

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ



DOKUMENTACJA PROJEKTU
**Przetwarzanie własnych typów danych
CLR UDT**

Aleksandra Poręba
nr. indeksu 290514

8 czerwca 2019

Spis treści

1	Opis problemu	3
2	Opis funkcjonalności udostępnianej przez API	3
2.1	Menu główne	3
2.2	Menu figury	4
2.3	Menu dla wszystkich figury	4
2.4	Dodawanie figur	5
2.5	Przeszukiwanie figur	5
3	Opis typów danych oraz metod udostępnionych w ramach API	5
3.1	Command	5
3.1.1	Pola klasy	5
3.1.2	Metody klasy	6
3.2	Projekt_UDT_aplikacja	6
3.2.1	Pola klasy	6
3.2.2	Metody klasy	6
4	Opis klas UDT	7
4.1	Pola klasy	7
4.2	Metody klasy	7
5	Testy jednostkowe	8
6	Instrukcja uruchomienia	8
6.1	Stworzenie bazy danych	8
6.2	Deploy UDT	8
6.3	Stworzenie tabel	9
6.4	Uruchomienie aplikacji	9
7	Bibliografia	9

1 Opis problemu

Celem projektu było stworzenie aplikacji konsolowej wykorzystującej złożone User Defined Types. Ma ona udostępniać opcję dodawania rekordów, wyszukiwania i tworzenie raportów oraz zawierać niezbędne testy.

Jako typy złożone zostały wybrane figury geometryczne dwuwymiarowe, takie jak koło, trójkąt, kwadrat, prostokąt, równoległobok oraz trapez.

2 Opis funkcjonalności udostępnianej przez API

Stworzona aplikacja jest aplikacją konsolową, poruszamy się po kolejnych funkcjonalnościach przez wybór odpowiednich klawiszy, zgodnie z informacjami wyświetlanymi w menu. Możemy dodawać, przeszukiwać oraz wyświetlać figury według rodzaju, albo wszystkie razem. Możliwe jest również wyświetlenie pola oraz obwodu figury.

2.1 Menu główne

Menu główne umożliwia nam wybór figury, z którą chcemy pracować:

- **k** - koło,
- **t** - trójkąt,
- **w** - kwadrat,
- **p** - prostokąt,
- **r** - równoległobok,
- **z** - trapez,
- **f** - wszystkie figury,
- **m** - wyświetlenie menu,
- **e** - wyjście z programu.

2.2 Menu figury

Gdy wybierzemy którąś z figur, możemy wybrać operacje:

- **d** - dodaj nową figurę,
- **w** - wyświetl wszystkie dodane figury (danego typu),
- **p** - wyświetl wszystkie dodane figury wraz z polami,
- **s** - wyświetl wszystkie dodane figury wraz z polami, posortowane według pola,
- **y** - przeszukaj figury według pola,
- **o** - wyświetl wszystkie dodane figury wraz z obwodami,
- **b** - wyświetl wszystkie dodane figury wraz z obwodami, posortowane,
- **u** - przeszukaj figury według obwodu,
- **m** - przejście do menu głównego,
- **e** - wyjście z programu.

2.3 Menu dla wszystkich figury

Dla opcji wszystkich figur mamy dostępne menu:

- **w** - wyświetl wszystkie dodane figury,
- **p** - wyświetl wszystkie dodane figury wraz z polami,
- **s** - wyświetl wszystkie dodane figury wraz z polami, posortowane według pola,
- **y** - przeszukaj figury według pola,
- **o** - wyświetl wszystkie dodane figury wraz z obwodami,
- **b** - wyświetl wszystkie dodane figury wraz z obwodami, posortowane,
- **u** - przeszukaj figury według obwodu,
- **m** - przejście do menu głównego,
- **e** - wyjście z programu.

2.4 Dodawanie figur

Podczas dodawania figury wyświetli się tekst pomocniczy, w jakim formacie i kolejności należy podać wartości. Powinny być one oddzielone spacją, a liczby dziesiętne zapisane z przecinkiem (nie kropką). Podajemy kolejne wierzchołki, najpierw współrzędną x , potem y (dla koła promień, a następnie współrzędne środka). Gdy podane zostanie więcej wartości, będą one pominięte, gdy za mało pojawi się błąd.

2.5 Przeszukiwanie figur

Możliwe jest wyszukanie figur spełniających podany przez użytkownika warunek dotyczący obwodu lub pola. Dostępne są operatory $=$, $>$ oraz $<$. Następnie po spacji trzeba podać liczbę, zapisaną w formacie z kropką. Gdy któreś z wymagań nie zostanie spełnione, pojawi się odpowiednia informacja.

3 Opis typów danych oraz metod udostępnionych w ramach API

Aplikacja konsolowa składa się z dwóch klas: `Projekt_UDT_aplikacja` oraz `Command`.

3.1 Command

Klasa `Command` przechowuje informacje na temat zapytania, które będzie wysyłane.

3.1.1 Pola klasy

- `_com` - przechowuje treść zapytania lub jego część, która jest stała w każdym wywołaniu,
- `_type` - przechowuje informację o typie zapytania, czy to będzie selekcja, dodawanie, czy wyszukiwanie - zależnie od typu, trzeba będzie pobrać od użytkownika dodatkowe informacje przy każdym zapytaniu,
- `_insertHelper` - wiadomość wyświetlana przy pobieraniu dodatkowych argumentów,

- **`_resultAttr`** - zawiera listę nazw kolumn, które będą zwrócone po zapytaniu.

3.1.2 Metody klasy

Metody klasy to funkcje zwracające wartości pól ("getter"). Pola inicjalizowane są w konstruktorze.

3.2 Projekt `_UDT_aplikacja`

Ta klasa jest główną klasą programu, realizującą funkcjonalności aplikacji konsolowej.

3.2.1 Pola klasy

- **`_currentFigure`** - zawiera informację o aktualnie przetwarzanej figurze,
- **`_commands`** - mapa zawierająca dostępne zapytania.

3.2.2 Metody klasy

- **`InitMap`** - inicjalizuje mapę z zapytaniami,
- **`PrintMenu`** - wypisuje główne menu programu, ze znakami odpowiadającymi figurom,
- **`HandleMenu`** - w zależności od wybranego znaku, zwraca odpowiednią figurę,
- **`PrintFigureMenu`** - wypisuje menu z dostępnymi operacjami dla pojedynczej figury,
- **`HandleFigureMenu`** - w zależności od wybranego znaku, wywołuje metodę wysyłającą odpowiednie zapytanie,
- **`Print2DFigureMenu`** - wypisuje menu z dostępnymi operacjami dla wszystkich figur,
- **`Handle2DFigureMenu`** - w zależności od wybranego znaku, wywołuje metodę wysyłającą odpowiednie zapytanie,

- **GetInsertData** - pobiera dane potrzebne dla zapytania typu insert,
- **GetSearchData** - pobiera dane potrzebne dla zapytania typu search,
- **SendCommand** - funkcja tworzy połączenie z bazą danych, wysyła zapytanie oraz obsługuje odpowiedź bądź wyjątek. Jeśli jest to konieczne, zleca pobranie dodatkowych argumentów do zapytania,
- **Main** - posiada w sobie główną pętlę programu, w zależności od obecnie wybranej figury wyświetla menu oraz zleca obsługę wybranych opcji.

4 Opis klas UDT

Klasy są zbudowane schematycznie - udostępniają zestaw metod, realizujących operacje dla danej figury. Zestaw funkcji może się różnić, jednak każda udostępnia konstruktor, przyjmujący wartości pól i metody **Parse**, **ToString**, **Pole** oraz **Obwod**.

4.1 Pola klasy

Każda z klas zawiera w sobie pola reprezentujące wierzchołki figur. Różnicą jest klasa **Kolo** zawiera współrzędne środka oraz promień.

4.2 Metody klasy

Klasy zawierają konstruktor przyjmujący wartości kolejnych wierzchołków (lub promienia i środka). Oprócz tego dostępne są metody:

- **ToString** - zwraca opis figury,
- **Parse** - parsuje łańcuch znaków, i gdy jest poprawny tworzy nowy obiekt,
- **Pole** - zwraca pole figury,
- **Obwód** - zwraca obwód figury,
- **SprawdzPunkty** - funkcja sprawdzająca, czy z podanych wartości można utworzyć wybraną figurę,

- **PoliczBoki** - liczy długości boków figury,
- **DlugosciBokow** - zwraca w postaci łańcucha znaków długości boków figury,
- **PoliczWysokosc** - liczy wysokość figury.

5 Testy jednostkowe

Została przetestowana część aplikacji konsolowej - opiera się ona na wypisywaniu na ekran, więc testy `assert` dla niektórych metod nie były możliwe. Dokładniej zostało przetestowane połączenie z bazą i wysyłanie zapytań. Łącznie wykonano 27 testów, sprawdzających przypadki poprawne i te rzucające wyjątki.

Testy można uruchomić w `Visual Studio 2008` otwierając projekt `Projekt_UDT_aplikacja`, a następnie wybierając `Test->Run->All tests in solution`.

6 Instrukcja uruchomienia

6.1 Stworzenie bazy danych

Na początku należy uruchomić za pomocą `Microsoft SQL Server Management Studio` pierwszą część skryptu `projekt_UDT_init.sql` (PUNKT 1) znajdującego się w folderze `sql-server`. Stworzy ona odpowiednią bazę danych.

6.2 Deploy UDT

Następnym krokiem jest stworzenie typów UDT w nowej bazie. Należy otworzyć projekt `Projekt_UDT` w `Visual Studio 2008` i wybrać `Build` a następnie `Deploy`. Oba powinny być zakończone `successful`.

Gdy pojawią się problemy z otwarciem projektu za pomocą pliku `.sln` można wejść do katalogu `Projekt_UDT` i utworzyć plik `.csproj`.

Po otwarciu projektu dobrze jest upewnić się, czy napewno mamy dodane referencje do bazy. Można to zrobić klikając PPM na nazwę projektu i wybrać `Properties->Database->Connection String->Browse`. Gdy na liście nie ma bazy projektUDT możemy ją dodać poprzez kliknięcie `Add New Reference` i wybrać `Server Name` na `MSSQLSERVER`, a następnie w tym samym oknie `Connect to a database->Select or enter database name` na projektUDT. Gdy nie ma opcji `MSSQLSERVER` na liście powinniśmy zmienić `SQL Server Browser` na `Running`, a gdy nie ma bazy projektUDT na liście, to znaczy że nie została poprawnie dodana.

6.3 Stworzenie tabel

Teraz należy wykonać drugą część (PUNKT 2) skryptu `projekt_UDT_init`, która stworzy odpowiednie tabele i funkcje.

6.4 Uruchomienie aplikacji

Ostatnim krokiem jest uruchomienie aplikacji. Należy otworzyć projekt `Projekt_UDT_aplikacja` w Visual Studio 2008 i wybrać `Build` a następnie uruchomić klikając na zieloną strzałkę.

7 Bibliografia

https://newton.fis.agh.edu.pl/~antek/read_pdf.php?file=BD2_L09_CLR.pdf