Współczesne biblioteki do budowania graficznego interfejsu użytkownika

Współcześnie każdy popularny język programowania ogólnego przeznaczenia posiada biblioteki graficzne umożliwiające tworzenie graficznego interfejsu użytkownika. Wymienię i scharakteryzuje niektóre z nich

* Język C++
  + Biblioteka Qt: Jest to bardzo popularna biblioteka do tworzenia złożonych interfejsów użytkownika. Biblioteka Qt jest biblioteką występującą nie tylko w języku C++ jednak to właśnie w nim jest najchętniej używana. Programy napisane w Qt mogą być uruchamiane na różnych platformach jednak wymaga to ich kompilacji
  + Biblioteka wxWidgets: Otwarto źródłowa biblioteka graficzna, wieloplatformowa. Zapewnia jednolity interfejs graficzny nie zależnie od systemu operacyjnego.
* Języki .NET
  + Biblioteka graficzna służąca do tworzenia graficznych interfejsów użytkownika w językach obsługiwanych przez środowisko .NET
* Java
  + AWT: Pierwsza biblioteka graficzna Javy. Korzysta z systemowych elementów graficznych. Obecnie nie stosowana ze względu na problemy z wydajnością
  + Swing: Biblioteka służąca do tworzenia lekkich programów graficznych. Nie korzysta z systemowych kontrolek, tylko z własnych. Obecnie wycofywana z użycia, nadal popularna w wielu projektach.
  + JavaFX: Nowa biblioteka graficzna której możliwości zostały zaprezentowane w 2006 roku. Ze względu na to iż stanowi ona część tematu niniejszej pracy, zostanie ona szczegółowo omówiona w osobnym rozdziale.
  + SWT: Biblioteka rozwijana przez fundację Eclipse. Korzysta z systemowych elementów graficznych.



Rysunek 3. Menu główne gry napisanej w Javie. Wykorzystuje bibliotekę Swing

# Język Java

## Rys historyczny języka

## Skrócony opis języka

## Język Java obecnie

# Biblioteka JavaFX

## Opis biblioteki

## Opis kontrolek dostępnych w bibliotece

# JavaFX jako widok we wzorcu projektowym Model-Widok-Kontroler

## Opis wzorca Model-Widok-Kontroler (MVC)

## Język XML stosowany do opisu graficznego interfejsu użytkownika}

## Opis języka XML

## Tworzenie interfejsu za pomocą dokumentów FXML

## Język CSS stosowany do dostosowania aplikacji od strony graficznej

## Inne rozwiązania łączące język Java z XML

# Rozwiązania istniejące na rynku

## Rozwiązania do biblioteki JavaFX

## Inne rozwiązania do języka Java

## Popularne rozwiązania do innych języków

# Aplikacja wykonana w ramach pracy inżynierskiej

## Założenia projektowe

## Schemat działania aplikacji

## Opis wybranych rozwiązań

# Instrukcja użytkowania aplikacji

## Tworzenie nowego projektu

## Dodawanie kontrolek do okna

## Edycja istniejących kontrolek

## Konfiguracja programu

# Podsumowanie/Wnioski

# Bibliografia:

* Rysunek 1. : http://eduinf.waw.pl/inf/hist/002\_gui/xeroxstar.php