Politechnika Gdańska

Wydział Elektroniki,
Telekomunikacji i Informatyki

Katedra: Algorytmów i Modelowania Systemów

Imię i nazwisko dyplomanta: Marcin Jurczak

Nr albumu: 171641

Forma i poziom studiów: Stacjonarne jednolite studia magisterskie

Kierunek studiów: Informatyka

Specjalność: Algorytmów i Technologii Internetowych

Praca dyplomowa magisterska

Temat pracy:
Wykorzystanie języka ELM do tworzenia aplikacji frontendowych.

Title of thesis:
Programming the front end applications with ELM language.

Opiekun pracy:
dr inż. Krzysztof Manuszewski

Data ostatecznego zatwierdzenia raportu podobieństw w JSA: //TODO

Gdańsk, 2022

Streszczenie

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer 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 aplikacji internetowych

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

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer 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

Wykaz najważniejszych skrótów

CI – ang. Continuous Integration, pol. ciągła integracja.
CD – ang. Continuous Deployment, pol. ciągłe wdrażanie.
HTML – ang. Hypertext Markup Language, pol. hipertekstowy język znaczników.
HTTP – ang. Hypertext Transfer Protocol, pol. protokół przesyłania hipertekstu.

Rozdział 1

Wstęp i cel pracy

Głównym celem niniejszej pracy jest zapoznanie się z funkcyjnym językiem programowania Elm oraz stworzenie przykładowej frontendowej aplikacji internetowej. Ponadto chciałbym przeprowadzić porównanie tej technologii z innymi, bardziej powszechnie używanymi 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

Trzeci rozdział poświęcam na

W czwartm rozdziale przedstawiam implementację przygotowanej aplikacji frontendowej napisanej w Elm'ie.

Piąty rozdział zawiera instrukcję laboratoryjną.

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

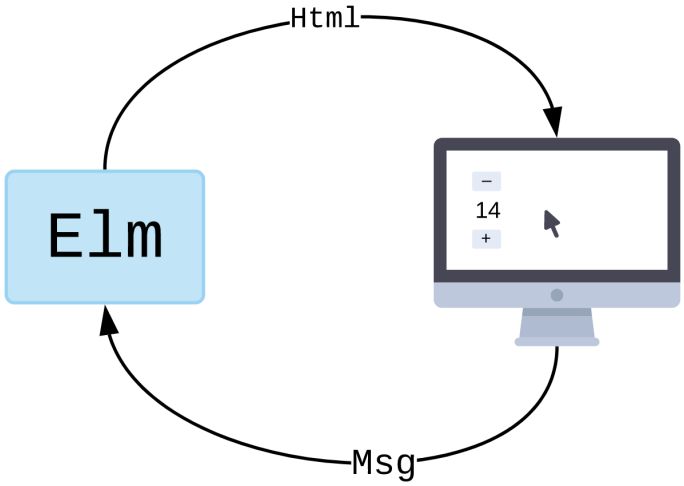
Rozdział 3

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.

3.1 The Elm Architecture

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



Rysunek 3.1: The Elm Architecture

Architektura Elma składa się z trzech podstawowych elementów:

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

3.1.1 Model

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

Listing 3.1: *The Elm Architecture* - Model

```
type alias Model = Int


init : Model
init =
  0
```

3.1.2 Update

Listing 3.2: *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.

3.1.3 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 3.3: *The Elm Architecture* - View

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

3.2 Narzędzia

elm repl

elm init

elm reactor

elm make

elm install

elm-live [7]

Rozdział 4

Implementacja

W ramach części praktycznej niniejszej pracy stworzona została aplikacja internetowa typu *startpage*, 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.

4.1 Zegar

4.2 Pogoda

4.3 Wyszukiwarka

4.4 Zakładki

4.5 Dokument hipertekstowy

4.6 Style

4.7 Całość

Rozdział 5

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.

5.1 Przygotowanie środowiska

Pierwszą rzeczą, którą należy się zająć przed rozpoczęciem nowego projektu jest przygotowanie odpowiedniego środowiska deweloperskiego. Należy upewnić się, że wszystkie narzędzia potrzebne do wykonania pracy są zainstalowane i prawidłowo skonfigurowane. W przypadku pracy z Elm'em będziemy korzystać przede wszystkim z kompilatora, edytora tekstu wspierającego podświetlanie składni, a także innych narzędzi wspomagających proces tworzenia oprogramowania z użyciem tej technologii.

5.1.1 Linux

Najprostszym sposobem instalacji platformy Elm na systemie operacyjnym Linux jest wykorzystanie narzędzia npm – powszechnie używanego menadżera pakietów służącego do zarządzania warstwą frontendową aplikacji internetowych. Aby zainstalować npm należy skorzystać z systemowego menadżera pakietów. Na przykładzie dystrybucji Ubuntu będą to komendy:

```
$ sudo apt update
$ sudo apt install npm
```

Kiedy narzędzie zostanie już pomyślnie zainstalowane, można przejść do instalacji platformy Elm. Posłuży do tego polecenie:

```
$ npm install –g elm
```

Zgodnie z dokumentacją npm[8], flaga `-g` oznacza, że pakiet zostanie zainstalowany globalnie, dzięki czemu będzie dostępny z każdego miejsca z systemu. Aby sprawdzić, czy rzeczywiście tak się stało, wystarczy uruchomić polecenie `elm --version`. Jeżeli w odpowiedzi otrzymamy prawidłową wersję kompilatora oznacza to, że platforma Elm jest gotowa do działania i można przejść do kolejnego kroku.

5.1.2 Windows

5.1.3 Edytor

5.1.4 Tworzenie projektu

5.2 Podstawy języka Elm

5.3 Aplikacja frontendowa

Rozdział 6

Podsumowanie

6.1 Wnioski

Spis rysunków

Spis listingów

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/`.

[8] Isaac Z. Schlueter. *npm documentation*. URL: `https://docs.npmjs.com/`.