



POLITECHNIKA GDAŃSKA
WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI I INFORMATYKI



Katedra:	Algorytmów i Technologii Internetowych
Imię i nazwisko dyplomanta:	Marcin Jurczak
Nr albumu:	171641
Forma i poziom studiów:	Stacjonarne jednolite studia magisterskie
Kierunek studiów:	Informatyka

Praca dyplomowa magisterska

Temat pracy:
Wykorzystanie języka ELM do tworzenia aplikacji frontendowych.

Promotor:
dr inż. Krzysztof Manuszewski

Gdańsk, 2022

Streszczenie

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Słowa kluczowe: Elm, programowanie funkcyjne, wytwarzane aplikacje internetowych

Dziedzina nauki i techniki: Nauki inżynierskie i techniczne, inżynieria informatyczna.

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Keywords: Elm, functional programming, web development

Field of Science and Technology: Engineering and Technology, Information engineering.

Spis treści

1	Wstęp i cel pracy	4
2	Aplikacje frontendowe	5
2.1	Powszechne rozwiązania	5
2.1.1	React	5
2.1.2	Angular	5
2.1.3	Podobieństwa i różnice	5
3	Programowanie funkcyjne	6
4	Elm	7
4.1	The Elm Architecture	7
4.1.1	Model	8
4.1.2	Update	8
4.1.3	View	8
4.2	Narzędzia	8
5	Implementacja	10
5.1	Zegar	10
5.2	Pogoda	10
5.3	Wyszukiwarka	10
5.4	Zakładki	10
6	Instrukcja laboratoryjna	11
6.1	Przygotowanie środowiska	11
6.1.1	Linux	11
6.1.2	Windows	11
6.2	Podstawy języka Elm	11
6.3	Aplikacja frontendowa	11
7	Podsumowanie	12
7.1	Wnioski	12

Rozdział 1

Wstęp i cel pracy

Głównym celem niniejszej pracy jest zapoznanie się z funkcyjnym językiem programowania Elm oraz stworzenie frontendowej aplikacji internetowej. Ponadto chciałbym przeprowadzić porównanie tej technologii z innymi, bardziej powszechnymi rozwiązaniami do tworzenia aplikacji internetowych. Ostatnim celem pracy jest przygotowanie części dydaktycznej w postaci instrukcji laboratoryjnej, która mogłaby zostać potencjalnie wykorzystana w ramach przedmiotu Współczesne Aplikacje Programowania Funkcyjnego, prowadzonego przez mojego promotora, dra inż. Krzysztofa Manuszewskiego.

W drugim rozdziale skupiam się na przedstawieniu technologii powszechnie używanych do tworzenia frontendowych aplikacji internetowych, t.j. React oraz Angular. Prezentuję cechy wyróżniające

Rozdział 2

Aplikacje frontendowe

2.1 Powszechne rozwiązania

2.1.1 React

[1] [2]

2.1.2 Angular

[3] [4]

2.1.3 Podobieństwa i różnice

Rozdział 3

Programowanie funkcyjne

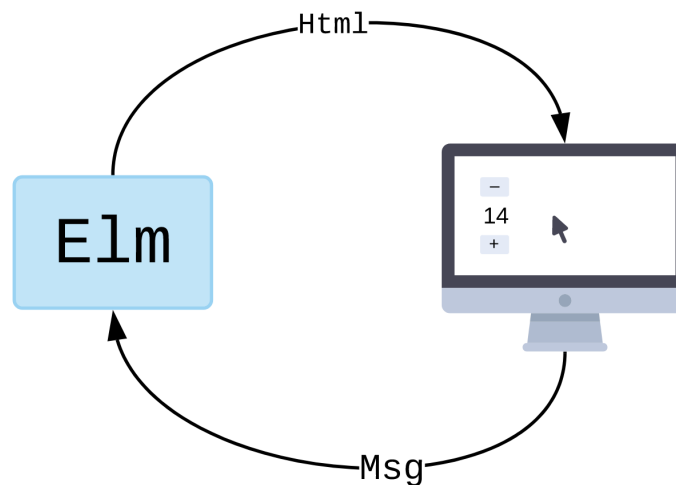
Rozdział 4

Elm

Elm [5] jest czysto funkcyjnym językiem programowania przeznaczonym do tworzenia graficznych interfejsów użytkownika. Powstał w roku 2012 wraz z opublikowaniem przez Evana Czaplickiego pracy „Elm: Concurrent FRP for Functional GUIs” [6]. Podczas jego tworzenia nacisk został położony na użyteczność, wydajność oraz niską podatność na błędy.

4.1 The Elm Architecture

The Elm Architecture jest schematem tworzenia interaktywnych aplikacji internetowych lub gier.



Rysunek 4.1: The Elm Architecture

Architektura Elma składa się z trzech podstawowych elementów:

- Model – opisujący stan aplikacji
- Update – opisujący logikę aplikacji
- View – opisujący wygląd aplikacji

4.1.1 Model

Celem modelu jest zdefiniowanie danych w naszej aplikacji. W tym przypadku model będzie bardzo prosty - jedna wartość liczbowa, która może zostać zwiększona lub zmniejszona.

Listing 4.1: *The Elm Architecture - Model*

```
type alias Model = Int

init : Model
init =
    0
```

4.1.2 Update

Listing 4.2: *The Elm Architecture - Update*

```
type Msg
    = Increment
    | Decrement

update : Msg -> Model -> Model
update msg model =
    case msg of
        Increment ->
            model + 1

        Decrement ->
            model - 1
```

Funkcja update ma za zadanie opisywać jak nasz model będzie się zmieniał w czasie. Może odebrać dwa typy wiadomości - Increment i Decrement. W wyniku operacji update otrzymujemy nowy, zaktualizowany model.

4.1.3 View

Funkcja view jako argument przyjmuje model i zwraca kod HTML. Wykorzystany został tutaj handler onClick, który po kliknięciu generuje odpowiednią wiadomość. Znak plusa generuje wiadomość Increment, znak minusa Decrement. Wybrana wiadomość trafia do funkcji update.

Listing 4.3: *The Elm Architecture - View*

```
view : Model -> Html Msg
view model =
    div []
        [ button [ onClick Decrement ] [ text "-" ]
        , div [] [ text (String.fromInt model) ]
        , button [ onClick Increment ] [ text "+" ]
        ]
```

4.2 Narzędzia

elm repl

elm init

`elm reactor`

`elm make`

`elm install`

`elm-live` [7]

Rozdział 5

Implementacja

W ramach części praktycznej niniejszej pracy stworzona została aplikacja internetowa typu *start-page*, czyli spersonalizowanej strony startowej przeglądarki zawierającej najpotrzebniejsze i najczęściej używane elementy oraz skróty. Na potrzeby aplikacji postanowiłem stworzyć cztery moduły:

- Cyfrowy zegar wskazujący aktualny czas w strefie czasowej użytkownika
- Pogoda w Gdańsku przedstawiona w formie krótkiego opisu oraz temperatury w stopniach Celsjusza
- Wyszukiwarka Google
- Zakładki zawierające odnośniki do wybranych stron internetowych

W poniższych podrozdziałach skupię się na opisie implementacji każdego modułu.

5.1 Zegar

5.2 Pogoda

5.3 Wyszukiwarka

5.4 Zakładki

Rozdział 6

Instrukcja laboratoryjna

W poniższym rozdziale przedstawiam przykładową instrukcję laboratoryjną, która krok po kroku przeprowadza czytelnika przez proces tworzenia aplikacji w Elmie, zaczynając od przygotowania środowiska deweloperskiego, przez podstawy języka wraz z ćwiczeniami pozwalającymi na lepsze zrozumienie składni, aż po stworzenie większej aplikacji frontendowej.

6.1 Przygotowanie środowiska

6.1.1 Linux

6.1.2 Windows

6.2 Podstawy języka Elm

6.3 Aplikacja frontendowa

Rozdział 7

Podsumowanie

7.1 Wnioski

Spis rysunków

4.1 The Elm Architecture	7
------------------------------------	---

Spis listingów

4.1	<i>The Elm Architecture</i> - Model	8
4.2	<i>The Elm Architecture</i> - Update	8
4.3	<i>The Elm Architecture</i> - View	8

Bibliografia

- [1] Alex Banks i Eve Porcello. *Learning React: functional web development with React and Redux*. " O'Reilly Media, Inc.", 2017.
- [2] Jordan Walke. *React documentation*. 2022. URL: <https://angular.io/docs>.
- [3] Ashok and Mehta Jain Nilesh and Bhansali. „AngularJS: A modern MVC framework in JavaScript”. W: *Journal of Global Research in Computer Science* (2014).
- [4] Google. *Angular documentation*. 2022. URL: <https://angular.io/docs>.
- [5] Evan Czaplicki. *Elm documentation*. 2022. URL: <https://elm-lang.org/docs>.
- [6] Evan Czaplicki. „Elm : Concurrent FRP for Functional GUIs”. W: *Elm : Concurrent FRP for Functional GUIs*. 2012.
- [7] Will King. *elm-live documentation*. 2022. URL: <https://www.elm-live.com/#documentation>.