

# LABORATORIUM SYSTEMÓW MOBILNYCH

## STWORZENIE BIBLIOTEKI SŁUŻĄCEJ DO WCZYTYWANIA DANYCH Z PLIKÓW MIF, SHP I ICH KONWERSJI DO POSTACI RELACYJNEJ

### I. Temat ćwiczenia

Stworzenie biblioteki służącej do wczytywania danych z plików MIF, SHP i ich konwersji do postaci relacyjnej

### II. Wymagania

Znajomość SQL-a

Znajomość sposobu odczytywania plików w języku C#

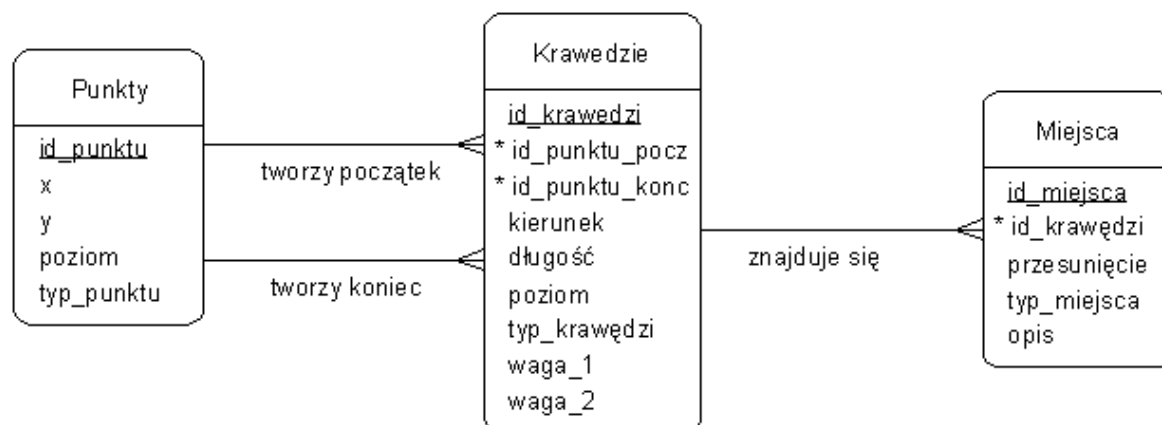
### III. Ćwiczenie

#### 1. Utworzenie modułu konwersji danych przestrzennych

Celem ćwiczenia jest zaprezentowanie studentom metod konwersji formatów przechowywania danych geograficznych MIF jak i SHP do modelu relacyjnego.

Tworzona aplikacja musi być w pełni funkcjonalna, a także przeznaczona na urządzenia mobilne (*PocketPC*) i korzystać z bibliotek łączonych dynamicznie (*dll*). Dla każdego formatu pliku powinna być stworzona osobna biblioteka z zestawem klas odpowiedzialny za jego konwersję.

W celu wykonania powyższej aplikacji należy w pierwszej kolejności stworzyć nową bazę danych ze schematem pokazanym na rysunku 1.



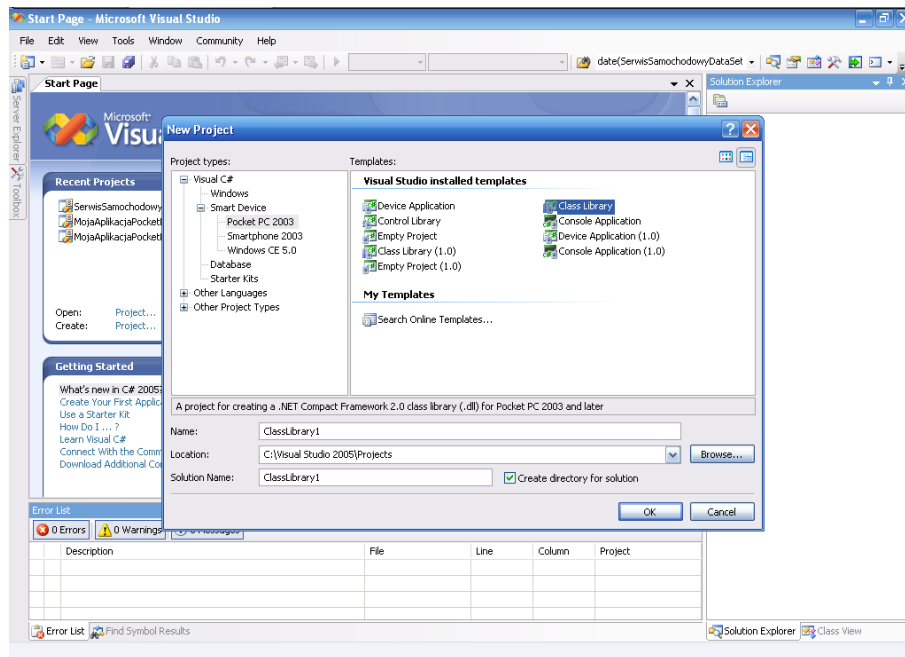
Rysunek 1 – Schemat relacji bazy danych przechowywania danych przestrzennych

Proces tworzenia nowej bazy danych na urządzenia mobilne z wykorzystaniem technologii Microsoft SQL Server Mobile Edition, jak i tworzenie nowego schematu relacji zostały opisane w poprzednich ćwiczeniach.

Przejdźmy do krótkiej charakterystyki wspomnianego schematu relacji. Składa się on z następujących tabel:

- **Punkty**
  - **id\_punktu** // identyfikator punktu, klucz podstawowy relacji
  - **x** // współrzędna x
  - **y** // współrzędna y
  - **poziom** // współrzędna z
  - **typ\_punktu**
- **Krawędzie**
  - **id\_krawedzi** // identyfikator krawędzi, klucz podstawowy relacji
  - **id\_punktu\_pocz** // klucz obcy wskazujący na relację Punkty
  - **id\_punktu\_konc** // klucz obcy wskazujący relację Punkty
  - **kierunek** // podawany w stopniach
  - **długość** // odległość euklidesowa początku i końca
  - **typ\_krawedzi**
- **Miejsca**
  - **id\_miejsca** // identyfikator miejsca, klucz podstawowy relacji
  - **id\_krawedzi** // klucz obcy wskazujący relację Krawędzie
  - **przesunięcie** // minimalna odległość pomiędzy miejscem a najbliższą krawędzią
  - **typ\_miejsca**
  - **opis**

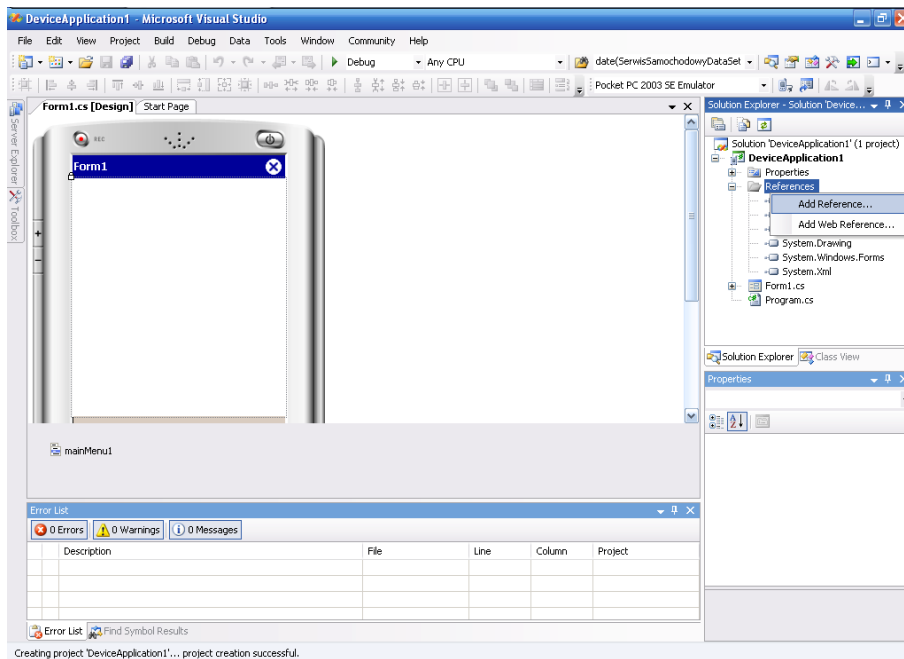
Kolejnym krokiem tworzenia aplikacji jest przygotowanie bibliotek dla każdego z formatów. W tym celu po uruchomieniu środowiska Microsoft Visual Studio 2005 tworzymy nowy projekt wybierając z menu **File -> New -> Project**, a następnie jako **Projekt Type** wskazujemy **Smart Device -> Pocket PC 2003 -> Class Library** (Rysunek 2).



Rysunek 2 – Okno dialogowe wyboru nowego projektu (biblioteki łączonej dynamicznie – dll)

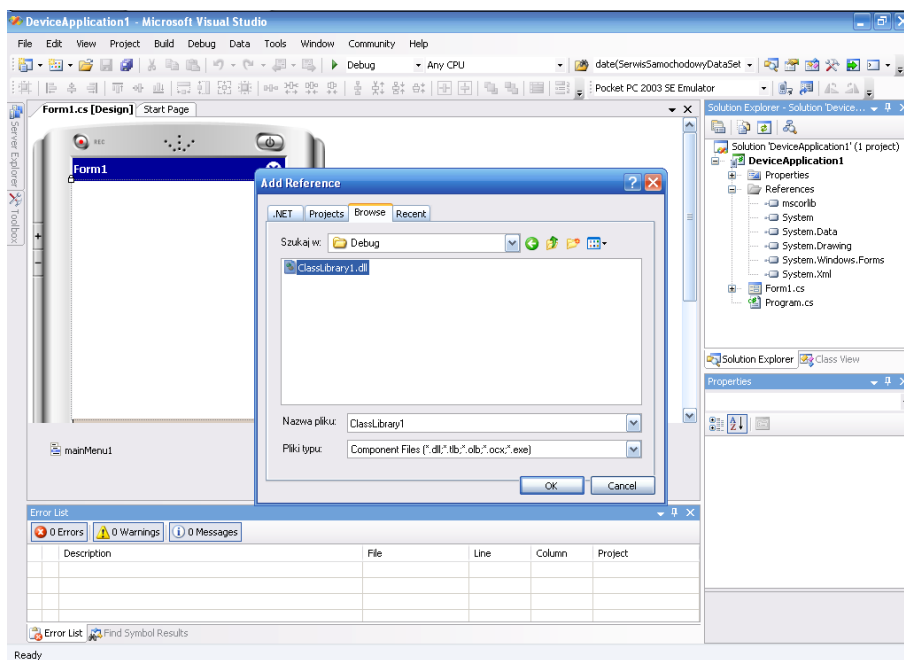
Po utworzeniu biblioteki niezbędne jest jej dodanie do tworzonej aplikacji.

Dodanie biblioteki dll do utworzonej aplikacji dla urządzenia Pocket PC dokonujemy poprzez wskazanie folderu **References** w gałęzi projektu w oknie **Solution Explorer** i wybranie z menu kontekstowego opcji **Add Reference** (Rysunek 3).



**Rysunek 3 – Dodanie referencji do biblioteki**

W wyświetlonym oknie dialogowym przechodzimy do zakładki **Browse**, a następnie wskazujemy utworzoną bibliotekę dll (Rysunek 4).



**Rysunek 4 – Okno dialogowe wyboru biblioteki**

Tworzenie biblioteki łączonej dynamicznej.

Aby możliwe było odwoływanie się do bazy danych z poziomu kodu źródłowego, do utworzonego projektu *Class Library* niezbędne jest dodanie odwołania do biblioteki *System.Data.SqlServerCe.dll* znajdującej się w folderze *Microsoft Visual Studio 8\Common7\IDE* (dokonujemy tego w podobny sposób jak w przypadku dodawania referencji do utworzonych bibliotek).

### Tworzenie bazy danych:

Klasa *SqlCeEngine* umożliwia zarządzanie bazą danych (jej tworzenie, modyfikowanie i usuwanie) na urządzeniu mobilnym (Listing 1).

```
SqlCeEngine engine = new SqlCeEngine("Data Source='Test.sdf'");
if (!(File.Exists("Test.sdf")))
    engine.CreateDatabase();
```

Listing 1 – Utworzenie bazy danych

W konstruktorze klasy podajemy nazwę pliku mobilnej bazy danych.

### Nawiązanie połączenia z bazą danych:

Nawiązanie połączenia z bazą danych możliwe jest dzięki klasie *SqlCeConnection*, w konstruktorze której podajemy adres bazy danych wykorzystując zdefiniowany wcześniej obiekt *engine* lub też podając adres istniejącej bazy danych (Listing 2).

```
SqlCeConnection connection = new SqlCeConnection(engine.LocalConnectionString);
connection.Open();
```

Listing 2 – Nawiązanie połączenia z bazy danych

### Wykonywanie poleceń:

Aby wykonać polecenie SQL wykorzystujemy obiekt klasy *SqlCeCommand*. Polecenie SQL definiujemy we własności *CommandText*, a następnie wywołujemy jedną z trzech metod (Listing 3):

- **ExecuteScalar()** – zwraca pojedynczą wartość liczbową (np. wynik funkcji COUNT, MAX itp.),
- **ExecuteQuery()** – zwraca zbiór krotek z bazy danych spełniających warunek klauzuli WHERE zapytania,
- **ExecuteNonQuery()** – wykonuje polecenie modyfikacji bazy danych (DML) takie jak: INSERT, DELETE, UPDATE,
- **ExecuteReader()** – zwraca kursor iterujący po źródle danych. Wymagane jest otwarte połączenie z bazą danych, a także zasycie kodu obsługi zwracanych danych w kodzie aplikacji.

```
SqlCeCommand command = connection.CreateCommand();
command.CommandText = "SELECT count(*) FROM products";
int result = (System.Int32)(command.ExecuteScalar());
```

Listing 3 – Wykonanie polecenia na bazie danych

Po wykonaniu operacji na bazie danych należy pamiętać o zamknięciu połączenia z bazą danych (metoda *Close()* obiektu klasy *SqlCeConnection*) w celu zwolnienia zajmowanych zasobów.

## **2. Zadanie**

Stworzyć aplikację konwertującą przestrzenne dane geograficzne zapisane w formatach MIF oraz SHP do postaci relacyjnej – informacje o obiektach geograficznych powinny być zapisane w bazie danych (Microsoft SQL Server Mobile Edition) wg podanego powyżej schematu relacji. Aplikacja powinna wykorzystywać biblioteki *dll* – po jednej dla każdego formatu (MIF oraz SHP).