|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | A logo with black text  Description automatically generated | **AKADEMIA GÓRNICZO-HUTNICZA**  **im. Stanisława Staszica w Krakowie**  **WYDZIAŁ ZARZĄDZANIA** | wz_znak | |
| Programowanie Obiektowe |
|  |
|  |
| Dokumentacja Techniczna projektu zaliczeniowego  Temat projektu: System rezerwacji biletów firmy Łęcz-Trans  **Informatyka i ekonometria** |
|  |

Wykonały:

Natalia Trąba

Andżelika Wójcik

Urszula Zastawna

Kraków, 22.01.2024

Spis treści

[1. Zaimplementowane klasy 3](#_Toc156744891)

[1.1 Klasa Bilet 3](#_Toc156744892)

[1.2 Klasa Kurs 3](#_Toc156744893)

[1.3 Klasa ZarzadzanieBiletami 4](#_Toc156744894)

[1.4 Klasa ZakupBiletu 4](#_Toc156744895)

[1.5 Klasa Zwrot 5](#_Toc156744896)

[1.6 Klasa Potwierdzenie 6](#_Toc156744897)

[2. Okna 6](#_Toc156744898)

[2.1 Okno MainWindow 6](#_Toc156744899)

[2.2 Okno ZakupBiletu 7](#_Toc156744900)

[2.3 Okno Potwierdzenie 7](#_Toc156744901)

[3. Instrukcja działania 7](#_Toc156744902)

# Zaimplementowane klasy

## Klasa Bilet

Klasa Bilet stanowi reprezentację biletu w systemie rezerwacji biletów. Zawiera informacje dotyczące klienta, rodzaju biletu, daty zakupu oraz związanej trasy kursu.

Składowe klasy:

1. Prywatne pola

idStatic (typ: int): Statyczne pole przechowujące unikalne identyfikatory biletów (inicjalizowane na 2137).

imieNazwisko (typ: string): Imię i nazwisko posiadacza biletu.

email (typ: string): Adres e-mail posiadacza.

idBiletu (typ: int): Unikalny identyfikator biletu.

nrTelefonu (typ: string): Numer telefonu posiadacza.

rodzaj (typ: RodzajBiletu): Rodzaj biletu (normalny/ulgowy).

dataKupna (typ: DateTime): Data zakupu.

idKursu (typ: string): Identyfikator kursu związany z biletem.

cena (typ: decimal): Cena biletu.

2. Właściwości

IdBiletu, ImieNazwisko, Email, NrTelefonu, Rodzaj, DataKupna, Cena, IdKursu: Dostęp do pól z użyciem właściwości.

3. Konstruktor

Bilet(): Inicjalizuje obiekt i nadaje unikalny identyfikator biletu.

Enumeracja: RodzajBiletu

normalny, ulgowy: Wartości określające rodzaj biletu.

## Klasa Kurs

Klasa Kurs reprezentuje pojedynczy kurs w systemie rezerwacji biletów. Zawiera informacje o autobusie, dostępnych miejscach, trasie, dacie i godzinie kursu.

Składowe klasy:

1. Prywatne pola

autobus (typ: Autobus): Pole przechowujące informację o przypisanym autobusie.

dostepneMiejsca (typ: int): Ilość dostępnych miejsc na kursie.

trasa (typ: Trasa): Pole przechowujące informację o trasie kursu.

dataKursu (typ: DateTime): Data kursu.

godzina (typ: string): Godzina rozpoczęcia kursu.

idKursu (typ: string): Unikalny identyfikator kursu, generowany automatycznie na podstawie daty, godziny, i trasy.

2. Właściwości

Autobus, DataKursu, DostepneMiejsca, Trasa, Godzina: Dostęp do pól z użyciem właściwości.

IdKursu: Dostęp do pola idKursu z możliwością pobrania złożonej wartości na podstawie innych pól.

3. Enumeracje

Autobus: Określa dostępne autobusy.

Trasa: Określa dostępne trasy kursów.

## Klasa ZarzadzanieBiletami

Klasa **ZarzadzanieBiletami** odpowiada za zarządzanie biletami w systemie rezerwacji. Zawiera listy biletów i kursów, a także metody do zakupu, zwrotu biletu, generowania biletów w formie PDF oraz operacji zapisu i odczytu danych z pliku XML. Klasa korzysta z biblioteki QuestPDF do generowania plików PDF.

Składowe klasy:

1. Pola

listaBiletow (typ: List<Bilet>): Lista przechowująca obiekty biletów.

listaKursow (typ: List<Kurs>): Lista przechowująca obiekty kursów.

sciezkaBilety (typ: string): Ścieżka do pliku XML, w którym przechowywane są dane biletów.

2. Konstruktor

ZarzadzanieBiletami(): Konstruktor inicjalizujący listy biletów i kursów oraz wczytujący dane z pliku XML.

3. Metody

**InicjalizujKursy()** - Inicjalizuje listę kursów danymi początkowymi.

**PobierzDostepneKursy()** - Zwraca listę kursów, na których są dostępne miejsca.

**KupBilet(Bilet nowyBilet)** - Realizuje zakup biletu, zmniejszając dostępne miejsca na kursie.

**ZwrocBilet(int biletId)** - Realizuje zwrot biletu, zwiększając dostępne miejsca na odpowiadającym kursie.

**ZapiszBilety()** - Zapisuje listę biletów do pliku XML. Serializacja XML wymaga, aby klasa Bilet była oznaczona atrybutem [Serializable].

**WczytajBilety()**  - Wczytuje listę biletów z pliku XML.

**GenerujPdf(Bilet nowyBilet)** - Generuje plik PDF z danymi biletu.

## Klasa ZakupBiletu

Klasa ZakupBiletu odpowiada za interakcję z użytkownikiem w procesie zakupu biletu. Zawiera logikę obsługi zdarzeń przycisków, weryfikacji pól wejściowych, generowania nowego biletu oraz odświeżania dostępnych kursów w zależności od wybranych parametrów.

Składowe klasy:

1. Pola

wybranyKurs (typ: Kurs): Pole przechowujące aktualnie wybrany kurs podczas zakupu biletu.

listaKursow (typ: List<Kurs>): Lista przechowująca kursy dostępne do wyboru.

2. Konstruktor

ZakupBiletu(): Konstruktor inicjalizujący okno zakupu biletu, ustawiający źródła danych dla ComboBoxów i przypisujący obsługę zdarzeń.

3. Metody

**DalejBtn\_Click()** - Obsługuje kliknięcie przycisku "Dalej". Sprawdza poprawność wprowadzonych danych, tworzy nowy bilet i uruchamia okno potwierdzenia zakupu.

**SprawdzPola()** - Sprawdza, czy wszystkie pola są wypełnione przed kontynuacją procesu zakupu. W przypadku braku danych, wyświetla komunikat błędu.

**PobierzDostepneKursy()** - Wykorzystuje instancję klasy ZarzadzanieBiletami do pobrania dostępnych kursów.

**ObliczCeneBiletu()** - Oblicza cenę biletu na podstawie wybranego rodzaju biletu.

**DatePicker\_SelectedDateChanged()** - Obsługuje zmianę wybranej daty w kontrolce DatePicker, co powoduje odświeżenie dostępnych kursów.

**TrasaCb\_SelectionChanged()** - Obsługuje zmianę wybranej trasy w ComboBoxie, co powoduje odświeżenie dostępnych kursów.

**OdswiezKursy()** - Aktualizuje dostępne kursy w ComboBoxie na podstawie wybranej daty i trasy.

**GodzinaCb\_SelectionChanged()** - Obsługuje zmianę wybranego kursu w ComboBoxie, co pozwala na przechowanie aktualnie wybranego kursu do dalszych operacji.

## Klasa Zwrot

Klasa Zwrot odpowiada za obsługę okna zwrotu biletu w systemie rezerwacji. Umożliwia użytkownikowi wybór biletu do zwrotu, potwierdzenie operacji oraz nawigację do głównego okna.

Składowe klasy:

1. Pola

zarzadzanieBiletami (typ: ZarzadzanieBiletami): Pole przechowujące instancję klasy ZarzadzanieBiletami do zarządzania biletami.

2. Konstruktor

Zwrot(): Konstruktor inicjalizujący okno zwrotu biletu, ustawiający źródło danych dla ComboBoxa oraz przypisujący obsługę zdarzeń.

3. Metody

**PowrotBtn\_Click()**- Obsługuje kliknięcie przycisku "Powrót". Otwiera główne okno programu i zamyka okno zwrotu.

**ZwrotBtn\_Click() -** Obsługuje kliknięcie przycisku "Zwrot". Sprawdza, czy wybrano bilet do zwrotu, a następnie próbuje zrealizować zwrot biletu za pomocą klasy ZarzadzanieBiletami. Wyświetla komunikaty o sukcesie lub błędzie i aktualizuje widok.

**ZwrotCb\_SelectionChanged()** -Obsługuje zmianę wybranego biletu w ComboBoxie.

## Klasa Potwierdzenie

Klasa Potwierdzenie odpowiada za interfejs użytkownika w oknie potwierdzenia zakupu biletu. Pozwala użytkownikowi sprawdzić cenę zakupionego biletu, a następnie potwierdzić lub anulować zakup.

Składowe klasy:

1. Pola

nowyBilet (typ: Bilet): Pole przechowujące informacje o nowo utworzonym biletu.

2. Właściwość

CzyPotwierdzono (typ: bool): Właściwość dostępowa do informacji, czy użytkownik potwierdził zakup.

3. Konstruktor

Potwierdzenie(Bilet nowyBilet): Konstruktor inicjalizujący okno potwierdzenia zakupu biletu, przypisujący informacje o nowym bilecie oraz wywołujący metodę WyswietlCene().

4. Metody

**WyswietlCene()** - Wyświetla na interfejsie użytkownika informacje o rodzaju biletu i cenie.

**AnulujBtn\_Click()** - Obsługuje kliknięcie przycisku "Anuluj". Ustawia flagę CzyPotwierdzono na false i zamyka okno.

**PotwierdzamBtn\_Click()** - Obsługuje kliknięcie przycisku "Potwierdzam". Ustawia flagę CzyPotwierdzono na true, ustawia DialogResult na true (co pozwala na zwrócenie wartości true po zamknięciu okna) i zamyka okno.

# Okna

## Okno MainWindow

A screenshot of a computer

Description automatically generated

## Okno ZakupBiletu

A screenshot of a computer

Description automatically generated

## 2.3 Okno Potwierdzenie

A screenshot of a computer

Description automatically generated

# Instrukcja działania

1. Uruchomienie Programu:

Uruchom aplikację, klikając dwukrotnie na plik wykonywalny programu.

2. Główne Okno ("MainWindow"):

Główne okno programu zawiera trzy przyciski:

"Kup bilet": Przechodzi do okna zakupu biletu.

"Zwróć bilet": Przechodzi do okna zwrotu biletu.

"Regulamin": Przechodzi do okna z regulaminem.

3. Zakup Biletu ("ZakupBiletu"):

W oknie zakupu biletu użytkownik wybiera trasę, godzinę i inne dane.

Wypełnia formularz danymi osobowymi (imię, nazwisko, e-mail, telefon).

Wybiera rodzaj biletu (normalny lub ulgowy).

Potwierdza zakup, co powoduje generowanie biletu w formie pliku PDF.

Przykładowy szablon drukowanego biletu:

A screenshot of a computer

Description automatically generated

4. Zwrot Biletu ("Zwrot"):

W oknie zwrotu biletu użytkownik wybiera bilet do zwrotu z listy zakupionych biletów.

Potwierdza zwrot, co powoduje zwiększenie dostępnych miejsc na wybranym kursie.

5. Regulamin:

W oknie z regulaminem użytkownik może zapoznać się z zasadami korzystania z systemu rezerwacji.