**Podłączenie do bazy danych:**

Konfiguracja ciągu połączenia do bazy danych

{

  "ConnectionStrings": {

    "DefaultConnection": "Server=localhost;Database=elearning;User Id=ed;Password=YourPassword;"

  }

}

Odczytanie ciągu połączenie do bazy danych z pliku appsettings.json, a następnie użycie go dla utworzenia obiektu klasy Orm:

IConfigurationBuilder builder =

    new ConfigurationBuilder().AddJsonFile("appsettings.json", optional

                                           : false, reloadOnChange

                                           : true);

IConfigurationRoot root = builder.Build();

string connectionString =

    root.GetSection("ConnectionStrings")["DefaultConnection"];

Orm orm = new ("PostgreSql", connectionString);

**Pobieranie danych**

**1. GetAll**<T>() – pobranie wszystkich danych z danej nazwy klasy

**Parametry:**

* <T> - generyczny parametr dotyczący nazwy klas

**Zwracane typy:**

* List<Dictionary<string, object>> - lista słowników, w której każdy słownik posiada klucz typu string oraz wartość typu object

**Przykład użycia:**

var getBooks = orm.GetAll<Book>();

foreach (var row in getBooks)

{

    foreach (var item in row)

    {

        Console.WriteLine($"{item.Key}: {item.Value}");

    }

}

Obraz zawierający tekst, zrzut ekranu, Czcionka, czarne i białe

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

**2. GetAll<T>(params string[] fields)** – pobranie wszystkich danych wraz z możliwością wyboru nazw atrybutów do wyświetlenia z danej nazwy klasy

**Parametry:**

* <T> - Generyczny parametr dotyczący nazwy klas
* params string[] fields – tablica nazw atrybutów, które mają zostać wyświetlone

**Zwracane typy:**

* List<Dictionary<string, object>> - lista słowników, w której każdy słownik posiada klucz typu string oraz wartość typu object

**Przykład użycia:**

var getBooks = orm.GetAll<Book>(„Date”, „Publisher”);

foreach (var row in getBooks)

{

    foreach (var item in row)

    {

        Console.WriteLine($”{item.Key}: {item.Value}”);

    }

}

**3. GetById<T>(int id)** – pobranie danych po danych identyfikatorze z danej nazwy klasy

**Parametry:**

* <T> - generyczny parametr dotyczący nazwy klas
* Int id – liczba całkowita dotycząca identyfikatora

**Zwracane typy:**

* List<Dictionary<string, object>> - lista słowników, w której każdy słownik posiada klucz typu string oraz wartość typu object

**Przykład użycia:**

int bookId = 2;

var secondBook = orm.GetById<Book>(bookId);

foreach (var row in secondBook)

{

    foreach (var item in row)

    {

        Console.WriteLine($”{item.Key}: {item.Value}”);

    }

}

**4. GetById<T>(int id, params string[] fields) –** pobranie danych po danych identyfikatorze wraz z możliwością wyboru nazw atrybutów do wyświetlenia z danej nazwy klasy

**Parametry:**

* <T> - generyczny parametr dotyczący nazwy klas
* Int id – liczba całkowita dotycząca identyfikatora
* params string[] fields – tablica nazw atrybutów, które mają zostać wyświetlone

**Zwracane typy:**

* List<Dictionary<string, object>> - lista słowników, w której każdy słownik posiada klucz typu string oraz wartość typu object

**Przykład użycia:**

var thirdBookWithChosenFields = orm.GetById<Book>(3, "Date", "Publisher");

foreach (var row in thirdBookWithChosenFields)

{

    foreach (var item in row)

    {

        Console.WriteLine($"{item.Key}: {item.Value}");

    }

}

**5. GetAllPagination<T>()** – pobranie określonej liczby zwróconych rekordów z możliwością pominięcia początkowych wyników z danej nazwy klasy

**Parametry:**

* <T> - generyczny parametr dotyczący nazwy klas
* Int limit – liczba całkowita dotycząca liczby zwróconych rekordów
* Int skip – liczba całkowita dotycząca liczby rekordów, które mają zostać pominięte podczas pobierania danych (domyślnie jest równe 0)

**Zwracane typy:**

* List<Dictionary<string, object>> - lista słowników, w której każdy słownik posiada klucz typu string oraz wartość typu object

**Przykład użycia:**

int limit = 6;

int skip = 0;

var firstSixBooks = orm.GetAllPagination<Book>(limit, skip);

foreach (var row in firstSixBooks)

{

    foreach (var item in row)

    {

        Console.WriteLine($"{item.Key}: {item.Value}");

    }

}

**Manipulowanie danymi**

**1. InsertData<T>(Dictionary<string, object> newData) -** dodanie jednego rekordu z danej nazwy klasy

**Parametry:**

* <T> - generyczny parametr dotyczący nazwy klas
* **Dictionary<string, object> newData** słownik posiada klucz typu string będącym nazwą atrybutu oraz wartość typu object jako wartość dla tego atrybuty

**Zwracane typy:**

* **void**

**Przykład użycia:**

var newBook = new Dictionary<string, object>

{

    { "id", 470 },

    { "Date", "12/05/1904 00:00:00" },

    { "Publisher", "2 Kroki i 4 pszczoly" }

};

orm.InsertData<Book>(newBook);

**2. InsertMany<T>(List<Dictionary<string, object>> manyData) –** dodanie listy nowych danych z danej nazwy klasy

**Parametry:**

* <T> - generyczny parametr dotyczący nazwy klas
* **List<Dictionary<string, object>> manyData** - lista słowników, w której każdy posiada klucz typu string będącym nazwą atrybutu oraz wartość typu object jako wartość dla tego atrybuty

**Zwracane typy:**

* **void**

**Przykład użycia:**

var newBooks = new List<Dictionary<string, object>>{

    new Dictionary<string, object>{{"Id", 800},

                                   {"Date", "12/05/1904 00:00:00"},

                                   {"Publisher", "20 krokow i mewa"}},

    new Dictionary<string, object>{{"Id", 801},

                                   {"Date", "15/05/1904 00:00:00"},

                                   {"Publisher", "24 krokow i mewa"}},

    new Dictionary<string, object>{{"Id", 802},

                                   {"Date", "18/05/1904 00:00:00"},

                                   {"Publisher", "42 krokow i mewa"}}};

orm.InsertMany<Book>(newBooks);

**3. DeleteById<T>(int id)** – usunięcie danych po jego identyfikatorze id z danej nazwy klasy

**Parametry:**

* <T> - generyczny parametr dotyczący nazwy klas
* Int id – liczba całkowita dotycząca identyfikatora

**Zwracane typy:**

* void

**Przykład użycia:**

int idToDelete = 340;

orm.DeleteById<Book>(idToDelete);