

User

```
protected int UserId;  
protected int UserPermId;  
protected bool isLoggedIn;  
protected Fields UserLogin;  
protected Fields UserPass;
```

```
public User();
```

Ustawia standardowe parametry obiektu User na niezalogowanego

```
public User(int GettingUserId, Fields GettingUserLogin, GettingUserPass);
```

Wywołuje funkcję, aby ustawić obiekt z podanymi argumentami

```
protected void UserInit(int GettingUserId, Fields GettingUserLogin, Fields GettingUserPass);
```

Funkcja wywoływana przez konstruktor do ustawienia właściwości klasy na te, podane w argumentach

```
public override string ToString();
```

Zwraca opis użytkownika w formacie: `UserId UserLogin UserPermId isLoggedIn`

```
public int returnUserId() {return UserId;}  
public Fields returnUserLogin(){return UserLogin;}  
public Fields returnUserPass(){return UserPass;}  
public int returnUserPermId(){return UserPermId;}  
public bool returnIsLoggedIn(){return isLoggedIn;}  
public bool setValueOfField ( Fields field, string value );  
public string returnNameOfField ( Fields field );  
public string returnNameOfField ( Fields field );  
public void setUserLoginStatusTrue();  
public void setUserPermId ( int PermId );
```

```
public bool setValueOfField ( Fields field, string value );
```

Zmienia wartość pola, które zostało podane w argumencie

```
public string returnNameOfField ( Fields field );
```

Zwraca nazwę pola, które zostało podane w argumencie

```
public string returnNameOfField ( Fields field );
```

Zwraca wartość pola, które zostało podane w argumencie

```
public void setUserLoginStatusTrue();
```

Zmienia status zalogowania użytkownika

```
public void setUserPermId ( int PermId );
```

Zmienia parametr uprawnień użytkownika

Admin : User

```
public Admin( User user );
```

Wywołuje funkcję `UserInit()` z klasy użytkownik i ustawia uprawnienia (`UserPermId`) użytkownika na 1 (standardowo Admin)

Fields

```
protected string name;
```

```
protected string databaseName;
```

```
public Fields();
```

Ustala właściwości na null

```
public Fields( string Gname, string GdbName );
```

Ustala właściwości na argumenty funkcji

```
public string returnName();
```

Zwraca `name`

```
public string returnDatabaseName();
```

Zwraca `databaseName`

```
virtual public bool validation ( string gValue );
```

Sprawdza walidację, którą wykorzystuje `setValue()` Standardowo zawsze zwraca true.

```
virtual public bool setValue ( string gValue, int UserId );
```

UserLoginField : Fields

```
public Fields();
```

Ustala właściwości

```
override public bool validation ( string gValue );
```

Sprawdza walidację, którą wykorzystuje `setValue()`. W tym przypadku nadpisujemy standardowe true na sprawdzenie czy `gValue` nie jest zajęty oraz czy jego długość mieści się w przedziale (3,15)

```
public bool isBusy ( string gValue );
```

Sprawdza czy `gValue` nie jest zajęty

UserPassField : Fields

```
public Fields();
```

Ustala właściwości

```
override public bool validation ( string gValue );
```

Sprawdza walidację, którą wykorzystuje `setValue()`. W tym przypadku nadpisujemy standardowe true na sprawdzenie czy długość `gValue` mieści się w przedziale (3,15)

```
virtual public bool setValue ( string gValue, int UserId );
```

Nadpisujemy główne `setValue()` aby móc zakodować hasło używając `PasswordHasher`

PasswordHasher

```
public bool comparePassword( string passNormal, string passHashed );
```

Porównywanie haseł

```
public string HashPassword ( string Password );
```

Zwraca zakodowane hasło

Login

```
public User LoginInit( string UserLogin, string UserPass );
```

Na podstawie loginu pobiera dane z bazy danych, a następnie przy pomocy klasy `Perm` zwraca obiekt klasy `User`.

Przykładowe użycie:

```
Login login = new Login();
```

```
User u = login.LoginInit( „login”, „hasło” );
```

Logout

```
public User LogoutInit( User user );
```

Zmienia status użytkownika na niezalogowany i zwraca obiekt klasy `User`.

Przykładowe użycie:

```
Logout logout = new Logout();
```

```
User u = logout.LogoutInit( „login”, „hasło” );
```

Register

```
private int UserLogin;
```

```
private bool UserPassword;
```

```
public Register ( string Login, string Password );
```

```
public bool checkDatas();
```

Sprawdza czy długość `UserLogin` jest w zakresie (3,15)

```
public bool RegisterInit();
```

Odwołuje się do klasy `PasswordHasher` a następnie wysyła informacje do bazy danych

```
public bool isBusy();
```

Sprawdza czy `UserLogin` nie jest zajęty

Przykładowe użycie:

```
Register register = new Register („login”, „hasło”);
```

```
register.RegisterInit();
```

Perm

```
static class Perms
```

Lista zdefiniowanych uprawnień

```
public User ReturnObjectWithPerm( User user );
```

Na podstawie zdefiniowanych uprawnień zwraca obiekt klasy `User` bądź jego potomka (np. `Admin`)

Database

```
private static Database instance;
```

```
private MySqlConnection conn;
```

```
private string conn;
```

```
static class DatabaseConfig
```

Dane bazy danych

```
private Database();
```

```
public static Database Instance;
```

```
private void Initialize();
```

Ustawienie danych bazy danych

```
private bool OpenConnection();
```

```
private bool CloseConnection();
```

```
public bool NonReturnQuery ( string query );
```

Wywołanie zapytania mysql typu Insert / Update / Delete

```
public int CountWhere ( string query );
```

Zwraca wystąpienie elementów dopasowanych do zapytania

```
public List < string > [] Select ( string query, string [] rows );
```

Zwraca wartości wywołania zapytania

DatabaseController : Database

```
protected Database db;
```

```
private void Initialize();
```

UserDatabaseController : DatabaseController

```
public UserDatabaseController();
```

```
public bool checkPassCompare ( string UserLogin, string UserPass );
```

Porównuje hasła

```
public int ReturnUserPermid ( int UserId );
```

Zwraca uprawnienia użytkownika po UserId

```
public int ReturnUserId ( string UserLogin );
```

Zwraca UserId po UserLogin

```
public string ReturnFieldValue ( int UserId, Fields field );
```

Zwraca wartość dla pola field

```
public bool SetFieldValue ( int UserId, Fields field, string value );
```

Zmienia wartość pola field

```
public void Insert ( string query );
```

Wykonuje zapytanie typu insert