

# Projekt zaliczeniowy



Programowanie obiektowe

Rok akademicki 2024/2025

## Autorzy:

Kacper Kwak

Adrian Marek

Paweł Modzelewski

Marek Geisler

## SYSTEM TWORZENIA ZAMÓWIEŃ W RESTAURACJI

System tworzenia zamówień w restauracji to aplikacja desktopowa, która umożliwia zarządzanie klientami, zamówieniami oraz pracownikami restauracji. Projekt składa się z części odpowiedzialnej za działanie aplikacji napisanej w języku C# oraz interfejsu graficznego opartego na technologii WPF.

### Podział ról:

**Kacper Kwak** – funkcje dodawania nowych obiektów, własne exceptions

**Adrian Marek** – stworzenie i zaprojektowanie okien GUI (DodajPracownika.xaml, DodajKonto.xaml, DodajKlienta.xaml, DodajDanie.xaml, admin.xaml, OknoKelner.xaml, panel.xaml), wykonanie dokumentacji technicznej.

**Paweł Modzelewski** – Stworzenie klas ich konstruktorów i właściwości, diagram UML między klasami, kod obsługi GUI

**Marek Geisler** – Serializacja do Xml, Testy Jednostkowe, napisanie instrukcji obsługi, Stworzenie przykładowych danych

### Struktura projektu

Projekt składa się z trzech głównych części:

- **Działanie aplikacji:** katalog ProjektProgOb
- **Interfejs graficzny:** katalog RestaurantGUI
- **Testy Jednostkowe:** katalog TestRestauracja

## Opis funkcjonalności klas:

### Klasa Danie

Klasa **Danie** reprezentuje pozycję w menu restauracji, zawierając informacje o nazwie, kategorii, składnikach i cenie. Jest niezbędna do zarządzania ofertą restauracji i umożliwia przechowywanie danych o posiłkach. Modyfikatory dostępu pól są ustawione jako *private*, aby zapewnić enkapsulację danych i kontrolowany dostęp poprzez konstruktor lub metody klasy.

### Klasa Osoba

**Osoba** to klasa abstrakcyjna, która definiuje podstawowe dane każdej osoby w systemie, takie jak imię, nazwisko, e-mail i numer telefonu. Modyfikatory dostępu pól są prywatne, aby chronić dane przed nieautoryzowanym dostępem. Konstruktor wymusza podanie wszystkich danych, sprawdzenie adresu e-mail i numeru telefonu zabezpiecza przed błędnym formatem danych. Klasa została oznaczona atrybutem [DataContract], co umożliwia jej serializację.

### Klasa Klient

Klasa **Klient** dziedziczy po **Osoba** i przechowuje informacje o kliencie oraz jego historii zamówień. Jest oznaczona atrybutami [DataContract] i [DataMember] dla serializacji, co umożliwia zapis i odczyt danych. Lista zamówień jest prywatna, aby zapewnić bezpieczeństwo danych, a konstruktor inicjalizuje podstawowe dane klienta.

### Klasa Pracownik

**Pracownik** dziedziczy po **Osoba** i dodaje specyficzne pola, takie jak pozycja, numer PESEL oraz status kucharza i aktywności. Sprawdzenie numeru PESEL zapewnia poprawność danych. Konstruktor wymaga podania wszystkich danych, a domyślnie ustawia pracownika jako aktywnego.

### Klasa Konto

Klasa **Konto** reprezentuje system zarządzania użytkownikami, umożliwiając przechowywanie informacji o logowaniu, hasle oraz poziomie uprawnień (klient, admin, pracownik). Kluczowym celem jej stworzenia jest zarządzanie dostępem do systemu restauracyjnego oraz przypisanie kont do odpowiednich osób (**Osoba**). Publiczne właściwości umożliwiają kontrolowany dostęp. Klasa oferuje dwa konstruktory umożliwiające tworzenie kont zarówno dla klientów, jak i pracowników.

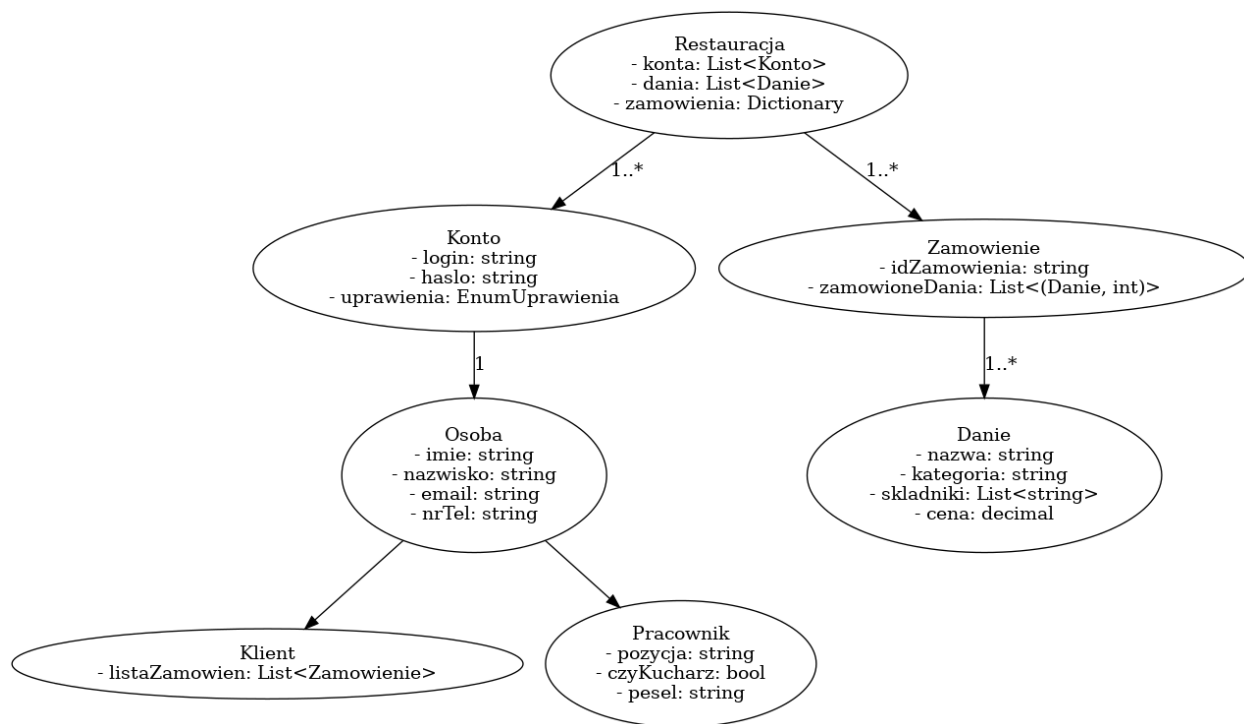
## Klasa Restauracja

Klasa **Restauracja** zarządza kluczowymi elementami restauracji, takimi jak lista zamówień, kont, pracowników oraz kategorii dań. Jest niezbędna do przechowywania danych i operacji związanych z restauracją. Prywatne pola zapewniają bezpieczeństwo danych, a publiczne właściwości umożliwiają dostęp w kontrolowany sposób. Klasa ma metody do dodawania kont klientów i pracowników, pobierania listy klientów oraz zapisu danych do pliku XML.

## Klasa Zamowienie

Klasa **Zamowienie** przechowuje informacje o zamówieniu, w tym jego unikalny identyfikator, listę zamówionych dań z ilościami, datę złożenia oraz opłaty. Konstruktor automatycznie generuje unikalny identyfikator na podstawie daty i wewnętrznego indeksu, co zapewnia unikalność zamówień. Klasa jest kluczowa dla realizacji procesu składania zamówień. Modyfikatory dostępu pól są prywatne, co chroni dane przed nieautoryzowanymi zmianami.

## Diagram klas:



## Opis testów jednostkowych:

W projekcie zawarte jest 5 testów sprawdzających działanie programu:

- TestPeselException() - Sprawdza czy poprawnie filtrowany jest pesel
- TestEmailException() //sprawdza czy poprawnie filtrowany jest adres email
- TestTelefonException() //sprawdza czy poprawnie filtrowany jest nr telefonu
- TestPowtorkaEmail() Sprawdza czy poprawnie odrzucane sa powtorzone adresy email
- TestSerializacjaDoXml() Sprawdza tworzenie pliku xml i czy zawartość się zgadza

## Instrukcja Obsługi:

Program posiada graficzny interfejs. Użytkownik ma do wyboru listy dań, pracowników, klientów i wszystkich kont. Do każdej z tych list można dodać nowe pozycje, usuwać istniejące lub je edytować.

By wprowadzone dane były zaakceptowane muszą mieć następujące formaty:

- Osoba (Pracownik i Klient) musi mieć prawidłowy format adresu email – [nazwa]@[domena].[rozszerzenie domeny], np. [agata@agh.pl](mailto:agata@agh.pl)
- Osoba musi mieć numer telefonu składający się z 9 cyfr
- Pracownik musi mieć numer pesel posiadający 11 cyfr
- Email dodawanego konta nie może już istnieć na liście

## Funkcje zawarte w programie:

- DodajKontoKlienta – używa się jej do dodania nowych klientów do restauracji
- DodajKontoPracownika - używa się jej do dodania nowych pracowników do restauracji
- PobierzListeKlientow – filtruje listę kont by wyświetlić tylko klientów
- UsunPracownika – usuwa pracownika z listy po peselu
- UsunKlienta – usuwa z listy klienta po emailu
- EdytujPracownika – Zmienia dane pracownika na liście
- EdytujKlienta – Zmienia dane klienta na liście
- ZapiszXML – Serializuje dane o restauracji do pliku formatu xml