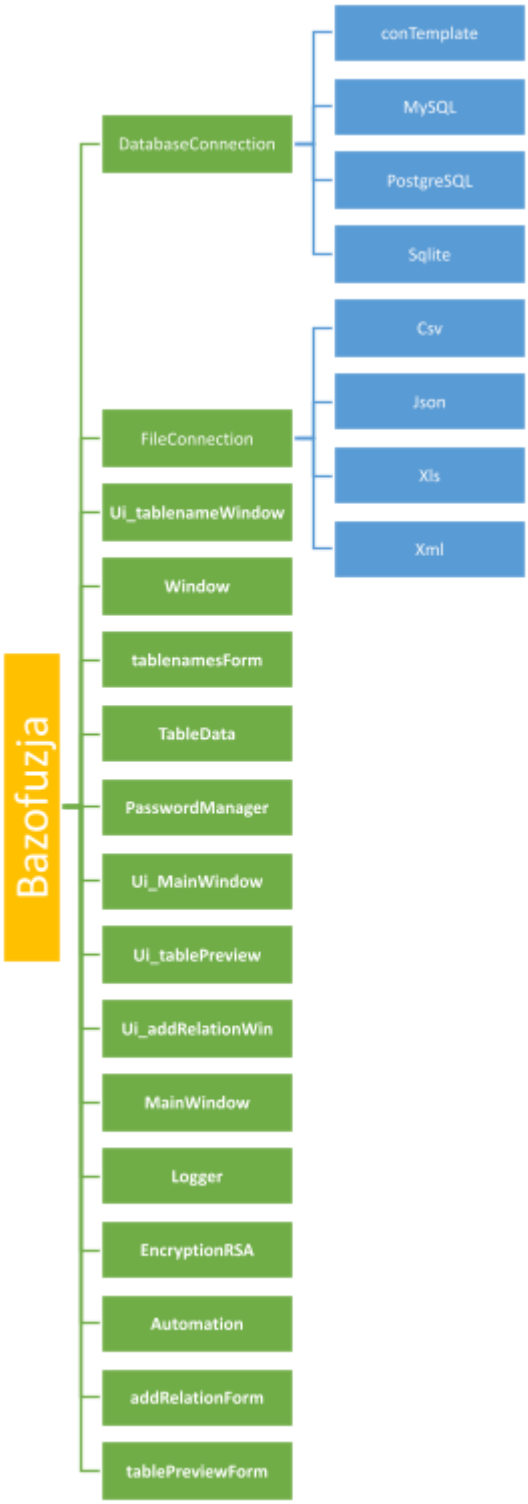
Dokumentacja programistyczna

1. Hierarchia klas



2. Lista klas i funkcji

Klasa conTemplate

Metody:

- def connect (self)
  Metoda wywołująca połączenie z bazą
- def getTableList (self)
  Metoda zwracająca listę nazw tabel z bazy

Klasa addRelationForm

Metody:

- def __init__ (self)
  inicjalizuje obiekty klasy addRelationForm, czyli klasy okna do dodawania relacji
- def add (self)
  dodaje relacje do głównej klasy MainWindow
- def show (self)
  wyświetla okno
- def clear (self)

  usuwa zapisane dane w polach przy ponownym włączeniu

Klasa tablePreviewForm

Metody:

- def __init__ (self)
  inicjalizuje obiekty klasy tablePreviewForm, czyli klasy okna dopodglądu danych wejściowych
- def show (self)
  wyświetla okno
- def clear (self)
  usuwa zapisane dane w polach przy ponownym włączeniu
- def getTableNames (self)

pobiera listę zaznaczonych tabel z klasy MainWindow

- def loadTable (self)

  wczytuje dane z tabeli, której nazwa aktualnie znajduje się na liście rozwijanej

Klasa MainWindow

Metody:

- def __init__ (self)
  inicjalizuje obiekty klasy MainWindow, czyli klasy głównego okna
- def show (self)
  wyświetla okno
- def exit (self)
  zamyka aplikacje
- def clearAll (self)
  czyści pozostałości danych po stronie interfejsu
- def cleanLoginInfo (self)
  czyści pola z danymi logowania
- def hideWindow (self)
  obsługuje minimalizacje okna
- def maxWindow (self)
  obsługuje pełny ekran okna
- def changeMode (self)
  zmienia styl kolorystyczny (jasny/ciemny)
- def showModulesDir (self)
  obsługuje otworzenie folderu plugins
- def saveRelations (self)
  transferuje relacje do bazy wyjściowej
- def fillRelationTable (self)
  jeżeli relacje lub klucze główne istnieją w bazie wejściowej, to metoda wypełnia nimi tabele
- def addRelation (self)
  uruchamia okno do dodawania relacji
- def setRow (self, l)
  pobiera l, czyli listę z jedną relacją w formie [nazwa tabeli, nazwa kolumny, nazwa tabeli, nazwa kolumny], i dodaje ją do tabeli.

- def deleteRelation (self)
  usuwa zaznaczoną relację z tabeli
- def showTablenames (self)
  wyświetla okno do zmiany nazw tabel
- def showPreview (self)
  wyświetla okno poglądowe danych wejściowych
- def showEditPage (self)
  wyświetla stronę edycji danych w opcjach zaawansowanych
- def showRelations (self)
  wyświetla stronę edycji relacji
- def showSystemInput (self)
  wyświetla stronę z dostępnymi systemami danych na wejściu
- def showSystemOutput (self)
  wyświetla stronę z dostępnymi systemami danych na wyjściu
- def showSelection (self)
  wyświetla stronę z tabelami do wyboru lub stronę do tworzenia tabel z danych płaskich
- def showSummary (self)
  wyświetla podsumowanie
- def showSaveOutput (self)
  wyświetla dla wyjścia stronę logowania w przypadku bazy lub stronę wyboru pliku w przypadku plików płaskich i sqlite
- def showSourceInput (self)
  wyświetla dla wejścia stronę logowania w przypadku bazy lub stronę wyboru pliku w przypadku plików płaskich i sqlite
- def setSystemInput (self, item)
  pobiera item jako nazwę wybranego systemu wejściowego
- def setSystemOutput (self, item)
  pobiera item jako nazwę wybranego systemu wyjściowego
- def getSelectedTables (self)
  zwraca listę zaznaczonych tabel
- def deleteNullsFun (self)
  wywołuje usunięcie pustych rekordów
- def deleteDupliFun (self)
  wywołuje usunięcie duplikatów rekordów
- def deleteByIDFun (self)
  wykonuje interpolację rekordów

- def loadFileInput (self)
  wczytuje plik wejściowy
- def saveFileOutput (self)
  zapisuje plik wyjściowy
- def loginDatabase (self)
  pobiera dane logowania z interfejsu i łączy z odpowiednią bazą
- def submitTables (self)
  pokazuje okno z wyborem systemu wyjściowego
- def submitAll (self)
  wykonuje transfer
- def createStr (self)
  tworzy tekst bazy danych do podsumowania
- def putDataSum (self)
  pobiera dane i wypisuje je na ekran do podsumowania
- def loadFlatPreview (self)
  wyświetla dane z plików płaskich
- def createNewRows (self, number)
  tworzy “number” puste wersy, gdzie number to ilość kolumn w pliku płaskich
- def splitFlatFileData (self)
  rozdziela tabele i kolumny z pliku płaskiego
- def collectOperations (self)

  pobiera wykonane operacje i wyświetla je w podsumowaniu

Klasa Automation

Metody:

- def __init__ (self, Win)
  konstruktor klasy przyjmujący obiekt klasy Window
- def setDatabases (self)
  zapisuje wszelkie dane dotyczące wejścia / wyjścia bazy danych
- def setLoginData (self, dict, Db)
  dodaje do słownika dane potrzebne do ponownego logowania
- def addOperation (self, newAcction,listOfArg)
  tworzy słownik zawierający listę operacji
- def save (self)

funkcja wykonująca zapis wykonywanych czynności do pliku binarnego

- def read(self,filepath)

  funkcja czytająca plik binarny i wykonująca wszystkie operacje, które zostały w nim zapisane

Klasa EncryptionRSA

Metody:

- def __init__ (self)
  konstruktor klasy kontrolujący pliki i klucz prywatny
- def generate (self)
  funkcja generująca i zwracająca klucz prywatny
- def setPrivKey (self)
  funkcja pobierająca klucz prywatny z pliku
- def encrypt (self)
  funkcja szyfrująca hasła do pliku
- def encryptPassword (self, password)
  funkcja szyfrująca hasło
- def decrypt (self)
  funkcja deszyfrująca hasła z pliku
- def decryptPassword (self, password)
  funkcja deszyfrująca hasło
- def exportPrivKey (self)
  funkcja zapisująca klucz prywatny do pliku
- def addPassword (self, password)
  funkcja dodająca zaszyfrowane hasło do pliku
- def readPasswords (self)

  funkcja zwracająca odszyfrowane hasła

Klasa PasswordManager

Metody:

- def __init__ (self)
  konstruktor dbający o plik z loginami i informacjami o zapisanych bazach

- def readLogins (self)
  funkcja zwracająca wszystkie informacje z pliku z loginami
- def addLogin (self, login, databaseName, databaseAddress)
  funkcja dodająca loginy i dane o bazach do pliku
- def readFittingPassword (self, login, databaseName, databaseAddress)

  funkcja deszyfrująca odpowiednie hasło dla odpowiedniego loginu w zapisanej bazie

Klasa DatabaseConnection

Metody:

- def __init__ (self, host, user, password, database="", uiOutput=None)
  Konstruktor tworzący obiekt klasy, przyjmujący jako parametry dane do logowania do bazy, dodatkowo obiekt ui będący wyjściem dla generowanych logów
- def getHost (self)
  Metoda zwracająca pole self.host
- def getUser (self)
  Metoda zwracająca pole self.user
- def getPassword (self)
  Metoda zwracająca pole self.password
- def getDatabase (self)
  Metoda zwracająca pole self.database
- def getIfLoginNeeded (self)
  Metoda zwracająca pole ifLoginNeeded, rozróżniające bazy do których należy połączyć się przez silnik SQLAlchemy od tych, które zawarte są w jednym pliku
- def setDatabase (self, database, ifReload=False)
  Metoda zmieniająca połączoną bazę w locie (ifReload = True wykonuje również automatyczne przeładowanie połączenia)
- def getFilePath (self)
  Metoda zwracająca ścieżkę pliku bazy (jedynie dla baz z polem ifLoginNeeded = False)
- def close (self)
  Metoda bezpiecznie zamykająca połączenie z bazą

- def commit (self)
  Metoda wykonująca commit w bazie do której połączony jest obiekt
- def getName (self)
  Metoda zwracająca z namespace'u nazwę klasy
- def executeQuery (self, query)
  Metoda wykonująca podane w argumencie query
- def getTableAsDataFrame (self, tableName)
  Metoda zwracająca dataframe zawierający strukturę i dane z tabeli o podanej nazwie
- def exportTableToDatabase (self, table, ifExists="append")
  Metoda eksportująca gotowy obiekt klasy dataframe do bazy. Argument ifExist (append, replace, fail) definiuje akcję, jaką metoda podejmie, gdy tabela w bazie już istnieje)
- def chunking (self, table)
  Metoda przygotowująca maksymalny rozmiar pojedynczego chunku, jaki można pobrać z danej tabeli przy dostępnej pamięci operacyjnej
- def connect (self)
  Metoda wywołująca połączenie z bazą
  def getTableList (self)
  Metoda zwracająca listę nazw tabel z bazy
- def getPrimaryKeys (self)
  Wykrywa wszystkie klucze główne w podanej relacyjnej bazie danych.
- def getForeignKeys (self)
  Funkcja wykrywająca wszystkie klucze obce w relacyjnej bazie danych.
- def addPrimaryKey (self, tableName, columnName)
  Funkcja, która dodaje klucze główne w docelowej bazie danych.
- def addForeignKey (self, constraintName, tableName, columnName, referencedTableName, referencedColumnName)
  Dodaje klucze obce w docelowej bazie danych.

Klasa FileConnection

Metody:

- def __init__ (self, filepath, uiOutput)
  Konstruktor przyjmujący ścieżkę do pliku i obiekt ui będący wyjściem dla generowanych logów

- def getName (self)
  Metoda zwracająca z namespace'u nazwę klasy
- def getFilePath (self)
  Metoda zwracająca ścieżkę do obsługiwanego pliku
- def getIfLoginNeeded (self)
  Metoda zwracająca pole ifLoginNeeded, rozróżniające rozwiązania do których należy połączyć się przez silnik SQLAlchemy od tych, które zawarte są w jednym pliku
- def setMultipleFilenames (self, data)
  Metoda umożliwiająca zmianę wielu nazw otwartych plików na raz
- def getFileAsDataFrame (self)
  Metoda zwracająca dataframe z pliku o podanej nazwie i rozszerzeniu
- def exportDataFrameToFile (self)
  Metoda eksportująca wybrany dataframe do pliku o wybranym rozszerzeniu

Klasa Logger

Metody:

- def __init__ (self, filename, uiOutput=None)
  Konstruktor tworzący obiekt loggera i otwierający plik logów o podanej nazwie. Dodatkowo obiekt może przyjąć referencję do obiektu ui, w celu
- def createLog (self, objectName, logString, ifPrint=False)
  Tworzy log i dodaje go do pliku (dla ifPrint = True wyrzuca go również do stdout oraz obiektu uiOutput)
- def closeFile (self)
  Zamyka bezpiecznie plik

Klasa TableData

Metody:

- def __init__ (self, tableName, tableDataFrame)
  Konstruktor przyjmujący nazwę tabeli i dataframe tabeli zawierający jej strukturę oraz zawarte w niej dane
- def getDataFrame (self)
  Metoda zwracająca przechowywany dataframe

- def getTableName (self)
  Metoda zwracająca nazwę tabeli
- def setTableName (self, name)
  Metoda zmieniająca nazwę tabeli
- def removeNullRows (self)
  Usuwa puste rekordy
- def removeNullColumns (self)
  Metoda usuwająca kolumny z wartościami null
- def removeByIndex (self, i)
  Metoda usuwająca rekord po indeksie
- def removeColumn (self, i)
  Metoda usuwająca kolumnę o podanej nazwie
- def removeDuplicates (self)
  Metoda usuwająca duplikaty rekordów
- def interpolateNullRecords (self)
  Metoda interpolująca liniowo puste rekordy

Klasa Csv

Metody:

- def getFileAsDataFrame (self, separator=',')
  Metoda zwracająca dataframe z pliku o podanej nazwie i rozszerzeniu
- def exportDataFrameToFile (self, data, separator=',')
  Metoda eksportująca wybrany dataframe do pliku o wybranym rozszerzeniu

Klasa Json

Metody:

- def getFileAsDataFrame (self)
  Metoda zwracająca dataframe z pliku o podanej nazwie i rozszerzeniu
- def exportDataFrameToFile (self, data)
  Metoda eksportująca wybrany dataframe do pliku o wybranym rozszerzeniu

Klasa MySQL

Metody:

- def connect (self)
  Metoda wywołująca połączenie z bazą
- def getTableList (self)
  Metoda zwracająca listę nazw tabel z bazy
- def getPrimaryKeys (self)
  Wykrywa wszystkie klucze główne w relacyjnej bazie danych MySQL.
- def getForeignKeys (self)
  Funkcja wykrywająca wszystkie klucze obce w relacyjnej bazie danych MySQL.
- def addPrimaryKey (self, tableName, columnName)
  Funkcja, która dodaje klucze główne do docelowej bazy danych MySQL.
- def addForeignKey (self, constraintName, tableName, columnName, referencedTableName, referencedColumnName)
  Dodaje klucze obce do MySQL, o ile taka opcja była podana przez użytkownika.

Klasa PostgreSQL

Metody:

- def connect (self)
  Metoda wywołująca połączenie z bazą
- def getTableList (self)
  Metoda zwracająca listę nazw tabel z bazy
- def getPrimaryKeys (self)
  Wykrywa wszystkie klucze główne w PostgreSQL.
- def getForeignKeys (self)
  Funkcja wykrywająca wszystkie klucze obce w PostgreSQL.
- def addPrimaryKey (self, tableName, columnName)
  Dodaje klucze główne w docelowej bazie danych PostgreSQL.
- def addForeignKey (self, constraintName, tableName, columnName, referencedTableName, referencedColumnName)

  Dodaje klucze obce w PostgreSQL.

Klasa Sqlite

Metody:

- def __init__ (self, filename, uiOutput)
  Konstruktor przyjmujący ścieżkę do pliku i obiekt GUI będący wyjściem dla generowanych logów
- def connect (self)
  Metoda wywołująca połączenie z bazą
- def getTableList (self)
  Metoda zwracająca listę nazw tabel z bazy

Klasa Xls

Metody:

- def getFileAsDataFrame (self, sheetName=0)
  Zwraca obiekt typu dataframe z pliku o podanej nazwie i rozszerzeniu
- def exportDataFrameToFile (self, data, sheetname="__default__")
  Metoda eksportująca wybrany dataframe do pliku o wybranym rozszerzeniu

Klasa Xml

Metody:

- def getFileAsDataFrame (self)
  Zwraca obiekt typu dataframe z pliku o podanej nazwie i rozszerzeniu
- def exportDataFrameToFile (self, data)
  Eksportuje wybrany dataframe do pliku o wybranym rozszerzeniu

Klasa Ui_MainWindow

Metody:

- def setupUi (self, MainWindow)
  Tworzy wszystkie elementy interfejsu
- def retranslateUi (self, MainWindow)

  Dodaje widoczne nazwy dla interfejsu

Klasa Window

Metody:

- def __init__ (self)
  Konstruktor tworzący obiekt klasy Window
- def clearAll (self)
  ustawia wartości początkowe dla obiektów bazy/pliku wejściowego i bazy/pliku wyjściowego oraz ich nazw
- def LoadPlugins (self)
  Znajduje ścieżkę do podkatalogu plugins w katalogu głównym naszej aplikacji, a następnie wykrywa wszystkie moduły znajdujące się w tym katalogu. Importuje biblioteki potrzebne do obsługi każdej wtyczki.
- def DatabaseList (self)
  zwraca listę dostępnych modułów
- def login (self, ui, dbNameStr)
  pobiera dane do logowania z panelu logowania aplikacji, a następnie tworzy obiekt DatabaseConnection, który wymaga logowania
- def isLoginFormNeeded (self, tech)
  Zwraca prawdę jeśli potrzeba wyświetlić panel logowania. Przyjmuje stringa będącego nazwą modułu
- def isFlatFile (self, db)
  Zwraca prawdę jeśli obiekt db jest obiektem klasy FileConnection
- def autoSave(self)
  funkcja wywołująca zapis wykonywanych na bazie/pliku operacji
- def autoRead(self,filepath)
  funkcja wywołująca odczyt i wykonanie operacji zapisanych w podanym pliku
- def isDatabase (self, db)
  Zwraca prawdę jeśli obiekt db jest obiektem klasy DatabaseConnection
- def returnName (self, Db)
  Zwraca nazwę modułu podanej w argumencie bazy danych. Np. Jeśli Db to obiekt klasy MySql, zwróci “MySql”
- def setDbIn (self, DbIn)
  Ustawia bazę/plik wejściowy
- def setDbOut (self, DbOut)

ustawia bazę/plik wyjściowy

- def setTechIn (self, techIn)
  ustawia nazwę systemu bazy/pliku wejściowego
- def setTechOut (self, techOut)
  ustawia nazwę systemu bazy/pliku wyjściowego
- def getDbIn (self)
  zwraca obiekt bazy danych/pliku z którego transferujemy dane
- def getDbOut (self)
  zwraca obiekt bazy danych/pliku do którego transferujemy dane
- def getTechIn (self)
  zwraca informacje o nazwie modułu wejściowego
- def getTechOut (self)
  zwraca informacje o nazwie modułu wyjściowego
- def getData (self, table)
  zwraca listę obiektów dataFrame powiązanych z tabelą o podanej nazwie
- def loadLogin (self, ui, Db)
  wczytuje dane logowania do interfejsu
- def loadTableList (self, tab)
  wczytuje listę dostępnych modułów
- def fileIn (self, dbNameStr, fileName, ui)
  tworzy obiekt klasy DatabaseConnection, który podawany jest z pliku lub obiekt klasy FileConnection
- def getTable (self, TableList, newTableList, ifExists="append")
  Tworzy obiekty dataFrame, na których wykonywane są późniejsze operacje przed zatwierdzeniem zmian, porcjuje tabele, gdy nie mieści się w RAM

- def getFilePath (self, db)
  Pobiera obiekt pliku i zwraca ścieżkę pliku danych
- def getPrimaryKeys (self)
  Wykrywa wszystkich klucze głównych w wejściowej bazie danych.
- def getForeignKeys (self)
  Funkcja wykrywająca wszystkie klucze obce w wejściowej bazie danych.
- def addPrimaryKey (self, tableName, columnName)
  Funkcja, która dodaje klucze główne w docelowej bazie danych.
- def addForeignKey (self, constraintName, tableName, columnName, referencedTableName, referencedColumnName)

Dodaje klucze obce w docelowej bazie danych, o ile taka opcja była zaznaczona przez użytkownika.

- def createTable (self, tempL, listOfTables)
  Rozdziela dane z pliku płaskiego na tabele. tempL zawiera listę tabel, a listOfTables zawiera listę krotek z nazwą tabeli i nazwą kolumny, która ma się w niej znajdować.
- def deleteNullsFun (self, tablename)
  Usuwa puste rekordy z obiektów dataFrame podanej tabeli
- def deleteDupliFun (self, tablename)
  Usuwa duplikaty z obiektów dataFrame podanej tabeli
- def closeAll (self, main_win)

  zamyka połączenia z bazami danych jeśli są otwarte

Klasa tablenamesForm

Metody:

- def __init__ (self)
  inicjalizuje obiekty klasy tablenamesForm, czyli klasy okna do zmiany nazw pól
- def show (self)
  wyświetla okno
- def loadData (self, listT)
  wczytuję zaznaczone nazwy tabel jako listT
- def saveModifNames (self)
  zapisuje zmodyfikowane nazwy tabel
- def discardAll (self)
  zamyka okno bez zapisu
- def getNewNames (self)

  zwraca listę z zmienionymi nazwami tabel

Klasa Ui_addRelationWin

Metody:

- def setupUi (self, addRelationWin)
  tworzy wszystkie obiekty interfejsu okna dodawania relacji
- def retranslateUi (self, addRelationWin)

zmienia widoczne nazwy obiektów

Klasa Ui_tablenameWindow

Metody:

- def setupUi (self, tablenameWindow)
  tworzy wszystkie obiekty okna zmiany nazw tabel
- def retranslateUi (self, tablenameWindow)

  dodaje widoczne nazwy obiektom w oknie

Klasa Ui_tablePreview

Metody:

- def setupUi (self, tablePreview)
  tworzy wszystkie obiekty okna podglądowego
- def retranslateUi (self, tablePreview)

  dodaje widoczne nazwy obiektom w oknie

3. Zakończenie

Licencje:

1. PyQT5

   GPL v3: GNU General Public License version 3 | Open Source Initiative

2. Psutils

   BSD 3-Clause License: The 3-Clause BSD License | Open Source Initiative

3. PyMySQL

   The MIT License: The MIT License | Open Source Initiative

4. Pandas

   BSD 3-Clause License: The 3-Clause BSD License | Open Source Initiative

5. SQLAlchemy

   The MIT License: The MIT License | Open Source Initiative

6. RSA

   The Apache License, Version 2.0: Apache License, Version 2.0

7. Crypto Tools

   The Apache License, Version 2.0: Apache License, Version 2.0

   BSD 3-Clause License: The 3-Clause BSD License | Open Source Initiative

8. XLWT

   BSD License (BSD): The 3-Clause BSD License | Open Source Initiative

9. LXML

   BSD License (BSD): The 3-Clause BSD License | Open Source Initiative

10. Psycopg2

    GNU Lesser General Public License v3 (LGPLv3) (GNU Lesser General Public License v3 (LGPLv3)): GNU Lesser General Public License version 3 | Open Source Initiative

11. OpenPyXL

    MIT License (MIT): The MIT License | Open Source Initiative

12. Xlrd

    BSD License (BSD): The 3-Clause BSD License | Open Source Initiative