

Wstępny projekt

Temat projektu: System w urzędzie pocztowym

Paulina Czapla SSI sem. 4, gr. 4, sekcja 2

Funkcjonalności oferowane przez program

W ramach projektu zrealizowany zostanie program okienkowy zaprojektowany w technologii Qt. Będzie on skierowany do użytku przez pracownika placówki pocztowej.

1. Nadanie przesyłki

Pracownik poczty będzie mógł wypełnić formularz danymi niezbędnymi do nadania przesyłki. Na podstawie informacji takich jak waga i rozmiar przesyłki wyznaczona zostanie cena przesyłki. Następnie, pracownik zostanie poproszony o wypełnienie formularza z danymi odbiorcy i nadawcy. Po wypełnieniu, będzie istniała możliwość wydrukowania etykiety.

Po zatwierdzeniu dane przesyłki pojawią się w listach mainDatabase i localDatabase, a następnie zapisane do plików.

2. Przegląd lokalnej bazy danych

Będzie możliwość przeglądania bazy przesyłek aktualnie znajdujących się w placówce pocztowej. Pojawią się one w tabeli. Będzie istniała możliwość wyszukiwania danych po numerze przesyłki.

Korzystający z programu dostanie możliwość zmiany statusów przesyłek. Dostępne statusy będą zależały od aktualnego statusu przesyłki. Dostępne będą:

- „w drodze”
- „oczekuje na odbiór w placówce” - jeśli ten status zostanie ustalony, to po 7 dniach bez zmiany wyskoczy okno z informacją o nieodebranej przesyłce. Status zostanie automatycznie zmieniony na „upłynął termin odbioru”.
- „oczekuje na wręczenie kurierowi”
- „wręczono kurierowi do doręczenia”
- „doręczono”
- „upłynął termin odbioru”
- „odmówiono odbioru”
- „odesłano do nadawcy”

Użytkownik będzie mógł również przenieść przesyłkę z lokalnej bazy danych do głównej bazy danych (tylko wtedy, gdy ma status „wręczono kurierowi do doręczenia”). Zniknie ona wtedy z lokalnej bazy danych.

3. Przegląd głównej bazy danych

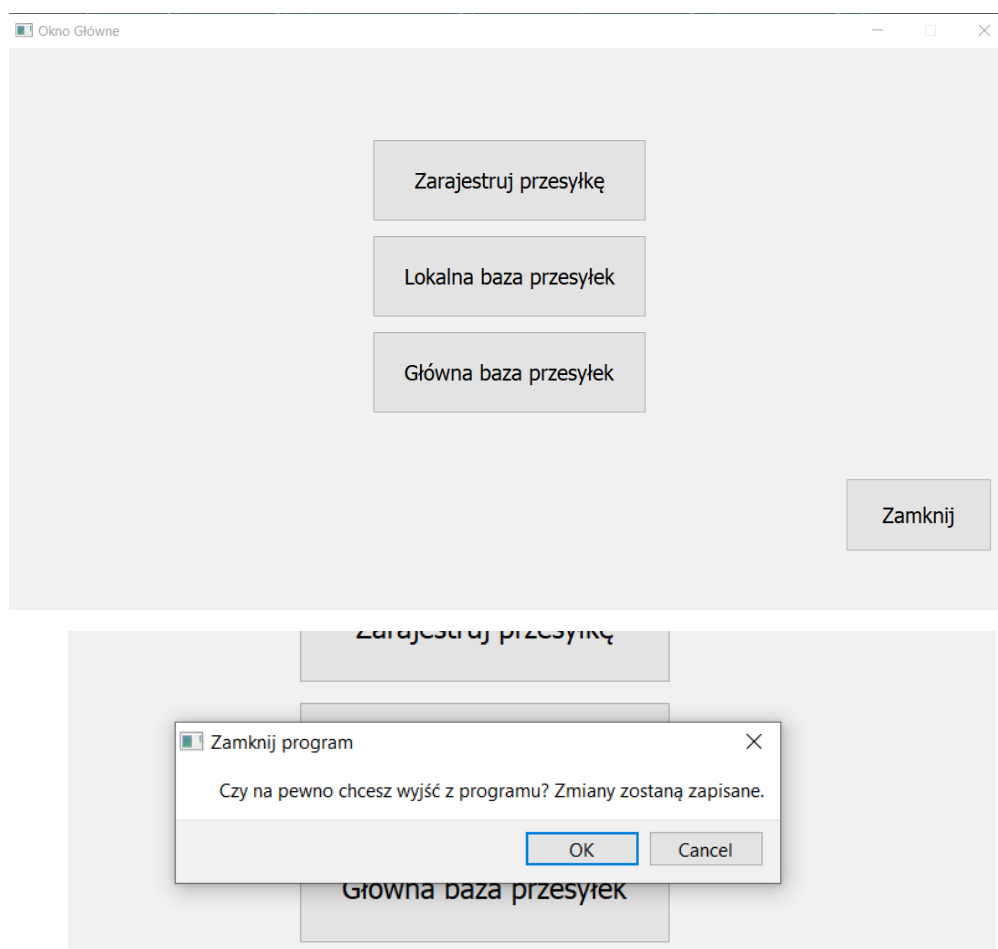
Użytkownik będzie miał możliwość przeglądania i wyszukiwania przesyłek w głównej bazie danych. Nie będzie mógł zmieniać ich statusów, ale będzie mógł przenieść przesyłkę do lokalnej bazy danych. Nie zniknie ona wtedy z głównej bazy.

Pliki wejściowe/wyjściowe

Program będzie przyjmował pliki wejściowe/wyjściowe

- plik maindatabase.txt, w którym będą przechowywane dane o wszystkich przesyłkach obsługiwanych aktualnie przez pocztę;
- plik localdatabase.txt, w którym będą się znajdowały dane o przesyłkach obsługiwanych przez daną placówkę pocztową (przesyłki znajdujące się w placówce, wydane do doręczenia kurierowi itd.)
- pliki z informacjami o kosztach wysyłki poszczególnych listów i paczek – parcelprices.txt i letterprices.txt. Na jego podstawie program wyznaczy cenę przy nadaniu przesyłki
- plik shipmentstatuses.txt ze statusami przesyłek oraz informacją o dostępnych statusach, na które można zmienić przy obecnym statusie. Każdy status ma swój identyfikator.

Interfejs programu



Okno Główne

Typ przesyłki

Polecony

Priorytet

Rozmiar

Waga

Do zapłaty:

TextLabel

Anuluj

Zatwierdź i przejdź dalej

Numer przesyłki: TextLabel

Dane nadawcy

Dane Odbiorcy

Imię i nazwisko

Nr telefonu

Kod pocztowy

-

Ulica i numer mieszkania

Miasto

Imię i nazwisko

Nr telefonu

Kod pocztowy

-

Ulica i numer mieszkania

Miasto

Drukuj etykietę

Wróć

Zatwierdź i zakończ

Okno Główne

Wyszukaj po

TextLabel

Przenieś do

Ustaw status

Wróć

Struktura dynamiczna

Projekt wykorzystuje własną klasę szablonową listy dwukierunkowej.

Wykorzystane techniki i zagadnienia laboratoryjne

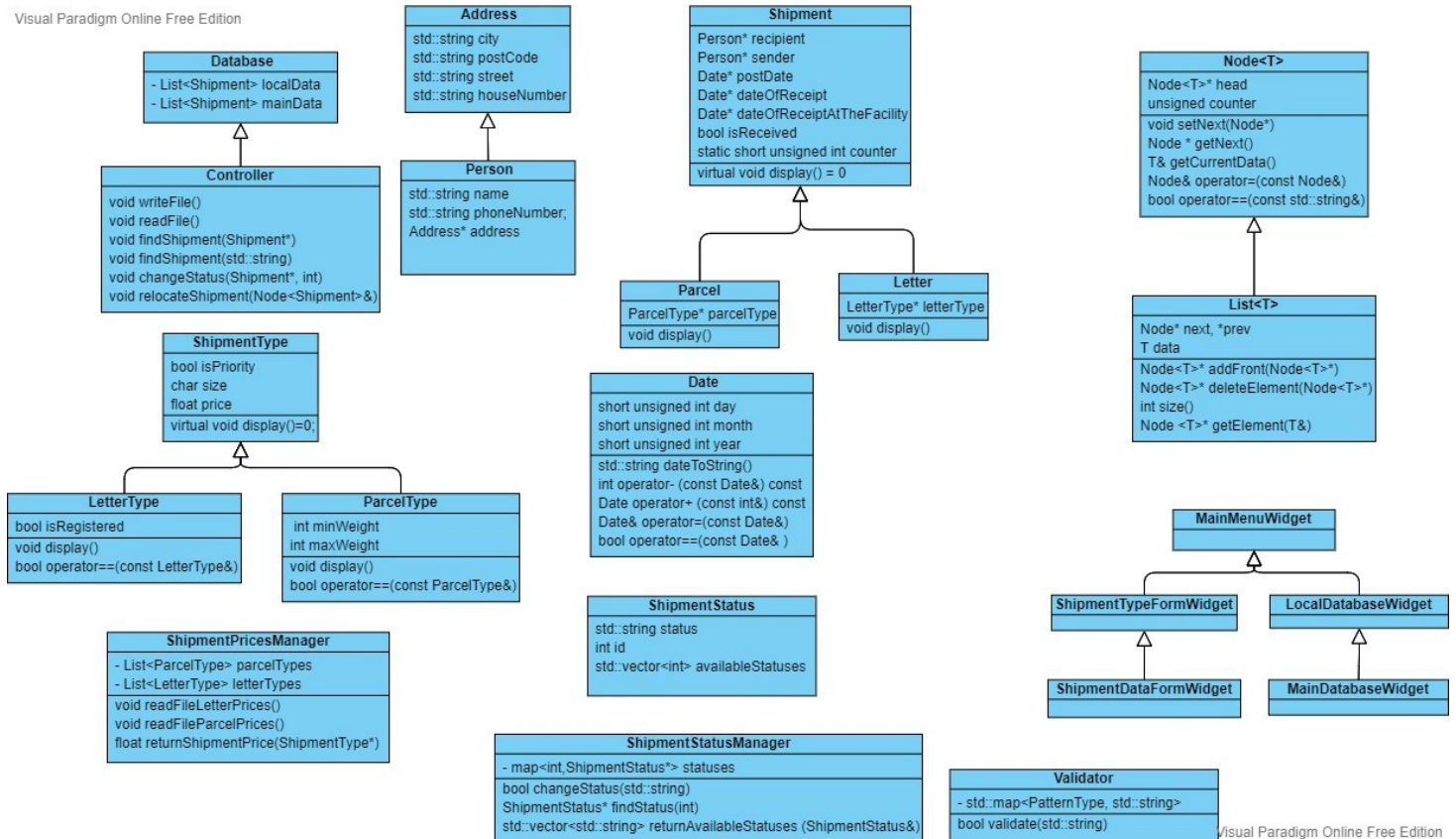
Wykorzystane zostaną techniki obiektowe takie jak:

- hermetyzacja;
- dziedziczenie;
- polimorfizm;
- operatory przeciążone.

Wykorzystane zostaną również zagadnienia poznane na laboratoriach:

- RTTI;
- wyrażenia regularne;
- szablony;
- algorytmy i iteratory STL;
- kontenery STL.

Diagram klas z atrybutami oraz wybranymi metodami



Opis wybranych klas

class Node<T> oraz class List<T>

Szablon węzła listy oraz szablon samej listy wykorzystane będą do tworzenia list przechowujących dane wysłanych przesyłek (pola w klasie Database) oraz do przechowywania informacji o typach paczek oraz listów (pola w klasie ShipmentPricesManager).

class Database

Klasa której polami są lokalna oraz główna baza danych. W jej konstruktorze wywołane zostaną metody do odczytywania plików z danymi i dodawania ich do dynamicznej struktury danych List<T>. W destruktorze zostaną wywołane metody do zapisu danych w plikach.

Po klasie Database będzie dziedziczyć klasa Controller, która będzie zarządzać rekordami w bazie.

class ShipmentType, class LetterType, class ParcelType

Po abstrakcyjnej klasie ShipmentType dziedziczą klasy LetterType oraz ParcelType. Obiekty tej klasy reprezentują typy przesyłek – listów lub paczek. Każdy typ charakteryzowany przez różne wartości – pewne są wspólne dla listów i paczek, inne unikalne. Wspólne wartości to cena, informacja czy paczka jest priorytetowa oraz wielkość.

class ShipmentPricesManager

Klasa ta przechowuje dwie listy. W jednej z nich znajdują się wszystkie dostępne typy paczek, a w drugiej wszystkie dostępne typy listów.

```
List<ParcelType> parcelTypes;  
List<LetterType> letterTypes;
```

W konstruktorze klasy wywołana zostanie metoda odczytująca informacje o typach przesyłek i ich cenach z plików. Tworzone będą odpowiednie obiekty LetterType i ParcelType oraz otrzymane dane wprowadzane będą do list. Inne metody klasy posłużą do wyznaczania ceny przesyłki na podstawie otrzymanych danych. Do wyznaczania ceny przesyłki o odpowiednich parametrach wykorzystane zostanie RTTI.

class ShipmentStatus, class ShipmentStatusManager

Obiekt klasy ShipmentStatus przechowuje informacje o konkretnym statusie przesyłki: nazwa statusu oraz id dostępnych do zmiany statusów w formie wektora. Klasa ShipmentStatusManager ma pole będące mapą, w której kluczem jest id typu przesyłki, a wartością wskaźnik na obiekt ShipmentStatus. Metody tej klasy posłużą do operacji związanych ze statusami przesyłek, wyświetlanie ich, zmianę statusu itd. Informacje o statusach pobierane są z pliku. W klasach ShipmentStatus i ShipmentStatusManager wykorzystane zostaną algorytmy STL i iteratory oraz kontenery map i vector.

class Validator

Polem klasy jest mapa, której kluczem jest typ wyliczeniowy PatternType, a wartością jest wzorzec, który zostanie wykorzystany do wyrażeń regularnych. Metoda klasy będzie walidować dane, zwracając false jeśli dane są niezgodne ze wzorcem lub true, jeśli są zgodne.

class MainMenuWidget oraz klasy dziedziczące po niej

Zaimplementowana zostanie klasa MainMenuWidget oraz klasy po niej dziedziczące widoczne na diagramie klas. Ich metody posłużą do obsługi stron stackedWidget oraz łączenia logiki programu z UI.