**Dokumentacja projektu zaliczeniowego   
z przedmiotu Bazy Danych 2**

Bartłomiej Leśnicki

1. Cel projektu:

Celem projektów jest opracowanie API, jego implementacja a następnie wykorzystanie w aplikacji napisanej w C# wykonującej testy jednostkowe dla udostępnionych funkcjonalności API. W ramach aplikacji można wykorzystać polecenie ASSERT do testowania udostępnionej funkcjonalności w API. Połączenie z bazą danych w przygotowanej na potrzeby testów aplikacji należy zrealizować wykorzystując interfejs ADO.NET.

2.Temat:

**Przetwarzanie własnych agregatów CLR UDA.** Opracować API oraz jego implementację obsługującą wybrany zestaw własnych agregatów CLR UDA (przynajmniej 5 oraz minimum dwa w wersji rozszerzonej). W ramach projektu należy przygotować odpowiedni zestaw danych umożliwiających poprawną weryfikację przygotowanych agregatów CLR UDA.

3.Agregaty:

Agregaty zostały napisane z użyciem frameworka .NET. Kod części projektowej realizującej stworzenie agregatów znajduje się w folderze Agregaty.

1.Ilosc - Oblicza liczbę rekordów zwróconych przez zapytanie **Ilosc(wyr)**

Symbol *zastępczy wyr.* reprezentuje wyrażenie tekstowe pola zawierającego dane, które mają zostać zliczone, lub wyrażenie wykonujące obliczenie przy użyciu danych w tym polu

2. Maksimum – Zwraca maksimum zestawu wartości określonych w polu zapytania. **Maksimum(wyr)**

Symbol zastępczy wyr reprezentuje wyrażenie tekstowe pola zawierającego dane, z których ma być wskazana wartość maksymalna.

3. Minimum - Zwraca minimum zestawu wartości określonych w polu zapytania. **Minimum(wyr)**

Symbol zastępczy wyr reprezentuje wyrażenie tekstowe pola zawierającego dane, z których ma być wskazana wartość maksymalna.

4.Mediana – Zwraca medianę zestawu wartości określonych w polu zapytania. **Mediana(wyr)**

Symbol zastępczy wyr reprezentuje wyrażenie tekstowe pola zawierającego dane, z których ma być wskazana mediana.

5.Moda – Zwraca mody(Dominanty) zestawu wartości określonych w polu zapytania. Wartość zwracana reprezentowana jest za pomocą tekstu gdzie mody poprzedzone są spacjami, jako znak separacji został użyty znak średnika ; **Moda(wyr)**

Symbol zastępczy wyr reprezentuje wyrażenie tekstowe pola zawierającego dane, z których mają być wskazane mody.

6.OdchylenieStandardowe – Zwraca odchylenie standardowe zestawu wartości w określonym polu zapytania. **OdchylenieStandardowe(wyr)**

Symbol zastępczy wyr reprezentuje wyrażenie tekstowe pola zawierającego dane, z których ma być wyliczone odchylenie standardowe.

7.Suma – Zwraca sumę zestawu wartości zawartych określonych w polu zapytania. **Suma(wyr)**

Symbol *zastępczy wyr.* reprezentuje wyrażenie tekstowe pola zawierającego dane liczbowe, które chcesz dodać, lub wyrażenie wykonujące obliczenie przy użyciu danych w tym polu.

8.Średnia – Zwraca średnią zestawu danych zawartych w określonym polu zapytania. **Średnia(wyr)**

Symbol zastępczy wyr reprezentuje wyrażenie tekstowe pola zawierającego dane liczbowe, z których ma zostać obliczona średnia.

9.Unikaty – Zwraca wartości unikalne z danych określonym w polu zapytania. Wartość zwracana reprezentowana jest za pomocą tekstu gdzie poszczególne wartości poprzedzone są spacjami, jako znak separacji został użyty znak średnika ; **Unikaty(wyr)**

Symbol zastępczy wyr reprezentuje wyrażenia tekstowe pola zawierającego dane liczbowe, na podstawie których mają zostać wybrane wartości unikalne.

10.Wariancja – Zwraca wariancję zestawu danych zawartych w określonym polu zapytania. **Wariancja(wyr)**

Symbol zastępczy wyr reprezentuje wyrażenie tekstowe pola zawierającego dane liczbowe, z których ma zostać wyliczona wariancja.

11.Wszystkie – Zwraca wszystkie wartości danych określonych w polu zapytania. Wartość zwracana reprezentowana jest za pomocą tekstu gdzie poszczególne wartości poprzedzone są spacjami, jako znak separacji został użyty znak średnika ; **Wszystkie(wyr)**

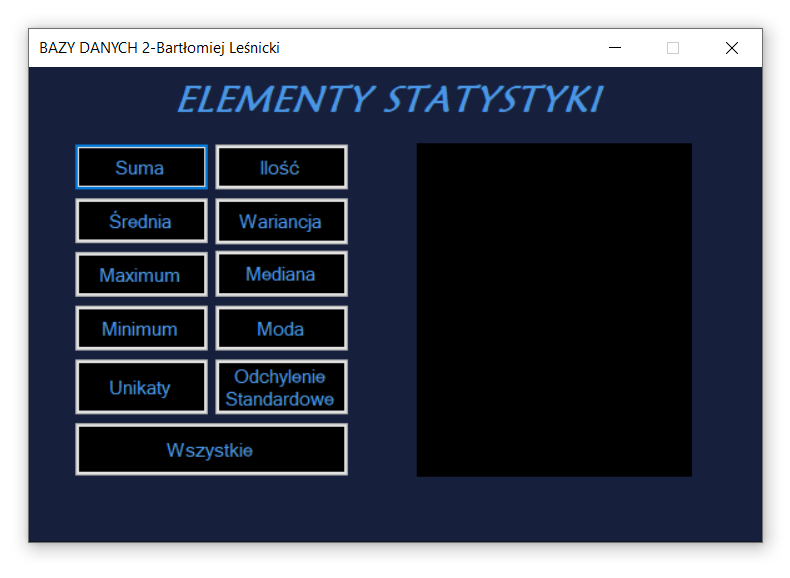
Symbol zastępczy wyr reprezentuje wyrażenie tekstowe pola zawierającego dane liczbowe, które mają zostać wypisane.

4.API i GUI

W ramach projektu zostało zaprojektowane API oraz jego graficzna interpretacja. Kod znajduje się w folderze API. Głównym punktem wejścia dla aplikacji jest statyczna klasa Program oraz metoda Main(). Za dostęp do bazy danych oraz wywoływanie agregatów odpowiedzialna jest klasa DataAcess oraz metody w niej zawarte. Za obsługę przycisków oraz innych aspektów GUI odpowiada klasa Form1. Implementacja testów jednostkowych została przeprowadzona w klasie DataAcessTests.

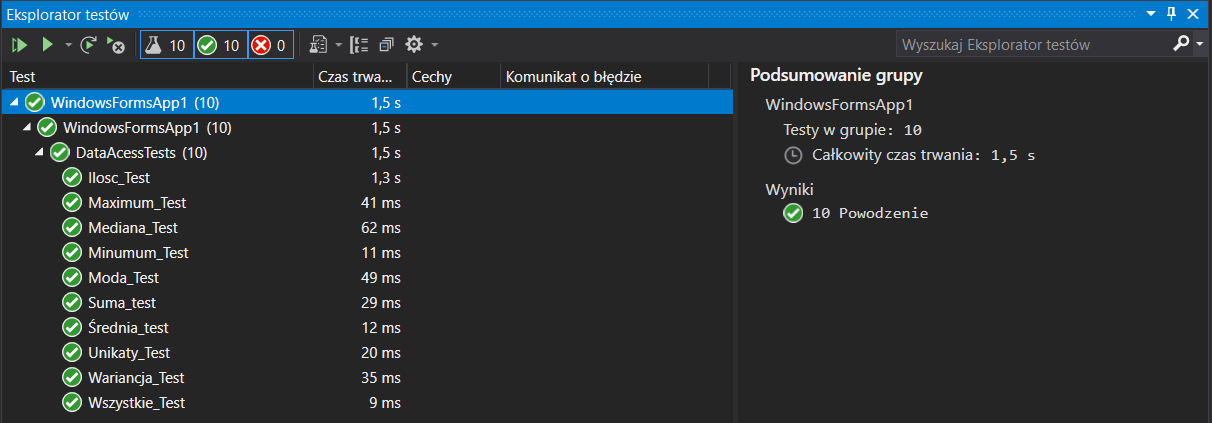
GUI zawiera:   
- 11 Przycisków obsługujących Agregaty.

- textBox na którym wypisywane są wartości zwracane agregatów.



5.Testy:

W ramach projektu kod został pokryty testami jednostkowymi. Każda z metod obsługująca agregat została przetestowana odpowiednim testem z Klasy DataAcessTests. Zostały przetestowane metody z klasy DataAcess. Wynik testów:



6.Uruchomienie Programu.

Najpierw trzeba stworzyć bazę danych a w niej testową tabelę „Dane\_testowe”

CREATE TABLE Dane\_Testowe ( Wartosc REAL NOT NULL) i zainicjować ją przykładowymi danymi.

INSERT INTO dbo.Dane\_Testowe VALUES (1),(2),(3),(4),(5),(6),(7),(8),(9),(1),(2),(5),(5),(5),(5),(5),(5),(5),(5),(4),(4), (4),(4),(4),(3.13)

Następnie tworzymy agregaty. W katalogu Agregaty znajduje się projekt Database1. Należy go skompilować i wdrożyć rozwiązanie po uprzednim podłączeniu się do bazy danych. W celu uruchomienia programu należ w pliku connectionString.txt znajdującym się w katalogu API\WindowsFormsApp1\bin\Release ustawić odpowiedni connection string. Teraz program można uruchomić z poziomu kodu lub bezpośrednio za pomocą pliku wykonywalnego WindowsFormsApp1.exe (w katalogu API\WindowsFormsApp1\bin\Release).

7.Podsumowanie i Wnioski

UDA – Pozwala programistom na zbudowanie własnej funkcji agregującej, którą można używać w połączeniu z klauzulą GROUP BY w zapytaniu TSQL. To pozwala na przeprowadzenie złożonych operacji statystycznych takich jak znajdowanie wariancji, odchylenia standardowego, mody czy mediany oraz analizę danych przy użyciu silnika baz danych. Co więcej, kodu agregatów nie można tworzyć inaczej, jak tylko używając technologii .NET. Nie ma możliwości stworzenia ich przy użyciu samego kodu języka T-SQL.

8.Literatura

Materiały dostępne w ramach kursu Bazy Danych 2 oraz:

<https://docs.microsoft.com/pl-pl/aspnet/core/?view=aspnetcore-5.0>

<https://www.microsoft.com/pl-pl/sql-server/sql-server-2019>

Bartłomiej Leśnicki