



**Politechnika Świętokrzyska w Kielcach**  
**Wydział Elektroniki, Automatyki i Informatyki**

PROJEKT PROGRAMOWANIE W JĘZYKU JAVA

TEMAT PROJEKTU:

Aplikacja do prowadzenia oraz zarządzania komunikacją miejską/bus

Rok: Drugi  
Semestr: Czwarty  
Grupa: **2ID15A**  
Kamil Świąder  
Paweł Pawelec

Data oddania:  
**27.06.2017**

Ocena:

## 1. TEMAT PROJEKTU:

Tematem naszego projektu jest **Aplikacja do zarządzania komunikacją miejską/busami**. Aplikacja tego typu może zostać rozwinięta na wiele różnych wariantów, jej celem podstawowym powinno być zarządzanie pojazdami, ich trasami etc.

Głównym założeniem naszego projektu było spełnienie tych funkcjonalności jak również dodanie innych. Przy projektowaniu aplikacji zdecydowaliśmy się na wariant z busami. Dodatkowo chcieliśmy zaimplementować nie tylko samo administrowanie komunikacją jak również dodać możliwość dokonywania transakcji (tj. kupno biletów), wydruk paragonu, potwierdzenie zakupu, jak również chcieliśmy zapewnić np. Możliwość sprawdzenia danemu kierowcy jego rozkładu jazdy. W tym celu musieliśmy zaimplementować różne typy okien obsługi zależnie od tego czy pracownik był Administratorem, Kierowcom czy Sprzedawcą. Cele, które sobie postawiliśmy zostały w pełni zrealizowane.

## 2. REALIZACJA PROJEKTU

### 2.1. Język programowania, środowisko, biblioteki:

Nasz projekt został całościowo napisany w języku Java zgodnie z paradygmatem obiektowym stosując elementy związane z tym paradygmatem programowania.

Projekt został napisany w środowisku **Eclipse** jako projekt typu **Maven** zgodnie z wymaganiami projektowymi. Środowisko pracy ułatwiło nam podstawowe prace nad projektem tj.: podpowiadanie i podświetlanie składni, generowanie konstruktorów, getterów, seterów czy import potrzebnych bibliotek.

W projekcie głównie korzystaliśmy z bibliotek:

**Swing** – biblioteka graficzna stosowana do tworzenia aplikacji z graficznym interfejsem użytkownika. Jest ona nowszą, ulepszoną wersją biblioteki AWT. Stosując bibliotekę Swinga w naszym projekcie użyliśmy jej komponenty oraz większość funkcjonalności jaką oferuje.

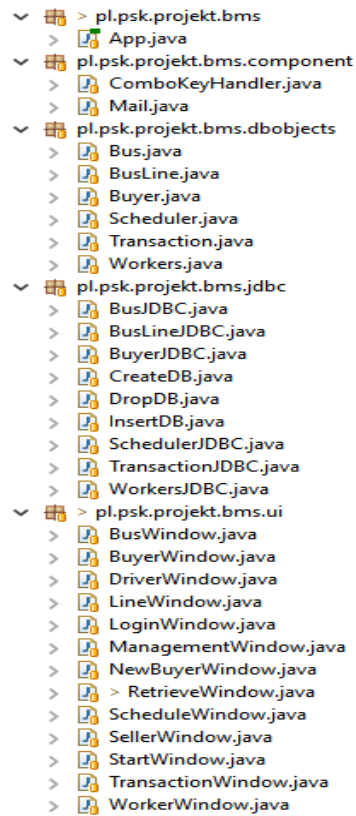
**AWT** – Abstract Window Toolkit, jest to pakiet zawierający niezależny od platformy systemowej zestaw klas do projektowania aplikacji w środowisku graficznym. Wykorzystywana między innymi do rysowania grafik 2D oraz obsługi myszy.

**SQL** - udostępnia interfejs API do uzyskiwania i przetwarzania danych przechowywanych w źródle danych (zazwyczaj relacyjnej bazy danych) przy użyciu języka programowania.

## 2.2. Struktura projektu:

Projekt został podzielony na 5 pakiety:

W wrzuconym poniżej screenie ukazane są klasy z różnych pakietów:



```

- pl.psk.projekt.bms
  - App.java
- pl.psk.projekt.bms.component
  - ComboKeyHandler.java
  - Mail.java
- pl.psk.projekt.bms.dbobjects
  - Bus.java
  - BusLine.java
  - Buyer.java
  - Scheduler.java
  - Transaction.java
  - Workers.java
- pl.psk.projekt.bms.jdbc
  - BusJDBC.java
  - BusLineJDBC.java
  - BuyerJDBC.java
  - CreateDB.java
  - DropDB.java
  - InsertDB.java
  - SchedulerJDBC.java
  - TransactionJDBC.java
  - WorkersJDBC.java
- pl.psk.projekt.bms.ui
  - BusWindow.java
  - BuyerWindow.java
  - DriverWindow.java
  - LineWindow.java
  - LoginWindow.java
  - ManagementWindow.java
  - NewBuyerWindow.java
  - RetrieveWindow.java
  - ScheduleWindow.java
  - SellerWindow.java
  - StartWindow.java
  - TransactionWindow.java
  - WorkerWindow.java

```

W pakiecie zakończonym na **bms** znajduje się klasa odpowiedzialna za rozruch.

W pakiecie zakończonym na **bms.dbobject** znajdują się definicje klas, które są przetożeniem zawartości tabel z bazy danych.

W pakiecie zakończonym na **bms.jdbc** znajduje się obsługa połączeń z bazą i operacji na niej jak dodawanie czy usuwanie tabel, rekordów etc.

W pakiecie **bms.ui** znajdują się wizualne części aplikacji naszego projektu wraz z metodami pomagającymi w ich obsłudze.

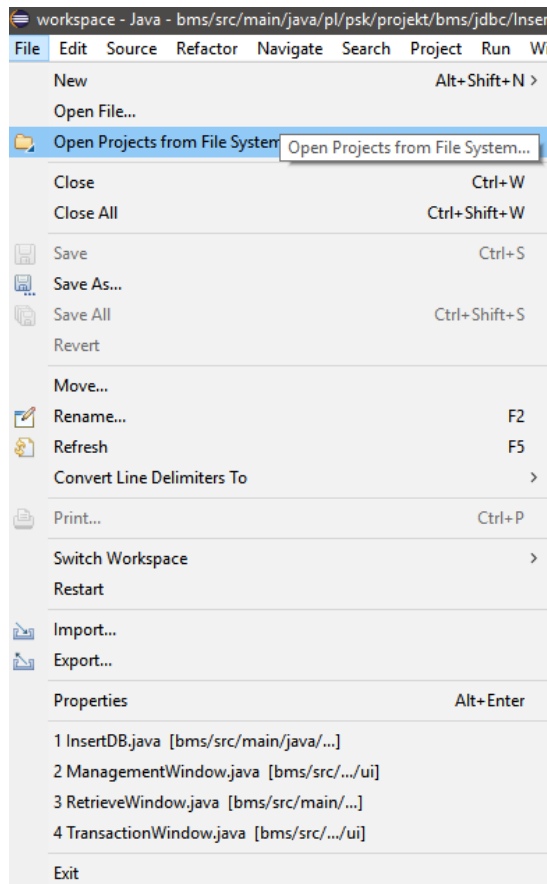
W pakiecie **bms.component** znajdują się dwie klasy:

- \* klasa **Mail** implementuje wysyłkę maili i sms-ów
- \* klasa **ComboKeyHandler** znajdują się metody do dynamicznego przeszukiwania tabel oraz okien comboBox ułatwiając prace w transakcjach, edycji etc.

### 3. IMPORTOWANIE ORAZ KOMPILACJA PROJEKTU

Projekt został napisany w IDE – Eclipse, lecz dzięki użyciu mechanizmu **Maven** można go w łatwy sposób zaimportować do innych środowisk.

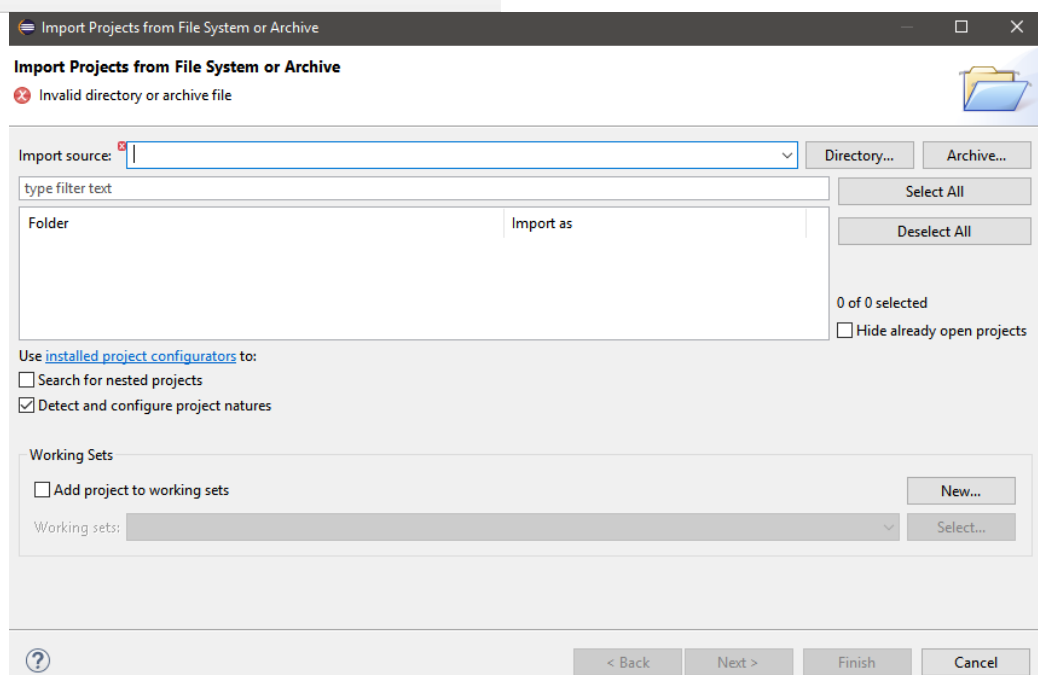
#### 3.1. Kompilacja projektu w Eclipse:



Należy uruchomić środowisko Eclipse. Następnie klikamy w lewym górnym rogu na przycisk „File” -> „Open Project from File System”:

Po wybraniu tej opcji ukaże nam się ekran z prośbą o wskazanie ścieżki docelowej do folderu z projektem lub archiwum. W przypadku pobrania projektu ze zdalnego repozytorium kodu jako archiwum należy wybrać opcję „Archive”, jeśli archiwum zostanie wypakowane należy wybrać opcję „Directory” wskazując ścieżkę docelową do folderu. Finalizując należy kliknąć na przycisk „Finish” i projekt zostanie poprawnie otwarty.

W celu jego skompilowania należy skorzystać z przycisku „Run”, który jest dostępny w pasku narzędzi lub klikając prawym przyciskiem na folder projektu i wybranie z menu kontekstowego Run Us → maven build z ustawionymi w polu goals: clean install -e, a następnie po zbudowaniu projektu użycie Run us → Java Application.



## 4. OPIS PROJEKTU

Projekt umożliwia zarządzanie komunikacją busów.

### 4.1. Pierwsze uruchomienie.

Przy uruchomieniu pojawi nam się okno powitalne z informacją, po wciśnięciu przycisku **"Move Forward"** ukazuje się okno Logowania. Przy pierwszym uruchomieniu wymagane jest wciśnięcie przycisku **"Create DataBase"**, aby utworzyć bazę, tabelę i użytkownika **admin** z hasłem **admin**. Wymagane, aby aplikacja działała poprawnie jest np. **Xampp** ze skonfigurowanym i uruchomionym serwerem **Apache** i **MySQL** – musi działać **na porcie 3306**, a **user** do bazy powinien być ustawiony jako **root** a **hasło** do niej jako **toor**. Po tym działaniu możemy wypełnić okna logowania username : **admin** i password : **admin** i wcisnąć przycisk Login.

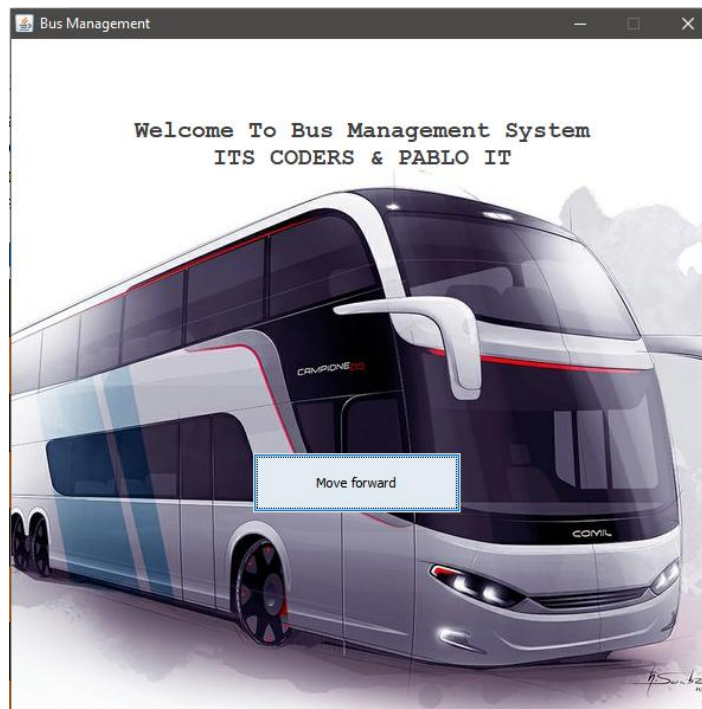
Na ekranie załaduje się okno administratora do zarządzania gdzie mamy dostępne przyciski Worker, Scheduler, Buyer itd. Jednak jeżeli nie chcemy wszystkiego wpisywać sami a testujemy proponujemy użyć przycisku **"Database Insert"**, który pomoże nam przetestować aplikację, po wciśnięciu tego przycisku załadowane zostaną do bazy dane do tabel Workers, Bus, BusLine, Schedule, puste natomiast zostaną Buyer i Transaction. Możemy teraz wchodzić w każdy z przycisków w celu dodawania, edycji itd.

W oknie można wylogować się przez Logout i zalogować na inny rodzaj konta:

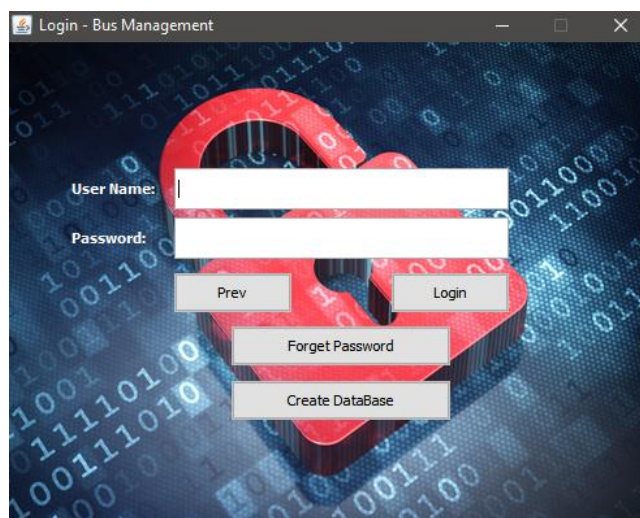
Kierowca: np. username: **pjan** password **pj**, Sprzedawca: username: **mjanuszek** password: **mj**

### 4.2. Wygląd poszczególnych okien aplikacji:

#### 1. Okno startowe: StartWindow

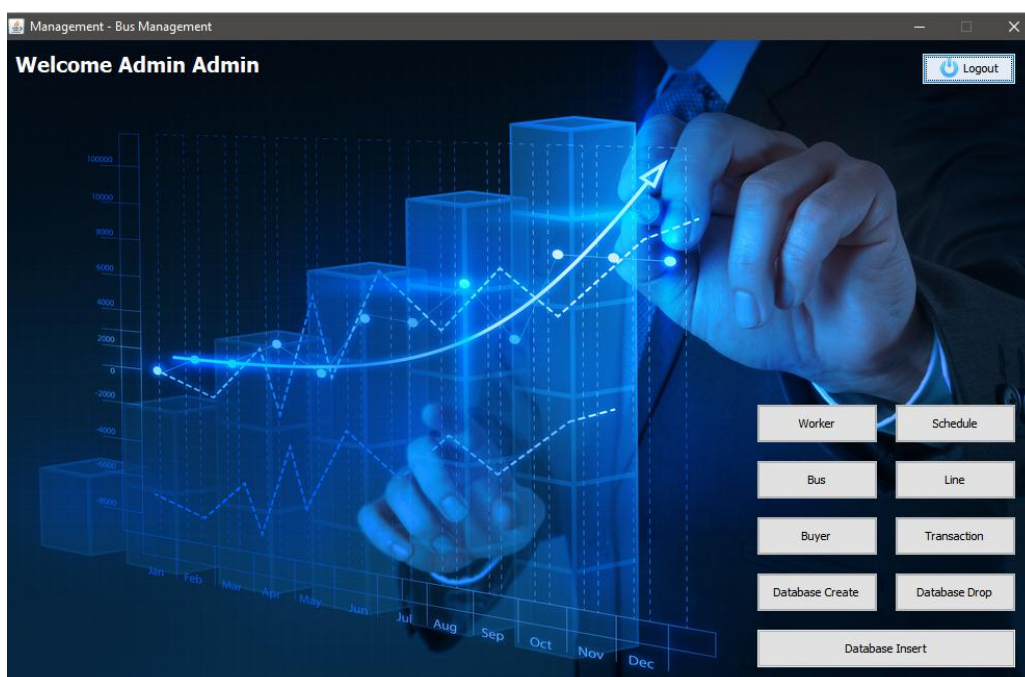


## 2. Okno Logowanie: LoginWindow



The Login window, titled "Login - Bus Management", features a dark blue background with a pattern of binary code (0s and 1s). A large, 3D red heart shape is centered on the screen. Overlaid on the heart are two white input fields for "User Name:" and "Password:". Below these fields are four buttons: "Prev", "Login", "Forget Password", and "Create DataBase".

## 3. Okno Administratora MenagmentWindow:



#### 4. Okno Kierowcy DriverWindow:



Driver Scheduler System

Welcome Pawel Jan

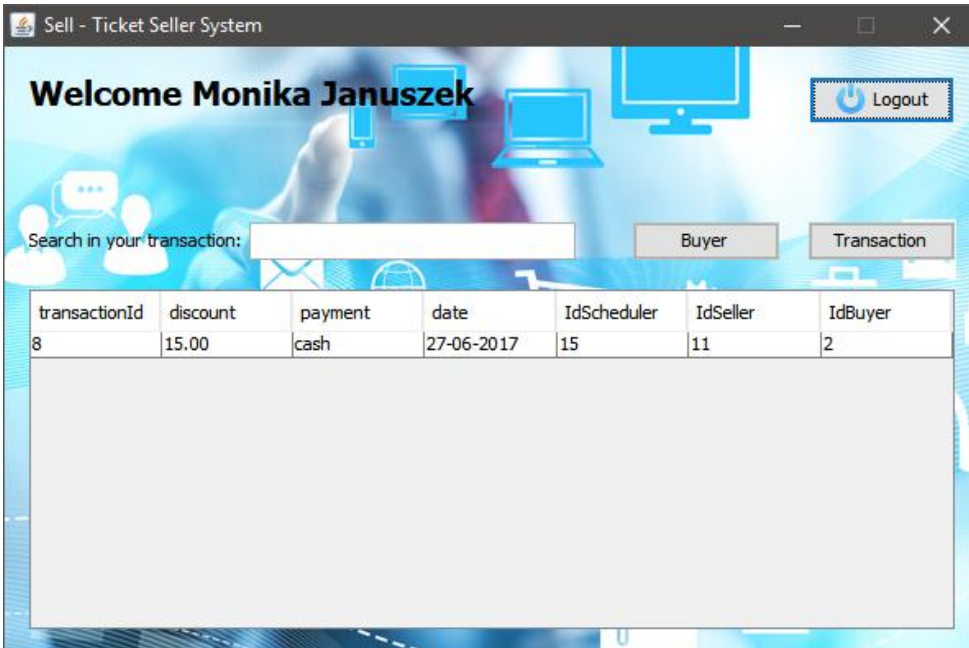
Logout

Search in your schedule:

Print Your Scheduler

schedulerID	departureTime	arrivalTime	IdBus	IdBusLine	IdDriver
1	05:00:00	07:00:00	1	2	4
2	07:15:00	09:15:00	1	1	4
3	09:30:00	11:30:00	1	2	4
4	11:45:00	13:45:00	1	1	4
5	14:00:00	16:00:00	1	2	4
6	16:15:00	18:15:00	1	1	4
7	18:30:00	20:30:00	1	2	4
8	20:45:00	22:45:00	1	1	4

#### 5. Okno Sprzedawcy SellerWindow



Sell - Ticket Seller System

Welcome Monika Januszek

Logout

Search in your transaction:

Buyer

Transaction


transactionId	discount	payment	date	IdScheduler	IdSeller	IdBuyer
8	15.00	cash	27-06-2017	15	11	2



## 6. Okno zaeządzania Kupującym BuyerWindow

**Buyer - Bus Management**

Name:  Surname:

Birthday:   Email:

Mobile Phone:  Street:

Apt/House Number:  Post Code:

City:  Insurance Number:

Search Buyer:


buyerId	name	surname	birthday	email	mobile	street	houseN...	postCode	city	insuranc..
1	Kamil	Swiader	01-Jun-1...	kamil.swi...	698741526	ddsfdsfds	87	25669	sadadas	258774
2	Janusz	Swiader	01-Jun-1...	j.s@m.pl	554455555	ddsfdsfds	87	25669	sadadas	988741
3	Pawel	Pawelc	12-Jun-1...	pablito15...	987423651	ddsfdsfds	87	25669	sadadas	555445


## 7. Okno zaeządzania Pracownikami WorkerWindow

**Worker - Bus Management**

User Name:  Password:

Name:  Surname:

Type:   Email:

Birthday:   Mobile Phone:

Adress:  Salary:

Search Worker:

workerId	accou...	userna...	passw...	name	surname	email	mobile	address	birthday	salary
1	Administ...	admin	admin	Admin	Admin	admin@...	000000...	Siedziba	16-Jun-...	25000.0
2	Administ...	pjanuszek	pj	Pawel	Januszek	pjanusz...	895412...	Warsza...	16-Jun-...	25000.0
3	Administ...	zjanuszek	zj	Zygfryd	Januszek	zjanusz...	897892...	Warsza...	20-Jun-...	25000.0
4	Driver	pjan	pj	Pawel	Jan	pjan@g...	895356...	Radom, ...	26-Jun-...	5000.0
5	kjura	Driver	kj	Krzysztof	Jura	kjuras@...	898512...	Kielce, u...	29-Jun-...	5000.0
6	Driver	zjuras	zj	Zygmunt	Juras	zjuras@...	815634...	Kielce, u...	26-Jun-...	5000.0
7	Driver	mslusar...	ms	Michal	Slusarczyk	mslusar...	741547...	Pepice 21	28-Jun-...	5000.0



## 5. PODSUMOWANIE

### 5.1. Realizacja celów projektu:

Celem projektu było stworzenie aplikacji umożliwiającej administrowanie komunikacją miejską / busami w języku Java. Zdecydowaliśmy się na bibliotekę graficzną Swing celem napisania interfejsu graficznego aplikacji. Ponadto naszym celem było wykorzystanie w projekcie bazy danych, co udało się zrealizować. Baza, którą wybraliśmy to MySQL(MariaDB). Mechanizm wątków wykorzystaliśmy by równocześnie można było pracować w kilku oknach jak worker, scheduler etc. Ponadto projekt mieliśmy zrealizować w środowiskach tj.: Eclipse, NetBeans czy IntelliJ, my wybraliśmy Eclipse, kolejnym punktem był wybór pomiędzy narzędziem Maven i Gradle, my posłużyliśmy się Mavenem. Wymagania pozwoliły nam zaznajomić się z usługą zdalnego repozytorium kodu, z podanych wybraliśmy usługę GitHub. Dokumentacja została wykonana narzędziem JavaDoc zgodnie z wymogiem. Projekt został napisany przez cały zespół, a my mieliśmy równomierny wkład w projekt. Wszystkie postawione cele zostały zrealizowane.



### 5.2. Kierunki rozwoju

Pierwszym kierunkiem rozwoju mogła by być adaptacja aplikacji do nowszej biblioteki graficznej JavaFX.

Napisanie aplikacji dla klientów którzy mogliby dokonywać sami zakupów, sprawdzeń rozkładów etc.

Aplikacja mogła by zostać rozwinięta do wersji sieciowej webserwisu. Wykorzystanie dodatkowych technologii.

Kolejnym krokiem mogłoby być zastosowanie mechanizmów śledzenia pojazdów, obsługi płatności etc.

## 6. BIBLIOGRAFIA I ŹRÓDŁA

- Bibliografia:

Cay S.Horstmann G. Cornell, Java Podstawy, Wyd. IX, Gliwice, Wydawnictwo Helion, 2014.

Cay S.Horstmann G. Cornell, Techniki zaawansowane. ,Wyd. IX, Gliwice, Wydawnictwo Helion, 2013.

Cay S.Horstmann, Java 8. Przewodnik doświadczonego programisty, Gliwice, Wydawnictwo Helion, 2015.

- Źródła

<http://docs.oracle.com/javase/8/docs/api/index.html>

<https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html>

<https://stackoverflow.com/questions/tagged/java>