

Technologie obiektowe Projekt		
27. Wizualizacja zawartości składu lub bazy danych 23. Konwersja danych pomiędzy różnymi modelami		
Github: https://github.com/Obertyn	E-mail: sikoramateusz1999@gmail.com	
Imię i nazwisko: Mateusz Sikora	Kierunek: Informatyka	Grupa: 21B

Spis treści	2
1. Cele projektu.....	3
2. Technologie użyte w projekcie	4
3. Opis kodu	5
4. Interfejs i praca programu	6
5. Podsumowanie	11
6. Bibliografia.....	12

1. Cele projektu

Głównymi celami projektu było stworzenie programu, który na podstawie skryptów SQL będzie umieć wizualizować zawartość bazy danych oraz konwertować dane pomiędzy różnymi modelami. System zarządzania bazami danych, który umożliwił mi korzystanie ze skryptów SQL, był Microsoft SQL Server. Skrypty SQL można uzyskać przez 2 sposoby: albo poprzez ściągnięcie ich z oficjalnej strony Microsoftu, albo poprzez wygenerowanie ich na podstawie posiadanych już baz, korzystając z programu Microsoft SQL Server Management [1].

Kolejnym celem było stworzenie graficznego interfejsu, dzięki któremu korzystanie z programu będzie proste dla użytkowników. Cel ten został osiągnięty dzięki bibliotece Guietta.

Co istotne, program musiał zostać napisany w sposób obiektowy. Programowanie Obiektowe jest metodą projektowania i implementowania systemów informatycznych polegającą na modelowaniu obiektów rzeczywistych przy użyciu obiektów programowych [2]. Językiem, który został wybrany do napisania programu był Python, który wspiera różne paradygmaty programowania: obiektowy, imperatywny oraz w mniejszym stopniu funkcyjny.

2. Technologie użyte w projekcie

Python jest interpretowanym językiem programowania wysokiego poziomu, który oferuje możliwości programowania obiektowego, strukturalnego i funkcyjnego [3].

MS SQL Server to system zarządzania bazą danych, wspierany i rozpowszechniany przez korporację Microsoft. Umożliwia dołączanie lub odłączanie baz danych do/z serwera, tworzenie struktury bazy danych i wypełnienie jej, usuwanie bazy danych, wykonywanie poleceń SQL i przygotowywanie funkcji wbudowanych SQL [4].

XML to język znaczników i format pliku do przechowywania, przesyłania i rekonstrukcji dowolnych danych. XML pozwala na zapisywanie danych w postaci, która jest łatwa do odczytu zarówno przez maszyny jak i przez ludzi [5].

XSD (XML Schema) to standard służący do definiowania struktury dokumentu XML. Głównym celem XSD jest przede wszystkim definiowanie elementów, atrybutów i typów danych, które może zawierać dokument [6].

CSV to format przechowywania danych, charakteryzujący się tym, że wartości w tych plikach tworzą tabelaryczną bazę danych. Wartości, które znajdują się w poszczególnych kolumnach, oddzielone są w tych plikach przecinkami. Pliki CSV można otwierać za pomocą Excela lub programów otwierających pliki tekstowe [7].

Matplotlib to popularna biblioteka zawierająca wiele modułów do sporządzania wykresów, wśród których najpopularniejszy jest użyty w tym projekcie pyplot [8].

Networkx to biblioteka służąca do tworzenia grafów [9].

Guietta to biblioteka, która pozwala tworzyć graficzny interfejs użytkownika [10].

3. Opis kodu

Na początku programu wczytywane są potrzebne biblioteki:

```
from guietta import _, Gui, Quit, CB #gui
import codecs #kopiuje zawartość pliku
import shutil #dzięki tej bibliotece zmienimy system kodowania Unicode
import re #zaawansowane replace
import networkx as nx #grafy
import matplotlib.pyplot as plt #wykresy
import os
```

Rys. 1. Wczytane biblioteki

Następnie implementowana jest klasa main, w której definiowane jest, jak będzie wyglądać graficzny interfejs użytkownika:

```
class main:
    global re

    opdict = {'tak': '1',
              'nie': '0'}

    gui = Gui(
        ["ścieżka do pliku:", "_a_", "separator:", "_b_", "1 linijka:", CB('op', opdict), ["CSV"]],
        ["ścieżka do pliku:", "_c_", "_", "_", "_", "_", ["XML"]],
        ["ścieżka do pliku:", "_d_", "_", "_", "_", "_", ["XSD"]],
        ["ścieżka do pliku:", "_e_", "_", "_", "_", "_", ["Wizualizacja"]],
        ["_", "_", "_", "_", "_", "_", Quit]
    )

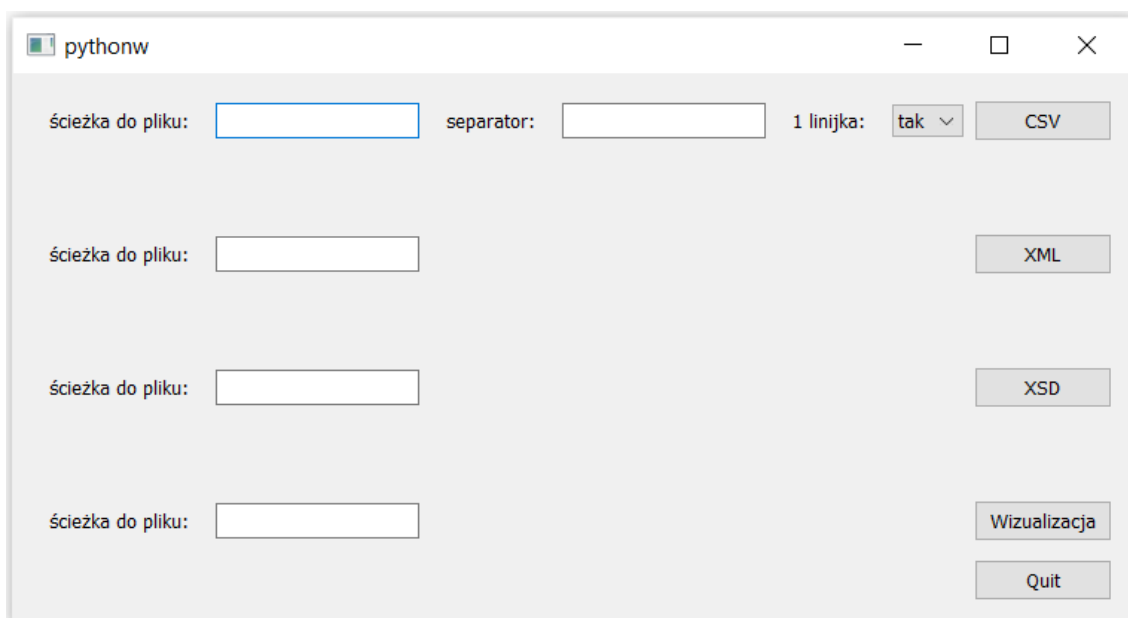
main=main()
```

Rys. 2. Klasa main

Pod klasą main znajdują się jeszcze 4 rozbudowane klasy: SQL_to_CSV, SQL_to_XML, SQL_to_XSD oraz visualisation. Pierwsze 3 służą do konwersji danych z języka SQL, a ostatnia odpowiada za wizualizację zawartości bazy danych. Pierwsze trzy klasy mają wspólną dużą część kodu, ponieważ żeby dokonać konwersji, potrzebują te same dane znajdujące się w skrypcie SQL. Działanie wszystkich klas opiera się na tworzeniu kilku dwuwymiarowych i jednowymiarowych tablic, w których przechowywane są dane pobrane ze skryptu SQL.












4. Interfejs i praca programu

Po włączeniu programu pojawia się graficzny interfejs, w którym możemy wybrać jakie czynności ma podjąć program. Po lewej stronie interfejsu są pola, w których należy podać ścieżkę do skryptu SQL. Jeśli skrypt SQL znajduje się w tym samym miejscu co program, to wystarczy podać tylko nazwę skryptu. Po prawej stronie interfejsu mamy przyciski, których naciśnięcie spowoduje stworzenie odpowiedniego pliku. Dodatkowo w górnej części interfejsu są dwie dodatkowe opcje dotyczące generowania plików CSV. Pierwsza z nich pozwala na wybranie separatora, który będzie oddzielał dane w plikach CSV. Druga opcja pozwala nam zdecydować czy chcemy, żeby w pierwszej linijce plików CSV znajdowały się nazwy kolumn.



Rys. 3. Interfejs programu

Korzystając z udostępnionej przez Microsoft bazy danych School zostanie zaprezentowane działanie programu [11]. Wybierając wszystkie opcje dostępne w programie, zostanie stworzonych 11 plików – 1 plik PNG, 1 plik XSD, 1 plik XML oraz 8 plików CSV. Ilość stworzonych plików CSV jest równa ilości tabel w bazie.

 dbo.Course	26.06.2022 19:49	Plik w formacie wa...	1 KB
 dbo.CourseInstructor	26.06.2022 19:49	Plik w formacie wa...	1 KB
 dbo.Department	26.06.2022 19:49	Plik w formacie wa...	1 KB
 dbo.OfficeAssignment	26.06.2022 19:49	Plik w formacie wa...	1 KB
 dbo.OnlineCourse	26.06.2022 19:49	Plik w formacie wa...	1 KB
 dbo.OnsiteCourse	26.06.2022 19:49	Plik w formacie wa...	1 KB
 dbo.Person	26.06.2022 19:49	Plik w formacie wa...	2 KB
 dbo.StudentGrade	26.06.2022 19:49	Plik w formacie wa...	1 KB
 SQL_to_XML	26.06.2022 19:49	Plik XML	25 KB
 SQL_to_XSD	26.06.2022 19:49	Plik XSD	3 KB
 School	26.06.2022 19:49	Plik PNG	37 KB

Rys. 4. Pliki wygenerowane przez program

Poniżej pokazane są dwa pliki CSV stworzone na podstawie tabeli Course. W pierwszym pliku ustawiono przecinek jako separator i zaznaczono brak pierwszej linii, a w drugim pliku ustawiono średnik jako separator i zaznaczono istnienie pierwszej linii.

```
1045, Calculus, 4, 7
1050, Chemistry, 4, 1
1061, Physics, 4, 1
2021, Composition, 3, 2
2030, Poetry, 2, 2
2042, Literature, 4, 2
3141, Trigonometry, 4, 7
4022, Microeconomics, 3, 4
4041, Macroeconomics, 3, 4
4061, Quantitative, 2, 4
```

Rys. 5. Plik CSV z tabelą Course z przecinkiem jako separator

```
CourseID; Title; Credits; DepartmentID
1045; Calculus; 4; 7
1050; Chemistry; 4; 1
1061; Physics; 4; 1
2021; Composition; 3; 2
2030; Poetry; 2; 2
2042; Literature; 4; 2
3141; Trigonometry; 4; 7
4022; Microeconomics; 3; 4
4041; Macroeconomics; 3; 4
4061; Quantitative; 2; 4
```

Rys. 6. Plik CSV z tabelą Course z średnikiem jako separator

Poniżej pokazane są fragmenty plików XML i XSD stworzone na podstawie bazy School. Plik XSD jest o wiele krótszy niż plik XML, ponieważ zawiera tylko strukturę bazy.

```
<School>
  <dbo.Course>
    <1>
      <CourseID>1045</CourseID>
      <Title>Calculus</Title>
      <Credits>4</Credits>
      <DepartmentID>7</DepartmentID>
    </1>
    <2>
      <CourseID>1050</CourseID>
      <Title>Chemistry</Title>
      <Credits>4</Credits>
      <DepartmentID>1</DepartmentID>
    </2>
    <3>
      <CourseID>1061</CourseID>
      <Title>Physics</Title>
      <Credits>4</Credits>
      <DepartmentID>1</DepartmentID>
    </3>
    <4>
      <CourseID>2021</CourseID>
      <Title>Composition</Title>
      <Credits>3</Credits>
      <DepartmentID>2</DepartmentID>
    </4>
  </dbo.Course>
</School>
```

Rys. 7. Plik XML z bazą School

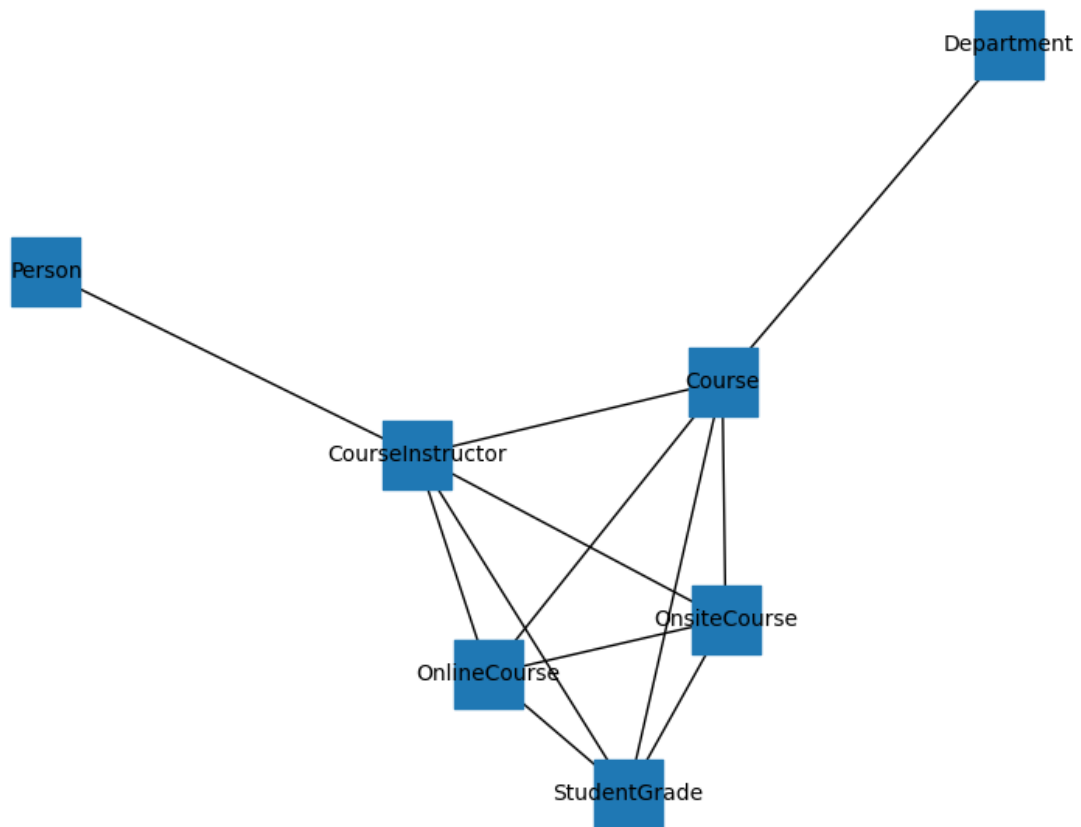

```

<xsd:element name="School">
  <xsd:complexType name="dbo.Course">
    <xsd:element name="CourseID" type="xs:int"/>
    <xsd:element name="Title" type="xs:string"/>
    <xsd:element name="Credits" type="xs:int"/>
    <xsd:element name="DepartmentID" type="xs:int"/>
  </xsd:complexType>
  <xsd:complexType name="dbo.CourseInstructor">
    <xsd:element name="CourseID" type="xs:int"/>
    <xsd:element name="PersonID" type="xs:int"/>
  </xsd:complexType>
  <xsd:complexType name="dbo.Department">
    <xsd:element name="DepartmentID" type="xs:int"/>
    <xsd:element name="Name" type="xs:string"/>
    <xsd:element name="Budget" type="xs:decimal"/>
    <xsd:element name="StartDate" type="xs:dateTime"/>
    <xsd:element name="Administrator" type="xs:int"/>
  </xsd:complexType>
  <xsd:complexType name="dbo.OfficeAssignment">
    <xsd:element name="InstructorID" type="xs:int"/>
    <xsd:element name="Location" type="xs:string"/>
    <xsd:element name="Timestamp" type="xs:dateTime"/>
  </xsd:complexType>
  <xsd:complexType name="dbo.OnlineCourse">
    <xsd:element name="CourseID" type="xs:int"/>
    <xsd:element name="URL" type="xs:string"/>
  </xsd:complexType>

```

Rys. 8. Plik XSD z bazą School

Ostatnim stworzonym plikiem jest plik graficzny w formacie PNG, który pokazuje strukturę bazy danych. Każda tabela w tej grafice jest oznaczona niebieskim kwadratem, a krawędzie łączące tabele pokazują, że te tabele posiadają wspólny klucz.



Rys. 9. Wizualizacja zawartości bazy danych School

5. Podsumowanie

Celem projektu było stworzenie programu, który potrafi wizualizować zawartość bazy danych oraz przeprowadzać konwersje danych ze skryptów SQL na pliki XML, XSD i CSV. Cele te zostały zrealizowane. Dodatkowo został zaprogramowany graficzny interfejs programu, dzięki czemu korzystanie z programu jest proste. Co istotne program został napisany w sposób obiektowy.

6. Bibliografia

- [1] <https://docs.microsoft.com/pl-pl/dotnet/framework/data/adonet/sql/linq/downloading-sample-databases>
- [2] <http://ii.uwb.edu.pl/rybnik/WdPO/WdPO%20W1pp.pdf>
- [3] <http://www.tu.kielce.pl/~stapor/materials-open/PodstawyInformatyki/python-wprowadzenie.pdf>
- [4] https://ssamolej.kia.prz.edu.pl/dydaktyka/inf_1EE_ZI/SQL_MS_SQL.pdf
- [5] <https://www.samouczekprogramisty.pl/xml-dla-poczatkujacych>
- [6] <https://www.techtarget.com/whatis/definition/XSD-XML-Schema-Definition>
- [7] <https://datatracker.ietf.org/doc/html/rfc4180>
- [8] <https://matplotlib.org>
- [9] <https://networkx.org>
- [10] <https://guietta.readthedocs.io/en/latest>
- [11] <https://docs.microsoft.com/en-us/ef/ef6/resources/school-database>