



# **—Wstęp teoretyczny—**

Nasz program został stworzony z myślą o kompleksowej obsłudze agencji mieszkaniowej, a nasze cele skupiają się na analizie oraz praktycznej implementacji kluczowych operacji związanych z zarządzaniem danymi. W ramach tego projektu koncentrujemy się na przeglądaniu list klientów, sprzedawców, mieszkań, oraz zarządzaniu zamówieniami.

Wykorzystaliśmy techniki programowania obiektowego, takie jak dziedziczenie, klasy wewnętrzne i abstrakcyjne, aby modelować rzeczywiste struktury danych związane z agencją mieszkaniową. W rezultacie nasza aplikacja oferuje klarowną reprezentację poszczególnych klas, takich jak klienci, sprzedawcy, mieszkania czy zamówienia, co znacznie ułatwia operacje na danych, sprawiając, że stają się intuicyjne.

Główne funkcje naszego programu obejmują:

* Płynne przeglądanie informacji: Umożliwiamy łatwe i intuicyjne przeglądanie danych dotyczących klientów, sprzedawców , mieszkań i zamówień.
* Efektywne zarządzanie zamówieniami: Zapewniamy operacje dodawania, usuwania zamówień oraz aktualizacji ich statusów, co przyczynia się do sprawnego zarządzania procesem.

Dodatkowo, nasza aplikacja opiera się na danych z plików:

* class Klient.txt: Lista klientów agencji mieszkaniowej.
* class Sprzedawca.txt: Lista sprzedawców agencji mieszkaniowej.
* class Mieszkanie.txt: Lista mieszkań oferowanych przez agencję mieszkaniową.
* class Zamowienie.txt: Lista zamówień (w trakcie realizacji oraz zrealizowanych).

To umożliwia elastyczne zarządzanie danymi i łatwą integrację z istniejącymi zbiorami informacji agencji mieszkaniowej.

Dodatkowo, nasza aplikacja została wzbogacona o funkcjonalność raportowania z działania programu. Aktywnie spisuje raporty, gromadząc istotne informacje na temat operacji wykonywanych w aplikacji. Te raporty nie tylko umożliwiają śledzenie bieżących działań, ale także stanowią wartościowe narzędzie do analizy danych oraz oceny efektywności agencji mieszkaniowej. Wszystkie te informacje są zapisywane w pliku log.txt.

# —**Opis implementacji z fragmentami kodu źródłowego**–

Nasz program został zorganizowany w kilka kluczowych klas, z których każda pełni określoną rolę w strukturze projektu:

**Klasy reprezentujące dane:**

Klient, Sprzedawca, Mieszkanie, oraz Zamowienie odpowiadają za reprezentację różnych rodzajów danych związanych z agencją mieszkaniową. Dane w każdej klasie są enkapsulowane za pomocą getterów i setterów, dzięki czemu nie zostaną one zmienione przez przypadek, lub niezgodnie z wymaganiami.

**Klasa zarządzająca bazą danych (BazaDanych):**

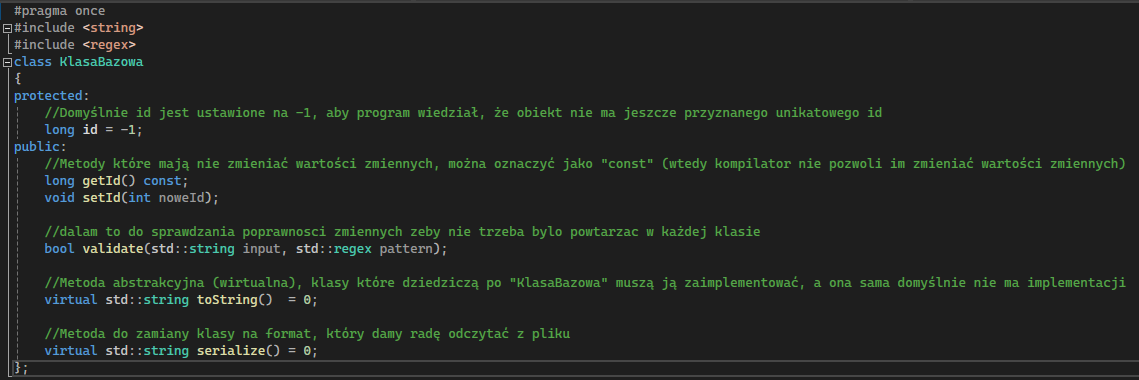
BazaDanych umożliwia efektywne zarządzanie danymi w programie, obsługując obiekty różnych typów. Klasy dziedziczące po KlasaBazowa muszą posiadać metody serialize i deserialize, co pozwala na skuteczny zapis i odczyt obiektów z pliku.

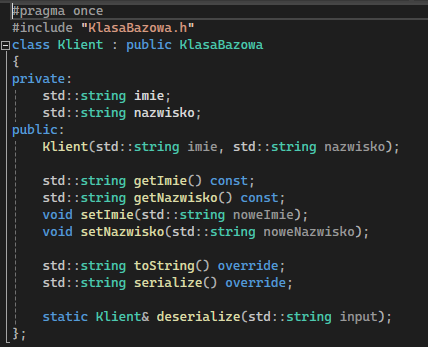
**KlasaBazowa:**

Jest klasą abstrakcyjną, nie pozwalającą na bezpośrednie tworzenie obiektów. Wymusza posiadanie numeru identyfikacyjnego (ID) oraz implementację metod:

* **toString** służy do czytelnego wypisania zawartości klasy,
* **serialize** przekształca dane klasy na string,
* **validate** pozwala na sprawdzenie poprawności przekazywanych danych.

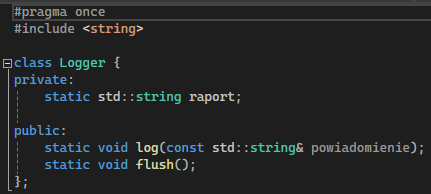
To sprawia, że klasy dziedziczące po *KlasaBazowa* są jednolite i spójne w strukturze.

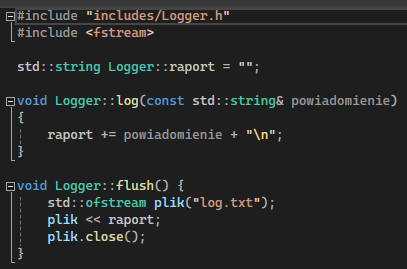




**Klasa obsługująca logowanie (Logger):**

Logger służy do tworzenia i zapisywania raportu z działania programu. Posiada statyczną metodę log, umożliwiającą łatwe dodawanie nowych wpisów do raportu w dowolnym miejscu projektu.





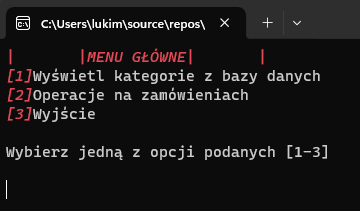
**Klasy spinające całość (Menu oraz ProjektS1):**

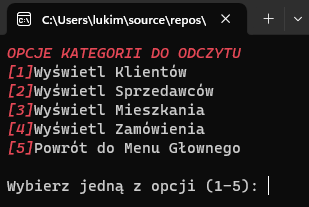
Menu i ProjektS1 odpowiadają za interakcję z użytkownikiem, umożliwiając przeglądanie zawartości bazy danych oraz zarządzanie zamówieniami za pomocą pętli i instrukcji switch-case.

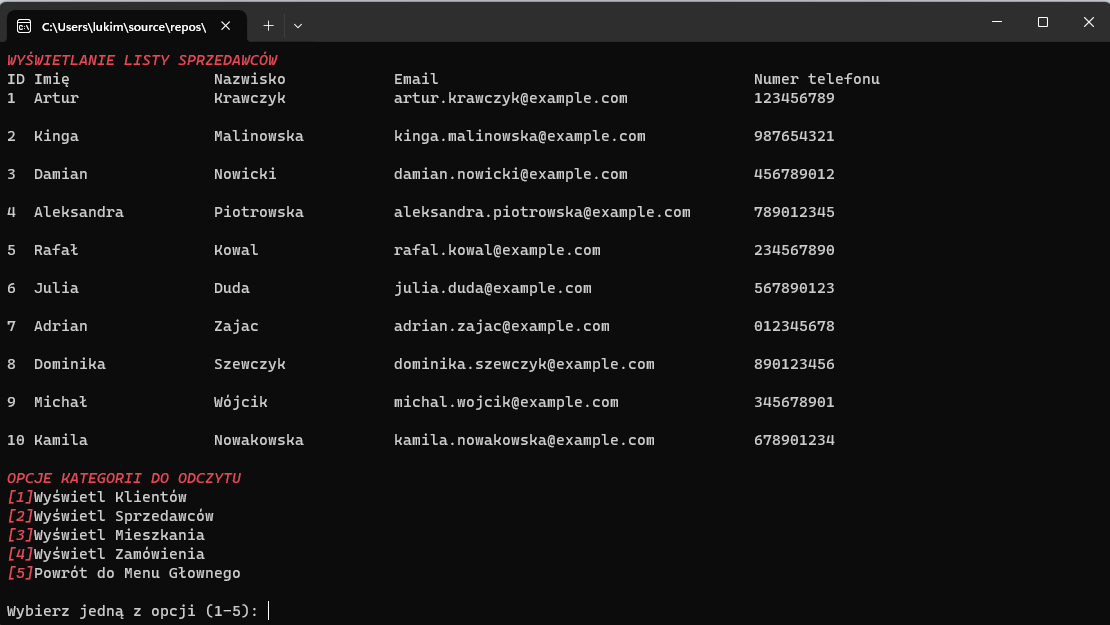
Dodatkowo, klasy "bazodanowe" dziedziczące po *KlasaBazowa* posiadają metodę **"deserialize",**  która umożliwia tworzenie nowych obiektów na podstawie stringa uzyskanego z metody **"serialize".**  To kluczowe dla *BazaDanych*, która używa tych metod do zapisu i odczytu obiektów z plików. Odczyt i zapis z plików są zapewniane przez klasę *BazaDanych*, opartą na szablonach, umożliwiającą zarządzanie obiektami dowolnego typu. Sposób zarządzania obiektami w *BazaDanych* pozwala na proste rozszerzanie funkcjonalności zapisu i odczytu o nowe klasy.

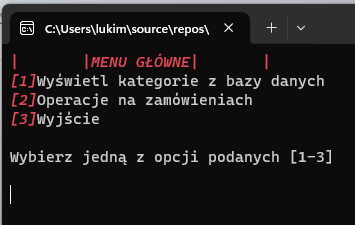
Wszystkie te elementy są spinane w funkcjonalną całość przez klasę Menu, która pozwala użytkownikowi na przeglądanie zawartości bazy danych oraz zarządzanie zamówieniami.

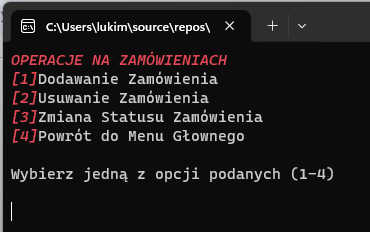
# **—Zrzuty ekranu z działania programu—**

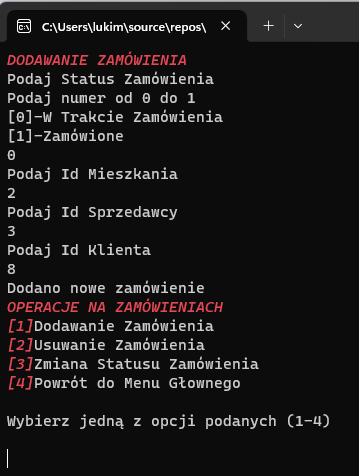


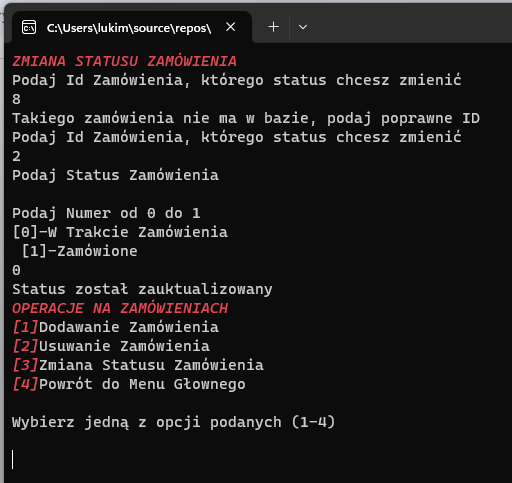


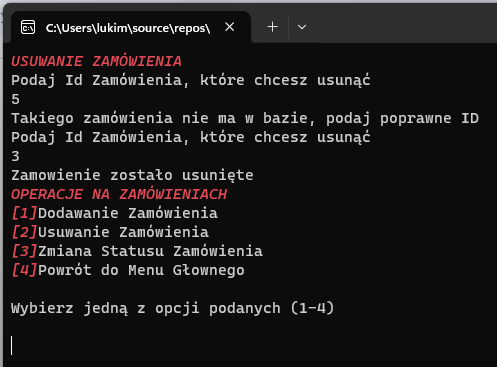




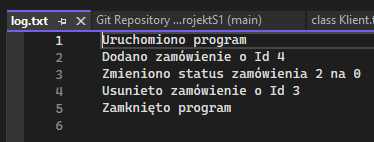




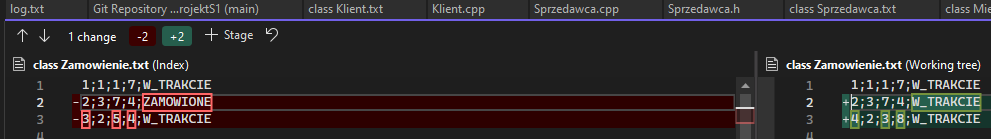




**Raport z działania:**



**Wynik zarządzania zamówieniami:**



# **—Wnioski—**

Program agencji mieszkaniowej jest rezultatem określonych wymagań, skupiających się na zaawansowanych technikach programowania obiektowego. Kod aplikacji obejmuje dziedziczenie, klasy wewnętrzne i abstrakcyjne.

Warto podkreślić, że nasza aplikacja jest przygotowana do dalszego rozwoju. Dzięki elastycznej architekturze możliwe jest łatwe rozszerzanie funkcjonalności o zarządzanie innymi rodzajami danych czy dodatkowe operacje. Aplikacja stanowi solidny fundament gotowy na ewolucję zgodnie z rozwojem agencji mieszkaniowej czy zmieniającymi się potrzebami użytkowników.

Tworząc program zarządzający mieszkaniami, zdobyliśmy praktyczne doświadczenie w wykorzystywaniu kluczowych założeń programowania obiektowego, takich jak dziedziczenie, klasy abstrakcyjne i szablony. To cenne doświadczenie, które przyczyniło się do naszego rozwoju jako programistów.