

WYDZIAŁ PODSTAWOWYCH PROBLEMÓW TECHNIKI  
POLITECHNIKA WROCŁAWSKA

# IMPLEMENTACJA ABSTRAKCYJNEJ MASZINY WARRENA

KAJETAN BILSKI

NR INDEKSU: 244942

Praca inżynierska napisana  
pod kierunkiem  
Przemysława Kobyłańskiego



Politechnika  
Wrocławska

WROCŁAW 2020



# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>1</b>
<b>2</b>	<b>Analiza problemu</b>	<b>3</b>
<b>3</b>	<b>Projekt systemu</b>	<b>5</b>
3.1	Grupy użytkowników i założenia . . . . .	5
3.2	Przypadki użycia i scenariusze . . . . .	5
3.3	Diagramy klas . . . . .	5
3.4	Diagramy aktywności . . . . .	5
3.5	Diagramy sekwencji . . . . .	5
3.6	Diagramy stanów . . . . .	6
3.7	Projekt bazy danych . . . . .	6
3.8	Opis protokołów . . . . .	6
3.9	Opis algorytmów . . . . .	7
<b>4</b>	<b>Implementacja systemu</b>	<b>9</b>
4.1	Opis technologii . . . . .	9
4.2	Omówienie kodów źródłowych . . . . .	9
<b>5</b>	<b>Instalacja i wdrożenie</b>	<b>11</b>
<b>6</b>	<b>Podsumowanie</b>	<b>13</b>
	<b>Bibliografia</b>	<b>15</b>
<b>A</b>	<b>Zawartość płyty CD</b>	<b>17</b>



# Wstęp

Celem pracy jest zaimplementowanie abstrakcyjnej maszyny Warrena dla czystego języka Prolog. Odpowiada on językowi trzeciego poziomu z opisu maszyny. Praca zawierać będzie, poza opisem implementacji, analizę efektywności jej działania na wybranych przykładach programów.



# Analiza problemu

W tym rozdziale należy przedstawić analizę zagadnienia, które podlega informatyzacji. Należy zidentyfikować i opisać obiekty składowe rozważanego wycinka rzeczywistości i ich wzajemne relacje (np. użytkowników systemu i ich role). Należy szczegółowo omówić procesy jakie zachodzą w systemie i które będą informatyzowane, takie jak np. przepływ dokumentów. Należy sprecyzować i wypunktować założenia funkcjonalne i poza funkcjonalne dla projektowanego systemu. Jeśli istnieją aplikacje realizujące dowolny podzbiór zadanych funkcjonalności realizowanego systemu należy przeprowadzić ich analizę porównawczą, wskazując na różnice bądź innowacyjne elementy, które projektowany w pracy system informatyczny będzie zawierał. Należy odnieść się do uwarunkowań prawnych związanych z procesami przetwarzania danych w projektowanym systemie. Jeśli zachodzi konieczność, należy wprowadzić i omówić model matematyczny elementów systemu na odpowiednim poziomie abstrakcji.

W niniejszym rozdziale omówiono koncepcję architektury programowej systemu .... W szczególny sposób .... Omówiono założenia funkcjonalne i нефункционалне podsystemów .... Przedstawiono mechanizmy .... Sklasyfikowano systemy ze względu na .... Omówiono istniejące rozwiązania informatyczne o podobnej funkcjonalności ... (zobacz [2]).





# Projekt systemu

W tym rozdziale przedstawiono szczegółowy projekt systemu w notacji UML uwzględniający wymagania funkcjonalne opisane w rozdziale 2. Do opisu relacji pomiędzy składowymi systemu wykorzystano diagramy .... Przedstawiono w pseudokodzie i omówiono algorytmy generowania ....

## 3.1 Grupy użytkowników i założenia

Architektura systemu ... jest wielowarstwowa i rozproszona, przy czym .... Podsystem ... jest systemem zbiorczym dla danych ... wysyłanych do serwera ....

Taka architektura jest zgodna z wzorcem projektowym MVC<sup>1</sup> (ang. Model-View-Controller). Przetwarzanie danych odbywa się ....

## 3.2 Przypadki użycia i scenariusze

W tej sekcji należy przedstawić przypadki użycia oraz odpowiadające im scenariusze dla poszczególnych grup użytkowników ....

## 3.3 Diagramy klas

W tej sekcji należy przedstawić diagramy klas dla odpowiednich elementów systemu zidentyfikowane na podstawie wcześniejszych rozważań

## 3.4 Diagramy aktywności

W tej sekcji należy przedstawić diagramy aktywności dla elementów systemu i odpowiednich procesów wynikające z wcześniejszej analizy.

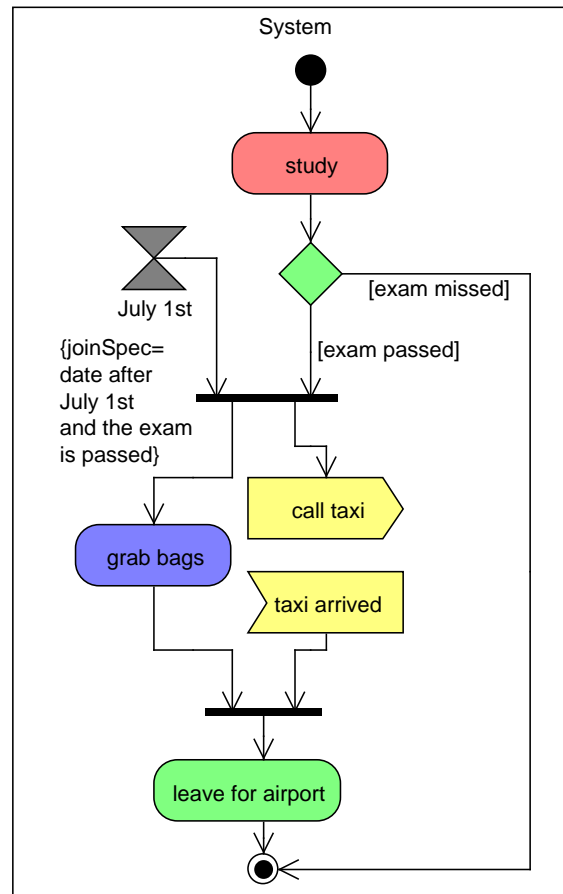
W niniejszym rozdziale przedstawiono diagramy aktywności .... Diagram na rysunku 3.1 przedstawia ....

## 3.5 Diagramy sekwencji

W tej sekcji należy przedstawić diagramy sekwencji dla obiektów systemu zidentyfikowanych na podstawie wcześniejszych rozważań. Należy wykorzystać nazewnictwo wprowadzone w poprzednich rozdziałach, w szczególności odpowiadające definicjom wprowadzonych klas.

---

<sup>1</sup>Należy odnieść się do wykorzystywanych wzorców projektowych



Rysunek 3.1: Diagram aktywności związany z procesem rejestracji dokumentu.

## 3.6 Diagramy stanów

W tej sekcji należy przedstawić diagramy stanów w których może znaleźć się system. Diagramy te są szczególnie istotne przy projektowaniu systemów czasu rzeczywistego.

## 3.7 Projekt bazy danych

W tej sekcji należy przedstawić projekt bazy danych. Należy omówić wycinek rzeczywistości i odpowiadające mu zidentyfikowane elementy systemu, których wartości będą podlegać utrwalaniu. Należy przedyskutować wybór typów danych dla atrybutów poszczególnych obiektów. Należy uzasadnić wybór platformy DBMS. Dla relacyjnych baz danych należy przedyskutować jej normalizację.

## 3.8 Opis protokołów

W tej sekcji należy omówić protokoły wykorzystywane przez komponenty systemu. Omówić formaty komunikatów i zilustrować je przykładami.

### 3.9 Opis algorytmów

W tej sekcji należy wymienić i przedyskutować algorytmy wykorzystywane w systemie. Algorytmy należy przedstawić w pseudokodzie (wykorzystać pakiet `algorithm2e`). Omówienia poszczególnych kroków algorytmów powinny zawierać odwołania do odpowiednich linii pseudokodu. Dla zaproponowanych autorskich algorytmów należy przeprowadzić analizę ich złożoności czasowej i pamięciowej.

Algorytm bąblowania jest przedstawiony w Pseudokodzie 3.1.

---

**Pseudokod 3.1:** Wyporność przez bąblowanie

---

**Input:** Zbiór bąbli  $B$

**Output:** Wyporność  $W$

```
1 foreach  $b \in B$  do
2   Process( $b$ );
3   for  $i \leftarrow 1$  to  $|B|$  do
4     if Calculate( $EW(i, b)$ )  $\leq 0$  then
5        $b \leftarrow 2 * b$ ;
6 while  $B \neq \emptyset$  do
7   for  $j \leftarrow 1$  to  $|B|$  do
8     if Calculate( $FT(j, \hat{b})$ )  $\leq 0$  then
9        $w \leftarrow 2 * \hat{b}$ ;
10       $W \leftarrow W \cup \{w\}$ ;
11       $B \leftarrow B \setminus \{b\}$ ;
```

---



# Implementacja systemu

## 4.1 Opis technologii

Należy tutaj zamieścić krótki opis (z referencjami) do technologii użytych przy implementacji systemu.

Do implementacji systemu użyto języka JAVA w wersji . . . , szczegółowy opis można znaleźć w [1]. Interfejs zaprojektowano w oparciu o HTML5 i CSS3 [3].

## 4.2 Omówienie kodów źródłowych

Kod źródłowy 4.1 przedstawia opisy poszczególnych metod interfejsu: WSPodmiotRejestracjaIF. Kompletne kody źródłowe znajdują się na płycie CD dołączonej do niniejszej pracy w katalogu Kody (patrz Dodatek A).

Kod źródłowy 4.1: Interfejs usługi Web Service: WSPodmiotRejestracjaIF.

---

```
package erejestracja.podmiot;
import java.rmi.RemoteException;
// Interfejs web serwisu dotyczącego obsługi podmiotów i rejestracji.
public interface WSPodmiotRejestracjaIF extends java.rmi.Remote{
// Pokazuje informacje o danym podmiocie.
// parametr: nrPeselRegon – numer PESEL podmiotu lub numer REGON firmy.
// return: Podmiot – obiekt transportowy: informacje o danym podmiocie.
public Podmiot pokazPodmiot(long nrPeselRegon) throws RemoteException;
// Dodaje nowy podmiot.
// parametr: nowyPodmiot – obiekt transportowy: informacje o nowym podmiocie.
// return: true – jeśli podmiot dodano, false – jeśli nie dodano.
public boolean dodajPodmiot(Podmiot nowyPodmiot) throws RemoteException;
// Usuwa dany podmiot.
// parametr: nrPeselRegon – numer PESEL osoby fizycznej lub numer REGON firmy.
// return: true – jeśli podmiot usunięto, false – jeśli nie usunięto.
public boolean usunPodmiot(long nrPeselRegon) throws RemoteException;
// Modyfikuje dany podmiot.
// parametr: podmiot – obiekt transportowy: informacje o modyfikowanym podmiocie.
// return: true – jeśli podmiot zmodyfikowano, false – jeśli nie zmodyfikowano.
public boolean modyfikujPodmiot(Podmiot podmiot) throws RemoteException;
// Pokazuje zarejestrowane podmioty na dany dowód rejestracyjny.
// parametr: nrDowoduRejestracyjnego – numer dowodu rejestracyjnego.
// return: PodmiotRejestracja[] – tablica obiektów transportowych: informacje o
// wszystkich zarejestrowanych podmiotach.
public PodmiotRejestracja[] pokazZarejestrowanePodmioty(
String nrDowoduRejestracyjnego) throws RemoteException;
// Nowa rejestracja podmiotu na dany dowód rejestracyjny.
// parametr: nrDowoduRejestracyjnego – numer dowodu rejestracyjnego.
// parametr: nrPeselRegon – numer PESEL podmiotu lub numer REGON firmy.
// parametr: czyWlasciciel – czy dany podmiot jest właścicielem pojazdu.
// return: true – jeśli zarejestrowano podmiot, false – jeśli nie zarejestrowano.
public boolean zarejestrujNowyPodmiot(String nrDowoduRejestracyjnego,
```



```

long nrPeselRegon, boolean czyWlasciciel) throws RemoteException;
// Usuwa wiązanie pomiędzy danym podmiotem, a dowodem rejestracyjnym.
// parametr: nrDowoduRejestracyjnego – numer dowodu rejestracyjnego.
// parametr: nrPeselRegon – numer PESEL podmiotu lub numer REGON firmy.
// return: true – jeśli podmiot wyrejestrowano, false – jeśli nie wyrejestrowano.
public boolean wyrejestrujPodmiot(String nrDowoduRejestracyjnego,
long nrPeselRegon) throws RemoteException;

```

Kod źródłowy 4.2 przedstawia procedurę przetwarzającą żądanie. Hasz utrwalany %granulacja wykorzystywany jest do komunikacji międzyprocesowej.

Kod źródłowy 4.2: Przetwarzanie żądania - procedura `process_req()`.

```

sub process_req() {
    my($r) = @_;
    $wyn = "";
    if ($r =~ /get/i) {
        @request = split("_", $r);
        $zad = $request[0];
        $ts1 = $request[1];
        $ts2 = $request[2];
        @date1 = split("/D/", $ts1);
        @date2 = split("/D/", $ts2);
        print "odebralem: _$r";
        $wyn = $wyn."zadanie: _$zad\n";
        $wyn = $wyn."czas_od: _"."$date1[0]". "-" . "$date1[1]". "-" . "$date1[2]". "-" . "$date1[3]". ":" . "$date1[4]";
        $wyn = $wyn."czas_do: _"."$date2[0]". "-" . "$date2[1]". "-" . "$date2[2]". "-" . "$date2[3]". ":" . "$date2[4]";
        $wyn = $wyn.&sym_sens($ts1, $ts2);
        return $wyn;
    }
    if ($r =~ /set gt/i) {
        @request = split("_", $r);
        $zad = $request[0];
        $ts1 = $request[1];
        $ts2 = $request[2];
        $gt = $request[3];
        dbmopen(%granulacja, "granulacja_baza", 0644);
        $granulacja{"gt"} = $gt;
        dbmclose(%granulacja);
        $wyn = "'GT' _zmienione_na: _$gt";
    }
}

```

# Instalacja i wdrożenie

W tym rozdziale należy omówić zawartość pakietu instalacyjnego oraz założenia co do środowiska, w którym realizowany system będzie instalowany. Należy przedstawić procedurę instalacji i wdrożenia systemu. Czynności instalacyjne powinny być szczegółowo rozpisane na kroki. Procedura wdrożenia powinna obejmować konfigurację platformy sprzętowej, OS (np. konfiguracje niezbędnych sterowników) oraz konfigurację wdrażanego systemu, m.in. tworzenia niezbędnych kont użytkowników. Procedura instalacji powinna prowadzić od stanu, w którym nie są zainstalowane żadne składniki systemu, do stanu w którym system jest gotowy do pracy i oczekuje na akcje typowego użytkownika.





# Podsumowanie

W podsumowanie należy określić stan zakończonych prac projektowych i implementacyjnych. Zaznaczyć, które z zakładanych funkcjonalności systemu udało się zrealizować. Omówić aspekty pielęgnacji systemu w środowisku wdrożeniowym. Wskazać dalsze możliwe kierunki rozwoju systemu, np. dodawanie nowych komponentów realizujących nowe funkcje.

W podsumowaniu należy podkreślić nowatorskie rozwiązania zastosowane w projekcie i implementacji (niebanalne algorytmy, nowe technologie, itp.).



# Bibliografia

- [1] Java technology. Web pages: <http://www.oracle.com/technetwork/java/>.
- [2] J. Cichoń, M. Klonowski, Łukasz Krzywiecki, B. Różański, P. Zieliński. On the number of nodes and their distribution in chord. 2006.
- [3] B. Frain. *Responsive Web Design with HTML5 and CSS3*. Packt Publishing, 2012.



# Zawartość płyty CD

W tym rozdziale należy krótko omówić zawartość dołączonej płyty CD.

