

# Dokumentacja końcowa projektu TKOM

---

## I. Temat

### XII. Zad HTML2

Napisać program scalający dwa pliki HTML, plik pierwotny oraz plik zmodyfikowany. Jeśli lokalna modyfikacja była wprowadzona w jednym z plików to zostaje wprowadzona do pliku docelowego. Przypadki konfliktowe rozstrzyga użytkownik.

## II. Opis

Program ma za zadanie wczytać podane na wejście dwa pliki w formacie HTML oraz dokonać ich scalenia wg pewnej opisaney poniżej tabeli z opisanymi przypadkami konfliktowymi. W celu wykonania takiego scalenia dokonywana jest analiza leksykalna i rozbiór składniowy obydwu plików poprzez wyróżnienie tokenów oraz zbudowania na ich podstawie drzew rozbioru zadanych plików. Pliki traktowane są w podobnej skali ważności, z lekką przewagą dla pliku podanego w kolejności jako drugiego, podczas rozstrzygania połączeń.

### A. Realizacja i działanie

Projekt został zrealizowany i napisany w języku Java. Wybór tego języka programowania został sprowokowany poprzez rozważenie różnic w możliwościach różnych języków. Postać pozornie łatwiejszej implementacji projektu oraz nieskomplikowana implementacja efektów graficznych, były głównymi cechami warunkującymi wybór. Rozważywszy różne możliwości, prostota tworzenia UI dająca wygodę użytkownikowi znacząco przeważała na korzyść tej właśnie technologii.

Pliki wynikowe zostają zbudowane z poprawnie stworzonego drzewa rozbioru, jednakże celem reprezentacji przyjaznej użytkownikowi, kod formatowany jest z użyciem wcięć. Sama analiza i generacja dokumentu typu HTML swoją trudność sprawia poprzez użycie białych znaków których powtórzenia są ignorowane, dlatego pliki wynikowe są zazwyczaj poprawnym odwzorowaniem plików z dokładnością do skrajnych pojedynczych białych znaków takich jak spacje, tabulacje czy znaki nowej linii.

### B. Założenia

Podczas uruchomienia algorytmu zaimplementowanego w programie zakładane jest, że pliki podane na wejście zostały poddane działaniu walidatora HTML <sup>(projekt wstępny pkt. VII)</sup>. Założenie to ma na celu wyeliminowanie braków w złożonej semantyce dokumentu HTML budowanego wg danego w jego źródle typu dokumentu (restrykcji) tj. wyeliminowanie semantycznie zabronionych zagnieżdżeń bądź niedozwolonych nazw atrybutów.

Jest to założenie pozwalające na wygenerowanie względnie poprawnego semantycznie pliku wynikowego z dokładnością do wyborów użytkownika dla nielicznych nielegalnych zagnieżdżeń wynikających z dosyć elastycznej i skomplikowanej składni dokumentów HTML. W przypadku błędnego użycia walidatora lub jego braku, błędy leksykalne zostają pokazane w postaci błędu zawierającego niedozwoloną konstrukcję, znacznik wraz z linią oraz numerem pozycji na którym znaleziony został przypadek.

## III. Struktura programu

### A. Podział na moduły

Program został podzielony na 4 główne moduły:

1. **Lexer** - Odpowiedzialny za analizę leksykalną podanych plików wejściowych. Na podstawie analizy ciągu wejściowego generuje listę tokenów<sup>(projekt wstępny pkt. V.A.1.b)</sup> rozpoznanych na wejściu. W przypadku znalezienia błędnych konstrukcji leksykalnych przerywa analizę zwracając błąd w postaci:  
Wiadomość, {nr linii}, {nr znaku}, fragment otaczającego kodu (+- 10 znaków)
2. **Parser** - Odpowiedzialny za budowę drzew rozbioru z podanej na wejściu listy rozpoznanych tokenów. Budowa odbywa się na podstawie uproszczonej gramatyki<sup>(projekt wstępny pkt. VI.B.1.b)</sup> języka HTML. W przypadku błędu zwracany jest komunikat o zaistniałej sytuacji wraz z informacją co było spodziewane na wejściu.
3. **Linker** - Odpowiednik generatora kodu<sup>(projekt wstępny VI.C.c)</sup>. Zawiera logikę działania programu. Poddaje analizie podane na wejście drzewa rozbioru plików wg zadanego w programie algorytmu<sup>(projekt wstępny pkt. VI.C.b)</sup>.
4. **View** - Odpowiedzialny za wyświetlenie prostego GUI programu wraz z obsługą zdarzeń generowanych przez użytkownika w postaci wyboru plików czy naciśnięcia przycisków.

## B. Struktura klas

Główne klasy reprezentujące logikę aplikacji z wyjątkiem interfejsu użytkownika opisanego w *pkt III.C*. Całość kodu źródłowego wraz z komentarzami w języku angielskim dostępne w postaci dokumentacji *JavaDoc*. Poniżej zostały przedstawione jedynie istotne fragmenty z punktu widzenia zewnętrznego API udostępnianego przez bibliotekę.

### 1. Lexer

```
public class Lexer {

    private int lineNumber;           // Aktualnie analizowana linia
    private int linePosition;         // Aktualny znak w linii
    private String illegalTag;        // Używane do propagacji znalezionego błędnego znacznika
    private String text;              // Tekst poddawany analizie
    private int pos;                  // Aktualna pozycja w analizowanym tekście
    private LinkedList<Token> tokens; // Lista generowanych tokenów
    private Stack<String> stack;      // Stos pomocniczy do utrzymania poprawnej kolejności
                                     // znaczników otwierających i zamykających

    //...

    public LinkedList<Token> extractTokens(); // Funkcja rozpoczynająca proces analizy
                                             // leksykalnej podanego w konstruktorze
                                             // tekstu. Zwraca wygenerowaną listę tokenów
                                             // lub rzuca RuntimeException gdy napotka
                                             // błąd.

    //...
}
```

### 2. Parser

```
public class Parser {

    private TreeElement tree;          // Generowane drzewo wynikowe
    private TagElement actualElement;  // Aktualny element używany do budowy drzewa
                                     // analizując tekst na różnych poziomach
                                     // zagnieżdżeń

    private LinkedList<Token> tokens;  // List podanych na wejściu tokenów do analizy
    private Iterator<Token> tokenIterator; // Iterator poruszający się po liście tokenów

    //...

    public TreeElement parseTokens(); // Funkcja rozpoczyna parsowanie podanej w konstruktorze
                                     // listy tokenów. Zwraca zbudowane drzewo rozbioru
                                     // lub rzuca RuntimeException gdy napotka błąd.

    //...
}
```

### 3. Linker

```
public class Linker {
    private Map<Integer, Integer> conflictMap; // Mapa użytych konstrukcji konfliktowych z talie
                                                // konfliktów w postaci nr konfliktu=ilość
                                                // wystąpień podczas złączenia

    private TreeElement first; // Drzewo rozbirodu pierwszego pliku
    private TreeElement second; // Drzewo rozbirodu drugiego pliku
    private TreeElement result; // Zmienna na drzewo wynikowe
    private TagElement actualResult; // Aktualnie rozważany element drzewa wynikowego

    //...

    public TreeElement generateResult(); // Funkcja rozpoczynająca wykonanie algorytmu złączenia
                                        // drzew rozbirodu wcześniej podanych plików. W
                                        // przypadku konfliktu wymaganego interakcji
                                        // użytkownika wyświetla okno dialogowe oczekujące
                                        // wyboru którejs z podanych możliwości rozwiązania.
                                        // Zwraca drzewo rozbirodu pliku wynikowego.

    //...
}
```

### 4. Użyte tokeny

```
public abstract class Token {
    //...
    private String value; // Wartość tokena
    //...
    public String toString() {
        return value;
    }
    //...
}

public class DoctypeToken extends Token {
    //...
    public String toString() {
        return "<!doctype " + value +
">";
    }
    //...
}

public class StartOpenTagToken extends Token {
    //...
    public String toString() {
        return "<" + value;
    }
    //...
}

public class EndOpenTagToken extends Token {
    //...
    public String toString() {
        return "</" + value;
    }
    //...
}

public class CloseTagToken extends Token {
    //...
    public String toString() {
        return ">";
    }
    //...
}
```

```
public class CloseEmptyTagToken extends Token {
    //...
    public String toString() {
        return " />";
    }
    //...
}

public class AttributeKeyToken extends Token {
    //...
    public String toString() {
        return " " + value;
    }
    //...
}

public class AttributeEqualsToken extends Token {
    //...
    public String toString() {
        return "=";
    }
    //...
}

public class AttributeValueToken extends Token {
    //...
    private char quote; // Ogranicznik
    używany do opakowania wartości
    //...
    public String toString() {
        return this.quote + value +
this.quote;
    }
    //...
}

public class TextToken extends Token {
    //...
}

public class CommentToken extends Token {
    //...
    public String toString() {
        return "<!--" + value + "-->";
    }
    //...
}
```

## 5. Drzewo rozbioru

```

public class TreeElement {
    //...
    private String doctype;                // Typ dokumentu
    private LinkedList<Element> elements;    // Zawsze powinien być jeden element jako
    korzeń(root) dokumentu
    //...
    public String toString() {
        // Generuje kod źródłowy dokumentu HTML oraz zawierających elementów
    }
    //...
}

public abstract class Element {
    protected String tag;                  // Nazwa znacznika
    protected boolean empty;              // Czy jest to pojedynczy znacznik
    protected Element parent = null;      // Element nadrzędny
    protected int level = 0;              // Używane do generacji wcięć
    //...
}

public class TagElement extends Element {
    private Map<String, String> attributes = new TreeMap<String, String>(); // Mapa atrybut-
    wartość znacznika
    private LinkedList<Element> elements = new LinkedList<>(); // Lista elementów dzieci
    //...
}

public class TextElement extends Element {
    protected String tag = "#text";        // Nazwa znacznika elementu tekstowego
    protected boolean empty = true;        // Zawsze bez znacznika końcowego
    protected String content;              // Zawartość tekstowa znacznika
    //...
}

public class CommentElement extends TextElement {
    protected String tag = "#comment";     // Nazwa znacznika elementu komentarza
    //...
}

```

## 6. Lista dozwolonych znaczników wraz z informacją o tym czy są to podwójne czy pojedyncze znaczniki

```

public enum TagQuantity {

    a(false), abbr(false), acronym(false), address(false), applet(false), article(false),
    //...
    param(true), source(true), track(true), wbr(true), nobr(false);

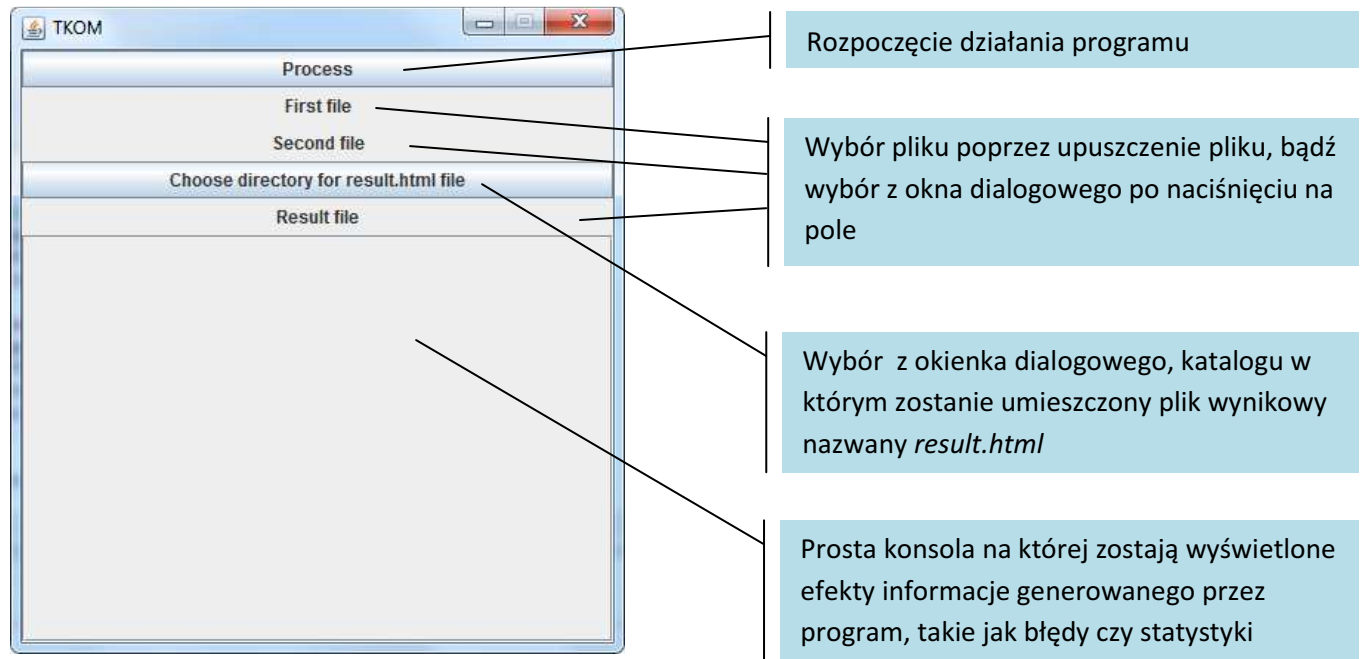
    private boolean single;                // Czy jest <tag /> czy <tag></tag>

    TagQuantity(boolean single) {
        this.single = single;
    }

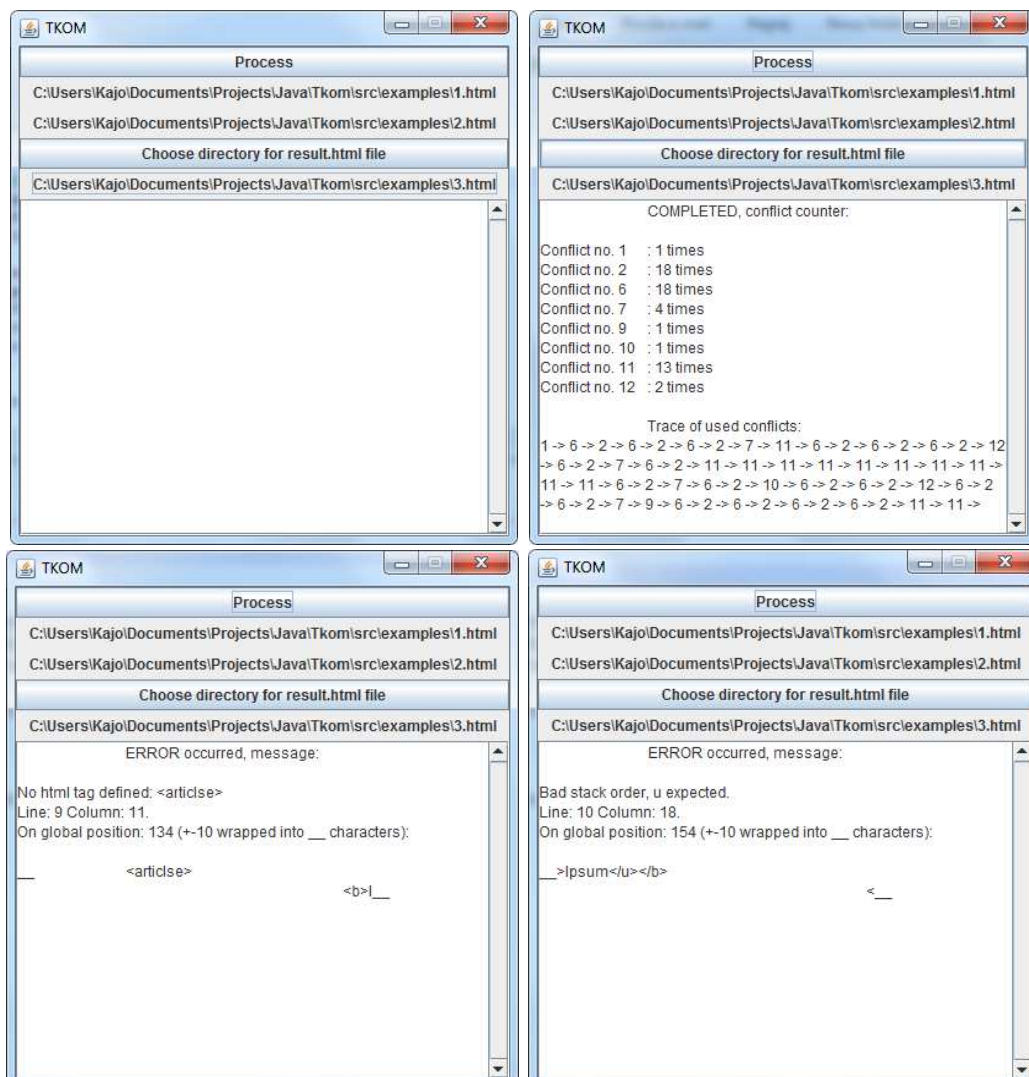
    public boolean isSingle() {
        return single;
    }
}

```

## C. GUI



Przykładowe wyniki uruchomienia:



## IV. Zmiany w stosunku do projektu wstępnego

### A. Tabela konfliktów (projekt wstępny pkt. VI.A.2)

W trakcie implementacji okazało się iż konflikt nr 8 jest efektywnie zastąpiony z użyciem konfliktu nr 12. Zostają porównane dwa różne znaczniki w efekcie czego jednym z możliwych wyborów jest wewnętrzna analiza znacznika (konflikt nr 12 opcja 4) dająca rezultat użycia konfliktu nr 8.

Na poziomie implementacyjnym konflikty o numerach od 2 do 5 włącznie ze względu na swoją konstrukcję, zostały efektywnie zaimplementowane jako jedna funkcja z użyciem mapy atrybutów: *klucz=wartość*. Nie zmienia to sensu założeń jednakże pozwoliło to uniknąć dublowania kodu.

#### 1. Zaktualizowana tabela konfliktów

##### a) Legenda

Text – ciąg znaków nie zawierający znaczników  
 Text1 ≠ Text  
 <tag> – jakiś znacznik  
 <tagX> ≠ <tag> ≠ <tagY>, X ≠ Y  
 ... – tekst lub znaczniki  
 ...1 ≠ ...2

##### b) Tabela

Lp.	Plik pierwotny	Plik zmodyfikowany	Plik docelowy
1	Znacznik typu dokumentu wybierany z przewagą dla drugiego pliku		
	<!doctype Text>	<!doctype Text1>	<!doctype Text1>
2	<b>Różne kombinacje występowania atrybutów zastąpione z przewagą dla pliku zmodyfikowanego przedstawione poniżej wraz z przykładami</b>		
	W pierwszym pliku brak atrybutu/atributów, w drugim występują. Atrybuty są sumowane.		
	<tag>	<tag a="b" ..>	<tag a="b" ..>
	W pierwszym pliku występują atrybuty, w drugim brak. Atrybuty są sumowane.		
	<tag a="b" ..>	<tag>	<tag a="b" ..>
	W obydwu plikach występują różne atrybuty. Atrybuty są sumowane.		
	<tag a="b" ..>	<tag c="d" ..>	<tag a="b" c="d" ..>
	W obydwu plikach występują atrybuty. Jeżeli nazwy atrybutów się powtarzają to wybierane są z pliku drugiego.		
	<tag a="b" ..>	<tag a="c" ..>	<tag a="c" ..>
6	Jeżeli tagi są takie same, algorytm scalania (VI.C.b) przepisuje znacznik i ewentualnie weryfikuje atrybuty wg tabeli oraz następnie sprawdza kolejno wnętrze tegoż znacznika		
	<tag ..1>...1</tag>	<tag ..2>...2</tag>	<tag ..3> // Wewnętrzna analiza </tag>
7	Jeżeli tagi są takie same i zawierają tylko tekst. Teksty są różne, to użytkownik ma do wyboru możliwe rozwiązania przedstawione poniżej.		
	<tag>Text</tag>	<tag>Text1</tag>	<b>Konflikt, wybór:</b> 1. <tag>Text</tag> 2. <tag>Text1</tag> // Konkatenacja tekstu: 3. <tag>TextText1</tag> 4. <tag>Text1Text</tag>
9	Jeżeli porównywany jest sam tekst z nim samym ale otagowanym pojedynczym znacznikiem to wynikiem jest		

	tenże otagowany tekst.		
	Text	<tag>Text</tag>	<tag>Text</tag>
10	Jeżeli porównywany jest tag zawierający jakieś elementy z elementem tekstowym, to wynikiem jest konkatenacja tagu i elementu tekstowego		
	<tag>...</tag>	Text	<tag>...</tag> Text
11	Jeżeli porównywane jest wnętrze znacznika i w jednym z plików nie ma znaczników a w drugim są lub w jednym jest więcej elementów niż w drugim to znaczniki są sumowane		
	<tag></tag>  -----np.- <tag> <tag1></tag1> </tag>	<tag> ... </tag> -----np.- <tag> <tag1></tag1> ... </tag>	<tag> ... </tag> -----np.- <tag> <tag1></tag1> ... </tag>
12	Jeżeli porównywane elementy to różne tagi które zawierają jakieś elementy, to użytkownik ma do wyboru możliwe rozwiązania przedstawione poniżej:		
	<tag> ...1 </tag>	<tag1> ...2 </tag1>	<b>Konflikt, wybór:</b> 1. <tag>...1</tag> <tag1>...2</tag1> 2. <tag>...1</tag> 3. <tag1>...2</tag> 4. Wybór <i>tag1</i> z pliku drugiego i rozstrzygnięcie konfliktu pomiędzy <i>tag</i> z pliku pierwszego i wnętrzem <i>tag1</i> pliku drugiego
12.1	Przykład do pkt. 11: Do pliku wynikowego dodawany jest znacznik otaczający <i>tag1</i> i następnym elementem do rozstrzygnięcia jest ten sam znacznik pliku pierwszego <i>tag</i> i wnętrzem znacznika pliku drugiego <i>tag1</i> -> <i>tag2</i> ...3		
	<tag> ...1 </tag>	<tag1> <tag2> ...2 </tag2> ...3 </tag1>	<b>Cd. opis:</b> Dodaj <tag1> do pliku docelowego i rozstrzygaj konflikt pomiędzy: <tag>...1</tag> i wnętrzem <tag1> <tag2>...2</tag2>

## B. Algorytmy

Działanie algorytmów analizy leksykalnej<sup>(projekt wstępny pkt. V.A.3)</sup>, algorytmu parsera<sup>(projekt wstępny pkt. VI.B.2.a)</sup> i algorytmu budowy docelowego drzewa<sup>(projekt wstępny pkt. VI.C.b)</sup> zostały zmienione nieznacznie, tylko w stopniu umożliwiającym efektywną implementację w wybranej technologii (*Java*) z użyciem funkcji pomocniczych w myśl dobrej praktyki programistycznej. Istotne zmiany widoczne są jedynie w funkcji **b** analizatora leksykalnego od kroku nr 4.

### 1. Analiza leksykalna

Analiza zadanego tekstu z pliku HTML. W jej trakcie budowana jest lista tokenów.

1. Jednorazowe wykonanie funkcji **a** w celu znalezienia typu dokumentu
2. Wykonywanie w pętli funkcji **b** w celu znalezienia znaczników
3. W przypadku znalezienia znacznika jednorazowe wykonanie funkcji **c** dla każdego znacznika otwierającego

#### a) Znalezienie opisu dokumentu - wywoływana jednorazowo

1. Znajdź '*<!doctype*' na początku pliku.
2. Jeśli brak lub nie na początku
  - 2.1. Zakończ działanie wyświetlając komunikat o błędzie

3. Znajdź znacznik kończący '>'
4. Wyodrębnij typ dokumentu i zapisz do listy tokenów -> `doctypeToken`

#### **b) Szukanie elementu - wywoływana w pętli**

1. Znajdź najbliższy początek znacznika '<'
2. Jeżeli występuje jakiś tekst do pozycji znalezionej znacznika
  - 2.1. Zapisz jako token -> `textToken`
3. Jeżeli brak znaku '<'
  - 3.1. Zakończ działanie
4. Jeżeli następny znak to '/' - oczekuj znacznika końcowego
  - 4.1. Znajdź najbliższy znak '>'
  - 4.2. Używając znaków do znalezionej znacznika zapisz tokeny -> `endOpenTagToken` i `closeTagToken`
  - 4.3. Zakończ działanie funkcji
5. Jeżeli następny znak to '!' - oczekuj komentarza
  - 5.1. Wszystkie znaki do najbliższego ciągu '-->' potraktuj jako komentarz i zapisz jako `CommentToken`
  - 5.2. Zakończ działanie funkcji
6. W przeciwnym przypadku wywołaj funkcję `c`
7. Znajdź najbliższy koniec znacznika '>'
8. Jeżeli znaleziony element jest pusty '... />' - {*single\_tag*}
  - 8.1. Zapisz token `closeEmptyTagToken`
9. W przeciwnym przypadku
  - 9.1. Zapisz token `closeTagToken`
10. Zakończ działanie funkcji

#### **c) Analiza znacznika pomiędzy znakami '<' i '>' - wywoływana przez b**

1. Znajdź pierwszy biały znak
2. Jeżeli brak
  - 2.1. Znajdź końcowy znak znacznika '>'
3. Potraktuj tekst od początku do znalezionej znaku jako nazwę znacznika i zapisz jako token -> `startOpenTagToken`
4. Pomiń kolejne białe znaki
5. Jeżeli znaleziono koniec znacznika '>' lub '/>'
  - 5.1. Zakończ działanie funkcji
6. Znajdź najbliższy znak '='
7. Tekst od ostatniego białego znaku do znaku '=' potraktuj jako nazwę atrybutu i zapisz jako token -> `attributeKeyToken` i `attributeEqualsToken`
  - 7.1. Jeżeli brak znaku " lub '
    - 7.1.1. Znajdź najbliższy biały znak
    - 7.1.2. Jeżeli brak to znajdź koniec znacznika '>' lub '/>'
      - 7.1.2.1. Zapisz tekst od znaku '=' do znalezionej znaku jako token -> `attributeValueToken`
  - 7.2. Jeżeli znaleziono " lub '
    - 7.2.1. Znajdź kończący znak " lub '
    - 7.2.2. Zapisz tekst pomiędzy znalezionymi znakami jako token -> `attributeValueToken`
8. Wróć do pkt. 4

## **2. Algorytm parsera**

1. Wczytaj token
2. Jeżeli `doctypeToken`



- 2.1. Wpisz do struktury drzewa
3. Jeżeli brak
  - 3.1. Zgłoś błąd braku typu dokumentu
4. Wczytaj kolejny token
5. Jeżeli `startOpenTagToken`
  - 5.1. Utwórz element w aktualnym węźle drzewa
  - 5.2. Wczytaj kolejny token
  - 5.3. Jeżeli `closeEmptyTagToken`
    - 5.3.1. Wróć do pkt. 4
  - 5.4. Jeżeli `closeTagToken`
    - 5.4.1. Przejdź do utworzonego elementu i wróć do pkt. 4
  - 5.5. Jeżeli `attributeKeyToken`
    - 5.5.1. Dodaj atrybut identyfikowany danym kluczem do węzła
    - 5.5.2. Wczytaj token
    - 5.5.3. Jeżeli nie `attributeEqualsToken`
      - 5.5.3.1. Zgłoś błąd i zakończ
    - 5.5.4. Wczytaj token
    - 5.5.5. Jeżeli nie `attributeValueToken`
      - 5.5.5.1. Zgłoś błąd i zakończ
    - 5.5.6. Zapisz wartość atrybutu
    - 5.5.7. Wróć do pkt. 5.2
6. Jeżeli `endOpenTagToken`
  - 6.1. Wejdź poziomo wyżej w drzewie
  - 6.2. Wróć do pkt. 4
7. Jeżeli `textToken`
  - 7.1. Utwórz element tekstowy w drzewie
  - 7.2. Wróć do pkt. 4
8. Jeżeli brak
  - 8.1. Zakończ budowę drzewa

### 3. Tworzenie pliku docelowego

1. Przechodź równolegle (w głąb) po kolejnych elementach obydwu zbudowanych drzew rozbioru na podstawie podanych na wejściu plików HTML
  - 1.1. Jeżeli elementy są takie same
    - 1.1.1. Utwórz taki element w drzewie docelowym
  - 1.2. Jeżeli znaczniki są takie same a atrybuty nie
    - 1.2.1. Rozwiąż konflikt atrybutów wg tabeli konfliktów
  - 1.3. Jeżeli znaczniki są różne
    - 1.3.1. Rozwiąż konflikt wg tabeli konfliktów
    - 1.3.2. Jeżeli konflikt wymaga decyzji użytkownika
      - 1.3.2.1. Wyświetl okno do decyzji jak rozwiązać konflikt
2. Analizuj wewnętrzne elementy w rozważanym elemencie wg algorytmu pkt. 1
3. Jeżeli brak elementów wewnętrznych
  - 3.1. Przejdź do następnego elementu i wykonuj algorytm wg pkt. 1
  - 3.2. Jeżeli brak następnego elementu to koniec

## V. Testy

Przeprowadzone zostały przypadki testowe pokazujące wykorzystanie rozstrzygnięć wszystkich wyspecyfikowanych konfliktów, a także testy pokazujące poprawne odwzorowanie dokumentów wczytanych na wejściu z plikiem wynikowym w przypadku scalania 1:1. Testowane były także niepoprawne konstrukcje plików wejściowych celem wykrycia poprawnej obsługi sytuacji wyjątkowych.

### A. Wynik uruchomienia przykładowego testu

Wynikiem przeprowadzenia testu programu jest plik wynikowy oraz informacja o użytych przypadkach rozwiązywania konfliktów<sup>(projekt wstępny pkt. VI.A.2)</sup> podczas dokonywania złączenia podanych plików, bądź informacja o błędzie. W przypadku pozytywnego zakończenia działania programu zostaje również wyświetlony ślad kolejno użytych przypadków z tabeli konfliktów<sup>(pkt. IV.A.1.b)</sup> jako ciąg numerów danych konstrukcji.

Przykładowy wynik działania programu (wydruk z konsoli):

```
COMPLETED, conflict counter:
Conflict no. 1 : 1 times
Conflict no. 2 : 18 times
Conflict no. 6 : 18 times
Conflict no. 7 : 4 times
Conflict no. 9 : 1 times
Conflict no. 10 : 1 times
Conflict no. 11 : 13 times
Conflict no. 12 : 2 times

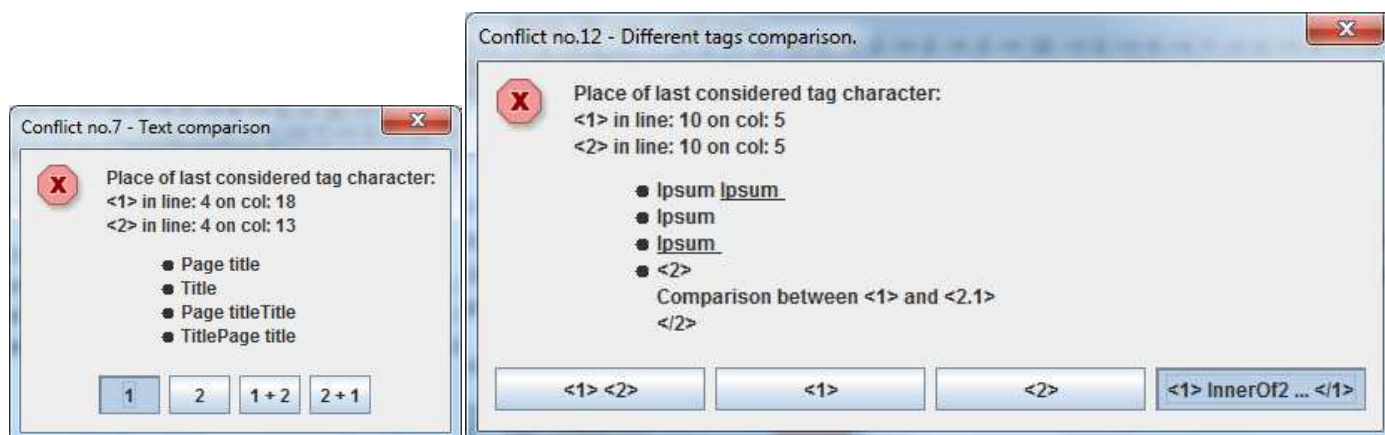
Trace of used conflicts:
1 -> 6 -> 2 -> 6 -> 2 -> 6 -> 2 -> 7 -> 11 -> 6 -> 2 -> 6 -> 2 -> 6 -> 2 -> 12 -> 6 -> 2 -> 7 -> 6 ->
2 -> 11 -> 11 -> 11 -> 11 -> 11 -> 11 -> 11 -> 11 -> 11 -> 6 -> 2 -> 7 -> 6 -> 2 -> 10 -> 6 ->
2 -> 6 -> 2 -> 12 -> 6 -> 2 -> 6 -> 2 -> 7 -> 9 -> 6 -> 2 -> 6 -> 2 -> 6 -> 2 -> 6 -> 2 -> 11 -> 11 ->
>
```

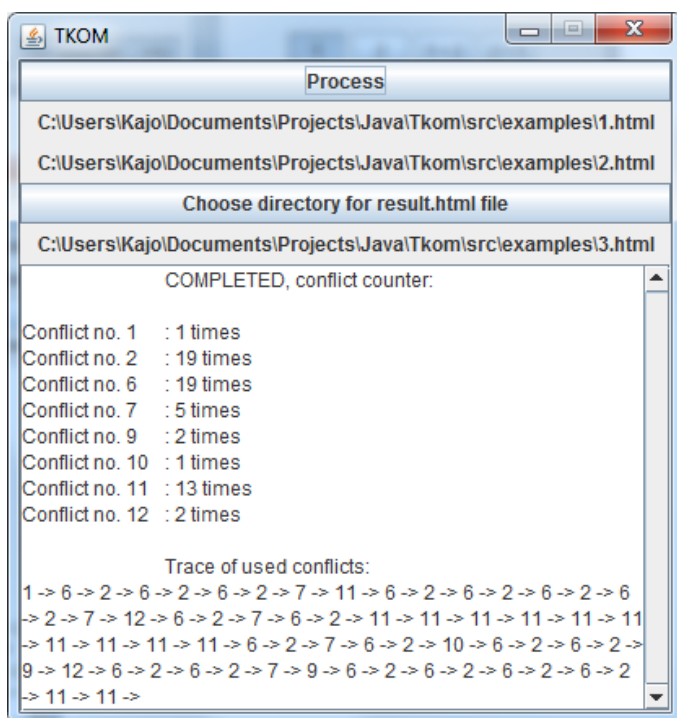
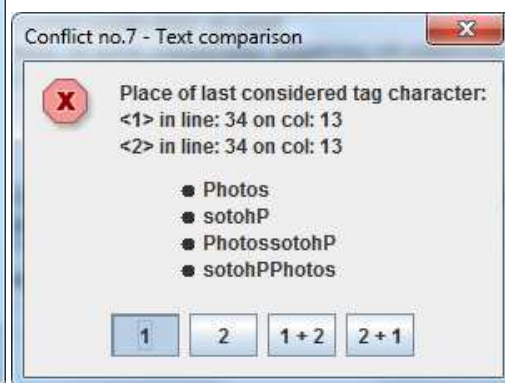
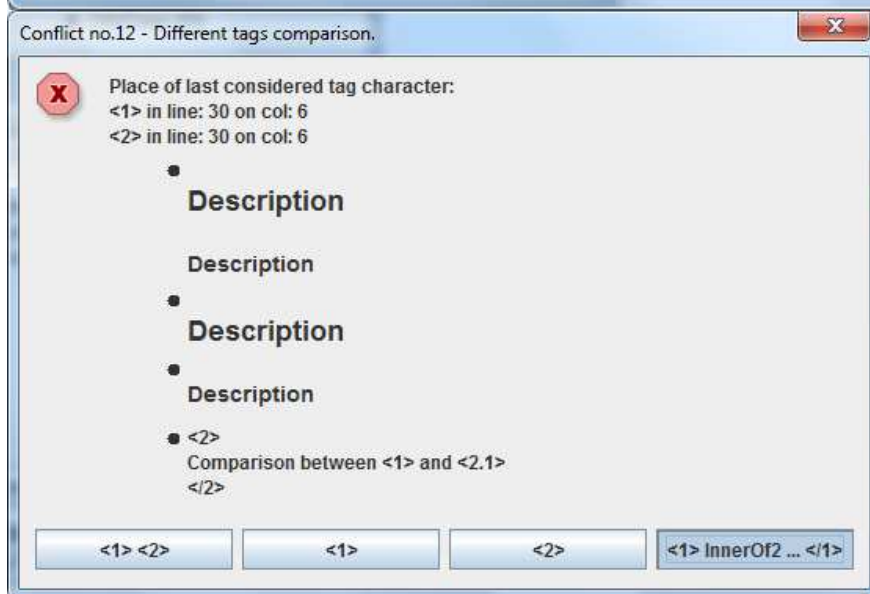
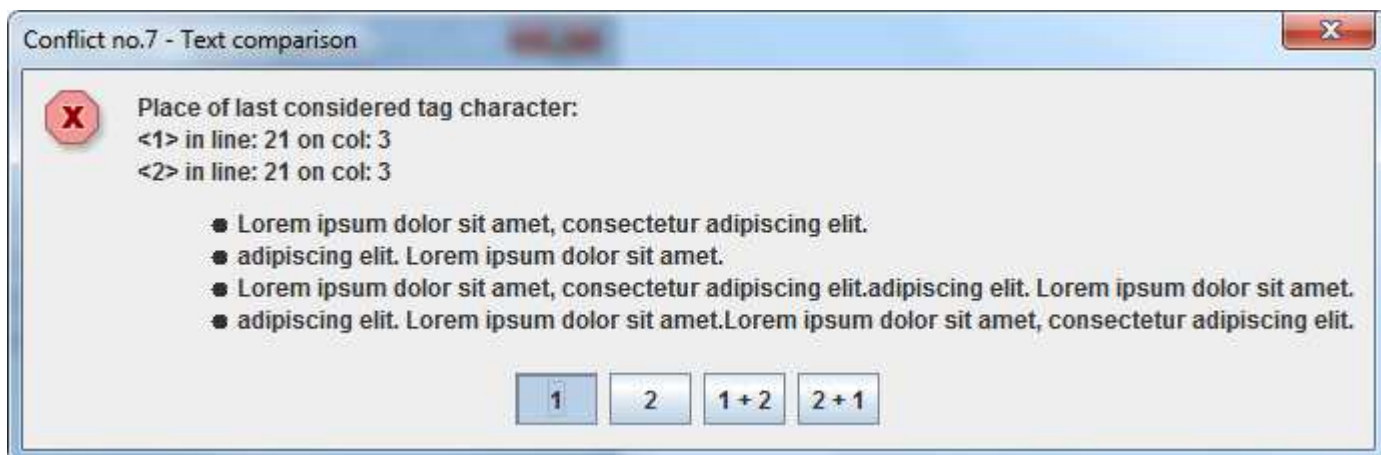
### B. Testy rozstrzygające konflikty

#### 1. Test wszystkich konfliktów przykładu<sup>(projekt wstępny pkt. VII.A)</sup>

Do testu użyte zostały pliki przedstawione na następnych stronach (pkt. a, pkt. b), oraz zamieszczony został kod wynikowy (pkt. c). W trakcie rozstrzygania konfliktów została użyta pierwsza z możliwych opcji oferowanych do wyboru bądź analiza wewnętrzna<sup>(projekt wstępny pkt. VI.A.2.12.4)</sup> w przypadku konfliktu nr 12.

Sekwencja wyborów:





Prosta konsola wynikowa pokazuje użyte w trakcie analizy konflikty, co oznacza że każdy z nich został użyty choć raz.

Wartym uwagi jest fakt, iż konflikt nr 6 implikuje swoim zachowaniem wywołanie konfliktu nr 2 co jest zachowaniem spodziewanym i w pełni poprawnym.

Wynik testu przedstawiony został na następnych stronach. Dla celów łatwiejszej analizy:

**a) Pierwszy plik**

```
1  <!doctype html>
2  <html>
3  <head>
4      <title>Page title</title>
5
6  </head>
7  <body>
8      <section class=nav style="color: #111;">
9          <article>
10             <b>Ipsum</b>
11             <br> - Lorem ipsum dolor sit amet. <br> -
12             Lorem ipsum dolor sit amet. <br> - Lorem ipsum dolor sit amet. <br>
13             Lorem ipsum dolor sit amet, consectetur adipiscing elit. <br />
14             <br /> Lorem ipsum dolor sit amet, consectetur adipiscing elit.
15             <hr />
16             Lorem ipsum dolor sit amet, consectetur adipiscing elit.
17          </article>
18
19          <article>
20             Lorem ipsum dolor sit amet, consectetur adipiscing elit.
21             <p>
22                 <u>Ttx</u>
23
24             </p>
25          </article>
26      </section>
27
28      <section class=cont>
29          <article>
30             <h2>Description</h2>
31          </article>
32
33          <article>
34             <h2>Photos</h2>
35             <span>(Click to do)</span>
36             <ul class=phot>
37                 <li><a href="#" target="_blank"> 
39                 </a></li>
40
41             </ul>
42          </article>
43      </section>
44
45
46
47
48
49  </body>
50  </html>
```

**b)     *Drugi plik***

```
1  <!doctype html>
2  <html lang=pl>
3  <head>
4      <title>Title</title>
5      <meta charset=utf-8 />
6  </head>
7  <body style="background-color: #CDE">
8      <section class=nav>
9          <article>
10             <u><b>Ipsum</b></u>
11             <br> - Lorem ipsum dolor sit amet. <br>
12
13
14
15
16
17         </article>
18
19         <article style="color: #00F;">
20             adipiscing elit. Lorem ipsum dolor sit amet.
21             <p>
22                 Olaboga
23
24             </p>
25         </article>
26     </section>
27
28     <section class=cont>
29         <article>
30             <h3>Description</h3>
31         </article>
32
33         <article>
34             <h2>sotohP</h2>
35             (Click to do)
36             <ul class=phot>
37                 <li><a href="#" target="_self"> 
39                 </a></li>
40                 <li>Txt</li>
41             </ul>
42         </article>
43     </section>
44
45     <div>
46         123.345 <i>.00</i>
47     </div>
48
49 </body>
50 </html>
```

**c) Plik wynikowy (pozbawiony wynikowego formatowania z wcięciami)**

```
1 <!doctype html>
2 <html lang="pl">
3 <head>
4     <title>Page title</title>
5     <meta charset="utf-8" />
6 </head>
7 <body style="background-color: #CDE">
8     <section class="nav" style="color: #111;">
9         <article>
10             <u><b>Ipsum</b></u>
11             <br />- Lorem ipsum dolor sit amet.<br />-
12             Lorem ipsum dolor sit amet.<br />- Lorem ipsum dolor sit amet.<br />
13             Lorem ipsum dolor sit amet, consectetur adipiscing elit.<br />
14             <br />Lorem ipsum dolor sit amet, consectetur adipiscing elit.
15             <hr />
16             Lorem ipsum dolor sit amet, consectetur adipiscing elit.
17         </article>
18
19         <article style="color: #00F;">
20             Lorem ipsum dolor sit amet, consectetur adipiscing elit.
21             <p>
22                 <u>Ttx</u>
23                 Olaboga
24             </p>
25         </article>
26     </section>
27
28     <section class="cont">
29         <article>
30             <h3><h2>Description</h2></h3>
31         </article>
32
33         <article>
34             <h2>Photos</h2>
35             <span>Click to do</span>
36             <ul class="phot">
37                 <li><a href="#" target="_self">
38                     
39                 </a></li>
40                 <li>Txt</li>
41             </ul>
42         </article>
43     </section>
44
45     <div>
46         123.345<i>.00</i>
47     </div>
48
49 </body>
50 </html>
```

## C. Testy obsługi sytuacji wyjątkowych

Ze względu na specyfikę przypadku testowego, na wejście podawany jest znacząco skrócony i zniekształcony (*ang. malformed*) dokument HTML.

### 1. Niedozwolony znacznik

```
<!doctype html>
<html>
<head>
  <title></title>
</head>
<body>
<znk>Ough</znk>
</body>
</html>
```

```
ERROR occurred, message:
No html tag defined: <znk>
Line: 7 Column: 4.
On global position: 72 (+-10 wrapped
into __ characters):
__ody>
<znk>Ough</znk>
```

### 2. Błędna kolejność domykanych znaczników

```
<!doctype html>
<html>
<head>
  <title></title>
</head>
<body>
<u><i></u></i>
</body>
</html>
```

```
ERROR occurred, message:
Bad stack order, u expected.
Line: 7 Column: 10.
On global position: 78 (+-10 wrapped
into __ characters):
__<u><i></u></i>
</bo__
```

### 3. Brak typu dokumentu

```
<html>
<head>
  <title></title>
</head>
<body>
</body>
</html>
```

```
ERROR occurred, message:
No doctype found!
Line: 1 Column: 2.
On global position: 2 (+-10 wrapped
into __ characters):
__<html>
<hea__
```

## D. Testy odwzorowania 1:1 (scalenia identycznych plików źródłowych)

Test ten ma na celu sprawdzenie działania analizatora, tj. czy ciąg wejściowy jest wczytany w całości, oraz czy generowany dokument jest poprawnie odwzorowany. W teście użyty został plik oraz jego dokładna kopia. Oczekiwany wynikiem, który został spełniony w trakcie tego rodzaju testowania, jest brak użycia konfliktów z numerami od 9 do 12. Użyte powinno być tylko rozstrzygnięcie konfliktów o numerach: 2, 6, 7 w dowolnych ilościach, konfliktu nr 1 w ilości równej 1 oraz ilość użytych konfliktów nr 2 i 6 powinna być równa<sup>(pkt. V.B.1)</sup>. Wynikiem testu jest plik HTML możliwie bardzo zbliżony do kodu pliku wejściowego.

Do testu zostały użyte kody źródłowe stron (*ich publicznie wygenerowana forma*) prawdziwych stron internetowych. Z uwagi na dosyć duże rozmiary analizowanego kodu, zostały przedstawione zrzuty ekranu stron pokazujące odwzorowanie strony z dokładnością do niektórych białych znaków<sup>(pkt. II.A)</sup> oraz aktualnie pobranych danych poprzez użycie skryptów na stronach (*głównie reklamy*).

Na kolejnych stronach pokazane zostaną pary wyświetlonych stron internetowych, odpowiednio: zapisanej na dysku i wygenerowanej po przebiegu programu. Strony zostały tak dobrane żeby można było gołym okiem zobaczyć odwzorowanie, gdyż niektóre strony posiadają względne odwołania do niektórych źródeł.

Na zrzutach ekranu wskazane zostały niektóre ww. punkty różniące się co do białych znaków.



onet.pl

Onet Wiadomości Sport Biznes Rozrywka Plotki Gry Tech Moto Styl życia Wiedza **Randki** zaloguj się

onet. enhanced by Google Szukaj

Oceń nową stronę Onet.pl NOWE

Sympatia Allegro Zumi Gry online VOD Sympatia+ Poczta

Samsung SMART CAMERA

NAJEDŹ I ZŁAP BOLTA

Dzisiaj Pogoda | Horoskop | Program TV | Kina | Kursy walut | Giełda online | Wyniki Lotto  
Schudnij Niedziela 19 maja, Augustyna i Piotra

Warszawa teraz 19°C jutro 24°C

Wiadomości

"Przyznałem, że mój syn jest gejem. Zdziwiły mnie reakcje ludzi"

- Rząd wypowiada wojnę "mafii kolnierzyków"
- Leszek Miller oskarża Kaczyńskiego o kłamstwo
- "Tusk ucieka do Brukseli. Rozwalił to państwo"
- Śmiertelny wypadek na trasie Warszawa-Lublin
- Płonący krzyż i erupcja wulkanu. Zobacz zdjęcia
- Koszmar w klasztorze, męczyl ją przez 20 lat
- Motor uderzył w tramwaj z prędkością 200 km/h
- Niespodziewany skutek operacji Angeliny Jolie
- Strach mieszkańców. Bóbr zabił wędkarza
- Dramat na peronie, wózek z dzieckiem na torach
- "Jackowski przechodził przez szalony okres"
- Jak zdobyć dofinansowanie dla firmy? REKLAMA

Rozwiń więcej

O tym się mówi

**To gorsze niż prostytutka! Kim są te kobiety?**  
Wystarczy wpisać w sieci parę słów, by odnaleźć katalogi żon na sprzedaż. Czasami cała wioska składa się na przekupienie urzędników, żeby przyszła panna młoda mogła lecieć do Europy

pl.wikipedia.org

## Strona główna

Z Wikipedii, wolnej encyklopedii  
Skocz do: [nawigacji](#), [wyszukiwania](#)

## Witaj w Wikipedii

wolnej encyklopedii, którą każdy może redagować.

Obecnie mamy **967 784** hasła, w tym **2205** wyróżnionych

Poznaj nasze zasady • [Testuj wiki!](#)

[Nauki ścisłe i przyrodnicze](#) • [Nauki społeczne i humanistyczne](#) • [Technika](#) • [Geografia](#) • [Historia](#) • [Kultura](#) • [Społeczeństwo](#) • [Sport](#) • [Religia](#) • [Polska](#)

[Wiecej portali](#) • [Indeks hasel](#) • [Przegląd kategorii](#)

Właśnie trwa [Tydzień z Eurovizją](#). [Dołącz do akcji!](#)

## Czy wiesz...

Z [nowych artykułów](#) w Wikipedii:

[Arctic Star](#)

...że [muszkatela miedziana](#) została po raz pierwszy opisana przez [Linneusza](#) w 1766 roku?

...za jaki czyn został skazany na śmierć św. [Rodrigo Aguilar Alemán](#)?

...[które brytyjskie](#) odznaczenie za udział w walkach [II wojny światowej](#) zostało ustanowione dopiero w 2012 roku?

...ile [pomników przyrody](#) znajduje się na terenie [gminy Goldap](#)?

## Artykuł na medal

**ORP Bałtyk**, pierwotnie **D'Entrecasteaux** – francuski [krajownik pancernopokładowy](#) z początku XX wieku, następnie w okresie międzywojennym polski [hulak szkolny](#). Krajownik „D'Entrecasteaux”, noszący imię na cześć francuskiego [XVIII-wiecznego żeglarsza](#), był jedynym zbudowanym okrętem swojego typu i unikatową jednostką we [francuskiej marynarce wojennej](#) – dużym [krajownikiem I klasy](#) skonstruowanym specjalnie do służby w koloniach na Dalekim Wschodzie. W czasie służby, między innymi, brał udział w tłumieniu [powstania bokserów](#) w Chinach. Podczas [I wojny światowej](#) służył aktywnie na Morzu Śródziemnym, odznaczając się zwłaszcza w obronie [Kanału Sueskiego](#) przed Turkami w 1915 roku. Po wojnie został w 1922 roku wycofany z czynnej służby i przekształcony w hulak bez napędu. W tym charakterze był przejściowo wypożyczony Belgii, a następnie w 1927 roku został zakupiony przez polską Marynarkę Wojenną, jako hulak szkolny. Był pierwszym polskim okrętem noszącym nazwę **ORP „Bałtyk”** i największym okrętem, jaki kiedykolwiek służył w **PMW**. Po wybuchu II wojny światowej został przejęty przez Niemców i po krótkim okresie wykorzystywania, złomowany do 1942 roku. [Czytaj więcej...](#)

[Inne artykuły na medal](#) • [Jak wybieramy najlepsze?](#)

## Dobry artykuł



onet.pl

Onet Wiadomości Sport Biznes Rozrywka Plotki Gry Tech Moto Styl życia Wiedza **Randki** zaloguj się

Oceń nową stronę Onet.pl

**onet.** enhanced by Google Szukaj

Sympatia Allegro Zumi Gry online VOD Sympatia+ Poczta

**Samsung SMART CAMERA**

**NAJEDŹ I ZŁAP BOLTA**

**Dzisiaj** Pogoda | Horoskop | Program TV | Kina | Kursy walut | Giełda online | Wyniki Lotto  
Schudnij Niedziela 19 maja, Augustyna i Piotra

**Warszawa** teraz 19°C jutro 24°C

**Wiadomości**

"Przyznałem, że mój syn jest gejem. Zdziwiły mnie reakcje ludzi"

- Rząd wypowiada wojnę "mafii kołnierzyków"
- Leszek Miller oskarża Kaczyńskiego o kłamstwo
- "Tusk ucieka do Brukseli. Rozwalił to państwo"
- Śmiertelny wypadek na trasie Warszawa-Lublin
- Płonący krzyż i erupcja wulkanu. Zobacz zdjęcia
- Koszmar w klasztorze, mężczyś ją przez 20 lat
- Motor uderzył w tramwaj z prędkością 200 km/h
- Niespodziewany skutek operacji Angeliny Jolie
- Strach mieszkańców. Bóbr zabił wędkarza
- Dramat na peronie, wózek z dzieckiem na torach
- "Jackowski przechodził przez szalony okres"
- Dotacje unijne dla firm w 2013! REKLAMA

Rozwiń więcej

O tym się mówi

**To gorsze niż prostytutka! Kim są te kobiety?**  
Wystarczy wpisać w sieci parę słów, by odnaleźć katalogi żon na sprzedaż. Czasami cała wioska składa się na przekupienie urzędników, żeby przyszła panna młoda mogła lecieć do Europy

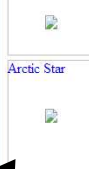
pl.wikipedia.org**Strona główna**Z Wikipedii, wolnej encyklopedii  
Skocz do: [nawigacji](#), [wyszukiwania](#)**Witaj w Wikipedii.**

wolnej encyklopedii, którą każdy może redagować.

Obecnie mamy **967 784** hasła, w tym **2205** wyróżnionych.Poznaj nasze zasady. • [Testuj wiki!](#)

[Nauki ścisłe i przyrodnicze](#) • [Nauki społeczne i humanistyczne](#) • [Technika](#) • [Geografia](#) • [Historia](#) • [Kultura](#)  
[Społeczeństwo](#) • [Sport](#) • [Religia](#) • [Polska](#)

[Więcej portali](#) • [Indeks hasła](#) • [Przegląd kategorii](#)

Właśnie trwa [Tydzień z Eurowizją](#). [Dołącz do akcji!](#)**Czy wiesz...**Z [nowych artykułów](#) w Wikipedii:...że [muszkatela miedziana](#) została po raz pierwszy opisana przez [Linneusza](#) w 1766 roku?...za jaki czyn został skazany na śmierć św. [Rodrigo Aguilar Alemán](#)?...[które brytyjskie](#) odznaczenie za udział w walkach [II wojny światowej](#) zostało ustanowione dopiero w 2012 roku?...ile [pomników przyrody](#) znajduje się na terenie [gminy Goldap](#)?**Artykuł na medal**

**ORP Bałtyk**, pierwotnie **D'Entrecasteaux** – francuski [krajownik pancernopokładowy](#) z początku XX wieku, następnie w okresie międzywojennym polski [hulak szkolny](#). Krajownik „D'Entrecasteaux”, noszący imię na cześć francuskiego [XVIII-wiecznego żeglarsza](#), był jedynym zbudowanym okrętem swojego typu i unikatową jednostką we [francuskiej marynarce wojennej](#) – dużym [krajownikiem](#) I klasy skonstruowanym specjalnie do służby w koloniach na Dalekim Wschodzie. W czasie służby, między innymi, brał udział w tłumieniu [powstania bokserów](#) w Chinach. Podczas [I wojny światowej](#) służył aktywnie na Morzu Śródziemnym, odznaczając się zwłaszcza w obronie [Kanału Sueskiego](#) przed Turkami w 1915 roku. Po wojnie został w 1922 roku wycofany z czynnej służby i przekształcony w hulak bez napędu. W tym charakterze był przejęciowo wypożyczony Belgii, a następnie w 1927 roku został zakupiony przez polską Marynarkę Wojenną, jako hulak szkolny. Był pierwszym polskim okrętem noszącym nazwę **ORP „Bałtyk”** i największym okrętem, jaki kiedykolwiek służył w [PMW](#). Po wybuchu II wojny światowej został przejęty przez Niemców i po krótkim okresie wykorzystywania, złomowany do 1942 roku. [Czytaj więcej...](#)

[Bałtyk d.jpg](#)[Imię artykułu na medal](#) • [Jak wybieramy najlepsze?](#)**Dobry artykuł**