Połatyńska Anna

**Zakład Metrologi Interdyscyplinarnej, Pracownia Wsparcia Metrologii**

Główny Urząd Miar

Warszawa

luty 2019

Program RemaGUM

Opis rozwiązania wersja 0.3

1. Zapotrzebowanie

Oprogramowanie napisanie dla potrzeb **Warsztatu Głównego Urzędu Miar.** Program ma umożliwić zarządzanie warsztatem i magazynem materiałów i normaliów a tym samym ułatwić pracę osobie zarządzającej warsztatem i technikowi.

1. Cel powstania

Program ewidencyjny maszyn i urządzeń w warsztacie GUM, oraz stanów magazynowych warsztatu. Stanowi on punkt wyjścia do stworzenia rozwiązań do zarządzania majątkiem poszczególnych laboratoriów GUM.

**Cel powstania**: ułatwienie ewidencjonowania i monitorowania stanu technicznego oraz napraw maszyn w warsztacie, oraz bieżące śledzenie stanów magazynowych .

Program składa się z dwóch modułów:

**1. spis maszyn -** poprzez bieżącą aktualizację np.: operatora maszyny, osoby zarządzającej, pomieszczenia, dodawanie nowego sprzętu i usuwania zużytego oraz monitorowania stanu technicznego i planu napraw i czasu wyłączenia z eksploatacji poszczególnych maszyn.

**2. spis magazynowy** - poprzez aktualizację, bieżącego śledzenia dostępnych ilości poszczególnych produktów, ich bieżącego zużycia, ilości odpadów itp., jak również informować o konieczności dokonania zakupów w przypadku przekroczenia stanu minimalnego. Spisy magazynowe powinny zawierać również stany magazynowe: wszelkich potrzebnych narzędzi   
i drobnych sprzętów warsztatowych jak: wiertła, klucze, frezy, gwintowniki itd. , z możliwością odpisywania jako zużyte, i w przypadku braku niezbędnej części – generowania informacji o konieczności zakupu.

1. Wykorzystanie programu

RemaGUM jest programem desktopowym, po umieszczeniu go w katalogu wspólnym program staje się dostępny do współdzielenia przez osobę zarządzającą warsztatem/magazynem, technologa, pracowników warsztatu oraz pracowników magazynu. Projekt może być rozwinięty o funkcję logowania użytkowników o różnych uprawnieniach, co dawałoby im różny dostęp do poszczególnych zakładek programu.

1. Środowisko pracy

**RemaGUM** może być uruchomiony na komputerze PC pod kontrolą Windows XP lub nowszym. Aktualna wersja programu skompilowana jest w środowisku Microsoft Visual Studio 2017.   
W większości przypadków program nie wymaga instalacji na komputerach z pakietem MS Office.

W obecnej konfiguracji program wykonawczy (RemaGUM.exe) zapisany jest w folderze sieciowym.

**RemaGUM** współpracuje z dowolną wersją bazy Access i umożliwia korzystanie z jej zasobów.

1. Komponenty programu

Program został zaprojektowany, napisany i wdrożony w Pracowni Wsparcia Metrologii Zakładu Metrologi Interdyscyplinarnej Głównego Urzędu Miar. Napisany w **języku C#** przy wykorzystaniu środowiska **Microsoft Visual Studio Enterprise 2017**. Dane obsługuje baza danych **MSAccess** przy użyciu języka zapytań **SQL**.

Komponenty programu:

1. pliki wykonawczy rozwiązania – **aplikacja RemaGUM** (.exe),
2. baza danych **MS Access - RemaGUM**,
3. plik ze ścieżką dostępu do bazy danych <ConnectionString> - **settings** (.xml),
4. skompilowany plik **pomocy bezpośredniej** w formacie HTML (.chm) **help** HTMLWorkshop,
5. plik konfiguracyjny - XML (exe.config)
6. Baza danych debugowania programu – RemaGUM.pdb.
7. Zawartość projektu

Cały projekt zawarty jest w katalogach:

* **…\RemaGUM** – projekt rozwiązania wytworzonego w środowisku Visual Studio zawierający pełny kod źródłowy **C#** oraz wszystkie niezbędne komponenty do modyfikacji i rozwinięcia, ponownej kompilacji oraz dystrybucji programu.
* **……\RemaGUM \Docs\Help** obejmuje dokumenty stanowiące składniki pomocy bezpośredniej w programie tzw. help.
* **…\RemaGUM \bin\Debug\RemaGUM.mbd** – baza danych MS Access (hasło: zlom).

1. Opis bazy danych MS ACCESS RemaGUM

Baza danych zosatała zaprojektowana z uwzglednieniem specyficznych potrzeb warsztatu GUM. Na dzień zamówienia programu RemaGUM nie było okreslone jakiego typu relacje mają być uwzglednione w bazie danych, z uwagi iż w przysłości może zaistnieć potrzeba zwiększenia elastyczności powiązań pomiędzy tabelami, wprowadzone zostały tabele pośrednie dające możliwość stworzenia realacji multi-multi.

### 7.1 Budowa modułu Magazyn

Dla uproszczenia w tabeli Materiały mieszczą się materiały i normalia warsztatowe (ci sami dostawcy dla obu kategorii). Rozróżnienie na poszczególne typy (normalia/materiały) występuje przy użyciu danych słownikowych. Główną relacją jest połaczenie Materiałów z Dostawcami dlatego przyjęto, iż relacje multi-multi będą się odbywały poprzez tabelę posrednią Dostawca\_Material. Umożliwia to przypisanie wybranemu materiałowi wielu dostawców. Dane słownikowe zawarte są w tabelach: Rodzaj\_mat, Jednostka\_miar.



*Rys 1. Tabela Materiały – relacje.*

### 7.2 Budowa modułu Maszyny

Tabela Maszyny przechowuje dane poszczególnych maszyn. Tabela ta komunikuje się z tabelą Dysponent za pomocą tabeli pośredniej Maszyny\_Dysponent (przy obecnych założeniach dysponent ma pod zarządzaniem kilka maszyn) oraz z tabelą Operator za pośrednictwem tabeli MaszynyOperator. Umożliwia to przypisanie konktetnej maszynie wielu operatorów.

Dane słownikowe zawarte są w tabelach: Propozycja, Wykorzystanie, Stan\_techniczny, Kategoria i Dzial.



*Rys 2. Tabela Maszyny – relacje.*

1. Opcje kompilacji

Program został napisany w C# w środowisku Microsoft Visual Studio Enterprise 2017.

Konfiguracja projekt rozwiązania programistycznego RemaGUM:

Aplikacja:

* Nazwa zestawu: RemaGUM
* Domyślna przestrzeń nazw: RemaGUM
* Platforma docelowa: .NET Framework 4 Client Profile
* Typ wyjściowy: Aplikacja Systemu Windows
* Obiekt uruchomieniowy: Not set

Kompilacja:

* Konfiguracja: Active (Debug)
* Platforma: Active (Any CPU)
* Symbole kompilacji warunkowej: Zdefiniuj stałą DEBUG, Zdefiniuj stałą TRACE
* Platforma docelowa: x86
* Zezwalaj na niebezpieczny kod: nie
* Optymalizuj kod: nie
* Poziom ostrzeżenia: 4
* Pomiń ostrzeżenia: brak
* Traktuj ostrzeżenia jako błędy: brak
* Ścieżka wyjściowa: bin\Debug\
* Plik dokumentacji XML: brak
* Zarejestruj w celu interakcji z modelem COM: brak
* Generuj zestaw serializacji: auto

Debug:

* Uruchom projekt: tak

1. Użyte komponenty środowiska VS

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.IO;

using System.Xml;

using System.Diagnostics;

using System.Threading;

using System.Data.SqlClient;

using System.Configuration;

using System.Data.OleDb;

1. Opis funkcji i klas

W niniejszym opisie przyjmuje się zasadę, że argumenty funkcji, które nie zostały opatrzone komentarzem są predefiniowane w VisualStudio.

### Klasa Main – start aplikacji.

namespace RemaGUM

{

/// <summary>

/// Class Program.

/// </summary>

static class Program

{

/// <summary>

/// Główny punkt wejścia dla aplikacji.

/// </summary>

[STAThread]

static void Main()

}

}

### Klasa Rest – hasło bazy danych MSACCESS.

namespace RemaGUM

{

/// <summary>

/// Class Rest.

/// </summary>

class Rest

{

/// <summary>

/// Połączenie z bazą danych.

/// </summary>

/// <param name="connString">The connection string.</param>

/// <returns>System.String.</returns>

public string dbConnection(string connString)

}// class Rest

}//namespace nsRest

### Obsług połączeń z bazą danych

namespace nsAccess2DB

{

/// <summary>

/// Class dbConnection.

/// </summary>

class dbConnection

{

/// <exclude />

private OleDbDataAdapter \_adapter;

/// <exclude />

private OleDbConnection \_conn;

/// <exclude />

public string \_error = string.Empty; //komunikat błedu

/// <summary>

/// Initializes a new instance of the <see cref="dbConnection"/> class.

/// </summary>

/// <param name="connString">The connection string.</param>

/// <constructor>

/// Połączenie z bazą.

/// </constructor>

public dbConnection(string connString)

/// <summary>

/// Opens the connection.

/// </summary>

/// <returns>OleDbConnection.</returns>

/// <method>

/// Otwiera połaczenie jeśli jest zamknięte lub przerwane.

/// </method>

private OleDbConnection openConnection()

/// <summary>

/// Executes the select query.

/// </summary>

/// <param name="query">The query.</param>

/// <param name="parameters">The parameters.</param>

/// <returns>DataTable.</returns>

/// <method>

/// Polecenie wyboru.

/// </method>

public DataTable executeSelectQuery(String query, OleDbParameter[] parameters)

/// <summary>

/// Executes the insert query.

/// </summary>

/// <param name="query">The query.</param>

/// <param name="parameters">The parameters.</param>

/// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>

/// <method>

/// Polecenie dodania.

/// </method>

public bool executeInsertQuery(String query, OleDbParameter[] parameters)

/// <summary>

/// Executes the delete query.

/// </summary>

/// <param name="query">The query.</param>

/// <param name="parameters">The parameters.</param>

/// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>

/// <method>

/// Polecenie usunięcia.

/// </method>

public bool executeDeleteQuery(String query, OleDbParameter[] parameters)

/// <summary>

/// Executes the update query.

/// </summary>

/// <param name="query">The query.</param>

/// <param name="parameters">The parameters.</param>

/// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>

/// <method>

/// Polecenie aktualizacji.

/// </method>

public bool executeUpdateQuery(String query, OleDbParameter[] parameters)

}// class dbConnection

//---------------------------------------------------------------> MaszynyVO

/// <summary>

/// Klasa wymiany danych z tabelą Maszyny

/// </summary>

public class MaszynyVO

{

/// <summary>

/// Konstruktor wymiany danych z tabelą Maszyny

/// </summary>

//public MaszynyVO (){ }

public MaszynyVO() { }

// gettery i settery

/// <summary>

/// Gets or sets the identyfikator.

/// </summary>

/// <value>The identyfikator.</value>

public int Identyfikator

/// <summary>

/// Gets or sets the kategoria.

/// </summary>

/// <value>The kategoria.</value>

public string Kategoria

/// <summary>

/// Gets or sets the nazwa.

/// </summary>

/// <value>The nazwa.</value>

public string Nazwa

/// <summary>

/// Gets or sets the typ.

/// </summary>

/// <value>The typ.</value>

public string Typ

/// <summary>

/// Gets or sets the nr inwentarzowy.

/// </summary>

/// <value>The nr inwentarzowy.</value>

public string Nr\_inwentarzowy

/// <summary>

/// Gets or sets the nr fabryczny.

/// </summary>

/// <value>The nr fabryczny.</value>

public string Nr\_fabryczny

/// <summary>

/// Gets or sets the rok produkcji.

/// </summary>

/// <value>The rok produkcji.</value>

public string Rok\_produkcji

/// <summary>

/// Gets or sets the producent.

/// </summary>

/// <value>The producent.</value>

public string Producent

/// <summary>

/// Gets or sets the zdjecie.

/// </summary>

/// <value>The zdjecie.</value>

public string Zdjecie

/// <summary>

/// Gets or sets the zawartosc pliku.

/// </summary>

/// <value>The zawartosc pliku.</value>

public byte[] Zawartosc\_pliku

/// <summary>

/// Gets or sets the rozszerz ZDJ.

/// </summary>

/// <value>The rozszerz ZDJ.</value>

public string Rozszerz\_zdj

/// <summary>

/// Gets or sets the nazwa dysponent.

/// </summary>

/// <value>The nazwa dysponent.</value>

public string Nazwa\_dysponent

/// <summary>

/// Gets or sets the nr pom.

/// </summary>

/// <value>The nr pom.</value>

public string Nr\_pom

/// <summary>

/// Gets or sets the dzial.

/// </summary>

/// <value>The dzial.</value>

public string Dzial

/// <summary>

/// Gets or sets the nr prot BHP.

/// </summary>

/// <value>The nr prot BHP.</value>

public string Nr\_prot\_BHP

/// <summary>

/// Gets or sets the data ost przegl.

/// </summary>

/// <value>The data ost przegl.</value>

public int Data\_ost\_przegl

/// <summary>

/// Gets or sets the data kol przegl.

/// </summary>

/// <value>The data kol przegl.</value>

public int Data\_kol\_przegl

/// <summary>

/// Gets or sets the uwagi.

/// </summary>

/// <value>The uwagi.</value>

public string Uwagi

/// <summary>

/// Gets or sets the wykorzystanie.

/// </summary>

/// <value>The wykorzystanie.</value>

public string Wykorzystanie

/// <summary>

/// Gets or sets the stan techniczny.

/// </summary>

/// <value>The stan techniczny.</value>

public string Stan\_techniczny

/// <summary>

/// Gets or sets the propozycja.

/// </summary>

/// <value>The propozycja.</value>

public string Propozycja

/// <summary>

/// Gets or sets the rok ost przeg.

/// </summary>

/// <value>The rok ost przeg.</value>

public int Rok\_ost\_przeg

/// <summary>

/// Gets or sets the mc ost przeg.

/// </summary>

/// <value>The mc ost przeg.</value>

public int Mc\_ost\_przeg

/// <summary>

/// Gets or sets the dz ost przeg.

/// </summary>

/// <value>The dz ost przeg.</value>

public int Dz\_ost\_przeg

/// <summary>

/// Gets or sets the rok kol przeg.

/// </summary>

/// <value>The rok kol przeg.</value>

public int Rok\_kol\_przeg

/// <summary>

/// Gets or sets the mc kol przeg.

/// </summary>

/// <value>The mc kol przeg.</value>

public int Mc\_kol\_przeg

/// <summary>

/// Gets or sets the dz kol przeg.

/// </summary>

/// <value>The dz kol przeg.</value>

public int Dz\_kol\_przeg

}// class MaszynyVO

//klasa dostępu (Data Access Object) do tabeli Maszyny ----------> DAO

/// <summary>

/// Class MaszynyDAO.

/// </summary>

public class MaszynyDAO

{

/// <summary>

/// The connection

/// </summary>

private dbConnection \_conn;

/// <summary>

/// The error

/// </summary>

public string \_error = string.Empty;

/// <summary>

/// Initializes a new instance of the <see cref="MaszynyDAO"/> class.

/// </summary>

/// <param name="connString">The connection string.</param>

/// <construktor>

/// Konstruktor

/// </construktor>

public MaszynyDAO(string connString)

/// <summary>

/// Selects the query.

/// </summary>

/// <param name="query">The query.</param>

/// <returns>DataTable.</returns>

public DataTable selectQuery(string query)

/// <summary>

/// Zwraca tabelę wszystkich Maszyn.

/// </summary>

/// <returns>Tabele.</returns>

public DataTable select()

/// <summary>

/// Zwraca tabelę wszystkich Maszyn po ID.

/// </summary>

/// <param name="Identyfikator">The identyfikator.</param>

/// <returns>Tabele.</returns>

public DataTable select(int Identyfikator)

/// <summary>

/// Selects the nazwa.

/// </summary>

/// <param name="Nazwa">The nazwa.</param>

/// <returns>DataTable.</returns>

public DataTable selectNazwa(string Nazwa)

/// <summary>

/// Zwraca z tabeli Maszyny Zdjęcie wybranej maszyny.

/// </summary>

/// <param name="Zdjecie">The zdjecie.</param>

/// <returns>DataTable.</returns>

public DataTable selectZdjecie(string Zdjecie)

/// <summary>

/// Wprowadza nowy rekord.

/// </summary>

/// <param name="VO">Obiekt wymiany danych.</param>

/// <returns>Wartość logiczna powodzenia operacji.</returns>

public bool insert(nsAccess2DB.MaszynyVO VO)

/// <summary>

/// Aktualizuje rekord z wyjątkiem ID.

/// </summary>

/// <param name="VO">Obiekt wymiany danych</param>

/// <returns>Wartość logiczna powodzenia operacji.</returns>

public bool update(nsAccess2DB.MaszynyVO VO)

/// <summary>

/// Deletes the specified identyfikator.

/// </summary>

/// <param name="Identyfikator">The identyfikator.</param>

/// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>

public bool delete(int Identyfikator)

// warstawa operacji biznesowych tabeli Maszyny ---> BUS

/// <summary>

/// Class MaszynyBUS.

/// </summary>

public class MaszynyBUS

{

/// <summary>

/// Konstruktor.

/// </summary>

/// <param name="connString">ConnectionString.</param>

public MaszynyBUS(string connString)

/// <summary>

/// Wypełnia tablice danych pozycjami.

/// </summary>

public void select()

/// <summary>

/// Wypełnia tablice danych pozycjami.

/// </summary>

/// <param name="Identyfikator">ID maszyny.</param>

public void select(int Identyfikator)

/// <summary>

/// Wypełnia tablice danych pozycjami.

/// </summary>

/// <param name="Nazwa">The nazwa.</param>

public void selectNazwa(string Nazwa)

/// <summary>

/// Wypełnia tablice danych pozycjami.

/// </summary>

/// <param name="Zdjecie">The zdjecie.</param>

public void selectZdjecie(string Zdjecie)

/// <summary>

/// Dowolne zapytanie z formularza.

/// </summary>

/// <param name="query">The query.</param>

public void selectQuery(string query)

/// <summary>

/// Wprowadza rekord do tabeli.

/// </summary>

/// <param name="VO">Obiekt wymiany danych.</param>

/// <returns>Wartość logiczna powodzenia akcji.</returns>

private bool insert(nsAccess2DB.MaszynyVO VO)

/// <summary>

/// Aktualizuje rekord z wyjątkiem Identyfikatora Maszyny.

/// </summary>

/// <param name="VO">Obiekt wymiany danych.</param>

/// <returns>Wartość logiczna powodzenia akcji.</returns>

private bool update(nsAccess2DB.MaszynyVO VO)

/// <summary>

/// Usuwa rekord po identyfikatorze.

/// </summary>

/// <param name="Identyfikator">The identyfikator.</param>

/// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>

public bool delete(int Identyfikator)

/// <summary>

/// Wypełnia tablice.

/// </summary>

/// <param name="dt">Tabela danych.</param>

private void fillTable(DataTable dt)

/// <summary>

/// Przemieszcza indeks w tablicy danych o jedną pozycję.

/// </summary>

public void skip()

/// <summary>

/// Przemieszcza indeks w tablicy danych na pozycję pierwszą.

/// </summary>

public void top()

/// <summary>

/// Zmienna logiczna osiągnięcia końca pliku.

/// </summary>

/// <value><c>true</c> if EOF; otherwise, <c>false</c>.</value>

public bool eof

/// <summary>

/// Zwraca liczbę pozycji tablicy.

/// </summary>

/// <value>The count.</value>

public int count

/// <summary>

/// Zwraca daną okrśloną wskaźnikiem pozycji.

/// </summary>

/// <value>The vo.</value>

public MaszynyVO VO

/// <summary>

/// Ustawia wskaźnik pozycji.

/// </summary>

/// <value>The index.</value>

public int idx

/// <summary>

/// Sprawdza istnienie rekordu.

/// </summary>

/// <param name="Identyfikator">ID maszyny.</param>

/// <returns>Wynik logiczny sprawdzenia.</returns>

private bool exists(int Identyfikator)

/// <summary>

/// Zwraca indeks pozycji.

/// </summary>

/// <param name="Identyfikator">ID maszyny.</param>

/// <returns>Indeks pozycji. -1 oznacza brak identyfikatora.</returns>

public int getIdx(int Identyfikator)

/// <summary>

/// Zwraca maszynę o wskazanym identyfikatorze.

/// </summary>

/// <param name="Identyfikator">Identyfikator maszyny.</param>

/// <returns>Maszyny. Jeśli ID==-1 to maszyny nie znaleziono.</returns>

public MaszynyVO GetVO(int Identyfikator)

/// <summary>

/// Dodaje lub aktualizuje rekord.

/// </summary>

/// <param name="VO">Obiekt wymiany danych.</param>

/// <returns>Wynik powodzenia akcji.</returns>

public bool write(nsAccess2DB.MaszynyVO VO)

}//public class MaszynyBUS

/// <summary>

/// Klasa wymiany danych z tabelą Czestotliwosc.

/// </summary>

public class WykorzystanieVO

//Klasa dostępu (Data Access Object) do tabeli Wykorzystanie.

/// <summary>

/// Class WykorzystanieDAO.

/// </summary>

public class WykorzystanieDAO

{

/// <summary>

/// Initializes a new instance of the <see cref="WykorzystanieDAO"/> class.

/// </summary>

/// <param name="connString">The connection string.</param>

/// <constructor>

/// Konstruktor.

/// </constructor>

public WykorzystanieDAO(string connString)

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <returns>DataTable.</returns>

public DataTable select()

}//class WykorzystanieDAO

//Warstwa operacji biznesowaych tabeli Wykorzystanie.

/// <summary>

/// Class WykorzystanieBUS.

/// </summary>

public class WykorzystanieBUS

{

/// <summary>

/// Konstruktor.

/// </summary>

/// <param name="connString">ConnectionStrine.</param>

public WykorzystanieBUS(string connString)

/// <summary>

/// Wypełnia tablice.

/// </summary>

/// <param name="dt">Tabela danych.</param>

private void fillTable(DataTable dt)

/// <summary>

/// Wypełnia tablice danych pozycjami.

/// </summary>

public void select()

/// <summary>

/// Przemieszcza indeks w tablicy danych o jedną pozycję.

/// </summary>

public void skip()

/// <summary>

/// Przemieszcza indeks w tablicy danych na pozycję pierwszą.

/// </summary>

public void top()

/// <summary>

/// Zmienna logiczna osiągnięcia końca pliku.

/// </summary>

/// <value><c>true</c> if EOF; otherwise, <c>false</c>.</value>

public bool eof

/// <summary>

/// Zwraca liczbę pozycji tablicy.

/// </summary>

/// <value>The count.</value>

public int count

/// <summary>

/// Zwraca daną okrśloną wskaźnikiem pozycji.

/// </summary>

/// <value>The vo.</value>

public WykorzystanieVO VO

/// <summary>

/// Ustawia wskaźnik pozycji.

/// </summary>

/// <value>The index.</value>

public int idx

/// <summary>

/// Sprawdza istnienie rekordu.

/// </summary>

/// <param name="Wykorzystanie">The wykorzystanie.</param>

/// <returns>Wynik logiczny sprawdzenia.</returns>

public bool exists(String Wykorzystanie)

/// <summary>

/// Zwraca indeks pozycji.

/// </summary>

/// <param name="Wykorzystanie">Identyfikator Wykorzystania maszyny</param>

/// <returns>Indeks pozycji. -1 oznacza brak identyfikatora Wykorzystania.</returns>

public int getIdx(string Wykorzystanie)

}//class WykorzystanieBUS

/////////////////////////////////////////////////// Kategoria - dane słownikowe

/// <summary>

/// Klasa wymiany danych z tabelą Kategoria.

/// </summary>

public class KategoriaVO

//Klasa dostępu (Data Access Object) do tabeli Kategoria.

/// <summary>

/// Class KategoriaDAO.

/// </summary>

public class KategoriaDAO

//Warstwa operacji biznesowaych tabeli Kategoria.

/// <summary>

/// Class KategoriaBUS.

/// </summary>

public class KategoriaBUS

{

/// <summary>

/// Konstruktor.

/// </summary>

/// <param name="connString">ConnectionString.</param>

public KategoriaBUS(string connString)

/// <summary>

/// Wypełnia tablice.

/// </summary>

/// <param name="dt">Tabela danych.</param>

private void fillTable(DataTable dt)

/// <summary>

/// Wypełnia tablice danych pozycjami.

/// </summary>

public void select()

/// <summary>

/// Przemieszcza indeks w tablicy danych na pozycję pierwszą.

/// </summary>

public void top()

/// <summary>

/// Zmienna logiczna osiągnięcia końca pliku.

/// </summary>

/// <value><c>true</c> if EOF; otherwise, <c>false</c>.</value>

public bool eof

/// <summary>

/// Zwraca liczbę pozycji tablicy.

/// </summary>

/// <value>The count.</value>

public int count

/// <summary>

/// Zwraca daną okrśloną wskaźnikiem pozycji.

/// </summary>

/// <value>The vo.</value>

public KategoriaVO VO

/// <summary>

/// Ustawia wskaźnik pozycji.

/// </summary>

/// <value>The index.</value>

public int idx

/// <summary>

/// Sprawdza istnienie rekordu.

/// </summary>

/// <param name="Nazwa">The nazwa.</param>

/// <returns>Wynik logiczny sprawdzenia.</returns>

public bool exists(String Nazwa)

/// <summary>

/// Zwraca indeks pozycji.

/// </summary>

/// <param name="Nazwa">Identyfikator Kategoria maszyn</param>

/// <returns>Indeks pozycji. -1 oznacza brak identyfikatora Kategori.</returns>

public int getIdx(string Nazwa)

}//class KategoriaBUS

/////////////////////////////////////////////// Propozycja - dane słownikowe

/// <summary>

/// Klasa wymiany danych z tabelą Propozycja.

/// </summary>

public class PropozycjaVO

{

/// <summary>

/// Konstruktor wymiany danych z tabelą Propozycja

/// </summary>

public PropozycjaVO() { }

}//class PropozycjaVO

//Klasa dostępu (Data Access Object) do tabeli Propozycja.

/// <summary>

/// Class PropozycjaDAO.

/// </summary>

public class PropozycjaDAO

{

/// <summary>

/// Initializes a new instance of the <see cref="PropozycjaDAO"/> class.

/// </summary>

/// <param name="connString">The connection string.</param>

/// <constructor>

/// Konstruktor.

/// </constructor>

public PropozycjaDAO(string connString)

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <returns>DataTable.</returns>

public DataTable select()

}//class PropozycjaDAO

//Warstwa operacji biznesowaych tabeli Propozycja.

/// <summary>

/// Class PropozycjaBUS.

/// </summary>

public class PropozycjaBUS

{

/// <summary>

/// Konstruktor.

/// </summary>

/// <param name="connString">ConnectionString.</param>

public PropozycjaBUS(string connString)

/// <summary>

/// Wypełnia tablice.

/// </summary>

/// <param name="dt">Tabela danych.</param>

private void fillTable(DataTable dt)

/// <summary>

/// Wypełnia tablice danych pozycjami.

/// </summary>

public void select()

/// <summary>

/// Przemieszcza indeks w tablicy danych o jedną pozycję.

/// </summary>

public void skip()

/// <summary>

/// Przemieszcza indeks w tablicy danych na pozycję pierwszą.

/// </summary>

public void top()

/// <summary>

/// Zmienna logiczna osiągnięcia końca pliku.

/// </summary>

/// <value><c>true</c> if EOF; otherwise, <c>false</c>.</value>

public bool eof

/// <summary>

/// Zwraca liczbę pozycji tablicy.

/// </summary>

/// <value>The count.</value>

public int count

/// <summary>

/// Zwraca daną okrśloną wskaźnikiem pozycji.

/// </summary>

/// <value>The vo.</value>

public PropozycjaVO VO

/// <summary>

/// Ustawia wskaźnik pozycji.

/// </summary>

/// <value>The index.</value>

public int idx

/// <summary>

/// Sprawdza istnienie rekordu.

/// </summary>

/// <param name="Nazwa">The nazwa.</param>

/// <returns>Wynik logiczny sprawdzenia.</returns>

public bool exists(String Nazwa)

/// <summary>

/// Zwraca indeks pozycji.

/// </summary>

/// <param name="Nazwa">Identyfikator Propozycja dot maszyn</param>

/// <returns>Indeks pozycji. -1 oznacza brak identyfikatora Propozycja.</returns>

public int getIdx(string Nazwa)

}//class PropozycjaBUS

/////////////////////////////////////////// Stan\_techniczny - dane słownikowe.

/// <summary>

/// Klasa wymiany danych z tabelą Stan\_techniczny.

/// </summary>

public class Stan\_technicznyVO

{

/// <summary>

/// Konstruktor wymiany danych z tabelą Stan\_techniczny

/// </summary>

public Stan\_technicznyVO() { }

/// <summary>

/// Gets or sets the nazwa.

/// </summary>

/// <value>The nazwa.</value>

public string Nazwa

}//class Stan\_technicznyVO

//Klasa dostępu (Data Access Object) do tabeli Stan\_techniczny.

/// <summary>

/// Class Stan\_technicznyDAO.

/// </summary>

public class Stan\_technicznyDAO

{

/// <summary>

/// Initializes a new instance of the <see cref="Stan\_technicznyDAO"/> class.

/// </summary>

/// <param name="connString">The connection string.</param>

/// <constructor>

/// Konstruktor.

/// </constructor>

public Stan\_technicznyDAO(string connString)

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <returns>DataTable.</returns>

public DataTable select()

}//class Stan\_technicznyDAO

//Warstwa operacji biznesowaych tabeli Stan\_techniczny.

/// <summary>

/// Class Stan\_technicznyBUS.

/// </summary>

public class Stan\_technicznyBUS

/// <summary>

/// Konstruktor.

/// </summary>

/// <param name="connString">ConnectionString.</param>

public Stan\_technicznyBUS(string connString)

/// <summary>

/// Wypełnia tablice.

/// </summary>

/// <param name="dt">Tabela danych.</param>

private void fillTable(DataTable dt)

/// <summary>

/// Wypełnia tablice danych pozycjami.

/// </summary>

public void select()

/// <summary>

/// Przemieszcza indeks w tablicy danych o jedną pozycję.

/// </summary>

public void skip()

/// <summary>

/// Przemieszcza indeks w tablicy danych na pozycję pierwszą.

/// </summary>

public void top()

/// <summary>

/// Zmienna logiczna osiągnięcia końca pliku.

/// </summary>

/// <value><c>true</c> if EOF; otherwise, <c>false</c>.</value>

public bool eof

/// <summary>

/// Zwraca liczbę pozycji tablicy.

/// </summary>

/// <value>The count.</value>

public int count

/// <summary>

/// Zwraca daną okrśloną wskaźnikiem pozycji.

/// </summary>

/// <value>The vo.</value>

public Stan\_technicznyVO VO

/// <summary>

/// Ustawia wskaźnik pozycji.

/// </summary>

/// <value>The index.</value>

public int idx

/// <summary>

/// Sprawdza istnienie rekordu.

/// </summary>

/// <param name="Nazwa">The nazwa.</param>

/// <returns>Wynik logiczny sprawdzenia.</returns>

public bool exists(String Nazwa)

/// <summary>

/// Zwraca indeks pozycji.

/// </summary>

/// <param name="Nazwa">Identyfikator Stan\_techniczny maszyn</param>

/// <returns>Indeks pozycji. -1 oznacza brak identyfikatora Propozycja.</returns>

public int getIdx(string Nazwa)

}//class Stan\_technicznyBUS

//////////////////////////////////////////////// Dzial - dane słownikowe.

/// <summary>

/// Klasa wymiany danych z tabelą Dzial.

/// </summary>

public class DzialVO

{

/// <summary>

/// Konstruktor wymiany danych z tabelą Dzial

/// </summary>

public DzialVO() { }

}//class DzialVO

//Klasa dostępu (Data Access Object) do tabeli Dzial.

/// <summary>

/// Class DzialDAO.

/// </summary>

public class DzialDAO

{

/// <summary>

/// Initializes a new instance of the <see cref="DzialDAO"/> class.

/// </summary>

/// <param name="connString">The connection string.</param>

/// <constructor>

/// Konstruktor.

/// </constructor>

public DzialDAO(string connString)

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <returns>DataTable.</returns>

public DataTable select()

}//class DzialDAO

//Warstwa operacji biznesowaych tabeli Dzial.

/// <summary>

/// Class DzialBUS.

/// </summary>

public class DzialBUS

{

/// <summary>

/// Konstruktor.

/// </summary>

/// <param name="connString">ConnectionString.</param>

public DzialBUS(string connString)

/// <summary>

/// Wypełnia tablice.

/// </summary>

/// <param name="dt">Tabela danych.</param>

private void fillTable(DataTable dt)

/// <summary>

/// Wypełnia tablice danych pozycjami.

/// </summary>

public void select()

/// <summary>

/// Przemieszcza indeks w tablicy danych o jedną pozycję.

/// </summary>

public void skip()

/// <summary>

/// Przemieszcza indeks w tablicy danych na pozycję pierwszą.

/// </summary>

public void top()

/// <summary>

/// Zmienna logiczna osiągnięcia końca pliku.

/// </summary>

/// <value><c>true</c> if EOF; otherwise, <c>false</c>.</value>

public bool eof

/// <summary>

/// Zwraca liczbę pozycji tablicy.

/// </summary>

/// <value>The count.</value>

public int count

/// <summary>

/// Zwraca daną okrśloną wskaźnikiem pozycji.

/// </summary>

/// <value>The vo.</value>

public DzialVO VO

/// <summary>

/// Ustawia wskaźnik pozycji.

/// </summary>

/// <value>The index.</value>

public int idx

/// <summary>

/// Sprawdza istnienie rekordu.

/// </summary>

/// <param name="Nazwa">The nazwa.</param>

/// <returns>Wynik logiczny sprawdzenia.</returns>

public bool exists(String Nazwa)

/// <summary>

/// Zwraca indeks pozycji.

/// </summary>

/// <param name="Nazwa">Identyfikator Dzial do którego należy maszyna</param>

/// <returns>Indeks pozycji. -1 oznacza brak identyfikatora Kategori.</returns>

public int getIdx(string Nazwa)

}//class DzialBUS

////////////////////////////////////////////////// Operator\_maszyny

/// <summary>

/// Klasa wymiany danych z tabelą Operator\_maszyny.

/// </summary>

public class OperatorVO

{

private string \_Op\_nazwisko = string.Empty; //255

/// <summary>

/// Konstruktor wymiany danych z tabelą Operator\_maszyny

/// </summary>

}//class OperatorVO

//Klasa dostępu (Data Access Object) do tabeli Nazwa\_operatora.

/// <summary>

/// Class OperatorDAO.

/// </summary>

public class OperatorDAO

{

/// <summary>

/// Initializes a new instance of the <see cref="OperatorDAO"/> class.

/// </summary>

/// <param name="connString">The connection string.</param>

/// <constructor>

/// Konstruktor.

/// </constructor>

public OperatorDAO(string connString)

//---> dowolne zapytanie z poziomu Form

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <param name="query">The query.</param>

/// <returns>DataTable.</returns>

public DataTable selectQuery(string query)

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <returns>Tabela Nazwa\_operatora.</returns>

public DataTable select()

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <param name="Identyfikator">The identyfikator.</param>

/// <returns>DataTable.</returns>

public DataTable select(int Identyfikator)

/// <summary>

/// Wprowadza nowy rekord do tabeli Operator\_maszyny.

/// </summary>

/// <param name="VO">Obiekt wymiany danych (insert)</param>

/// <returns>Wartość logiczna powodzenia operacji</returns>

public bool insert(nsAccess2DB.OperatorVO VO)

/// <summary>

/// Aktualizuje rekord z wyjątkiem Identyfikatora Operatora\_maszyny.

/// </summary>

/// <param name="VO">The vo.</param>

/// <returns>Wartość logiczna powodzenia operacji.</returns>

public bool update(nsAccess2DB.OperatorVO VO)

/// <summary>

/// Usuwa rekord.

/// </summary>

/// <param name="Identyfikator">Identyfikator.</param>

/// <returns>Wartość logiczna powodzenia operacji.</returns>

public bool delete(int Identyfikator)

}//class OperatorDAO

//Warstwa operacji biznesowaych tabeli Nazwa\_operatora BUS.

/// <summary>

/// Class OperatorBUS.

/// </summary>

public class OperatorBUS

{

/// <summary>

/// Konstruktor.

/// </summary>

/// <param name="connString">ConnectionString.</param>

public OperatorBUS(string connString)

/// <summary>

/// Wypełnia tablicę pozycjami danych

/// </summary>

/// <param name="query">The query.</param>

public void selectQuery(string query)

/// <summary>

/// Wypełnia tablice pozycjami danych .

/// </summary>

public void select()

/// <summary>

/// Wypełnia tablicę.

/// </summary>

/// <param name="Identyfikator">The identyfikator.</param>

public void select(int Identyfikator)

/// <summary>

/// Wprowadza rekord tabeli.

/// </summary>

/// <param name="VO">Obiekt wymiany danych.</param>

/// <returns>Wartość logiczna powodzenia akcji.</returns>

private bool insert(nsAccess2DB.OperatorVO VO)

/// <summary>

/// Aktualizuje rekord z wyjątkiem numeru protokołu.

/// </summary>

/// <param name="VO">Obiekt wymiany.</param>

/// <returns>Wartość logiczna powodzenia akcji.</returns>

private bool update(nsAccess2DB.OperatorVO VO)

/// <summary>

/// Usuwa rekord.

/// </summary>

/// <param name="Identyfikator">The identyfikator.</param>

public void delete(int Identyfikator)

/// <summary>

/// Wypełnia tablice.

/// </summary>

/// <param name="dt">Tabela danych.</param>

private void fillTable(DataTable dt)

/// <summary>

/// Przemieszcza indeks w tablicy danych o jedną pozycję.

/// </summary>

public void skip()

/// <summary>

/// Przemieszcza indeks w tablicy danych na pozycję pierwszą.

/// </summary>

public void top()

/// <summary>

/// Zmienna logiczna osiągnięcia końca pliku.

/// </summary>

/// <value><c>true</c> if EOF; otherwise, <c>false</c>.</value>

public bool eof

/// <summary>

/// Zwraca liczbę pozycji tablicy.

/// </summary>

/// <value>The count.</value>

public int count

/// <summary>

/// Zwraca daną okrśloną wskaźnikiem pozycji.

/// </summary>

/// <value>The vo.</value>

public OperatorVO VO

/// <summary>

/// Ustawia wskaźnik pozycji.

/// </summary>

/// <value>The index.</value>

public int idx

/// <summary>

/// Sprawdza istnienie rekordu.

/// </summary>

/// <param name="Identyfikator">ID operatora maszyny</param>

/// <returns>Wynik logiczny sprawdzenia.</returns>

public bool exists(int Identyfikator)

/// <summary>

/// Zwraca indeks pozycji.

/// </summary>

/// <param name="Identyfikator">The identyfikator.</param>

/// <returns>Indeks pozycji. -1 oznacza brak identyfikatora.</returns>

public int getIdx(int Identyfikator)

/// <summary>

/// Zwraca indeks pozycji.

/// </summary>

/// <param name="Identyfikator">Identyfikator Operator maszyny</param>

/// <returns>Indeks pozycji. -1 oznacza brak identyfikatora dla Operatora maszyny.</returns>

public OperatorVO GetVO(int Identyfikator)

/// <summary>

/// Dodaje lub aktualizuje rekord.

/// </summary>

/// <param name="VO">Obiekt wymiany danych.</param>

/// <returns>Wynik powodzenia akcji.</returns>

public bool write(nsAccess2DB.OperatorVO VO)

}//class OperatorBUS

///////////////////////////////////////////////// Operator\_maszyny\_Maszyny

/// <summary>

/// Klasa wymiany danych z tabelą Operator\_maszyny\_Maszyny.

/// </summary>

public class Maszyny\_OperatorVO

{

/// <summary>

/// Konstruktor wymiany danych z tabelą Operator\_maszyny\_Maszyny

/// </summary>

public Maszyny\_OperatorVO() { }

}//class Maszyny\_OperatorVO

//Klasa dostępu (Data Access Object) do tabeli Operator\_maszyny\_Maszyny.

/// <summary>

/// Class Maszyny\_OperatorDAO.

/// </summary>

public class Maszyny\_OperatorDAO

{

/// <summary>

/// Initializes a new instance of the <see cref="Maszyny\_OperatorDAO"/> class.

/// </summary>

/// <param name="connString">The connection string.</param>

/// <constructor>

/// Konstruktor.

/// </constructor>

public Maszyny\_OperatorDAO(string connString)

}//Maszyny\_OperatorDAO

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <param name="query">The query.</param>

/// <returns>DataTable.</returns>

public DataTable selectQuery(string query)

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <returns>Tabela Maszyny\_OperatorDAO.</returns>

public DataTable select()

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <returns>DataTable.</returns>

public DataTable select(int ID\_maszyny)

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <param name="ID\_op\_maszyny">The identifier op maszyny.</param>

/// <returns>DataTable.</returns>

public DataTable select(int ID\_maszyny, int ID\_op\_maszyny)

/// <summary>

/// Selects the operator.

/// </summary>

/// <returns>DataTable.</returns>

public DataTable selectOperator()

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <param name="ID\_op\_maszyny">The identifier op maszyny.</param>

/// <returns>DataTable.</returns>

public DataTable selectOperator(int ID\_op\_maszyny)

/// <summary>

/// Wprowadza nowy rekord

/// </summary>

/// <param name="VO">Obiekt wymiany danych (insert)</param>

/// <returns>Wartość logiczna powodzenia operacji</returns>

public bool insert(nsAccess2DB.Maszyny\_OperatorVO VO)

/// <summary>

/// Aktualizuje rekord z wyjątkiem ID.

/// </summary>

/// <param name="VO">The vo.</param>

/// <returns>Wartość logiczna powodzenia operacji.</returns>

public bool update(nsAccess2DB.Maszyny\_OperatorVO VO)

/// <summary>

/// Kasuje rekord po podanym Identyfikatorze.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>

public bool delete(int ID\_maszyny)

/// <summary>

/// Kasuje rekord po podanym Identyfikatorze.

/// </summary>

/// <param name="ID\_op\_maszyny">The identifier op maszyny.</param>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <returns>Wartość logiczna powodzenia operacji.</returns>

public bool delete(int ID\_op\_maszyny, int ID\_maszyny)

}//class Maszyny\_OperatorDAO

//Warstwa operacji biznesowaych tabeli Maszyny\_Operator ---> BUS.

/// <summary>

/// Class Maszyny\_OperatorBUS.

/// </summary>

public class Maszyny\_OperatorBUS

/// <summary>

/// Konstruktor.

/// </summary>

/// <param name="connString">The connection string.</param>

public Maszyny\_OperatorBUS(string connString)

/// <summary>

/// Wypełnia tablice pozycjami danych.

/// </summary>

public void select()

/// <summary>

/// Wypełnia tablice pozycjami danych.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

public void select(int ID\_maszyny)

/// <summary>

/// Wypełnia tablicę pozycjami danych.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <param name="ID\_op\_maszyny">The identifier op maszyny.</param>

public void select(int ID\_maszyny, int ID\_op\_maszyny)

/// <summary>

/// Wypełnia tablicę pozycjami danych.

/// </summary>

internal void selectOperator()

/// <summary>

/// Wypełnia tablicę pozycjami danych.

/// </summary>

/// <param name="ID\_operator">The identifier operator.</param>

public void selectOperator(int ID\_operator)

/// <summary>

/// Wypełnia tablicę pozycjami danych

/// </summary>

/// <param name="query">The query.</param>

public void selectQuery(string query)

/// <summary>

/// Wprowadza rekord do tabeli.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <param name="ID\_op\_maszyny">The identifier op maszyny.</param>

/// <param name="Maszyny\_nazwa">The maszyny nazwa.</param>

/// <returns>Wartość logiczna powodzenia operacji.</returns>

public bool insert(int ID\_maszyny, int ID\_op\_maszyny, string Maszyny\_nazwa)

/// <summary>

/// Aktualizuje rekord z wyjątkiem numeru protokołu.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <param name="ID\_op\_maszyny">The identifier op maszyny.</param>

/// <param name="Maszyny\_nazwa">The maszyny nazwa.</param>

/// <returns>Wartość logiczna powodzenia akcji.</returns>

private bool update(int ID\_maszyny, int ID\_op\_maszyny, string Maszyny\_nazwa)

/// <summary>

/// Usuwa z tabeli pozycję o wskazanych parametrach.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <returns>Wartość logiczna powodzenia akcji.</returns>

public bool delete(int ID\_maszyny)

/// <summary>

/// Usuwa z tabeli pozycję o wskazanych parametrach.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <param name="ID\_op\_maszyny">The identifier op maszyny.</param>

/// <returns>Wartość logiczna powodzenia akcji.</returns>

public bool delete(int ID\_maszyny, int ID\_op\_maszyny)

/// <summary>

/// Wypełnia tablicę.

/// </summary>

/// <param name="dt">Tabela danych.</param>

private void fillTable(DataTable dt)

// <summary>

/// <summary>

/// Skips this instance.

/// </summary>

public void skip()

/// <summary>

/// Przemieszcza indeks w tablicy danych na pozycję pierwszą.

/// </summary>

public void top()

/// <summary>

/// Zmienna logiczna osiągnięcia końca pliku.

/// </summary>

/// <value><c>true</c> if EOF; otherwise, <c>false</c>.</value>

public bool eof

/// <summary>

/// Zwraca liczbę pozycji tablicy.

/// </summary>

/// <value>The count.</value>

public int count

/// <summary>

/// Zwraca daną okrśloną wskaźnikiem pozycji.

/// </summary>

/// <value>The vo.</value>

public Maszyny\_OperatorVO VO

/// <summary>

/// Ustawia wskaźnik pozycji.

/// </summary>

/// <value>The index.</value>

public int idx

/// <summary>

/// Sprawdza istnienie pozycji w tabeli uzyskanej po ostanim poleceniu select.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <param name="ID\_op\_maszyny">The identifier op maszyny.</param>

/// <returns>Wartość logiczna istnienia pozycji.</returns>

public bool exists(int ID\_maszyny, int ID\_op\_maszyny)

/// <summary>

/// Zwraca indeks pozycji.

/// </summary>

/// <param name="ID\_maszyny">ID\_maszyny z taleli Maszyny\_Operator.</param>

/// <returns>Indeks pozycji. -1 oznacza brak identyfikatora.</returns>

public int getIdx(int ID\_maszyny)

/// <summary>

/// Zwraca maszynę o podanym indeksie z tabeli Maszyny\_Operator.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <returns>Maszyny\_OperatorVO.</returns>

public Maszyny\_OperatorVO GetVO(int ID\_maszyny)

/// <summary>

/// Writes the specified vo.

/// </summary>

/// <param name="VO">The vo.</param>

/// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>

public bool write(nsAccess2DB.Maszyny\_OperatorVO VO)

/// <summary>

/// Dodaje pozycję do tabeli.

/// </summary>

/// <param name="VO">The vo.</param>

/// <param name="VOs">The v os.</param>

private void add(Maszyny\_OperatorVO VO, ref Maszyny\_OperatorVO[] VOs)

}//class Maszyny\_OperatorBUS

///////////////////////////////////////////////////// Dysponent

/// <summary>

/// Klasa wymiany danych z tabelą Dysponent .

/// </summary>

public class DysponentVO

{

/// <summary>

/// Konstruktor wymiany danych z tabelą Dysponent.

/// </summary>

public DysponentVO() { }

/// <summary>

/// Gets or sets the identyfikator.

/// </summary>

/// <value>The identyfikator.</value>

public int Identyfikator

/// <summary>

/// Gets or sets the dysp dane.

/// </summary>

/// <value>The dysp dane.</value>

public string Dysp\_dane

/// <summary>

/// Gets or sets the dzial.

/// </summary>

/// <value>The dzial.</value>

public string Dzial

/// <summary>

/// Gets or sets the dysp nazwisko.

/// </summary>

/// <value>The dysp nazwisko.</value>

public string Dysp\_nazwisko

/// <summary>

/// Gets or sets the dysp imie.

/// </summary>

/// <value>The dysp imie.</value>

public string Dysp\_imie

/// <summary>

/// Gets or sets the dysp nazwa.

/// </summary>

/// <value>The dysp nazwa.</value>

public string Dysp\_nazwa

}//class DysponentVO

//Klasa dostępu (Data Access Object) do tabeli Dysponent . ----------> DAO

/// <summary>

/// Class DysponentDAO.

/// </summary>

public class DysponentDAO

{

/// <summary>

/// The connection

/// </summary>

private dbConnection \_conn;

/// <summary>

/// The error

/// </summary>

public string \_error = string.Empty;

/// <summary>

/// Initializes a new instance of the <see cref="DysponentDAO"/> class.

/// </summary>

/// <param name="connString">The connection string.</param>

/// <constructor>

/// Konstruktor.

/// </constructor>

public DysponentDAO(string connString)

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <param name="query">The query.</param>

/// <returns>DataTable.</returns>

public DataTable selectQuery(string query)

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <returns>DataTable.</returns>

public DataTable select()

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <param name="Identyfikator">The identyfikator.</param>

/// <returns>DataTable.</returns>

public DataTable select(int Identyfikator)

/// <summary>

/// Wprowadza nowy rekord do tabeli osoba zatrządzająca.

/// </summary>

/// <param name="VO">Obiekt wymiany danych.</param>

/// <returns>Wartość logiczna powodzenia operacji.</returns>

public bool insert(nsAccess2DB.DysponentVO VO)

/// <summary>

/// Aktualizuje rekord z wyjątkiem Identyfikatora.

/// </summary>

/// <param name="VO">The vo.</param>

/// <returns>Wartość logiczna powodzenia operacji.</returns>

public bool update(nsAccess2DB.DysponentVO VO)

/// <summary>

/// Kasuje rekord po podanym Identyfikatorze.

/// </summary>

/// <param name="Identyfikator">The identyfikator.</param>

/// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>

public bool delete(int Identyfikator)

}//class DysponentDAO

// Warstwa operacji biznesowaych tabeli Dysponent ---> BUS.

/// <summary>

/// Class DysponentBUS.

/// </summary>

public class DysponentBUS

{

/// <summary>

/// Konstruktor.

/// </summary>

/// <param name="connString">ConnectionString.</param>

public DysponentBUS(string connString)

/// <summary>

/// Wypełnia tablice danych pozycjami.

/// </summary>

public void select()

/// <summary>

/// Wypełnia tablicę danych pozycjami.

/// </summary>

/// <param name="Identyfikator">The identyfikator.</param>

public void select(int Identyfikator)

/// <summary>

/// Dowolne zapytanie z formularza.

/// </summary>

/// <param name="query">The query.</param>

public void selectQuery(string query)

/// <summary>

/// Wprowadza rekord do tabeli Osoba zarządzająca.

/// </summary>

/// <param name="VO">Obiekt wymiany danych.</param>

/// <returns>Wartość logiczna powodzenia akcji.</returns>

private bool insert(nsAccess2DB.DysponentVO VO)

/// <summary>

/// Aktualizuje rekord z wyjątkiem ID osoby zarządzającej.

/// </summary>

/// <param name="VO">Obiekt wymiany danych.</param>

/// <returns>Wartość logiczna powodzenia akcji.</returns>

private bool update(nsAccess2DB.DysponentVO VO)

/// <summary>

/// Usuwa rekord po Identyfikatorze.

/// </summary>

/// <param name="Identyfikator">The identyfikator.</param>

/// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>

public bool delete(int Identyfikator)

/// <summary>

/// Wypełnia tablice.

/// </summary>

/// <param name="dt">Tabela danych.</param>

private void fillTable(DataTable dt)

/// <summary>

/// Przemieszcza indeks w tablicy danych o jedną pozycję.

/// </summary>

public void skip()

/// <summary>

/// Przemieszcza indeks w tablicy danych na pozycję pierwszą.

/// </summary>

public void top()

/// <summary>

/// Zmienna logiczna osiągnięcia końca pliku.

/// </summary>

/// <value><c>true</c> if EOF; otherwise, <c>false</c>.</value>

public bool eof

/// <summary>

/// Zwraca liczbę pozycji tablicy.

/// </summary>

/// <value>The count.</value>

public int count

/// <summary>

/// Zwraca daną okrśloną wskaźnikiem pozycji.

/// </summary>

/// <value>The vo.</value>

public DysponentVO VO

/// <summary>

/// Ustawia wskaźnik pozycji.

/// </summary>

/// <value>The index.</value>

public int idx

/// <summary>

/// Sprawdza istnienie rekordu.

/// </summary>

/// <param name="Identyfikator">Identyfikator Dysponenta.</param>

/// <returns>Wynik logiczny sprawdzenia.</returns>

public bool exists(int Identyfikator)

/// <summary>

/// Zwraca indeks pozycji.

/// </summary>

/// <param name="Identyfikator">Identyfikator Dysponenta.</param>

/// <returns>Indeks pozycji. -1 oznacza brak identyfikatora Dysponenta.</returns>

public int getIdx(int Identyfikator)

/// <summary>

/// Zwraca Osobę zarządzającą o podanym Identyfikatorze.

/// </summary>

/// <param name="Identyfikator">The identyfikator.</param>

/// <returns>Jeżeli Identyfikator == -1 oznacza że Osoby Zarządzającej nie znaleziono.</returns>

public DysponentVO GetVO(int Identyfikator)

/// <summary>

/// Dodaje lub aktualizuje rekord.

/// </summary>

/// <param name="VO">The vo.</param>

/// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>

public bool write(nsAccess2DB.DysponentVO VO)

}//class DysponentBUS

/////////////////////////////////////////////// Maszyny\_Dysponent

/// <summary>

/// Klasa wymiany danych z tabelą Maszyny\_Dysponent.

/// </summary>

public class Maszyny\_DysponentVO

{

/// <summary>

/// Konstruktor wymiany danych z tabelą Maszyny\_Dysponent.

/// </summary>

public Maszyny\_DysponentVO() { }

/// <summary>

/// Gets or sets the identifier dysponent.

/// </summary>

/// <value>The identifier dysponent.</value>

public int ID\_dysponent

/// <summary>

/// Gets or sets the identifier maszyny.

/// </summary>

/// <value>The identifier maszyny.</value>

public int ID\_maszyny

/// <summary>

/// Gets or sets the maszyny nazwa d.

/// </summary>

/// <value>The maszyny nazwa d.</value>

public string Maszyny\_nazwa\_D

}//class Maszyny\_DysponentVO

//Klasa dostępu (Data Access Object) do tabeli Maszyny\_Dysponent.

/// <summary>

/// Class Maszyny\_DysponentDAO.

/// </summary>

public class Maszyny\_DysponentDAO

{

/// <summary>

/// Initializes a new instance of the <see cref="Maszyny\_DysponentDAO"/> class.

/// </summary>

/// <param name="connString">The connection string.</param>

/// <constructor>

/// Konstruktor.

/// </constructor>

public Maszyny\_DysponentDAO(string connString)

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <param name="query">The query.</param>

/// <returns>DataTable.</returns>

public DataTable selectQuery(string query)

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <returns>Tabela Maszyny\_DysponentDAO.</returns>

public DataTable select()

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <returns>DataTable.</returns>

public DataTable select(int ID\_maszyny)

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <param name="ID\_dysponent">The identifier dysponent.</param>

/// <returns>DataTable.</returns>

public DataTable select(int ID\_maszyny, int ID\_dysponent)

/// <summary>

/// Selects the dysponent.

/// </summary>

/// <returns>DataTable.</returns>

public DataTable selectDysponent()

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <param name="ID\_dysponent">The identifier dysponent.</param>

/// <returns>DataTable.</returns>

public DataTable selectDysponent(int ID\_dysponent)

/// <summary>

/// Wprowadza nowy rekord.

/// </summary>

/// <param name="VO">Obiekt wymiany danych (insert)</param>

/// <returns>Wartość logiczna powodzenia operacji</returns>

public bool insert(nsAccess2DB.Maszyny\_DysponentVO VO)

/// <summary>

/// Aktualizuje rekord z wyjątkiem ID.

/// </summary>

/// <param name="VO">The vo.</param>

/// <returns>Wartość logiczna powodzenia operacji.</returns>

public bool update(nsAccess2DB.Maszyny\_DysponentVO VO)

/// <summary>

/// Kasuje rekord po podanym Identyfikatorze.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>

public bool delete(int ID\_maszyny)

/// <summary>

/// Kasuje rekord po podanym Identyfikatorze.

/// </summary>

/// <param name="ID\_dysponent">The identifier dysponent.</param>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <returns>Wartość logiczna powodzenia operacji.</returns>

public bool delete(int ID\_dysponent, int ID\_maszyny)

}//class Maszyny\_DysponentDAO

//Warstwa operacji biznesowaych tabeli Maszyny\_Dysponent ---> BUS.

/// <summary>

/// Class Maszyny\_DysponentBUS.

/// </summary>

public class Maszyny\_DysponentBUS

{

/// <summary>

/// Konstruktor.

/// </summary>

/// <param name="connString">The connection string.</param>

public Maszyny\_DysponentBUS(string connString)

/// <summary>

/// Wypełnia tablice pozycjami danych.

/// </summary>

public void select()

/// <summary>

/// Wypełnia tablice pozycjami danych.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

public void select(int ID\_maszyny)

/// <summary>

/// Wypełnia tablicę pozycjami danych.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <param name="ID\_dysponent">The identifier dysponent.</param>

public void select(int ID\_maszyny, int ID\_dysponent)

/// <summary>

/// Wypełnia tablicę pozycjami danych.

/// </summary>

/// <param name="ID\_dysponent">The identifier dysponent.</param>

public void selectDysponent(int ID\_dysponent)

/// <summary>

/// Wypełnia tablicę pozycjami danych

/// </summary>

/// <param name="query">The query.</param>

public void selectQuery(string query)

/// <summary>

/// Wprowadza rekord do tabeli.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <param name="ID\_dysponent">The identifier dysponent.</param>

/// <param name="Maszyny\_nazwa\_D">The maszyny nazwa d.</param>

/// <returns>Wartość logiczna powodzenia operacji.</returns>

public bool insert(int ID\_maszyny, int ID\_dysponent, string Maszyny\_nazwa\_D)

/// <summary>

/// Aktualizuje rekord z wyjątkiem numeru protokołu.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <param name="ID\_dysponent">The identifier dysponent.</param>

/// <param name="Maszyny\_nazwa\_D">The maszyny nazwa d.</param>

/// <returns>Wartość logiczna powodzenia akcji.</returns>

private bool update(int ID\_maszyny, int ID\_dysponent, string Maszyny\_nazwa\_D)

/// <summary>

/// Usuwa z tabeli pozycję o wskazanych parametrach.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <returns>Wartość logiczna powodzenia akcji.</returns>

public bool delete(int ID\_maszyny)

/// <summary>

/// Usuwa z tabeli pozycję o wskazanych parametrach.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <param name="ID\_dysponent">The identifier dysponent.</param>

/// <returns>Wartość logiczna powodzenia akcji.</returns>

public bool delete(int ID\_maszyny, int ID\_dysponent)

/// <summary>

/// Wypełnia tablicę.

/// </summary>

/// <param name="dt">Tabela danych.</param>

private void fillTable(DataTable dt)

// <summary>

/// <summary>

/// Skips this instance.

/// </summary>

public void skip()

/// <summary>

/// Przemieszcza indeks w tablicy danych na pozycję pierwszą.

/// </summary>

public void top()

/// <summary>

/// Zmienna logiczna osiągnięcia końca pliku.

/// </summary>

/// <value><c>true</c> if EOF; otherwise, <c>false</c>.</value>

public bool eof

/// <summary>

/// Zwraca liczbę pozycji tablicy.

/// </summary>

/// <value>The count.</value>

public int count

/// <summary>

/// Zwraca daną okrśloną wskaźnikiem pozycji.

/// </summary>

/// <value>The vo.</value>

public Maszyny\_DysponentVO VO

/// <summary>

/// Ustawia wskaźnik pozycji.

/// </summary>

/// <value>The index.</value>

public int idx

/// <summary>

/// Sprawdza istnienie pozycji w tabeli uzyskanej po ostanim poleceniu select.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <param name="ID\_dysponent">The identifier dysponent.</param>

/// <returns>Wartość logiczna istnienia pozycji.</returns>

public bool exists(int ID\_maszyny, int ID\_dysponent)

/// <summary>

/// Zwraca indeks pozycji.

/// </summary>

/// <param name="ID\_maszyny">ID\_maszyny z taleli Maszyny\_Dysponent.</param>

/// <returns>Indeks pozycji. -1 oznacza brak identyfikatora.</returns>

public int getIdx(int ID\_maszyny)

/// <summary>

/// Zwraca maszynę o podanym indeksie z tabeli Maszyny\_Dysponent.

/// </summary>

/// <param name="ID\_maszyny">The identifier maszyny.</param>

/// <returns>Maszyny\_DysponentVO.</returns>

public Maszyny\_DysponentVO GetVO(int ID\_maszyny)

/// <summary>

/// Writes the specified vo.

/// </summary>

/// <param name="VO">The vo.</param>

/// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>

public bool write(nsAccess2DB.Maszyny\_DysponentVO VO)

/// <summary>

/// Dodaje pozycję do tabeli.

/// </summary>

/// <param name="VO">The vo.</param>

/// <param name="VOs">The v os.</param>

private void add(Maszyny\_DysponentVO VO, ref Maszyny\_DysponentVO[] VOs)

}//class Maszyny\_DysponentBUS

////////////////////////////// klasa wymiany danych z tabelą Materialy

// ---------------------------------------------------------- MateriałyVO

/// <summary>

/// Klasa wymiany danych z tabelą Materialy

/// </summary>

public class MaterialyVO

/// <summary>

/// Konstruktor wymiany danych z tabelą Materialy

/// </summary>

public MaterialyVO() { }

// gettery i settery

/// <summary>

/// Gets or sets the identyfikator.

/// </summary>

/// <value>The identyfikator.</value>

public int Identyfikator

/// <summary>

/// Gets or sets the nazwa mat.

/// </summary>

/// <value>The nazwa mat.</value>

public string Nazwa\_mat

/// <summary>

/// Gets or sets the typ mat.

/// </summary>

/// <value>The typ mat.</value>

public string Typ\_mat

/// <summary>

/// Gets or sets the rodzaj mat.

/// </summary>

/// <value>The rodzaj mat.</value>

public string Rodzaj\_mat

/// <summary>

/// Gets or sets the jednostka miar mat.

/// </summary>

/// <value>The jednostka miar mat.</value>

public string Jednostka\_miar\_mat

/// <summary>

/// Gets or sets the dostawca mat.

/// </summary>

/// <value>The dostawca mat.</value>

public string Dostawca\_mat

/// <summary>

/// Gets or sets the stan mat.

/// </summary>

/// <value>The stan mat.</value>

public int Stan\_mat

/// <summary>

/// Gets or sets the zuzycie mat.

/// </summary>

/// <value>The zuzycie mat.</value>

public int Zuzycie\_mat

/// <summary>

/// Gets or sets the odpad mat.

/// </summary>

/// <value>The odpad mat.</value>

public int Odpad\_mat

/// <summary>

/// Gets or sets the stan minimum mat.

/// </summary>

/// <value>The stan minimum mat.</value>

public int Stan\_min\_mat

/// <summary>

/// Gets or sets the zapotrzebowanie mat.

/// </summary>

/// <value>The zapotrzebowanie mat.</value>

public int Zapotrzebowanie\_mat

/// <summary>

/// Gets or sets the stan mag po mat.

/// </summary>

/// <value>The stan mag po mat.</value>

public int Stan\_mag\_po\_mat

/// <summary>

/// Gets or sets the magazyn.

/// </summary>

/// <value>The magazyn.</value>

public string Magazyn

} //class MaterialyVO

// -------------------klasa dostępu (Data Access Object) do tabeli Materiały DAO

/// <summary>

/// Class MaterialyDAO.

/// </summary>

public class MaterialyDAO

{

/// <summary>

/// Initializes a new instance of the <see cref="MaterialyDAO"/> class.

/// </summary>

/// <param name="connString">The connection string.</param>

/// <construktor>

/// Konstruktor

/// </construktor>

public MaterialyDAO(string connString)

/// <summary>

/// Selects the query.

/// </summary>

/// <param name="query">The query.</param>

/// <returns>DataTable.</returns>

public DataTable selectQuery(string query)

/// <summary>

/// Zwraca tabelę wszystkich Materiałów.

/// </summary>

/// <returns>Tabela Materiałów.</returns>

public DataTable select()

/// <summary>

/// Zwraca tabelę wszystkich Materiałów po ID.

/// </summary>

/// <param name="Identyfikator">The identyfikator.</param>

/// <returns>Tabela Materiałów.</returns>

public DataTable select(int Identyfikator)

/// <summary>

/// Wprowadza nowy rekord.

/// </summary>

/// <param name="VO">Obiekt wymiany danych.</param>

/// <returns>Wartość logiczna powodzenia operacji.</returns>

public bool insert(nsAccess2DB.MaterialyVO VO)

/// <summary>

/// Aktualizuje rekord z wyjątkiem Identyfikatora

/// </summary>

/// <param name="VO">Obiekt wymiany danych.</param>

/// <returns>Wartość logiczna powodzenia operacji.</returns>

public bool update(nsAccess2DB.MaterialyVO VO)

/// <summary>

/// Deletes the specified identyfikator.

/// </summary>

/// <param name="Identyfikator">The identyfikator.</param>

/// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>

public bool delete(int Identyfikator)

}//clasa MaterialyDAO

// warstawa operacji biznesowych tabeli Materialy ---> BUS

/// <summary>

/// Class MaterialyBUS.

/// </summary>

public class MaterialyBUS

/// <summary>

/// Konstruktor.

/// </summary>

/// <param name="connString">The connection string.</param>

public MaterialyBUS(string connString)

/// <summary>

/// Wypełnia tablice danych pozycjami.

/// </summary>

public void select()

/// <summary>

/// Wypełnia tablice danych pozycjami.

/// </summary>

/// <param name="Identyfikator">Identyfikator Materiału.</param>

public void select(int Identyfikator)

/// <summary>

/// Dowolne zapytanie z formularza.

/// </summary>

/// <param name="query">The query.</param>

public void selectQuery(string query)

/// <summary>

/// Wprowadza rekord do tabeli.

/// </summary>

/// <param name="VO">Obiekt wymiany danych.</param>

/// <returns>Wartość logiczna powodzenia akcji.</returns>

private bool insert(nsAccess2DB.MaterialyVO VO)

/// <summary>

/// Aktualizuje rekord z wyjątkiem Identyfikatora czynności.

/// </summary>

/// <param name="VO">The vo.</param>

/// <returns>Wartość logiczna powodzenia akcji.</returns>

private bool update(nsAccess2DB.MaterialyVO VO)

/// <summary>

/// Usuwa rekord po identyfikatorze.

/// </summary>

/// <param name="Identyfikator">The identyfikator.</param>

/// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>

public bool delete(int Identyfikator)

/// <summary>

/// Wypełnia tablice.

/// </summary>

/// <param name="dt">Tabela danych.</param>

private void fillTable(DataTable dt)

/// <summary>

/// Przemieszcza indeks w tablicy danych o jedną pozycję.

/// </summary>

public void skip()

/// <summary>

/// Przemieszcza indeks w tablicy danych na pozycję pierwszą.

/// </summary>

public void top()

/// <summary>

/// Zmienna logiczna osiągnięcia końca pliku.

/// </summary>

/// <value><c>true</c> if EOF; otherwise, <c>false</c>.</value>

public bool eof

/// <summary>

/// Zwraca liczbę pozycji tablicy.

/// </summary>

/// <value>The count.</value>

public int count

/// <summary>

/// Zwraca daną okrśloną wskaźnikiem pozycji.

/// </summary>

/// <value>The vo.</value>

public MaterialyVO VO

/// <summary>

/// Ustawia wskaźnik pozycji.

/// </summary>

/// <value>The index.</value>

public int idx

/// <summary>

/// Sprawdza istnienie rekordu.

/// </summary>

/// <param name="Identyfikator">Identyfikator Materiału</param>

/// <returns>Wynik logiczny sprawdzenia.</returns>

private bool exists(int Identyfikator)

/// <summary>

/// Zwraca indeks pozycji.

/// </summary>

/// <param name="Identyfikator">Identyfikator Materiału.</param>

/// <returns>Indeks pozycji. -1 oznacza brak identyfikatora.</returns>

public int getIdx(int Identyfikator)

/// <summary>

/// Zwraca material o wskazanym identyfikatorze.

/// </summary>

/// <param name="Identyfikator">Identyfikator Materiału.</param>

/// <returns>Materialy. Jeśli ID == -1 to materialu nie znaleziono.</returns>

public MaterialyVO GetVO(int Identyfikator)

/// <summary>

/// Dodaje lub aktualizuje rekord.

/// </summary>

/// <param name="VO">Obiekt wymiany danych.</param>

/// <returns>Wynik powodzenia akcji.</returns>

public bool write(nsAccess2DB.MaterialyVO VO)

}// clasa MaterialyBUS

/////////////////////////////////////////////////// Magazyn - dane słownikowe.

/// <summary>

/// Klasa wymiany danych z Wybor\_magazynu.

/// </summary>

public class Wybor\_magazynuVO

{

/// <summary>

/// The magazyn

/// </summary>

private string \_Magazyn = string.Empty;

/// <summary>

/// Konstruktor wymiany danych z tabelą Wybor\_magazynu.

/// </summary>

public Wybor\_magazynuVO() { }

/// <summary>

/// Gets or sets the magazyn.

/// </summary>

/// <value>The magazyn.</value>

public string Magazyn

}//class Wybor\_magazynuVO

/// <summary>

/// Class Wybor\_magazynuDAO.

/// </summary>

public class Wybor\_magazynuDAO

/// <summary>

/// Class Wybor\_magazynuBUS.

/// </summary>

public class Wybor\_magazynuBUS

{

/// <summary>

/// Konstruktor.

/// </summary>

/// <param name="connString">ConnectionString.</param>

public Wybor\_magazynuBUS(string connString)

/// <summary>

/// Wypełnia tablice.

/// </summary>

/// <param name="dt">Tabela danych.</param>

private void fillTable(DataTable dt)

/// <summary>

/// Wypełnia tablice danych pozycjami.

/// </summary>

public void select()

/// <summary>

/// Przemieszcza indeks w tablicy danych o jedną pozycję.

/// </summary>

public void skip()

/// <summary>

/// Przemieszcza indeks w tablicy danych na pozycję pierwszą.

/// </summary>

public void top()

/// <summary>

/// Zmienna logiczna osiągnięcia końca pliku.

/// </summary>

/// <value><c>true</c> if EOF; otherwise, <c>false</c>.</value>

public bool eof

/// <summary>

/// Zwraca liczbę pozycji tablicy.

/// </summary>

/// <value>The count.</value>

public int count

/// <summary>

/// Zwraca daną okrśloną wskaźnikiem pozycji.

/// </summary>

/// <value>The vo.</value>

public Wybor\_magazynuVO VO

/// <summary>

/// Ustawia wskaźnik pozycji.

/// </summary>

/// <value>The index.</value>

public int idx

/// <summary>

/// Sprawdza istnienie rekordu.

/// </summary>

/// <param name="Magazyn">The magazyn.</param>

/// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>

public bool exists(String Magazyn)

/// <summary>

/// Zwraca indeks pozycji.

/// </summary>

/// <param name="Magazyn">The magazyn.</param>

/// <returns>System.Int32.</returns>

public int getIdx(string Magazyn)

}//Wybor\_magazynuBUS

/////////////////////////////////////////////// Jednostka\_miar - dane słownikowe.

/// <summary>

/// Klasa wymiany danych z tabelą Jednostka\_miar

/// </summary>

public class Jednostka\_miarVO

{

/// <summary>

/// Konstruktor wymiany danych z tabelą Jednostka\_miar

/// </summary>

public Jednostka\_miarVO() { }

/// <summary>

/// Gets or sets the nazwa jednostka miar.

/// </summary>

/// <value>The nazwa jednostka miar.</value>

public string Nazwa\_jednostka\_miar

}//class Jednostak\_miarVO

/// <summary>

/// Class Jednostka\_miarDAO.

/// </summary>

public class Jednostka\_miarDAO

{

/// <summary>

/// Initializes a new instance of the <see cref="Jednostka\_miarDAO"/> class.

/// </summary>

/// <param name="connString">The connection string.</param>

/// <constructor>

/// Konstruktor.

/// </constructor>

public Jednostka\_miarDAO(string connString)

/// <summary>

/// Selects this instance.

/// </summary>

/// <returns>DataTable.</returns>

public DataTable select()

}//class Jednostka\_miarDAO

/// <summary>

/// Class Jednostka\_miarBUS.

/// </summary>

public class Jednostka\_miarBUS

{

/// <summary>

/// Konstruktor.

/// </summary>

/// <param name="connString">ConnectionString.</param>

public Jednostka\_miarBUS(string connString)

/// <summary>

/// Wypełnia tablice.

/// </summary>

/// <param name="dt">Tabela danych.</param>

private void fillTable(DataTable dt)

/// <summary>

/// Wypełnia tablice danych pozycjami.

/// </summary>

public void select()

/// <summary>

/// Przemieszcza indeks w tablicy danych o jedną pozycję.

/// </summary>

public void skip()

/// <summary>

/// Przemieszcza indeks w tablicy danych na pozycję pierwszą.

/// </summary>

public void top()

/// <summary>

/// Zmienna logiczna osiągnięcia końca pliku.

/// </summary>

/// <value><c>true</c> if EOF; otherwise, <c>false</c>.</value>

public bool eof

/// <summary>

/// Zwraca liczbę pozycji tablicy.

/// </summary>

/// <value>The count.</value>

public int count

/// <summary>

/// Zwraca daną okrśloną wskaźnikiem pozycji.

/// </summary>

/// <value>The vo.</value>

public Jednostka\_miarVO VO

/// <summary>

/// Ustawia wskaźnik pozycji.

/// </summary>

/// <value>The index.</value>

public int idx

/// <summary>

/// Sprawdza istnienie rekordu.

/// </summary>

/// <param name="Nazwa\_jednostka\_miar">The nazwa jednostka miar.</param>

/// <returns>Wynik logiczny sprawdzenia.</returns>

public bool exists(String Nazwa\_jednostka\_miar)

/// <summary>

/// Zwraca indeks pozycji.

/// </summary>

/// <param name="Nazwa\_jednostka\_miar">Identyfikator Jednostka\_miar</param>

/// <returns>Indeks pozycji. -1 oznacza brak identyfikatora jednostki miar.</returns>

public int getIdx(string Nazwa\_jednostka\_miar)

}//class Jednostka\_miarBUS

////////////////////////////////////////////// Rodzaj mat - dane słownikowe.

/// <summary>

/// Klasa wymiany danych z tabelą Rodzaj materiału.

/// </summary>

public class Rodzaj\_matVO

{

/// <summary>

/// Konstruktor wymiany danych z tabelą Rodzaj\_mat

/// </summary>

public Rodzaj\_matVO() { }

/// <summary>

/// Gets or sets the nazwa rodzaj mat.

/// </summary>

/// <value>The nazwa rodzaj mat.</value>

public string Nazwa\_rodzaj\_mat

}//class Rodzaj\_matVO

//Klasa dostępu (Data Access Object) do tabeli Rodzaj\_mat.

/// <summary>

/// Class Rodzaj\_matDAO.

/// </summary>

public class Rodzaj\_matDAO

{

/// <summary>

/// Konstruktor.

/// </summary>

/// <param name="connString">The connection string.</param>

public Rodzaj\_matDAO(string connString)

/// <summary>

/// Selects this instance.

/// </summary>

/// <returns>DataTable.</returns>

public DataTable select()

}// class Rodzaj\_matDAO

// Warstwa operacji biznesowaych tabeli Rodzaj\_mat --> BUS.

/// <summary>

/// Class Rodzaj\_matBUS.

/// </summary>

public class Rodzaj\_matBUS

{

/// <summary>

/// Konstruktor.

/// </summary>

/// <param name="connString">ConnectionString.</param>

public Rodzaj\_matBUS(string connString)

/// <summary>

/// Wypełnia tablice.

/// </summary>

/// <param name="dt">Tabela danych.</param>

private void fillTable(DataTable dt)

/// <summary>

/// Wypełnia tablice danych pozycjami.

/// </summary>

public void select()

/// <summary>

/// Przemieszcza indeks w tablicy danych o jedną pozycję.

/// </summary>

public void skip()

/// <summary>

/// Przemieszcza indeks w tablicy danych na pozycję pierwszą.

/// </summary>

public void top()

/// <summary>

/// Zmienna logiczna osiągnięcia końca pliku.

/// </summary>

/// <value><c>true</c> if EOF; otherwise, <c>false</c>.</value>

public bool eof

/// <summary>

/// Zwraca liczbę pozycji tablicy.

/// </summary>

/// <value>The count.</value>

public int count

/// <summary>

/// Zwraca daną okrśloną wskaźnikiem pozycji.

/// </summary>

/// <value>The vo.</value>

public Rodzaj\_matVO VO

/// <summary>

/// Ustawia wskaźnik pozycji.

/// </summary>

/// <value>The index.</value>

public int idx

/// <summary>

/// Sprawdza istnienie rekordu.

/// </summary>

/// <param name="Nazwa\_rodzaj\_mat">The nazwa rodzaj mat.</param>

/// <returns>Wynik logiczny sprawdzenia.</returns>

public bool exists(String Nazwa\_rodzaj\_mat)

/// <summary>

/// Zwraca indeks pozycji.

/// </summary>

/// <param name="Nazwa\_rodzaj\_mat">Identyfikator Nazwa\_rodzaj\_mat</param>

/// <returns>Indeks pozycji. -1 oznacza brak identyfikatora Nazwa\_rodzaj\_mat.</returns>

public int getIdx(string Nazwa\_rodzaj\_mat)

}//class Rodzaj\_matBUS

/// <summary>

/// </summary>

public class Dostawca\_MaterialVO

{

/// Konstruktor wymiany danych z tabelą Dostawca\_MaterialVO

/// </summary>

public Dostawca\_MaterialVO() { }

/// <summary>

/// Gets or sets the identyfikator.

/// </summary>

/// <value>The identyfikator.</value>

public int Identyfikator

/// <summary>

/// Gets or sets the identifier material.

/// </summary>

/// <value>The identifier material.</value>

public int ID\_material

/// <summary>

/// Gets or sets the identifier dostawca mat.

/// </summary>

/// <value>The identifier dostawca mat.</value>

public int ID\_dostawca\_mat

/// <summary>

/// Gets or sets the material nazwa.

/// </summary>

/// <value>The material nazwa.</value>

public string Material\_nazwa

}//class Dostawca\_MaterialVO

// // // // // // // klasa dostępu (Data Access Object) do tabeli Dostawca\_Material.

/// <summary>

/// Class Dostawca\_MaterialDAO.

/// </summary>

public class Dostawca\_MaterialDAO

{

/// <summary>

/// Initializes a new instance of the <see cref="Dostawca\_MaterialDAO"/> class.

/// </summary>

/// <param name="connString">The connection string.</param>

/// <constructor>

/// Konstruktor.

/// </constructor>

public Dostawca\_MaterialDAO(string connString)

// -------------------------------------> dowolne zapytanie z poziomu Form

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <param name="query">The query.</param>

/// <returns>DataTable.</returns>

public DataTable selectQuery(string query)

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <returns>Tabela Dostawca\_Material.</returns>

public DataTable select()

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <param name="ID\_material">The identifier material.</param>

/// <returns>Tabela Dostawca\_Material</returns>

public DataTable select(int ID\_material)

/// <summary>

/// Selects the specified identifier material.

/// </summary>

/// <param name="ID\_material">The identifier material.</param>

/// <param name="ID\_dostawca\_mat">The identifier dostawca mat.</param>

/// <returns>DataTable.</returns>

public DataTable select(int ID\_material, int ID\_dostawca\_mat)

/// <summary>

/// Selects the dostawca.

/// </summary>

/// <param name="ID\_dostawca">The identifier dostawca.</param>

/// <returns>DataTable.</returns>

public DataTable selectDostawca(int ID\_dostawca)

/// <summary>

/// Wprowadza nowy rekord

/// </summary>

/// <param name="VO">Obiekt wymiany danych (insert)</param>

/// <returns>Wartość logiczna powodzenia operacji</returns>

public bool insert(nsAccess2DB.Dostawca\_MaterialVO VO)

/// <summary>

/// Aktualizuje rekord z wyjątkiem ID.

/// </summary>

/// <param name="VO">obiekt wymiany danych</param>

/// <returns>Wartość logiczna powodzenia operacji.</returns>

public bool update(nsAccess2DB.Dostawca\_MaterialVO VO)

/// <summary>

/// usuwa rekord

/// </summary>

/// <param name="ID\_material">The identifier material.</param>

/// <returns>Wartośc logiczna powodzenia operacji.</returns>

public bool delete(int ID\_material)

}//class Dostawca\_MaterialDAO

// // // // Warstwa operacji biznesowych tabeli Dostawca\_Material BUS.

/// <summary>

/// Class Dostawca\_MaterialBUS.

/// </summary>

public class Dostawca\_MaterialBUS

{

/// <summary>

/// Konstruktor.

/// </summary>

/// <param name="connString">The connection string.</param>

public Dostawca\_MaterialBUS(string connString)

/// <summary>

/// Wypełnia tablicę danych pozycjami.

/// </summary>

public void select()

/// <summary>

/// Wypełnia tablicę danych pozycjami.

/// </summary>

/// <param name="ID\_material">The identifier material.</param>

public void select(int ID\_material)

/// <summary>

/// Wypełnia tablicę danych pozycjami.

/// </summary>

/// <param name="ID\_material">The identifier material.</param>

/// <param name="ID\_dostawca\_mat">The identifier dostawca mat.</param>

public void select(int ID\_material, int ID\_dostawca\_mat)

/// <summary>

/// Selects the dostawca.

/// </summary>

/// <param name="ID\_dostawca">The identifier dostawca.</param>

public void selectDostawca(int ID\_dostawca)

/// <summary>

/// Wypełnia tablicę pozycjami danych ---&gt; dowolne zapytanie z poziomu Form

/// </summary>

/// <param name="query">The query.</param>

public void selectQuery(string query)

/// <summary>

/// Wprowadza rekord do tabeli.

/// </summary>

/// <param name="ID\_material">The identifier material.</param>

/// <param name="ID\_dostawca\_mat">The identifier dostawca mat.</param>

/// <returns>Wartośc logiczna powodzenia operacji.</returns>

public bool insert(int ID\_material, int ID\_dostawca\_mat)

/// <summary>

/// Inserts the specified identifier material.

/// </summary>

/// <param name="ID\_material">The identifier material.</param>

/// <param name="ID\_dostawca\_mat">The identifier dostawca mat.</param>

/// <param name="Material\_nazwa">The material nazwa.</param>

/// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>

public bool insert(int ID\_material, int ID\_dostawca\_mat, string Material\_nazwa)

/// <summary>

/// Aktualizuje rekord z wyjatkiem numeru protokołu.

/// </summary>

/// <param name="VO">The vo.</param>

/// <returns>Wartośc logiczna powodzenia akcji.</returns>

private bool update(nsAccess2DB.Dostawca\_MaterialVO VO)

/// <summary>

/// Usuwa z tabeli pozycję o wskazanych parametrach.

/// </summary>

/// <param name="ID\_material">The identifier material.</param>

/// <returns>Wartość logiczna powodzenia akcji.</returns>

public bool delete(int ID\_material)

/// <summary>

/// Wypełnia tablicę

/// </summary>

/// <param name="dt">Tabela danych.</param>

private void fillTable(DataTable dt)

/// <summary>

/// <summary>

/// Skips this instance.

/// </summary>

public void skip()

/// <summary>

/// Przemieszcza indeks w tablicy danych na pozycję pierwszą.

/// </summary>

public void top()

/// <summary>

/// Zmienna logiczna osiągnięcia końca pliku.

/// </summary>

/// <value><c>true</c> if EOF; otherwise, <c>false</c>.</value>

public bool eof

/// <summary>

/// Zwraca liczbę pozycji tablicy.

/// </summary>

/// <value>The count.</value>

public int count

/// <summary>

/// Zwraca dane okreslone wskaźnikiem pozycji.

/// </summary>

/// <value>The vo.</value>

public Dostawca\_MaterialVO VO

/// <summary>

/// Ustawia wskaźnik pozycji.

/// </summary>

/// <value>The index.</value>

public int idx

/// <summary>

/// Exists the specified identifier material.

/// </summary>

/// <param name="ID\_material">The identifier material.</param>

/// <param name="ID\_dostawca\_mat">The identifier dostawca mat.</param>

/// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>

public bool exist(int ID\_material, int ID\_dostawca\_mat)

/// <summary>

/// Zwraca indeks pozycji.

/// </summary>

/// <param name="ID\_material">ID\_materialu z tabeli Dostawca\_Material.</param>

/// <returns>System.Int32.</returns>

public int getIdx(int ID\_material)

/// <summary>

/// Gets the vo.

/// </summary>

/// <param name="ID\_material">The identifier material.</param>

/// <returns>Dostawca\_MaterialVO.</returns>

public Dostawca\_MaterialVO GetVO(int ID\_material)

/// <summary>

/// Dodaje pozycje tabeli Dostawca\_Material.

/// </summary>

/// <param name="VO">The vo.</param>

/// <param name="VOs">The v os.</param>

private void add(Dostawca\_MaterialVO VO, ref Dostawca\_MaterialVO[] VOs)

}// class Dostawca\_MaterialBUS

// // // // // // // // // // // // tabela Dostawca\_mat

/// <summary>

/// Class Dostawca\_matVO.

/// </summary>

public class Dostawca\_matVO

{

/// <summary>

/// Konstruktor wymiany danych z tabelą Dostawca\_mat

/// </summary>

public Dostawca\_matVO() { }

/// <summary>

/// Gets or sets the identyfikator.

/// </summary>

/// <value>The identyfikator.</value>

public int Identyfikator

/// <summary>

/// Gets or sets the nazwa dostawca mat.

/// </summary>

/// <value>The nazwa dostawca mat.</value>

public string Nazwa\_dostawca\_mat

/// <summary>

/// Gets or sets the link dostawca mat.

/// </summary>

/// <value>The link dostawca mat.</value>

public string Link\_dostawca\_mat

/// <summary>

/// Gets or sets the dod information dostawca mat.

/// </summary>

/// <value>The dod information dostawca mat.</value>

public string Dod\_info\_dostawca\_mat

}// class Dostawca\_matVO

/// <summary>

/// Klasa dostępu (Data Access Object) do tabeli Dostawca\_mat

/// </summary>

public class Dostawca\_matDAO

{

/// <summary>

/// Initializes a new instance of the <see cref="Dostawca\_matDAO"/> class.

/// </summary>

/// <param name="connString">The connection string.</param>

public Dostawca\_matDAO(string connString)

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <param name="query">The query.</param>

/// <returns>DataTable.</returns>

public DataTable selectQuery(string query)

/// <summary>

/// Zwraca tabelę spełniającą wartości parametrów.

/// </summary>

/// <returns>DataTable.</returns>

public DataTable select()

/// <summary>

/// Selects the specified identyfikator.

/// </summary>

/// <param name="Identyfikator">The identyfikator.</param>

/// <returns>DataTable.</returns>

public DataTable select(int Identyfikator)

/// <summary>

/// Inserts the specified vo.

/// </summary>

/// <param name="VO">The vo.</param>

/// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>

public bool insert(nsAccess2DB.Dostawca\_matVO VO)

/// <summary>

/// Aktualizuje rekord z wyjątkiem Identyfikatora.

/// </summary>

/// <param name="VO">The vo.</param>

/// <returns>Wartość logiczna powodzenia operacji.</returns>

public bool update(nsAccess2DB.Dostawca\_matVO VO)

/// <summary>

/// Deletes the specified identyfikator.

/// </summary>

/// <param name="Identyfikator">The identyfikator.</param>

/// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>

public bool delete(int Identyfikator)

/// <summary>

/// Class Dostawca\_matBUS.

/// </summary>

public class Dostawca\_matBUS

{

/// <summary>

/// Konstruktor.

/// </summary>

/// <param name="connString">ConnectionString.</param>

public Dostawca\_matBUS(string connString)

/// <summary>

/// Wypełnia tablice danych pozycjami.

/// </summary>

public void select()

/// <summary>

/// Wypełnia tablice danych pozycjami.

/// </summary>

/// <param name="Identyfikator">Identyfikator Dostawcy materiału.</param>

/// <summary>

/// Dowolne zapytanie z formularza.

/// </summary>

/// <param name="query">The query.</param>

public void selectQuery(string query)

/// <summary>

/// Wprowadza rekord do tabeli.

/// </summary>

/// <param name="VO">Obiekt wymiany danych.</param>

/// <returns>Wartość logiczna powodzenia akcji.</returns>

public bool insert(nsAccess2DB.Dostawca\_matVO VO)

/// <summary>

/// Aktualizuje rekord z wyjątkiem Identyfikatora Dostawcy materiału.

/// </summary>

/// <param name="VO">Obiekt wymiany danych.</param>

/// <returns>Wartość logiczna powodzenia akcji.</returns>

public bool update(nsAccess2DB.Dostawca\_matVO VO)

/// <summary>

/// Usuwa rekord po identyfikatorze.

/// </summary>

/// <param name="Identyfikator">The identyfikator.</param>

/// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>

public bool delete(int Identyfikator)

/// <summary>

/// Wypełnia tablice Dostawca Materiału.

/// </summary>

/// <param name="dt">The dt.</param>

private void fillTable(DataTable dt)

/// <summary>

/// Przemieszcza indeks w tablicy danych o jedną pozycję.

/// </summary>

public void skip()

/// <summary>

/// Przemieszcza indeks w tablicy danych na pozycję pierwszą.

/// </summary>

public void top()

/// <summary>

/// Zmienna logiczna osiągnięcia końca pliku.

/// </summary>

/// <value><c>true</c> if EOF; otherwise, <c>false</c>.</value>

public bool eof

/// <summary>

/// Zwraca liczbę pozycji tablicy.

/// </summary>

/// <value>The count.</value>

public int count

/// <summary>

/// Gets the vo.

/// </summary>

/// <value>The vo.</value>

public Dostawca\_matVO VO

/// <summary>

/// Ustawia wskaźnik pozycji.

/// </summary>

/// <value>The index.</value>

public int idx

/// <summary>

/// Sprawdza istnienie rekordu.

/// </summary>

/// <param name="Identyfikator">ID dostawcy materiału.</param>

/// <returns>Wynik logiczny sprawdzenia.</returns>

private bool exists(int Identyfikator)

/// <summary>

/// Zwraca indeks pozycji.

/// </summary>

/// <param name="Identyfikator">ID dostawcy materiału.</param>

/// <returns>Indeks pozycji. -1 oznacza brak identyfikatora.</returns>

public int getIdx(int Identyfikator)

/// <summary>

/// Zwraca maszynę o wskazanym identyfikatorze.

/// </summary>

/// <param name="Identyfikator">Identyfikator dostawcy materiału.</param>

/// <returns>Maszyny. Jeśli ID==-1 to maszyny nie znaleziono.</returns>

public Dostawca\_matVO GetVO(int Identyfikator)

/// <summary>

/// Dodaje lub aktualizuje rekord.

/// </summary>

/// <param name="VO">Obiekt wymiany danych.</param>

/// <returns>Wynik powodzenia akcji.</returns>

public bool write(nsAccess2DB.Dostawca\_matVO VO)

}// Dostawca\_matBUS

}//namespace nsAccess2DB

### Obsługa działania programu – Klasa główna.

namespace RemaGUM

{

/// <summary>

/// Class SpisForm.

/// Implements the <see cref="System.Windows.Forms.Form" />

/// </summary>

/// <seealso cref="System.Windows.Forms.Form" />

public partial class SpisForm : Form

{

//ustawienia połączenie z bazą poprzez xml.

private settings \_settings;

/// <summary>

/// The connection string

/// </summary>

private string \_connString;

/// <summary>

/// The rest

/// </summary>

private Rest \_rest = new Rest();

/// <summary>

/// The help file- plik pomocy RemaGUM.

/// </summary>

private string \_helpFile = Application.StartupPath + "\\help.chm";

/// <summary>

/// zawartosc pliku-dane odczytane z pliku zdjęcia.

/// </summary>

private byte[] \_zawartoscPliku;

/// <summary>

/// dir nazwa -nazwa katalogu tymczasowego.

/// </summary>

string \_dirNazwa = "C:\\tempRemaGUM";

/// <summary>

/// dir pelna nazwa katalogu tymczasowego - pelna nazwa.

/// </summary>

string \_dirPelnaNazwa = string.Empty;

/// <summary>

/// Enum \_status- status działania formularza.

/// </summary>

enum \_status { edycja, nowy };

/// <summary>

/// status form - wartośc statusu formularza.

/// </summary>

private byte \_statusForm;

/// <summary>

/// The maszyna tag przechowuje identyfikatory maszyn dla operatora.

/// </summary>

private int[] \_maszynaTag;

/// <summary>

/// The maszyna identifier

/// </summary>

int \_maszynaId = -1; // identyfikator maszyny przy starcie programu.

/// <summary>

/// indeks szukanej maszyny.

/// </summary>

int \_maszynaSzukajIdx = 0;

/// <summary>

/// indeks szukanego dysponenta.

/// </summary>

int \_dysponentSzukajIdx = 0;

/// <summary>

/// indeks szukanego materiału.

/// </summary>

int \_materialSzukajIdx = 0;

/// <summary>

/// </summary>

private ToolTip \_tt;

/// <summary>

/// The interwal przegladow

/// </summary>

private int \_interwalPrzegladow = 365; //w dniach = 1 rok

private nsAccess2DB.OperatorBUS \_OperatorBUS;

private nsAccess2DB.MaszynyBUS \_maszynyBUS;

private nsAccess2DB.MaterialyBUS \_materialyBUS;

/// <summary>

/// Klasa umożliwiająca połączenie bazy danych za pomocą connectionStringa zapisanego w pliku xml

/// </summary>

public class settings

{

/// <summary>

/// Konstruktor wymiany danych ustawień programu w XML.

/// </summary>

public settings()

/// <summary>

/// Czyta plik.

/// </summary>

public void read()

/// <summary>

/// Czyta plik.

/// </summary>

/// <param name="XMLfile">Nazwa pliku xml</param>

public void read(string XMLfile)

/// <summary>

/// Zapisuje do pliku.

/// </summary>

public void write()

/// <summary>

/// Connection string.

/// </summary>

public string connectionString

/// <summary>

/// Connection string.

/// </summary>

public string XMLfile

/// <summary>

/// Pełna ścieżka kopii bazy danych.

/// </summary>

public string dBcopyFile

/// <summary>

/// Interwał w minutach sporzadzania kopii bazy danych.

/// </summary>

public int copyInterval

/// <summary>

/// Pełna nazwa katalogu historii bazy.

/// </summary>

public string dBhistoryDir

/// <summary>

/// Interwał w dniach sporządzania kopii bazy w katalogu historii.

/// </summary>

public int historyInterval

}//class settings

/// <summary>

/// Konstruktor formularza.

/// </summary>

/// <param name="connStr">Połaczenie z bazą.</param>

public SpisForm()

{

InitializeComponent();

// Indeksowanie w zakładkach

// tool tipy

\_tt = new ToolTip();

}//public SpisForm()

// Wyświetla komunikaty chwilowe w programie.

private void pokazKomunikat(string tresc)

/// <summary>

///Obsługuje tiki timera.

/// </summary>

/// <param name="sender">The source of the event.</param>

/// <param name="e">The <see cref="EventArgs"/> instance containing the event data.</param>

private void timer1\_Tick(object sender, EventArgs e)

/// <summary>

/// Ustawia datę i czas. 2018-02-14 11:06:21 format dat godzina indeks spacji dzieli ciąg na 2 podciagi data + godzina

/// </summary>

private void SetDataAndTime()

/// <summary>

/// Wyświetla komponenty w zależności od indeksu zakładki.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void tabControlZakladki\_SelectedIndexChanged(object sender, EventArgs e)

/// <summary>

/// Przycisk uruchamiający okienko o programie.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void toolStripButtonOProgramie\_Click(object sender, EventArgs e)

/// <summary>

/// Przycisk uruchomiający pomoc programu RemaGUM.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void toolStripButtonHelp\_Click(object sender, EventArgs e)

/// <summary>

/// Czyści dane z formularza maszyny.

/// </summary>

private void CzyscDaneMaszyny()

/// <summary>

/// zmiana indeksu w list box maszyny

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void listBoxMaszyny\_SelectedIndexChanged(object sender, EventArgs e)

/// <summary>

/// Odświeża listę maszyn w listBoxMaszyny.

/// </summary>

private void OdswiezListeMaszyn()

{

/// <summary>

/// Wypelnij operatorow maszyn - checked -boxy.

/// </summary>

/// <param name="v">The v.</param>

private void WypelnijOperatorow\_maszyn(CheckedListBox v)

/// <summary>

/// wypełnia listbox kategoriami maszyn

/// </summary>

private void WypelnijKategorie()

/// <summary>

/// Handles the SelectedIndexChanged event of the comboBoxKategoria control.

/// </summary>

/// <param name="sender">The source of the event.</param>

/// <param name="e">The <see cref="EventArgs"/> instance containing the event data.</param>

private void comboBoxKategoria\_SelectedIndexChanged(object sender, EventArgs e)

/// <summary>

/// Wypełnia danymi Dysponentów w zakładce Maszyny.

/// </summary>

private void WypelnijDysponent()

/// <summary>

/// Obsługa comboBox\_Dysponent control.

/// </summary>

/// <param name="sender">The source of the event.</param>

/// <param name="e">The <see cref="EventArgs"/> instance containing the event data.</param>

private void comboBox\_Dysponent\_SelectedIndexChanged(object sender, EventArgs e)

/// <summary>

/// Sortuje po nazwie maszyny.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void radioButtonNazwa\_CheckedChanged(object sender, EventArgs e)

/// <summary>

/// Sortuje po numerze inwentarzowym maszyny.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void radioButton\_Nr\_Inwentarzowy\_CheckedChanged(object sender, EventArgs e)

/// <summary>

/// Sortuje po typie maszyny.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void radioButton\_Typ\_CheckedChanged(object sender, EventArgs e)

/// <summary>

/// Sortuje po numerze fabrycznym maszyny.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void radioButtonNr\_fabrycznyCheckedChanged(object sender, EventArgs e)

/// <summary>

/// Sortuje po numerze pomieszczenia, w którym znajduje się maszyna.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void radioButton\_Nr\_Pomieszczenia\_CheckedChanged(object sender, EventArgs e)

/// <summary>

/// Sortuje po dacie ostatniego przeglądu maszyny.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void radioButtonData\_ost\_przegladu\_CheckedChanged(object sender, EventArgs e)

/// <summary>

/// wyszukuje maszynę po wpisaniu dowolnego ciągu wyrazów

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

///

/// <summary>

/// Wypełnia listbox Działami w których znajdują się maszyny.

/// </summary>

private void WypelnijDzial()

/// <summary>

/// Handles the SelectedIndexChanged event of the comboBoxDzial control.

/// </summary>

/// <param name="sender">The source of the event.</param>

/// <param name="e">The <see cref="EventArgs"/> instance containing the event data.</param>

private void comboBoxDzial\_SelectedIndexChanged(object sender, EventArgs e)

/// <summary>

/// Wypełnia listbox pozycjami częstotliwości Wykorzystania.

/// </summary>

private void WypelnijCzestotliwosc()

// przypisuje w combo identyfikator = nazwę częstotliwości wykorzystania

/// <summary>

/// Handles the SelectedIndexChanged event of the ComboBoxWykorzystanie control.

/// </summary>

/// <param name="sender">The source of the event.</param>

/// <param name="e">The <see cref="EventArgs"/> instance containing the event data.</param>

private void ComboBoxWykorzystanie\_SelectedIndexChanged(object sender, EventArgs e)

/// <summary>

/// wypełnia listbox propozycjami co zrobić z maszyną (zachować/złomować itp)

/// </summary>

private void WypelnijPropozycje()

/// <summary>

/// Handles the SelectedIndexChanged event of the comboBoxPropozycja control.

/// </summary>

/// <param name="sender">The source of the event.</param>

/// <param name="e">The <see cref="EventArgs"/> instance containing the event data.</param>

private void comboBoxPropozycja\_SelectedIndexChanged(object sender, EventArgs e)

/// <summary>

/// Wypełnia stan techniczy maszyny.

/// </summary>

private void WypelnijStan\_techniczny()

/// <summary>

/// Handles the SelectedIndexChanged event of the comboBoxStan\_techniczny control.

/// </summary>

/// <param name="sender">The source of the event.</param>

/// <param name="e">The <see cref="EventArgs"/> instance containing the event data.</param>

private void comboBoxStan\_techniczny\_SelectedIndexChanged(object sender, EventArgs e)

/// <summary>

/// Klawisz Nowa czyści formularz (maszyna).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void ButtonNowa\_Click(object sender, EventArgs e)

/// <summary>

/// Klawisz Anuluj (maszyna).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonAnuluj\_Click(object sender, EventArgs e)

/// <summary>

/// Klawisz Usuń (maszyna).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonUsun\_Click(object sender, EventArgs e)

/// <summary>

/// Działania po wciśnięciu klawisza Zapisz (maszyny).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonZapisz\_Click(object sender, EventArgs e)

/// <summary>

/// Pokazuje nazwę zdjęcia w formie linku.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void linkLabelNazwaZdjecia\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

/// <summary>

/// Przycisk Wgraj/pokaż zdjęcie maszyny.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonPokazZdj\_Click(object sender, EventArgs e)

/// <summary>

/// Przycisk Usuń zdjęcie.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonUsunZdj\_Click(object sender, EventArgs e)

/// <summary>

/// Wypełnia listBoxMaterialy.

/// </summary>

private void WypelnijMaterialyNazwami()

/// <summary>

/// Zmiana indeksu w list box Materialy.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void listBoxMaterialy\_SelectedIndexChanged(object sender, EventArgs e)

/// <summary>

/// Wypełnia CheckedListBoxDostawcyMat (zakładka Materiały).

/// </summary>

/// <param name="v">CheckedListBoxDostawcyMat</param>

private void WypelnijCheckedBoxDostawcow(CheckedListBox v)

/// <summary>

/// Zmiana indeksu dostawcy w checkedListBoxDostawcyMat (aktualizacja pól do odczytu danych wybranego dostawcy).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void checkedListBoxDostawcyMat\_SelectedIndex(object sender, EventArgs e)

/// <summary>

/// Czyści dane w formularzu Materiały.

/// </summary>

private void CzyscDaneMaterialy()

/// <summary>

/// Odswieża dane w zakładce materiały.

/// </summary>

private void OdswiezMaterialy()

/// <summary>

/// Odświeża checked boxy dostawców w zakładce materiały.

/// </summary>

private void OdswiezDostawcowWMaterialach()

/// <summary>

/// Wybiera rodzaj magazynu (materiały lub normalia).

/// </summary>

private void WybierzMagazyn()

/// <summary>

/// Handles the SelectedIndexChanged event of the comboBoxWyborMagazyn control.

/// </summary>

/// <param name="sender">The source of the event.</param>

/// <param name="e">The <see cref="EventArgs"/> instance containing the event data.</param>

private void comboBoxWyborMagazyn\_SelectedIndexChanged(object sender, EventArgs e)

/// <summary>

/// Wypełnia jednostkę miar dla Materiałów i Normaliów.

/// </summary>

private void WypelnijJednostka\_miar()

/// <summary>

/// Handles the SelectedIndexChanged event of the comboBoxJednostka\_mat control.

/// </summary>

/// <param name="sender">The source of the event.</param>

/// <param name="e">The <see cref="EventArgs"/> instance containing the event data.</param>

private void comboBoxJednostka\_mat\_SelectedIndexChanged(object sender, EventArgs e)

/// <summary>

/// Wypełnia rodzaj materiału (zakładka materiały).

/// </summary>

private void WypelnijRodzaj\_mat()

/// <summary>

/// Handles the SelectedIndexChanged event of the comboBoxRodzaj control.

/// </summary>

/// <param name="sender">The source of the event.</param>

/// <param name="e">The <see cref="EventArgs"/> instance containing the event data.</param>

private void comboBoxRodzaj\_SelectedIndexChanged(object sender, EventArgs e)

/// <summary>

/// Klinknięcie w link otwiera stronę z linka.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void linkLabelDostawcaMat2\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

/// <summary>

/// Sortowanie po nazwie materiału (zakładka materiał).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void radioButtonNazwa\_mat\_CheckedChanged(object sender, EventArgs e)

/// <summary>

/// Sortowanie po typie materiału (zakładka materiał).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void radioButtonTyp\_mat\_CheckedChanged(object sender, EventArgs e)

/// <summary>

/// Sortowanie po stanie minimalnym materiału (zakładka materiał).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void radioButtonStan\_min\_mat\_CheckedChanged(object sender, EventArgs e)

/// <summary>

/// Sortowanie po ilości materiału w magazynie (zakładka materiał).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void radioButtonMagazyn\_ilosc\_mat\_CheckedChanged(object sender, EventArgs e)

/// <summary>

/// Działania po wciśnięciu klawisza Nowa (materiały).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void ButtonNowa\_mat\_Click(object sender, EventArgs e)

/// <summary>

/// Aktywuje pola panelu materiałów bez pola stan magazynowy

/// </summary>

private void AktywujPanelMaterial()

/// <summary>

/// Działania po wciśnięciu klawisza Anuluj (materiały).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonAnuluj\_mat\_Click(object sender, EventArgs e)

/// <summary>

/// Działania po wciśnięciu klawisza Usuń (materiały).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonUsun\_mat\_Click(object sender, EventArgs e)

/// <summary>

/// Działania po wciśnięciu klawisza Zapisz (materiały).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonZapiszMat\_Click(object sender, EventArgs e)

//TODO wyszukiwarka materiałów.

/// <summary>

/// Wyszukuje materiał po dowolnym ciągu znaków w nazwie lub rodzaju materiału.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonSzukaj\_mat\_Click(object sender, EventArgs e)

/// <summary>

/// Wypełnia listę dostawców wg nazwy.

/// </summary>

private void WypelnijListeDostawcowDanymi()

/// <summary>

/// Zmiana indeksu

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void listBoxDostawcy\_SelectedIndexChanged(object sender, EventArgs e)

/// <summary>

/// Tworzy nowego dostawcę (wymaga przycisku zapisz).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonNowyDostawca\_Click(object sender, EventArgs e)

/// <summary>

/// otwiera link przy kliknięciu.

/// </summary>

/// <param name = "sender" ></ param >

/// < param name= "e" ></ param >

private void linkLabelDostawcaMat\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

/// <summary>

/// Usuwa link z obiektu (wymaga uzycia klawisza zapisz).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonUsunLink\_Click(object sender, EventArgs e)

/// <summary>

/// Akcje po wciśnięciu przycisku Zapisz.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonZapiszDostawca\_Click(object sender, EventArgs e)

/// <summary>

/// działania po wciśnięciu przycisku Anuluj.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonAnulujDostawca\_Click(object sender, EventArgs e)

/// <summary>

/// Czyści pola dane dostawcy w zakładce Dostawca.

/// </summary>

private void CzyscDaneDostawcy()

/// <summary>

/// Usuwa wybranego dostawcę po Identyfikatorze.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonUsunDostawca\_Click(object sender, EventArgs e)

/// <summary>

/// wyświetla listę operatorów maszyn po imieniu i nazwisku.

/// </summary>

private void WypelnijOperatorowDanymi()

/// <summary>

/// zmiana indeksu w list box operator maszyny

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void listBoxOperator\_maszyny\_SelectedIndexChanged(object sender, EventArgs e)

/// <summary>

/// wyświetla listę maszyn dla danego operatora.

/// </summary>

private void WypelnijOperatorowMaszynami()

/// <summary>

/// Zmiana indeksu dla comboBoxOperator

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void comboBoxOperator\_SelectedIndexChanged(object sender, EventArgs e)

/// <summary>

/// Wypełnia dział wskazanego Operatora.

/// </summary>

private void WypelnijDzialOperatora()

/// <summary>

/// Handles the SelectedIndexChanged event of the comboBoxDzial\_operator\_maszyny control.

/// </summary>

/// <param name="sender">The source of the event.</param>

/// <param name="e">The <see cref="EventArgs"/> instance containing the event data.</param>

private void comboBoxDzial\_operator\_maszyny\_SelectedIndexChanged(object sender, EventArgs e)

/// <summary>

/// Działania po wciśnięciu klawisza Nowa (zakładka operator).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonNowaOperator\_Click(object sender, EventArgs e)

/// <summary>

/// Działania po wciśnięciu klawisza Zapisz (zakładka operator).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonZapiszOperator\_Click(object sender, EventArgs e)

/// <summary>

/// Działania po wciśnięciu klawisza Anuluj (zakładka operator).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonAnulujOperator\_Click(object sender, EventArgs e)

/// <summary>

/// Działania po wciśnięciu klawisza Usuń (zakładka operator).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonUsunOperator\_Click(object sender, EventArgs e)

/// <summary>

/// Działania po wciśnięciu klawisza Szukaj (zakładka operator).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonSzukajOperator\_Click(object sender, EventArgs e)

/// <summary>

/// wyświetla listę dysponentów maszyn - nazwisko i imię.

/// </summary>

private void WypelnijDysponentowDanymi()

/// <summary>

/// Czyści dane w zakładce dysponent maszyn.

/// </summary>

private void CzyscDaneDysponenta()

/// <summary>

/// Odświeża listBoxDysponent.

/// </summary>

private void OdswiezDysponentowMaszyn()

/// <summary>

/// Zmiana indeksu w list box dysponent.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void listBoxDysponent\_SelectedIndexChanged(object sender, EventArgs e)

/// <summary>

/// Wypełnia dane listBoxMaszynyDysponenta

/// </summary>

private void WypelnijDysponentowMaszynami()

/// <summary>

/// Wypełnia dział zakładka dysponent.

/// </summary>

private void WypelnijDzialDysponenta()

/// <summary>

/// Handles the SelectedIndexChanged event of the comboBoxDzial\_dysponent control.

/// </summary>

/// <param name="sender">The source of the event.</param>

/// <param name="e">The <see cref="EventArgs"/> instance containing the event data.</param>

private void comboBoxDzial\_dysponent\_SelectedIndexChanged(object sender, EventArgs e)

/// <summary>

/// Działania po wciśnięciu klawisza Nowa (zakładka dysponent).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonNowaDysponent\_Click(object sender, EventArgs e)

/// <summary>

/// Działania po wciśnięciu klawisza Zapisz (zakładka dysponent).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonZapiszDysponent\_Click(object sender, EventArgs e)

/// <summary>

/// Działania po wciśnięciu klawisza Anuluj (zakładka dysponent).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonAnulujDysponent\_Click(object sender, EventArgs e)

/// <summary>

/// Działania po wciśnięciu klawisza Usuń (zakładka dysponent).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonUsunDysponent\_Click(object sender, EventArgs e)

/// <summary>

/// Działania po wciśnięciu klawisza Szukaj (zakładka dysponent).

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonSzukajDysponent\_Click(object sender, EventArgs e)

}// public partial class SpisForm : Form

}//namespace RemaGUM

### Komponenty wizualne – okienko „o RemaGUM”.

namespace RemaGUM

{

public partial class FormORemaGUM : Form

{

/// <summary>

/// Initializes a new instance of the <see cref="FormORemaGUM" /> class.

/// </summary>

public FormORemaGUM()

{

InitializeComponent();

}//FormORemaGUM()

}//FormORemaGUM : Form

}//FormORemaGUM : Form

1. Główne komponenty programu

### Moduł Maszyny – komponenty

Tabela Maszyny zawiera następujące pozycje:

Identyfikator (klucz własny)

1. Kategoria
2. Nazwa
3. Typ
4. Nr\_inwentarzowy
5. Nr\_fabryczny
6. Rok\_produkcji
7. Producent
8. Zdjecie
9. Zawartośc pliku
10. Rozszerz\_zdj
11. Nazwa\_os\_zarzadzajaca
12. Nazwa\_op\_maszyny
13. Nr\_pom
14. Dzial
15. Nr\_prot\_BHP
16. Data\_ost\_przegl (badania BHP)
17. Data\_kol\_przegl(badania BHP)
18. Uwagi
19. Wykorzystanie – jak często jest wykorzystywana (SŁOWNIK -> kilka razy w miesiącu
20. Stan\_techniczny (SŁOWNIK -> dobry/ do naprawy / złom)
21. Propozycja (SŁOWNIK wymiana/złom)
22. Rok\_ost\_przeg
23. Mc\_ost\_przeg
24. Dz\_ost\_przeg
25. Rok\_kol\_przeg
26. Mc\_kol\_przeg
27. Dz\_kol\_przeg
    * sortowanie tabeli maszyn po:
      + Nazwie,
      + Typie
      + Nr inwentarzowy
      + Nr fabryczny
      + Nr pomieszczenia
      + Data ostatniego przeglądu
      + Data kolejnego przeglądu

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identyfikator | Liczba |  |
|  | Kategoria | Txt 255 | 100 |
|  | Nazwa | Txt 255 | 255 |
|  | Typ | Txt 255 | 255 |
|  | Nr\_inwentarzowy | Txt 255 | 255 |
|  | Nr\_fabryczny | Txt 255 | 255 |
|  | Rok\_produkcji | Txt 255 | 50 |
|  | Producent | Txt 255 | 255 |
|  | Zdjecie | Txt 255 | 255 |
|  | Zawartośc pliku | ObiektOLE | 255 |
|  | Rozszerz\_zdj | Txt 255 | 255 |
|  | Nazwa\_dysponent | Txt 255 | 255 |
|  | Nazwa\_op\_maszyny | Txt 255 | 255 |
|  | Nr\_pom | Txt 255 | 20 |
|  | Dzial | Txt 255 | 20 |
|  | Nr\_prot\_BHP | Txt 255 | 255 |
|  | Data\_ost\_przegl (badania BHP) | data |  |
|  | Data\_kol\_przegl(badania BHP) | data |  |
|  | Uwagi | Txt słownik | 20 |
|  | Wykorzystanie | Txt słownik | 20 |
|  | Stan\_techniczny | Txt słownik | 20 |
|  | Propozycja | Txt 255 | 100 |
|  | Rok\_ost\_przeg | Int |  |
|  | Mc\_ost\_przeg | Int |  |
|  | Dz\_ost\_przeg | Int |  |
|  | Rok\_kol\_przeg | Int |  |
|  | Mc\_kol\_przeg | Int |  |
|  | Dz\_kol\_przeg | Int |  |

*tabela 1. Nazwy komponentów, kolumn i ich typy w tabeli Maszyny*

|  |  |  |
| --- | --- | --- |
| sortowanie | groupBoxSortowanie | radioButtonNazwa |
| radioButtonTyp |
| radioButtonNr\_fabryczny |
| radioButtonNr\_inwentarzowy |
| radioButtonNr\_pomieszczenia |
| radioButtonData\_ost\_przegladu |
| wyszukiwanie |  | textBoxWyszukiwanie |
|  | buttonSzukaj |
| Spis dostępnych maszyn | Lista nazw | listBoxMaszyny |
| Dane maszyny | Kategoria | comboBoxKategoria |
| Nazwa | textBoxNazwa |
| Typ | textBoxTyp |
| Rok\_produkcji | textBoxRok\_produkcji |
| Data\_ost\_przegl | dateTimePickerData\_ost\_przegl |
| Data\_kol\_przegl | dateTimePickerData\_kol\_przegl |
| Nr\_inwentarzowy | textBoxNr\_inwentarzowy |
| Nr\_fabryczny | textBoxNr\_fabryczny |
| Producent | textBoxProducent |
| Dysponent | comboBoxDysponent |
| Operator maszyny  WypelnijOperatorow\_maszyny | checkedListBoxOperator\_maszyny |
| Nr\_pom | comboBoxNr\_pom |
| Dzial | comboBoxDzial |
| Nr\_prot\_BHP | textBoxNr\_prot\_BHP |
| Uwagi | richTextBoxUwagi |
| zdjęcie | Zdjecie1 - nazwa |  |
| Rozszerz\_zdj1 |  |
| **Zawartość pliku** | pictureBox1 |
|  | buttonPokazZdj |
|  | buttonUsunZdj |
| Otwieranie po linku nazwy zdjęcia (?) | linkLabelNazwaZdjecia |
| Stan techniczny i częstotliwość wykorzystania | Stan\_techniczny | comboBoxStan\_techniczny |
| Wykorzystanie | comboBoxWykorzystanie |
| Propozycja | comboBoxPropozycja |
| Edycja spisu | nowa | buttonNowa |
| zapisz | buttonZapisz |
| anuluj | buttonAnuluj |
| usuń | buttonUsun |

*tabela. 2 Nazwy komponentów modułu maszyny*

Tabela **Kategoria (dane słownikowe** jaką kategorię stanowi**)** powinna zawierać następujące pozycje:

1. Identyfikator
2. **Kategoria** – (SŁOWNIK -> laboratoryjne / warsztatowe)

Tabela **Czestotliwosc (dane słownikowe** jak często jest wykorzystywana**)** powinna zawierać następujące pozycje:

1. Identyfikator
2. **Czestotliwosc** – (SŁOWNIK -> kilka razy w miesiącu / kwartale / w okresie półrocznym/w roku / rzadziej / nieużywana)

Tabela **Stan\_techniczny(dane słownikowe** jaki stan techniczny) powinna zawierać następujące pozycje:

1. Identyfikator
2. Stan\_techniczny (SŁOWNIK -> dobry/ do naprawy / do złomowania)

Tabela **Propozycja (dane słownikowe** jak wykorzystać**)** powinna zawierać następujące pozycje:

1. Identyfikator
2. Propozycja – (SŁOWNIK -> zachować, do remontu, do likwidacji)

### Moduł Operator maszyny

Tabela **Operator\_maszyny** zawiera następujące pozycje:

Identyfikator (ID\_operator) użytkownik maszyny klucz własny

1. Nazwa\_op\_maszyny (Imię i Nazwisko)
2. ID\_dzial
3. Nazwa\_dzial
4. Uprawnienie
5. Data\_konca\_upr (data po której wygasają uprawnienia np. 2 lata)
6. Rok
7. Mc
8. Dzien
9. Op\_nazwisko
10. Op\_imie

|  |  |  |
| --- | --- | --- |
| Identyfikator |  | Int |
| Nazwa\_op\_maszyny | Txt 255 | string |
| Op\_imie | Txt 255 | string |
| Op\_nazwisko | Txt 255 | string |
| ID\_dzial | liczba | Int |
| Nazwa\_dzial | Txt 255 | string |
| Uprawnienie | Txt 255 | string |
| Data\_konca\_upr | Data | string |
| Rok | liczba | Int |
| Mc | liczba | Int |
| Dzien | liczba | Int |

*tabela 3. Nazwy komponentów, kolumn i ich typy w tabeli Operator*

|  |  |  |
| --- | --- | --- |
| sortowanie | comboBoxOperator | Nazwisko |
| Dział |
| Uprawnienia |
| Data końca uprawnień |
| wyszukiwanie |  | textBoxWyszukiwanieOperator |
|  | buttonSzukajOperator |
| Spis operatorów maszyn | Lista nazw  listBoxOperator\_maszyny \_SelectedIndexChanged | listBoxOperator |
| Dane operatora maszyny | Op\_imie | textBoxImieOperator |
| Op\_nazwisko | textBoxNazwiskoOperator |
| Dzial  comboBoxDzial\_operator\_ maszyny\_SelectedIndexChanged | comboBoxDzialOperator |
| Uprawnienie | textBoxUprawnienieOperator |
| Data końca uprawnień | dateTimePickerDataKoncaUprOperator |
| Lista obsługiwanych przez wybranego operatora maszyn | listBoxMaszynyOperatora |
|  | ID operatora | toolStripStatusLabelIDOperatora |
| Edycja spisu | nowa | buttonNowaOperator |
| zapisz | buttonZapiszOperator |
| anuluj | buttonAnulujOperator |
| usuń | buttonUsunOperator |

*tabela. 4 Nazwy komponentów modułu operator.*

Tabela **Operator\_maszyny\_Maszyny (multi)** określa prawa do użytkowania maszyn, powinna zawierać następujące pozycje:

1. Identyfikator (klucz własny)
2. ID\_masz (klucz obcy)
3. ID\_op\_maszyny (klucz obcy)
4. Maszyny\_nazwa

### Moduł Dysponenci maszyny

**tabela Dysponenci maszyn** określa prawa do użytkowania maszyn powinna zawierać następujące pozycje:

1. Identyfikator (ID\_dysponent) opiekuna maszyny (klucz własny)
2. ID\_dzial
3. Nazwa\_dzial
4. Dysp\_nazwisko
5. Dysp\_imie
6. Dysp\_dane (dane kontaktowe, stanowisko)
7. Dysp\_nazwa

|  |  |  |
| --- | --- | --- |
| Identyfikator |  | Int |
| Dzial | liczba | Int |
| Dysp\_imie | Txt 255 | string |
| Dysp\_nazwisko | Txt 255 | string |
| Dysp\_dane | Txt 255 | string |
| Dysp\_nazwa | Txt 255 | string |

*tabela 5. Nazwy komponentów, kolumn i ich typy w tabeli dysponent.*

Nazwy komponentów zakładki Dysponenci maszyn

|  |  |  |
| --- | --- | --- |
| wyszukiwanie |  | textBoxWyszukiwanieDysponent |
|  | buttonSzukajDysponent |
| Spis dysponentów maszyn | Lista nazw  listBoxDysponent\_ SelectedIndexChanged | listBoxDysponent |
| Dane dysponenta maszyny | Dysp\_imie | textBoxImieDysponent |
| Dysp\_nazwisko | textBoxNazwiskoDysponent |
| Dzial comboBoxDzial\_dysponent\_ SelectedIndexChanged | comboBoxDzialDysponent |
| Dysp\_dane | richTextBoxDysponent\_dane |
| lista maszyn z tabeli maszyny | listBoxMaszynyDysponenta |
|  | Identyfikator | toolStripStatusLabelIDdysponenta |
| Edycja spisu | nowa | buttonNowaDysponent |
| zapisz | buttonZapiszDysponent |
| anuluj | buttonAnulujDysponent |
| usuń | buttonUsunDysponent |

*tabela. 6 Nazwy komponentów modułu dysponent.*

**Tabela Maszyny\_Dysponent (multi)** - nadmiarowa – do wykorzystania gdy kilku dysponentów 1 maszyny – obecnie nie ma takiej potrzeby. Określa dysponentów maszyn, powinna zawierać następujące pozycje:

1. Identyfikator (klucz własny)
2. ID\_masz (klucz obcy)
3. ID\_dysponent (klucz obcy)

### Moduł Materiały

Tabela Materiały zawiera następujące pozycje:

Identyfikator (ID\_materiału) (klucz własny)

1. **Nazwa\_mat** - nazwa materiału
2. **Typ\_mat** - typ (np. blacha, rury)
3. **Rodzaj\_mat** - rodzaj (dane słownikowe np. mosiężne, aluminiowe itp)
4. **Jednostka\_miar\_mat** - wartość jednostkowa
5. **Stan\_mat -** Ilość na stanie magazynowym (szt/m)
6. **Zuzycie\_mat** - bieżące zużycie
7. **Odpad\_mat -** Odpad
8. **Stan\_min\_mat** - Stan minimalny materiału (szt/m)
9. **Zapotrzebowanie\_mat** – bieżące zapotrzebowanie
10. **Stan\_mag\_po\_mat** - stan magazynowy po zużyciu/zapotrzebowaniu
11. **Dostawca\_mat -** (dane dostawcy nazwa, link do strony, Dodatkowe informacje)

|  |  |  |
| --- | --- | --- |
| Identyfikator |  | Int |
| **Nazwa\_mat** | Txt 255 | string |
| **Typ\_mat** | Txt 255 | string |
| **Rodzaj\_mat** | Txt 255 | string |
| **Jednostka\_miar\_mat** | Txt 255 | string |
| **Stan\_mat** | liczba | Int |
| **Zuzycie\_mat** | liczba | Int |
| **Odpad\_mat** | liczba | Int |
| **Stan\_min\_mat** | liczba | Int |
| **Zapotrzebowanie\_mat** | liczba | Int |
| **Stan\_mag\_po\_mat** | liczba | Int |
| **Dostawca\_mat** | Txt 255 | string |

*tabela 7. Nazwy komponentów, kolumn i ich typy w tabeli materiały.*

|  |  |  |
| --- | --- | --- |
| sortowanie | groupBoxSortowanie | radioButtonNazwa\_mat |
| radioButtonTyp\_mat |
| radioButtonCena\_mat |
| radioButtonMagazyn\_ilosc\_mat |
| wyszukiwanie |  | textBoxWyszukaj\_mat |
|  | buttonSzukaj\_mat |
| Spis dostępnych materiałów | Lista nazw | listBoxMaterialy |
| Dane materiału | Typ | textBoxTypMat |
| Rodzaj | comboBoxRodzajMat |
| Nazwa Materiału | textBoxNazwaMat |
| Wartość jednostkowa | comboBoxJednostkaMat |
| Stan magazynowy materiałów | textBoxMagazynMat |
| Bieżące zużycie | textBoxZuzycieMat |
| Odpad | textBoxOdpadMat |
| Stan minimalny | textBoxMinMat |
| Zapotrzebowanie | textBoxZapotrzebowanieMat |
| Dane dostawców materiałów | Dostawcy | checkedListBoxDostawcyMat |
| Link do strony dostawcy głównego | linkLabelDostawca |
| Dodatkowe informacje | richTextBoxDostawca |
| Edycja spisu | nowa | buttonNowaMat |
| zapisz | buttonZapiszMat |
| anuluj | buttonAnulujMat |
| usuń | buttonUsunMat |

*tabela 8. Nazwy komponentów modułu materiały.*

Program umożliwia:

* + Informowanie o niskim stanie magazynowym,
  + dodawanie, odejmowanie, edycję pozycji spisu materiałów,
  + wyszukiwanie po ciągu znaków w nazwie materiału,
  + sortowanie tabeli magazynu po:
    - Nazwie,
    - Typie
    - Cenie
    - Ilości na stanie.

Tabela multi **Dostawca\_Materiał** powinna zawierać następujące pozycje:

1. Identyfikator (ID\_dostawcy) (klucz własny)
2. ID\_material – id materiału (klucz obcy tabeli materiały)
3. ID\_dostawca\_mat – id dostawcy materiału (klucz obcy tabeli dostawcy)

### Moduł Dostawca

Tabela **Dostawca** zawiera następujące pozycje:

1. Identyfikator (ID\_dostawcy) (klucz własny)
2. Nazwa\_dostawca\_mat – nazwa dostawcy materiału
3. Link\_dostawca\_mat – link do strony dostawcy materiału
4. Dod\_info\_dostawca\_mat – dodatkowe informacje np. upusty o sposobie dostawy/płatności itp.

|  |  |  |
| --- | --- | --- |
| Identyfikator |  | Int |
| **Nazwa\_dostawca\_mat** | Txt 255 | string |
| **Link\_dostawca\_mat** | Txt 255 | string |
| **Dod\_info\_dostawca\_mat** | Txt 255 | string |

*tabela 9. Nazwy komponentów, kolumn i ich typy w tabeli dostawca.*

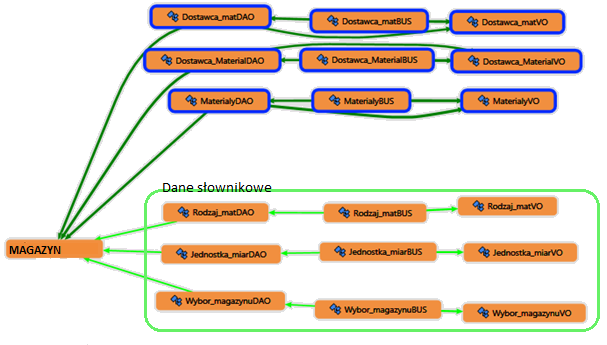
|  |  |  |
| --- | --- | --- |
| Spis dostępnych dostawców | Lista nazw | listBoxDostawcy |
| Dane dostawcy | Nazwa dostawcy | textBoxNazwaDostawcy |
| Dodatkowe informacje | richTextBoxDostawca |
| Materiału | textBoxNazwaMat |
| Wartość jednostkowa | comboBoxJednostkaMat |
| Stan magazynowy materiałów | textBoxMagazynMat |
| Bieżące zużycie | textBoxZuzycieMat |
| Odpad | textBoxOdpadMat |
| Stan minimalny | textBoxMinMat |
| Zapotrzebowanie | textBoxZapotrzebowanieMat |
| Dane dostawców materiałów | Dostawcy | checkedListBoxDostawcyMat |
| Link do strony dostawcy głównego | linkLabelDostawca |
| Dodatkowe informacje | richTextBoxDostawca |
| Link dostawcy | textBoxLink |
|  | UsuńLink | buttonUsunLink |
| Edycja spisu | nowa | buttonNowaMat |
| zapisz | buttonZapiszMat |
| anuluj | buttonAnulujMat |
| usuń | buttonUsunMat |

*tabela 10. Nazwy komponentów modułu dostawca.*

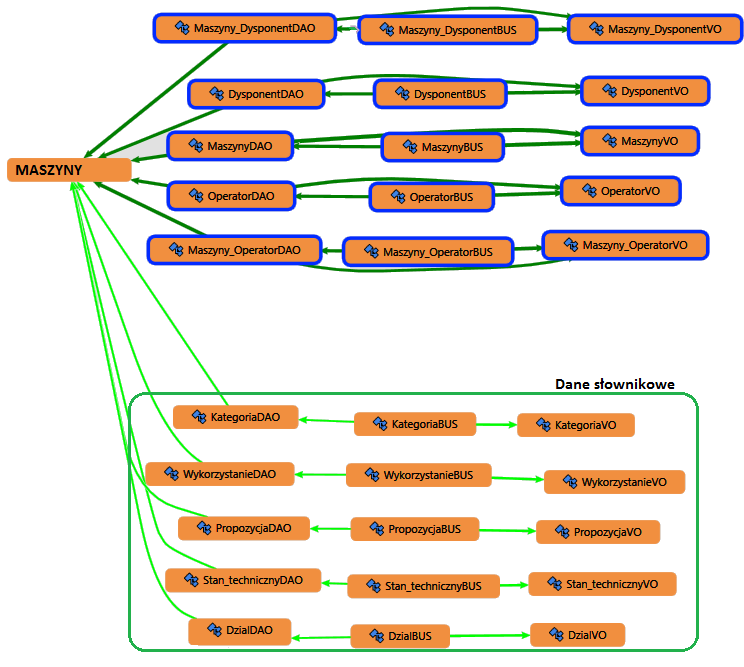
1. Opis funkcji i klas połączenia z bazą danych

Połączenie z bazą danych MS Access odbywa się trojstopniowo poprzez:

1. klasa z rozszerzeniem VO - wymiany danych z tabelą (pola tabeli, gettery i settery);
2. klasa z rozszerzeniem DAO - dostęp (Data Access Object) do tabeli (connectionstring oraz zapytania SQL);
3. klasa z rozszerzeniem BUS - obsługująca warstwę operacji biznesowych tabeli.



*Rys.3 Powiązania klas modułu magazyn.*



*Rys.4 Powiązania klas modułu maszyny.*

Spis treści

[1. Zapotrzebowanie 1](#_Toc536695395)

[2. Cel powstania 1](#_Toc536695396)

[3. Wykorzystanie programu 1](#_Toc536695397)

[4. Środowisko pracy 1](#_Toc536695398)

[5. Komponenty programu 1](#_Toc536695399)

[6. Zawartość projektu 2](#_Toc536695400)

[7. Opis bazy danych MS ACCESS RemaGUM 2](#_Toc536695401)

[7.1 Budowa modułu Magazyn 2](#_Toc536695402)

[7.2 Budowa modułu Maszyny 3](#_Toc536695403)

[8. Opcje kompilacji 3](#_Toc536695404)

[9. Użyte komponenty środowiska VS 4](#_Toc536695405)

[10. Opis funkcji i klas 4](#_Toc536695406)

[10.1. Klasa Main – start aplikacji. 4](#_Toc536695407)

[10.2. Klasa Rest – hasło bazy danych MSACCESS. 5](#_Toc536695408)

[10.3. Obsług połączeń z bazą danych 5](#_Toc536695409)

[10.4. Obsługa działania programu – Klasa główna. 53](#_Toc536695410)

[10.5. Komponenty wizualne – okienko „o RemaGUM”. 66](#_Toc536695411)

[11. Główne komponenty programu 66](#_Toc536695412)

[11.1. Moduł Maszyny – komponenty 66](#_Toc536695413)

[11.2. Moduł Operator maszyny 69](#_Toc536695414)

[11.3. Moduł Dysponenci maszyny 70](#_Toc536695415)

[11.4. Moduł Materiały 71](#_Toc536695416)

[11.5. Moduł Dostawca 72](#_Toc536695417)

[12. Opis funkcji i klas połączenia z bazą danych 73](#_Toc536695418)