****

Politechnika Świętokrzyska

Wydział Elektrotechniki, Automatyki i Informatyki

**Filip Tosnowiec, Szymon Śmiglarski, Mariusz Śmistek**

Rozwiązywanie Problemu Komiwojażera

**Projekt zespołowy**

studia stacjonarne

kierunek Informatyka

Opiekun projektu:

Dr inż. Ludomir Tuszyński

Kielce, 2022

**SPIS TREŚCI**

[1. Charakterystyka zadania projektowego 3](#_Toc96528962)

[2. Podstawa teoretyczna 3](#_Toc96528963)

[2.1. Opis matematyczny problemu 3](#_Toc96528964)

[2.2. Rozwiązanie problemu matematycznego 3](#_Toc96528965)

[3. Algorytm obliczeniowy 3](#_Toc96528966)

[4. Opis aplikacji 3](#_Toc96528967)

[5. Podsumowanie i wnioski 4](#_Toc96528968)

[6. Instrukcja obsługi aplikacji 4](#_Toc96528969)

[7. Literatura 4](#_Toc96528970)

# Charakterystyka zadania projektowego

System podejmuje rozwiązywania problemu Komiwojażera tj. szukanie najkrótszej ścieżki miedzy punktami. Pozwala na zapoznanie się z opisem rozwiązania, obliczenie wyniku dla własnych lub wygenerowanych danych.

**Słownik** – definicja wszystkich technicznych i specyficznych terminów użytych w dokumencie.

• **Problem Komiwojażera –** problem optymalizacyjny polegający na znalezieniu najkrótszej drogi pomiędzy n miejscami, która odwiedzi każde miejsce dokładnie raz i wróci do miejsca startowego

**• Graf Pełny –** graf, w którym każda para wierzchołków jest połączona krawędzią

**• Macierz sąsiedztwa –** Sposób reprezentacji grafu za pomocą dwuwymiarowej tablicy, w której w komórce (i,j) umieszczana jest waga krawędzi między wierzchołkami *i* a *j* (lub 0, jeśli takiej krawędzi nie ma)

**• Lista sąsiedztwa –** Sposób reprezentacji grafu za pomocą tablicy, w której każdy wierzchołek ma przypisaną listę sąsiadów

**• Algorytm najbliższego sąsiada –** heurystyczny algorytm rozwiązujący problem komiwojażera

**• Algorytm 2-opt -** Algorytm lokalnego przeszukiwania rozwiązania w problemie komiwojażera

# Podstawa teoretyczna

## Opis matematyczny problemu

Problem komiwojażera polega na znalezieniu najkrótszej drogi, która odwiedzi każde z n punktów i wróci do punktu startowego. Może to być przedstawione za pomocą grafu, gdzie każdy punkt jest wierzchołkiem, a krawędzie łączą punkty, między którymi jest bezpośrednie połączenie. Dla każdej krawędzi określona jest odległość, którą trzeba pokonać, aby przejść z jednego punktu do drugiego. Problem komiwojażera można zatem opisać za pomocą grafu pełnego, w którym każdy wierzchołek łączy się z każdym innym wierzchołkiem, i macierzy kosztów, która zawiera odległości między każdą parą wierzchołków.

## Rozwiązanie problemu matematycznego

Nie istnieje znany algorytm rozwiązujący problem komiwojażera w czasie wielomianowym dla każdego przypadku. W związku z tym, wiele różnych algorytmów jest stosowanych w celu znalezienia przybliżonego rozwiązania.

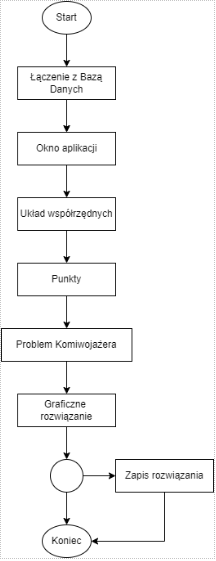
Jednym z popularnych algorytmów jest algorytm zachłanny, w którym wybierana jest najkrótsza krawędź z bieżącego wierzchołka do jakiegokolwiek innego nieodwiedzonego wierzchołka. Następnie wybierany jest kolejny nieodwiedzony wierzchołek, który jest najbliższy bieżącemu wierzchołkowi, i tak dalej, aż wszystkie wierzchołki zostaną odwiedzone. Ostatecznie zwraca się cykl Hamiltona, który stanowi przybliżone rozwiązanie problemu.

Innym popularnym algorytmem jest algorytm symulowanego wyżarzania, który naśladuje proces wyżarzania metalu. Rozpoczyna się od losowej permutacji wierzchołków, a następnie cykle są zamieniane miejscami w sposób losowy. Jeśli nowy cykl jest krótszy niż obecny cykl, to jest on zastępowany. Jeśli nowy cykl jest dłuższy, to jest on zastępowany z pewnym prawdopodobieństwem, które zmniejsza się wraz z czasem. Algorytm kończy się, gdy osiągnięty zostanie ustalony czas lub nie będzie poprawy przez pewien czas.

Innym podejściem jest programowanie liniowe, które polega na zapisaniu problemu jako funkcji celu i ograniczeń, które muszą być spełnione. Rozwiązanie problemu polega na znalezieniu wartości funkcji celu, która minimalizuje koszty podróży, spełniając jednocześnie wszystkie ograniczenia. Metoda ta jest dokładna, ale może być niepraktyczna dla dużych instancji problemu.

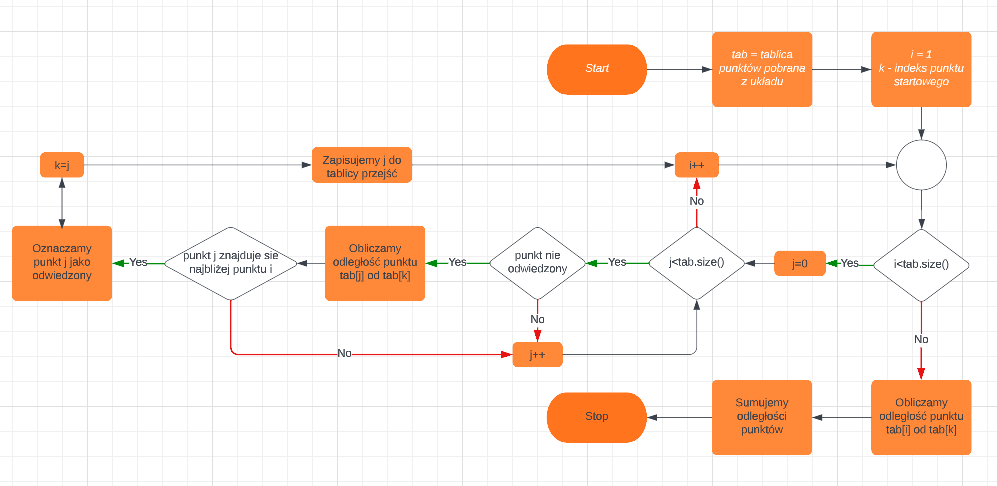
# STRUKTURA APLIKACJI, Algorytm obliczeniowy

## Moduły aplikacji



* Łączenie z Bazą Danych - nawiązanie połączenia z Bazą Danych za pomocą JDBC.
* Okno aplikacji - wyświetlanie wszystkich przycisków okna, nasłuchiwanie na akcję użytkownika. Odpowiada za wywołanie odpowiednich funkcji programu po działaniu użytkownika.
* Układ współrzędnych - wyświetlanie układu współrzędnych, punktów i ścieżek w oknie aplikacji. Pobiera pozycję kursora i zamienia ją na odpowiednie współrzędne X Y, jednocześnie dodając nowy punkt do układu.
* Punkty - pobieranie, generowanie, wczytywanie i przechowywanie współrzędnych punktów.
* Problem Komiwojażera - używa danych na temat punktów  w celu wyznaczenia najkrótszej ścieżki zaczynając od początkowego punktu. Oblicza ścieżkę przejść przez punkty szukając najkrótszej z nich. Powtarza tą  operację dla każdego z punktów i łączy wyniki każdej z nich tworząc jedną ścieżkę będącą rozwiązaniem problemu.
* Graficzne rozwiązanie - odpowiada za wyświetlanie ścieżek między punktami na układzie współrzędnych zgodnie z rozwiązaniem.
* Zapis rozwiązania - umożliwia zapis ścieżki i punktów do bazy danych w celu wczytania ich w przyszłości.

## Algorytm obliczeniowy



# Opis aplikacji

4.1. Analiza systemowa:

a. kontekst systemu,

* Kolorystyka:

1. Kolor układu współrzędnych – szary
2. Kolor punktów – niebieski
3. Kolor wybranego punktu – czerwony
4. Kolor startowego punktu – róż
5. Kolor ścieżki - zielony

Granice działania oprogramowania:

Zakres dodatniej wielkości układu współrzędnych: 5-15

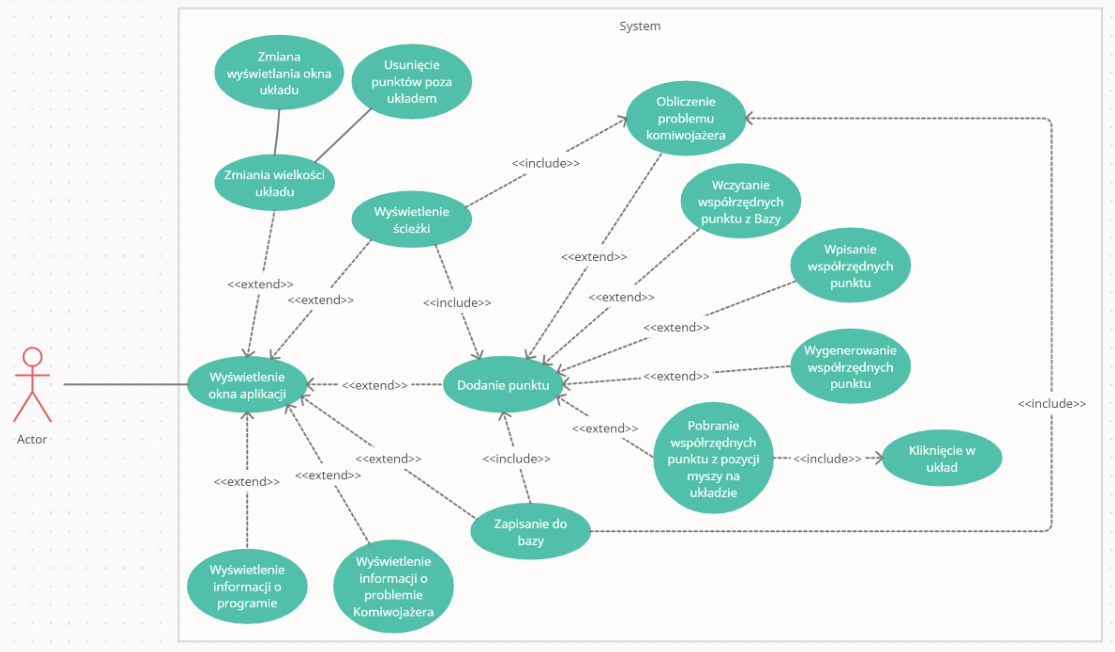
**Wymagania funkcjonalne**

* Dodawanie punktów za pomocą wpisania współrzędnych
* opis – użytkownik dodaje punkt i ustawia mu dokładne współrzędne
* wejście – liczby zmiennoprzecinkowe w zakresie wielkości układu współrzędnych z skokiem 0.5
* wyjście – powstanie punktu o określonych współrzędnych
* efekty uboczne – wyświetlenie punktu na układzie, dodanie go do listy punktów, uwzględnienie go w opracowywaniu rozwiązania
* Dodawanie punktów za pomocą kliknięcia w układ współrzędnych
* opis – użytkownik dodaje punkt z współrzędnymi pobranymi z pozycji myszy względem układu współrzędnych
* wejście – pozycja kursora
* wyjście – powstanie punktu o przeliczonych współrzędnych
* efekty uboczne – wyświetlenie punktu na układzie, dodanie go do listy punktów, uwzględnienie go w opracowywaniu rozwiązania
* Generowanie punktów
* opis – użytkownik dodaje punkt z współrzędnymi wygenerowanymi przez program w przedziale wielkości układu współrzędnych
* wejście – ilość punktów i zakres.
* wyjście – powstanie punktu o wygenerowanych współrzędnych
* efekty uboczne – wyświetlenie punktu na układzie, dodanie go do listy punktów, uwzględnienie go w opracowywaniu rozwiązania
* Wyznaczenie ścieżki – Rozwiązanie problemu Komiwojażera
* opis – wywołanie algorytmu rozwiązującego problem Komiwojażera na zdefiniowanych punktach układu współrzędnych.
* wejście – punkty układu współrzędnych.
* wyjście – wyznaczenie kolejności odwiedzin punktów
* efekty uboczne – wyświetlenie ścieżki na układzie, obliczenie długości trasy.
* Wyświetlenie informacji na temat działania programu
* opis – Wyświetlenie informacji na temat działania programu
* wyjście – wyświetlenie panelu zawierającego informacje na temat działania programu
* Wczytanie danych z Bazy
* opis – pobranie współrzędnych punktów z wcześniej zapisanego układu
* wejście – wcześniej zapisane punkty układu współrzędnych
* wyjście – dodanie punktów z pobranymi danymi
* efekty uboczne – wyświetlenie punktu na układzie, dodanie go do listy punktów, uwzględnienie go w opracowywaniu rozwiązania
* Zapis danych do Bazy
* opis – zapis punktów układu współrzędnych i ścieżki do bazy danych
* wejście – wygenerowana ścieżka i punkty w obecnie obsługiwanym układzie
* wyjście – implementacja danych do bazy danych
* Zmiana wielkości układu
* opis – zmiana zakresu wyświetlania układu współrzędnych
* wejście – definicja wielkości układu współrzędny
* wyjście – zmiana zakresu układu współrzędnych
* efekty uboczne – zmiana skali układu, usunięcie punktów poza zakresem

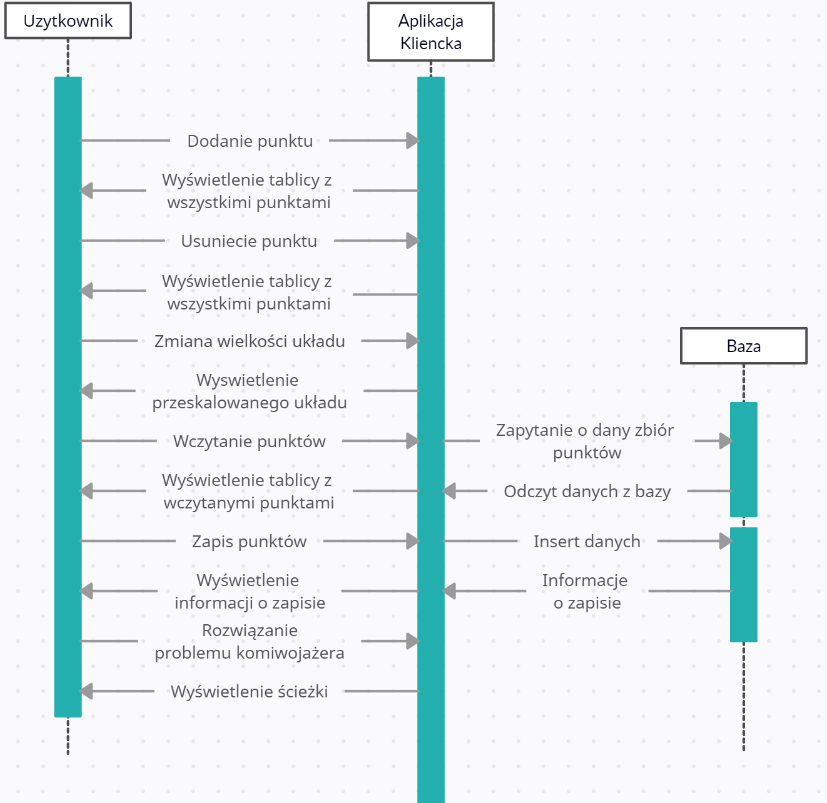
**Podział prac**:

* Filip Tosnowiec – struktura bazy danych, zapytania do bazy, algorytm komiwojażera, dokumentacja
* Szymon Śmiglarski – wygląd okna, pobieranie danych od użytkownika
* Mariusz Śmistek – Rest Api, rysowanie ścieżki na ekranie

b. scenariusze przypadków użycia,



c. diagram przypadków użycia, ew. diagram sekwencji (UML).



4.3. Projekt architektury:

a. wybór technologii,

* System Windows
* Java

Java to obiektowy język programowania, który został stworzony przez firmę Sun Microsystems (teraz należącą do Oracle) w latach 90. XX wieku. Jego głównym celem było zapewnienie platformy, która pozwoli programistom na pisanie aplikacji, które będą działać na różnych systemach operacyjnych i platformach sprzętowych.

Język Java jest wykorzystywany w wielu dziedzinach, takich jak tworzenie aplikacji webowych, mobilnych, desktopowych, a także w programowaniu systemów wbudowanych. Java charakteryzuje się bezpieczeństwem, niezależnością platformową, skalowalnością, łatwością nauki i obsługi oraz wydajnością.

W Javie programowanie jest prowadzone w oparciu o obiekty, czyli elementy, które reprezentują konkretne rzeczy lub koncepcje, takie jak użytkownik, plik lub połączenie sieciowe. Każdy obiekt ma swoje własne właściwości i zachowanie, które są definiowane przez jego klasę. Programista tworzy klasy, które definiują typy obiektów, a następnie tworzy instancje tych klas, czyli konkretnych obiektów, na podstawie których program jest tworzony.

Ważnym aspektem Javy jest także jej wirtualna maszyna (JVM - Java Virtual Machine), która pozwala na uruchomienie kodu Javy na różnych platformach sprzętowych bez konieczności modyfikacji kodu. Dzięki temu, program napisany w Javie jest przenośny i może być uruchomiony na dowolnej platformie, która ma zainstalowaną JVM.

* JDBC

JDBC (Java Database Connectivity) to interfejs programowania aplikacji (API), który umożliwia komunikację między językiem Java a różnymi systemami zarządzania bazami danych (DBMS), takimi jak Oracle, MySQL, Microsoft SQL Server i wiele innych.

JDBC zapewnia możliwość nawiązania połączenia z bazą danych, przesyłania zapytań SQL do bazy danych i odczytywania odpowiedzi. Dzięki temu programiści mogą tworzyć aplikacje, które korzystają z danych przechowywanych w bazach danych, np. do przetwarzania danych, generowania raportów lub prowadzenia transakcji biznesowych.

JDBC definiuje szereg klas i interfejsów, które są wykorzystywane w procesie komunikacji z bazą danych. Klasa DriverManager umożliwia nawiązanie połączenia z bazą danych, a klasa Connection zapewnia dostęp do połączenia. Interfejs Statement umożliwia wykonywanie zapytań SQL, a interfejs ResultSet umożliwia odczytywanie wyników zapytań.

JDBC jest często wykorzystywane w połączeniu z ORM (Object-Relational Mapping) - narzędziami, które pozwalają na mapowanie danych przechowywanych w bazie danych na obiekty języka Java. Popularnymi bibliotekami ORM są Hibernate, MyBatis i EclipseLink.

Dzięki JDBC programiści mogą tworzyć aplikacje, które korzystają z różnych baz danych i łatwo integrować je z innymi systemami informatycznymi.

* Oracle 18 DataBase

Oracle Database 18c to relacyjny system zarządzania bazami danych, wyprodukowany przez firmę Oracle. 18c to wersja, która została wydana w 2018 roku i wprowadziła wiele nowych funkcjonalności oraz poprawek w stosunku do poprzednich wersji.

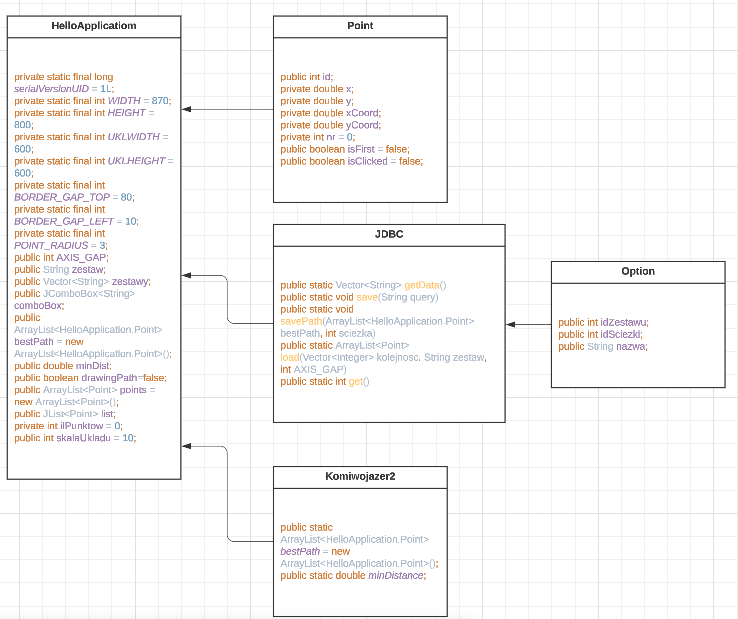
Wśród najważniejszych nowości w Oracle 18c znajduje się mechanizm samoregulacji bazy danych, który pozwala na automatyczne dostosowywanie się do obciążenia, co przekłada się na wyższą wydajność. Zwiększono także możliwości związane z chmurą, co pozwala na łatwiejszą integrację z usługami chmurowymi i migrację danych do chmury.

Oracle 18c wprowadza również nowe funkcjonalności związane z bezpieczeństwem, takie jak uwierzytelnianie wieloczynnikowe i szyfrowanie danych. Udoskonalono również mechanizmy replikacji danych i ich synchronizacji między różnymi instancjami bazy danych.

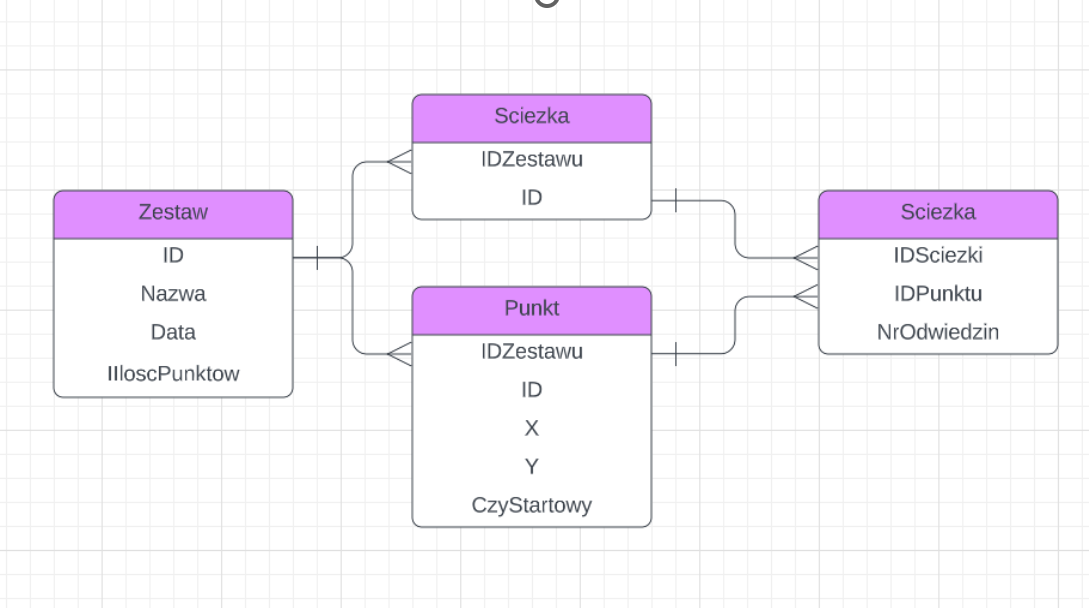
W Oracle 18c udoskonalono także narzędzia analityczne i raportowe, dzięki czemu możliwe jest szybsze i łatwiejsze przetwarzanie dużych ilości danych oraz generowanie raportów biznesowych. Wprowadzono również nowe funkcje związane z programowaniem proceduralnym, takie jak obsługa wyjątków, procedur anonimowych i wiele innych.

Oracle Database 18c to rozbudowany i nowoczesny system zarządzania bazami danych, który oferuje wiele funkcjonalności i możliwości dostosowania się do potrzeb biznesowych.

b. projekt architektury aplikacji,



c. projekt bazy danych – diagram ERD



d. przypadki użycia w modułach,

|  |  |
| --- | --- |
| Tytuł | Zmiana wielkości układu |
| Moduł | Układ współrzędnych |
| Wejście | --- |
| Wyjście | Zmiana przedziału układu współrzędnych |
| Scenariusz główny | 1. Użytkownik przesuwa suwakiem zmiany skali układu 2. Zmiana odległości między pozycjami układu 3. Zmiana ilości wyświetlanych pozycji |
| Scenariusz poboczny | 3a.  Usunięcie punktów znajdujących się poza układem |

|  |  |
| --- | --- |
| Tytuł | Wyświetlenie informacji o problemie Komiwojażera |
| Moduł | Problem Komiwojażera |
| Wejście | --- |
| Wyjście | Wyświetlenie panelu przedstawiającego założenia problemu Komiwojażera |
| Scenariusz główny | 1. Użytkownik klika dany przycisk. 2. Następuje przysłonięcie ekranu nowym panelem. 3. Element z opisem jest wyświetlany na panelu. |
| Scenariusz poboczny | Brak |

|  |  |
| --- | --- |
| Tytuł | Wyświetlanie ścieżki |
| Moduł | Układ współrzędnych |
| Wejście | Punkty, Obliczenie problemu komiwojażera |
| Wyjście | Wyświetlenie ścieżki rozwiązania problemu |
| Scenariusz główny | 1. Sprawdź, czy ścieżka nie jest pusta. 2. Dla każdej pary miast na ścieżce, wyświetl odległość między nimi. 3. Wyświetl informacje o łącznej długości ścieżki. 4. Wyświetl kolejno nazwy punktow wraz z ich pozycjami na mapie. 5. Zakończ wyświetlanie. |
| Scenariusz poboczny | 1. Jeśli ścieżka jest pusta, wyświetl komunikat o błędzie i zakończ. |

|  |  |
| --- | --- |
| Tytuł | Wyświetlenie informacji o programie |
| Moduł | Problem Komiwojażera |
| Wejście | --- |
| Wyjście | Wyświetlenie panelu przedstawiającego założenia projektu grupowego oraz jego składu |
| Scenariusz główny | 1. Uzytkownik klika odpowiedni przycisk 2. Element przysłania obecny widok ekranu 3. Element wyświetla założenia projektu oraz skład grupowy |
| Scenariusz poboczny | Brak |

|  |  |
| --- | --- |
| Tytuł | Pobieranie współrzędnych punktu z pozycji myszy na układzie |
| Moduł | Układ współrzędnych |
| Wejście | --- |
| Wyjście | Dodanie nowego punktu z przeskalowanymi współrzędnymi |
| Scenariusz główny | 1. Użytkownik klika w układzie współrzędnych. 2. Pobierana jest pozycja myszy na układzie współrzędnych. 3. Następuje przybliżenie wartości X i Y do zakresu .5 4. Przypisane zostają nowe wartości do zaznaczonych współrzędnych. 5. Dodanie nowego punktu |
| Scenariusz poboczny | 4a. Punkt o takich współrzędnych istnieje -  powiadom  tym użytkownika.        4c. Nie dodawaj punktu. |

|  |  |
| --- | --- |
| Tytuł | Zapisanie do Bazy |
| Moduł | Zapis rozwiązania |
| Wejście | Punkty, wygenerowana ścieżka |
| Wyjście | Zapisanie zestawu punktów do bazy i dodanie ich ścieżki |
| Scenariusz główny | 1. Użytkownik klika przycisk zapisz rozwiązanie 2. Pobranie danych na temat punktów i ich kolejności w ścieżce 3. Stworzenie nowego zestawu punktów 4. Przypisanie ścieżki do rozwiązania |
| Scenariusz poboczny | 3a.  Dany zestaw już istnieje – Przejdź do 4. |

|  |  |
| --- | --- |
| Tytuł | Wygenerowanie punktu |
| Moduł | Punkty |
| Wejście | --- |
| Wyjście | Stworzenie zestawu punktów o wygenerowanych danych |
| Scenariusz główny | 1. Użytkownik wpisuje ile punktów chce wygenerować 2. Sprawdzenie czy wpisana ilość nie przekracza maksymalnej 3. Wygenerowanie punktów w zakresie układu współrzędnych 4. Dodanie punktów do rozwiązania |
| Scenariusz poboczny | 2a. Wygenerowanie ilości punktów do maksymalnej lub wyświetlenie informacji o maksymalnej ilości punktów |

|  |  |
| --- | --- |
| Tytuł | Wpisanie współrzędnych punktu ręcznie |
| Moduł | Układ współrzędnych |
| Wejście | --- |
| Wyjście | Dodanie nowego punktu za pomocą panelu bocznego. |
| Scenariusz główny | 1. Użytkownik na panelu bocznym wybiera dany przycisk. 2. Wyświetlenie dwóch wejść dla układu współrzędnych. 3. Sprawdzenie poprawności składniowej. 4. Zaokrąglenie wartości X i Y do zakresu .5 5. Dodanie nowego punktu. |
| Scenariusz poboczny | 3a. Wprowadzono  niedozwolone znaki lub spoza zakresu - powiadomienie użytkownika o błędnym wejściu.         3b. Wyczyść dane wprowadzane         5a. Istnieją już punkty o danych współrzędnych - powiadomienie użytkownika o obecnych punktach.         5b. Wyczyść dane wprowadzane |

e. projekt API backend'u,

public static Vector<String> getData() – pobieranie danych z Bazy przy rozpoczęciu programu

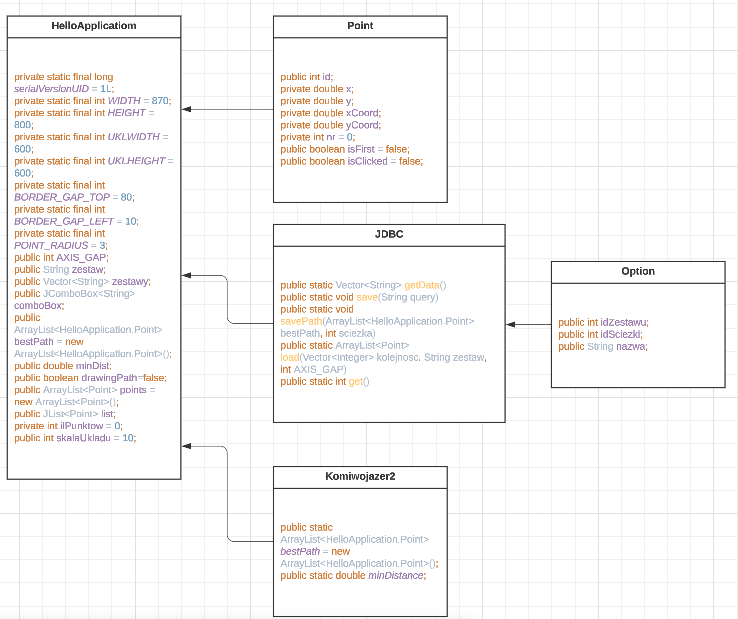
public static void save(String query) – zapisuje dane o punktach i zestawach do bazy

public static void savePath(ArrayList<HelloApplication.Point> bestPath, int sciezka) - zapisuje dane o sciezkach i kolejnościach do bazy

public static ArrayList<Point> load(Vector<Integer> kolejnosc, String zestaw, int AXIS\_GAP) – wczytanie danych z bazy

public static int get() – pobranie ostatniego identyfikatora zestawu (pomocne przy dodaniu nowego rekordu)

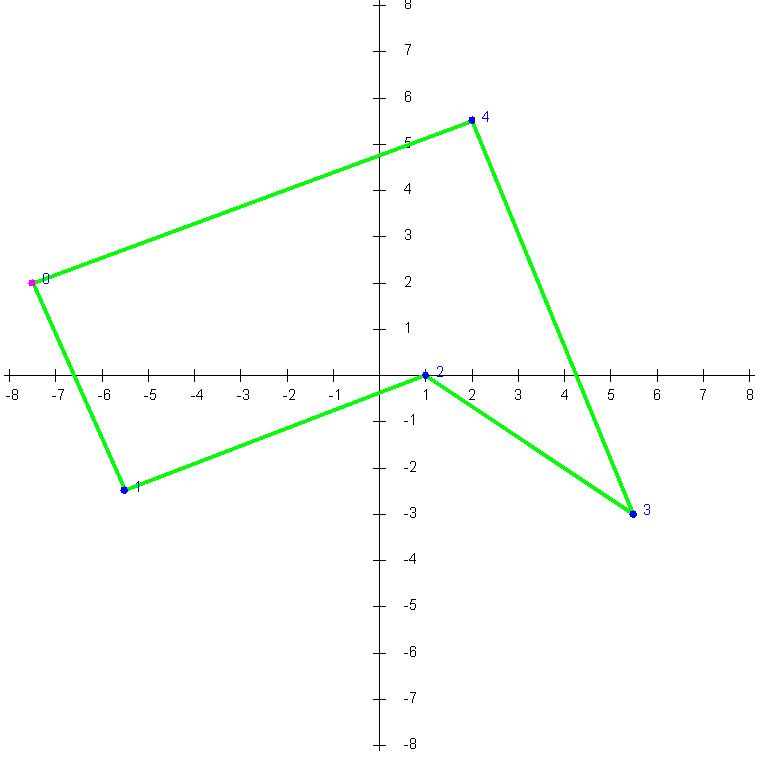
f. diagram klas.



4.4. Testowanie aplikacji.

Porównanie przykładu z www2.seas.gwu.edu z programem.

# 



# Podsumowanie i wnioski

Praca nad programem okazała się długotrwałym projektem przy którym spędziliśmy wiele wieczorów. Ucząc się niektórych technologii tj. javaFX, JDBC oraz baz Oracle powiększyliśmy swoje kompetencje w tych dziedzinach.

# Instrukcja obsługi aplikacji

Aplikacja zajmuje rozwiązaniem się problemu komiwojażera, dając użytkownikowi możliwość wprowadzania bądź wygenerowania zestawu punktów, na podstawie których po wybraniu punktu startowego zostanie obliczona droga przechodząca po wszystkich z punktach i powracająca do punktu startowego. Skala jak i ilość punktów jest modyfikowalna, lecz punkty mogą oscylować w połowach (np. 1,5). Użytkownik uruchamiając aplikację widzi okno aplikacje, które podzielone jest na trzy główne elementy:

* Na samej górze umieszczone są opcje związane z dodawaniem, edycją i usuwaniem punktów biorących udział w problemie.
  + Dodaj losowy punkt – generuje punkt w wyznaczonym przedziale punktowym.
  + Dodaj – dodanie punktu wpisując jego pozycję X i Y w wyskakującym oknie modalnym.
  + Modyfikuj – edycja wybranego punktu wprowadzając jego docelową pozycję.
  + Usuń – usuniecie wybranego punktu biorących udział w procesie obliczenia ścieżki
  + Oznacz jako startowy – po wybraniu punktu musimy ustawić punkt startowy, od którego program rozpocznie przeliczanie całej drogi w problemie Komiwojażera, i na którym skończy on swoją drogę.
  + Rozwiąż problem – przycisk, po którego kliknięciu uruchamia proces rozwiązywania problemu Komiwojażera. Działa on jedynie wtedy, gdy wyznaczony jest punkt startowy
* Środkowy panel z widocznym graficzny:
  + Głównym elementem programu jest układ współrzędnych, na którym graficznie będą przedstawione m.in. rozmieszczenie punktów, numeracja punktów jak i droga po rozwiązaniu problemu Komiwojażera. Na nim również użytkownik ma możliwość ustawieniu punktu ręcznie klikając na wybrany fragment.
  + Dolny suwak odpowiada za ustawienie zakresu pracy programu. Mając gotową listą punktów należy mieć pod uwagą to, że po zmniejszeniu zakresy pracy punkty spoza niego zostaną usunięte.
* Panel na prawo od interfejsu graficznego:
  + Lista punktów – na liście punktów wyświetlane będą dodane bądź wygenerowane punkty. To na tej liście użytkownik dokonuje zmian, takich jak modyfikacja, usunięcie bądź oznaczenie punktu jako startowy.
  + Zapisz rozwiązanie – przycisk powoduje zapisanie gotowego już rozwiązania do bazy danych programu.
  + Wczytaj rozwiązanie – mając zapisane rozwiązanie w bazie danych użytkownik może załadować już gotową ścieżkę.
  + Rozwijany panel – odpowiada za wybranie rozwiązania do późniejszego wczytania przyciskiem ‘Wczytaj rozwiązanie’

Włączając program po raz pierwszy by utworzyć przykładowy problem i go rozwiązać użytkownik powinien postępować zgodnie z instrukcją:

1. Dostosuj skalę wedle upodobań.
2. Dodaj losowe punkty klikając odpowiedni przycisk na górnym panelu.
3. Mając minimum 3 punkty zaznacz jeden z nich, który będzie początkiem jak i końcem drogi pokonanej w problemie.
4. Kliknij przycisk do rozwiązania problemu na górnym panelu programu.

# Literatura

Brak zewnętrznych literatur użytych w sprawozdaniu jak i przy wykonaniu projektu.