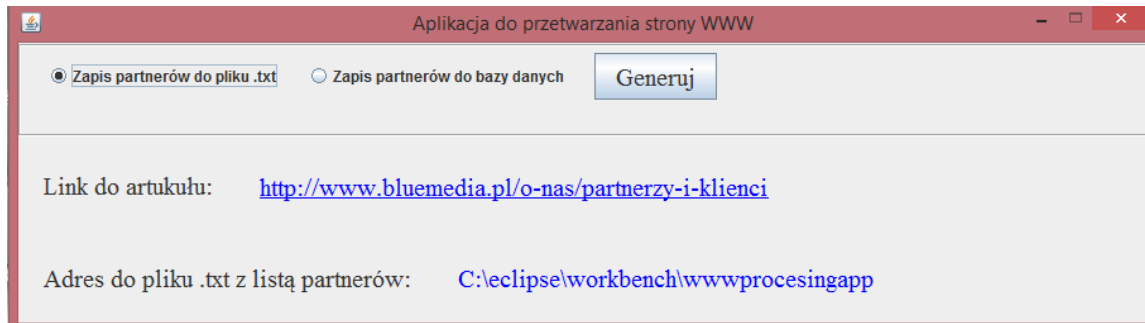


Aplikacja do przetwarzania strony WWW

1. Wygląd aplikacji



Aplikacja składa się z dwóch części. Pierwsza część jest odpowiedzialna za pobieranie listy partnerów ze strony WWW. Użytkownik musi wybrać czy dane mają zostać zapisane do pliku tekstowego czy do bazy danych. Druga część jest odpowiedzialna za wyświetlenie informacji o adresie strony WWW oraz ścieżce, w której jest zapisany plik tekstowy z listą partnerów. Podane adresy są to hiperłącza.

2. Struktura projektu

Do budowania projektu zostało wykorzystane narzędzie Maven. Projekt składa się z czterech klas (Main, MainFrame, Parser, DBconnection).

a. Klasa Main

Jest to główna klasa projektu. Jest odpowiedzialna za utworzenie okna aplikacji.

b. Klasa MainFrame

Klasa ta jest odpowiedzialna za inicjalizację komponentów, które są umieszczone w aplikacji. Są to takie elementy jak: panele, etykiety, przyciski.

c. Klasa Parser

Służy do pobrania źródła strony internetowej a następnie pobrania listy partnerów znajdujących się na tej stronie.

d. DBconnection

Klasa odpowiedzialna za połączenie z bazą danych oraz dodawanie nowych rekordów do bazy.

3. Opis metody pobierającej listę partnerów

Metodę można podzielić na dwie części. Pierwsza jest odpowiedzialna za znalezienie początku listy partnerów. Poniżej znajduje się kod odpowiedzialny za tą część.

```

// searching the first partner
do {
    // searching the text: "<ul> <li>"
    isStartOfPartners = text.(searchingStartOfList, index);

    if (isStartOfPartners) // if "<ul> <li>" is found
    {
        index += 36; // moving the index after <ul><li>

        isStartOfPartners = text.startsWith(charOfEnd, index); // checking if the next char is not '<'

        if (isStartOfPartners == true) // if the next char is '<', we do not have the first partner
            isStartOfPartners = false;
        else // if the next char is not '<', we have got the first partner, exit the 'do...while' loop
            isStartOfPartners = true;
    } else
        index++;
} while (isStartOfPartners == false);

```

W pętli do...while zwiększany jest indeks początkowy tak długo, aż zostanie znaleziony ciąg znaków „”. Jeżeli ciąg ten zostanie znaleziony, sprawdzone będzie, czy znak po wspomnianym ciągu nie jest znakiem specjalnym tzn. ‘<’. Jeżeli nie, oznacza to, że jest to pierwsza litera pierwszego partnera i wykonywanie pętli do...while zostanie zakończone. W przeciwnym wypadku, pętla będzie szukać nowego ciągu „”.

Druga część jest odpowiedzialna za pobranie listy partnerów oraz zapisanie ich do pliku tekstowego lub bazy danych.

```

// searching other partners
do {
    // if char is not '<', we have the first letter of partner
    while (text.charAt(index) != '<')
    {
        partnerName += text.charAt(index); // get the next letter and add it to the partnerName variable
        index++;
        isFoundNewPartner = true;
    }

    if(isFoundNewPartner)
    {
        System.out.println(partnerName);

        if (whereSave == 1)
            save.println(partnerName); // add the partner name to the .txt file
        else
            dbConnection.addNewPartner(partnerName);

        isFoundNewPartner = false;
        partnerName = "";
    }

    isEndOfPartners = text.startsWith(endOfPartnerList, index); // checking if this is not the end of partner list

    if (isEndOfPartners == false) // if is not
    {
        do {
            index++; // increase the index until not find the new partner ('<li>')
        } while (text.startsWith(newPartner, index) == false);

        index += 4; // moving the index after <li>
    }
} while (isEndOfPartners == false);

```

Jeżeli napotkany znak nie jest znakiem '<' oznacza to, że jest to litera związana z nazwą partnera. Dodajemy ją do zmiennej *partnerName*. Flaga *isFoundNewPartner* jest ustawiana na true, aby po zakończeniu pobierania całej nazwy przekazać ją do pliku tekstowego lub bazy danych. Po każdym znalezionym partnerze sprawdzane jest, czy nie był to już ostatni partner. Odpowiedzialne za to jest: *isEndOfPartners = text.startsWith(endOfPartnerList, index)*; Jeżeli są jeszcze partnerzy do pobrania, szukamy ciągu , a następnie ustawiamy wskaźnik za tym ciągiem, aby można było ponownie pobrać nazwę partnera.

4. Baza danych

Baza danych składa się z jednej tabeli, w której są dwie kolumny: ID oraz nazwa partnera. ID jest kluczem głównym. Do bazy danych użyto PostgreSQL.

5. Testy jednostkowe

Dla aplikacji powinny zostać napisane testy jednostkowe. Przykładowe testy jednostkowe to:

- Brak partnerów na stronie internetowej
- Brak strony www o podanym adresie
- Brak połączenia z bazą danych
- Baza danych nie istnieje
- Baza danych zawiera już partnerów, które aplikacja chce dodać po raz kolejny