

# SPECYFIKACJA IMPLEMENTACYJNA

## *Projekt zespołowy*

Mateusz Smoliński

Łukasz Knigawka

23 grudnia 2018

### Spis treści

<b>1</b>	<b>Wstęp</b>	<b>2</b>
<b>2</b>	<b>Diagram klas</b>	<b>2</b>
<b>3</b>	<b>Opis klas/metod</b>	<b>4</b>
3.1	Klasa MainWindow . . . . .	4
3.1.1	Metoda 1 . . . . .	4
3.1.2	Metoda 2 . . . . .	4
3.2	Klasa 2 . . . . .	4
3.2.1	Metoda 1 . . . . .	4
3.3	Klasa 3 . . . . .	4
3.3.1	metoda . . . . .	4
3.4	Klasa 4 . . . . .	4
3.5	Klasa 5 . . . . .	4
3.6	Klasa 6 . . . . .	4
3.6.1	Metoda 1 . . . . .	4
3.6.2	Metoda 2 . . . . .	4
<b>4</b>	<b>Działanie programu i zastosowanie algorytmu</b>	<b>4</b>
<b>5</b>	<b>Testy jednostkowe</b>	<b>4</b>
5.1	Testy klas 1 i 2 . . . . .	5
5.2	Testy klasy 3 . . . . .	5
5.3	Testy klasy 4 . . . . .	5
5.4	Testy klasy 5 . . . . .	5
5.5	Testy klasy 6 . . . . .	5

# 1 Wstęp

Celem projektu jest napisanie programu prezentującego interaktywną mapę na podstawie danych z pliku wejściowego. Program otrzymuje punkty tworzące kontur mapy, punkty kluczowe, na podstawie których kontur zostaje podzielony na obszary, a także definicje obiektów oraz listę takowych obiektów, które mają być brane pod uwagę przy analizie wyżej wspomnianych obszarów.

Jeśli dane będą podane prawidłowo, program na ich podstawie generuje planszę, która dalej może być modyfikowana przez użytkownika. Może on dodawać i usuwać elementy konturu, zmieniając kształt mapy, a także dodawać i usuwać punkty kluczowe, wpływając tym samym na podział na obszary. Po kliknięciu na jeden z punktów kluczowych użytkownik otrzyma informacje o obiektach znajdujących się na wyznaczonym przez niego obszarze.

Projekt „LUPA” zostanie napisany w języku C#, w środowisku Microsoft Visual Studio 15.9.4 na wersji C#???. Implementacja oraz testowanie programu odbywać się będą na komputerach o następujących parametrach:

- system operacyjny Windows 10 Home ver. 1803,
- procesor Intel Core i5-7200U,
- pamięć RAM 8,00 GB,
- karty graficzne Intel HD Graphics 620 + NVIDIA GeForce 940MX,

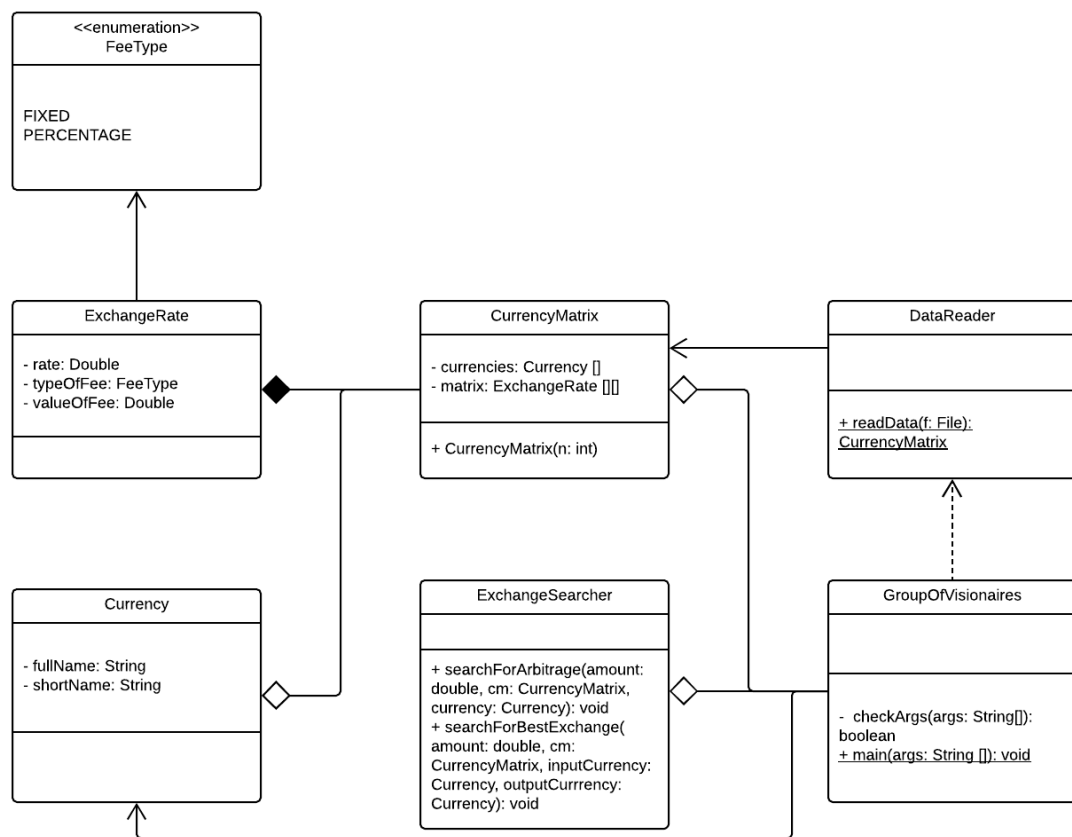
oraz:

tu komp Łukasza

# 2 Diagram klas

Program składać się będzie z  $n$  klas i  $n$  typów wyliczeniowych. Zależności pomiędzy poszczególnymi klasami obrazuje Rysunek 1.

W celu poprawy czytelności na diagramie klas nie zostały uwzględnione metody dostępne dla klas  $x$ ,  $y$ ,  $z$  oraz konstruktory  $xy$ ,  $xz$ ,  $yz$ .



Rysunek 1: Diagram klas

## 3 Opis klas/metod

### 3.1 Klasa MainWindow

Jest to główna klasa tego programu. opisać ją

#### 3.1.1 Metoda 1

Coś to robi

#### 3.1.2 Metoda 2

Pq

### 3.2 Klasa 2

wq

#### 3.2.1 Metoda 1

Mxq

### 3.3 Klasa 3

#### 3.3.1 metoda

- `public CurrencyMatrix (int n)`

standard listingu

### 3.4 Klasa 4

coś robi

### 3.5 Klasa 5

xxx

### 3.6 Klasa 6

W tej klasie coś

#### 3.6.1 Metoda 1

#### 3.6.2 Metoda 2

## 4 Działanie programu i zastosowanie algorytmu

Pełna sekwencja. Opis algorytmu może być problematyczny na tym etapie

## 5 Testy jednostkowe

Projekty testów zakładają użycie narzędzia ???. Poniżej znajduje się lista testów planowanych dla kluczowych metod programu, dla każdej z nich przewidziane są różne przypadki otrzymanych danych.

### **5.1 Testy klas 1 i 2**

W tych dwóch nie trzeba wiele testować

### **5.2 Testy klasy 3**

coś trzeba tu stestować

### **5.3 Testy klasy 4**

Tu testujemy coś

### **5.4 Testy klasy 5**

Ta klasa ma coś

### **5.5 Testy klasy 6**

Ta klasa odpowiada za coś