



# Bazy danych II

## Obsługa danych przestrzennych dwuwymiarowych - dokumentacja

Akademia Górniczo-Hutnicza w Krakowie  
Wydział Fizyki i Informatyki Stosowanej

Tomasz Kozieł

Czerwiec 2023

## Spis treści

1	Opis problemu i opis funkcjonalności udostępnianej przez API	3
2	Opis typów danych oraz metod ( funkcji ) udostępnionych w ramach API	3
3	Opis implementacji udostępnionego API przez bibliotekę	3
4	Testy jednostkowe	4
5	Opis aplikacji terminalowej	4
6	Uruchomienie projektu	10
7	Podsumowanie i wnioski	11
8	Literatura	11

# 1 Opis problemu i opis funkcjonalności udostępnianej przez API

Tematem projektu była obsługa danych przestrzennych dwuwymiarowych. W ramach zadania należało przygotować API i jego implementację umożliwiającą przetwarzanie danych dwuwymiarowych z wykorzystaniem własnych typów danych CLR (Common Language Runtime) UDT (User Defined Types) oraz metod. Przygotowane metody powinny umożliwiać wyznaczenie odległości pomiędzy punktami, wyznaczenie pola określonego obszaru oraz sprawdzenie czy punkt należy do danego obszaru. Opracowane typy danych i metody porównano z typami przetwarzającymi dane przestrzenne udostępnionymi w ramach SQL Server.

## 2 Opis typów danych oraz metod ( funkcji ) udostępnionych w ramach API

W ramach projektu stworzono dwa obiekty UDT Point oraz Figure. Point reprezentuje punkt, natomiast Figure reprezentuje figurę składającą się z listy punktów. Obydwa klasy posiadają następujące metody:

- konstruktory, getterzy oraz setterzy
- public override string ToString()
- public static Point(Figure Parse(SqlString s))
- public void Read(System.IO.BinaryReader r) - metoda serializująca obiekty
- public void Write(System.IO.BinaryWriter w) - metoda deserializująca obiekty

Ponadto klasa Point posiadała:

- public double x - współrzędna x
- public double y - współrzędna y
- private bool IsNull - pole określające czy obiekt został zainicjalizowany
- public SqlDouble Distance(Point p) - metoda obliczająca odległość pomiędzy dwoma punktami

Natomiast klasa Figure posiadała:

- public List<Point> points - lista punktów, które wchodziły w skład figury
- public void AddPoint(Point point) - metoda służąca do dodawania pojedynczych punktów (końcowo nie została wykorzystana)
- public Point ImpotPoints - setter-only property pozwala na dodawanie punktów do figury, jest wykorzystywana przy dodawaniu punktów w aplikacji konsolowej
- public int PoinCount - getter-only property, zwraca ilość punktów w figurze
- public double CalculateArea() - metoda zwracająca pole figury
- public bool IsInside(Point point) - metoda sprawdzająca czy przekazany punkt znajduje się wewnątrz figury

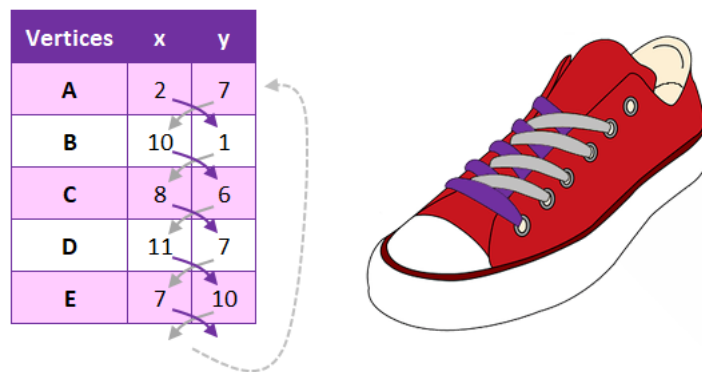
## 3 Opis implementacji udostępnionego API przez bibliotekę

W celu obliczenia odległości pomiędzy punktami wykorzystano wzór na odległość euklidesową

$$d = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

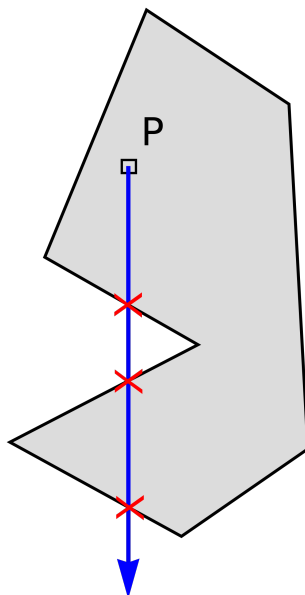
W celu obliczenia pola figury skorzystano z alorytmu Shoelace. Metoda polega na przemnożeniu współrzędnych odpowiadających sobie wierzchołków różnych boków wielokąta w celu znalezienia jego pola. Nazywana jest wzorem Shoelace ze względu na stałe przemnażanie współrzędnych tworzących wielokąt, jak wiązanie sznurówek:

$$Area = \frac{1}{2} \left| \sum_{i=1}^{n-1} x_i y_{i+1} + x_n y_1 - \sum_{i=1}^{n-1} x_{i+1} y_i - x_1 y_n \right|$$



Rysunek 1: Shoelace algorithm

W celu sprawdzenia czy punkt znajduje się wewnątrz figury skorzystano z algorytmu Ray Castingu (rzutowanie promieni). Algorytm polega na "rzucaniu" promienia z badanego punktu w nieskończoność i sprawdzaniu, ile razy ten promień przecina krawędzie obiektu. Jeśli liczba przecięć jest parzysta, to punkt jest poza obiektem, jeśli jest nieparzysta, to punkt jest wewnątrz obiektu.



Rysunek 2: Ray Casting algorithm

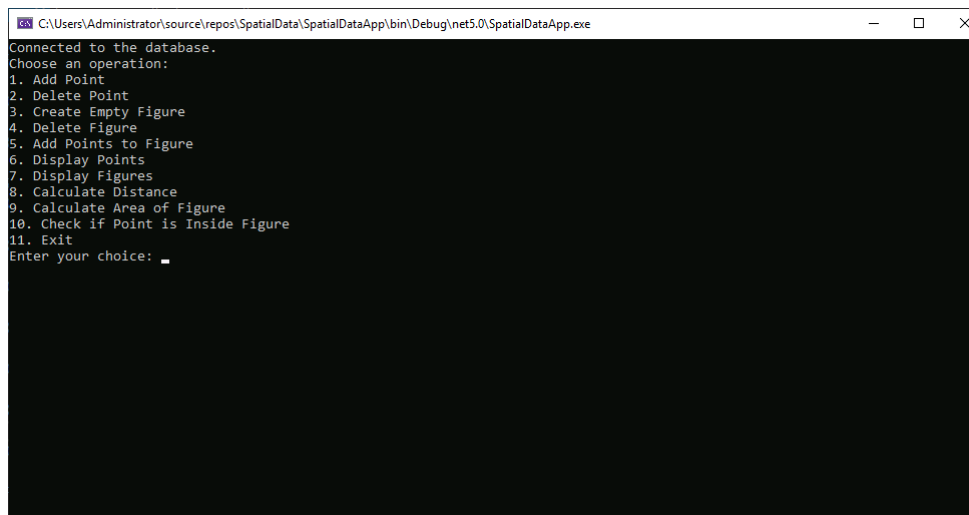
## 4 Testy jednostkowe

Ze względu na prostotę przygotowanych metod zdecydowano się pominąć testy jednostkowe. Aplikacja terminalowa została wykorzystana w celu sprawdzenia poprawności przygotowanych funkcjonalności.

## 5 Opis aplikacji terminalowej

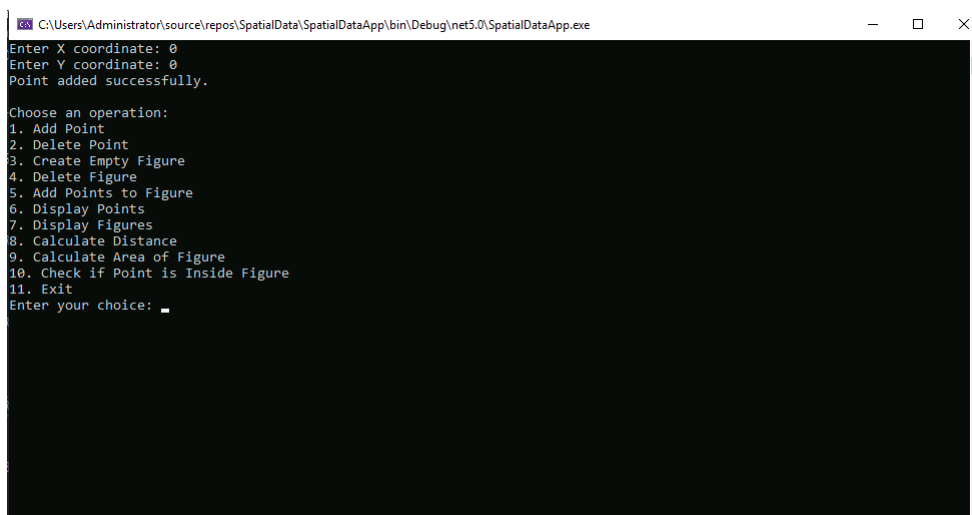
W celu prezentacji możliwości biblioteki przygotowano aplikację konsolową. Użytkownik porusza się po niej wybierając konkretne opcje (numery) oraz zatwierdzając operację klawiszem ENTER.

Początkowo użytkownikowi ukazuje się prosty ekranem powitalny:



Rysunek 3: Ekran główny aplikacji

Wybierając opcję 1: użytkownik zostaje poproszony o podanie współrzędnych X oraz Y nowego punktu. ID punktu zostaje nadane automatycznie, po wypełnieniu współrzędnych zostaje wyświetlona wiadomość "Point added successfully.":



Rysunek 4: 1) Add Point

Wybierając opcję 2: zostaje wyświetlona lista wszystkich punktów, użytkownik zostaje poproszone o podanie id punktu, który chce usunąć, po wypełnieniu pola id zostaje wyświetlona wiadomość "Point deleted successfully."

```
C:\Users\Administrator\source\repos\SpatialData\SpatialDataApp\bin\Debug\net5.0\SpatialDataApp.exe
Displaying points:
ID: 1, Point: 0,0
Enter Point ID: 1
Point deleted successfully.

Choose an operation:
1. Add Point
2. Delete Point
3. Create Empty Figure
4. Delete Figure
5. Add Points to Figure
6. Display Points
7. Display Figures
8. Calculate Distance
9. Calculate Area of Figure
10. Check if Point is Inside Figure
11. Exit
Enter your choice: _
```

Rysunek 5: 2) Delete Point

Wybierając opcję 3: zostaje stworzona figura bez jakiegokolwiek punktu, przydaje się to do użycia funkcji 5) Add Points to Figure, po stworzeniu figury zostaje wyświetlona wiadomość "Empty figure created successfully."

```
C:\Users\Administrator\source\repos\SpatialData\SpatialDataApp\bin\Debug\net5.0\SpatialDataApp.exe
Empty figure created successfully.

Choose an operation:
1. Add Point
2. Delete Point
3. Create Empty Figure
4. Delete Figure
5. Add Points to Figure
6. Display Points
7. Display Figures
8. Calculate Distance
9. Calculate Area of Figure
10. Check if Point is Inside Figure
11. Exit
Enter your choice: _
```

Rysunek 6: 3) Create Empty Figure

Wybierając opcję 4: zostaje wyświetlona lista wszystkich figur, użytkownik zostaje poproszony o wybranie id figury, którą chce usunąć, po wypełnieniu pola id zostaje wyświetlona wiadomość "Figure deleted successfully."

```
C:\Users\Administrator\source\repos\SpatialData\SpatialDataApp\bin\Debug\net5.0\SpatialDataApp.exe
ID: 1, Figure:
Enter Figure ID: 1
Figure deleted successfully.

Choose an operation:
1. Add Point
2. Delete Point
3. Create Empty Figure
4. Delete Figure
5. Add Points to Figure
6. Display Points
7. Display Figures
8. Calculate Distance
9. Calculate Area of Figure
10. Check if Point is Inside Figure
11. Exit
Enter your choice: _
```

Rysunek 7: 4) Delete Figure

Wybierając opcję 5: zostają wyświetlone figury oraz punkty, użytkownik zostaje poproszony o wybranie id figury oraz id punktów które chce dodać do wybranej figury, punkty muszą zostać wypisane jeden obok drugiego i muszą być oddzielone przecinkiem, jest to niezbędne z uwagi na funkcję konwertującą. Końcowo zostaje wyświetlona wiadomość "Points added to the Figure successfully."

```
C:\Users\Administrator\source\repos\SpatialData\SpatialDataApp\bin\Debug\net5.0\SpatialDataApp.exe
Displaying points:
ID: 2, Point: 0,0
ID: 3, Point: 5,0
ID: 4, Point: 5,5
ID: 5, Point: 0,5

ID: 2, Figure:
Enter Figure ID: 2
Enter Point IDs (comma-separated):
2,3,4,5
Points added to the figure successfully.

Choose an operation:
1. Add Point
2. Delete Point
3. Create Empty Figure
4. Delete Figure
5. Add Points to Figure
6. Display Points
7. Display Figures
8. Calculate Distance
9. Calculate Area of Figure
10. Check if Point is Inside Figure
11. Exit
Enter your choice: _
```

Rysunek 8: 5) Add Points to Figure

Wybierając opcję 6: zostają wyświetlone wszystkie punkty znajdujące się w bazie

```
C:\Users\Administrator\source\repos\SpatialData\SpatialDataApp\bin\Debug\net5.0\SpatialDataApp.exe
Displaying points:
ID: 2, Point: 0,0
ID: 3, Point: 5,0
ID: 4, Point: 5,5
ID: 5, Point: 0,5

Choose an operation:
1. Add Point
2. Delete Point
3. Create Empty Figure
4. Delete Figure
5. Add Points to Figure
6. Display Points
7. Display Figures
8. Calculate Distance
9. Calculate Area of Figure
10. Check if Point is Inside Figure
11. Exit
Enter your choice: _
```

Rysunek 9: 6) Display Points

Wybierając opcję 7: zostają wyświetlone wszystkie figury znajdujące się w bazie

```
C:\Users\Administrator\source\repos\SpatialData\SpatialDataApp\bin\Debug\net5.0\SpatialDataApp.exe
ID: 2, Figure: 0,0
5,0
5,5
0,5

Choose an operation:
1. Add Point
2. Delete Point
3. Create Empty Figure
4. Delete Figure
5. Add Points to Figure
6. Display Points
7. Display Figures
8. Calculate Distance
9. Calculate Area of Figure
10. Check if Point is Inside Figure
11. Exit
Enter your choice: _
```

Rysunek 10: 7) Display Figures

Wybierając opcję 8: zostają wypisane wszystkie punkty, użytkownik zostaje poproszony o podanie id dwóch punktów pomiędzy którymi chce obliczyć odległość



```
C:\Users\Administrator\source\repos\SpatialData\SpatialDataApp\bin\Debug\net5.0\SpatialDataApp.exe
Displaying points:
ID: 2, Point: 0,0
ID: 3, Point: 5,0
ID: 4, Point: 5,5
ID: 5, Point: 0,5
Enter Point 1 ID: 2
Enter Point 2 ID: 5
Distance between Point 2 and Point 5: 5

Choose an operation:
1. Add Point
2. Delete Point
3. Create Empty Figure
4. Delete Figure
5. Add Points to Figure
6. Display Points
7. Display Figures
8. Calculate Distance
9. Calculate Area of Figure
10. Check if Point is Inside Figure
11. Exit
Enter your choice: _
```

Rysunek 11: 8) Calculate Distance

Wybierając opcję 9: zostają wypisane wszystkie figury, użytkownik zostaje poproszony o podanie id figury, której chce obliczyć pole

```
C:\Users\Administrator\source\repos\SpatialData\SpatialDataApp\bin\Debug\net5.0\SpatialDataApp.exe
ID: 2, Figure: 0,0
5,0
5,5
0,5
Enter Figure ID: 2
Area of Figure 2: 25

Choose an operation:
1. Add Point
2. Delete Point
3. Create Empty Figure
4. Delete Figure
5. Add Points to Figure
6. Display Points
7. Display Figures
8. Calculate Distance
9. Calculate Area of Figure
10. Check if Point is Inside Figure
11. Exit
Enter your choice: _
```

Rysunek 12: 9) Calculate Area

Wybierając opcję 10: zostają wypisane wszystkie punkty oraz figury, użytkownik zostaje poproszony o podanie id punktu, który chce sprawdzić oraz id figury którą będzie badać

```
C:\Users\Administrator\source\repos\SpatialData\SpatialDataApp\bin\Debug\net5.0\SpatialDataApp.exe
Displaying points:
ID: 2, Point: 0,0
ID: 3, Point: 5,0
ID: 4, Point: 5,5
ID: 5, Point: 0,5
ID: 6, Point: 1,1

ID: 2, Figure: 0,0
5,0
5,5
0,5

Enter Point ID: 6
Enter Figure ID: 2
Point 6 is inside Figure 2.

Choose an operation:
1. Add Point
2. Delete Point
3. Create Empty Figure
4. Delete Figure
5. Add Points to Figure
6. Display Points
7. Display Figures
8. Calculate Distance
9. Calculate Area of Figure
10. Check if Point is Inside Figure
11. Exit
Enter your choice: _
```

Rysunek 13: 10) Check if Point is Inside Figure

Wybierając opcję 11: użytkownik zostaje poinformowany o rozłączeniu z bazą danych oraz zostaje wyświetlona wiadomość "Exiting the program."

```
Microsoft Visual Studio Debug Console
Exiting the program.
Disconnected from the database.

C:\Users\Administrator\source\repos\SpatialData\SpatialDataApp\bin\Debug\net5.0\SpatialDataApp.exe (process 8704) exited
with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console
when debugging stops.
Press any key to close this window . . .
```

Rysunek 14: 11) Exit

## 6 Uruchomienie projektu

Projekt został napisany w języku C# przy użyciu środowiska Microsoft Visual Studio 2019 (.NET Framework 4.7.2). Projekt od strony bazodanowej został utworzony w Microsoft SQL Server 18.

- Utworzenie bazy danych, pierwsze 5 linijek w pliku DatabaseSetup.sql
- Uruchomienie projektu "SpatialData" oraz połączenie się z odpowiednią bazą danych "SpatialDataDB"
- Wykonanie reszty skryptów z pliku DatabaseSetup.sql: utworzenie **assembly** - należy zmienić ścieżkę do pliku dll!, stworzenie potrzebnych typów oraz tabel.
- Uruchomienie aplikacji "SpatialDataApp" należy zmienić **connectionString** w pliku Program.cs na odpowiedni, który wyświetli się po połączeniu z bazą danych.

Projekt można znaleźć na Githubie:

<https://github.com/tomekkoziel/SpatialData>

## 7 Podsumowanie i wnioski

Funkcjonalność UDT udostępnia możliwość stworzenia elastycznych i przygotowanych pod własne potrzeby typów danych, które można dowolnie edytować według własnych upodobań. Jest to potężna funkcjonalność, która daje ogromne możliwości. Warto jednak zauważyć iż SQL Server posiada predefiniowane typy przechowujące dane "Geometry" i "Geography" oferują one bardziej rozbudowane funkcje i zoptymalizowane operacje dla danych przestrzennych, co może być korzystne w przypadku zaawansowanej analizy i przetwarzania danych przestrzennych.

## 8 Literatura

- <https://newton.fis.agh.edu.pl/antek/index.php?sub=dbase2>
- <https://www.101computing.net/the-shoelace-algorithm/>
- [http://0x80.pl/articles/point\\_in\\_polygon.html](http://0x80.pl/articles/point_in_polygon.html)
- <https://stackoverflow.com/>