## Projekt\_1

Projekt przywiduje stworzenie kontenera na podstawie techniki szablonów C++ przy użyci interfejsu graficznego GDI pod systemem operacyjnym Windows.

## 1. Generalna struktura projektu.

Solution składa się przynajmniej z trzech projektów (modułów) Visual Studio (VS) – Rys. 1.



Rys. 1 Struktura rozwiązania

Moduł Project\_MFC zabezpiecza obsługę GDI w postaci okna graficznego, menu, paneli narzędzi (toolbars), wierszy tekstowych oraz wszystkich niezbędnych dialogów dla wprowadzenia, modyfikacji, usunięcia danych oraz wszystkich operacji wymienionych w Rozdziale 2. Ten projekt generuje plik z rozszerzeniem \*.exe.

Moduł MyData realizuje obsługę danych (serwer danych), nie posiada interfejsu GDI, jednak realizuje wszystkie operacji nad danymi. Generuje dynamicznie wiązaną bibliotekę \*dll, która eksportuje swoje klasy do wszystkich innych modułów.

Moduł Except1 realizuje obsługę błędów i komunikatów i eksportuje się do modułów MY\_DATA oraz Project\_MFC. Jest to MFC dll. W przypadku błędu fatalnego Except1 powinien skończyć wykonanie aplikacji. W przypadku komunikatu posyła komunikat windowsowy do CMainFrame modułu Project\_MFC, który powinien być przechwycony i opracowany. Jako skutek, ten komunikat powinien pojawić się w oknie Output.

Na Rys. 2 jest przedstawiony przykład aplikacji GDI dla obsługi tablicy dynamicznej przechowującej punkty na płaszczyźnie ze współrzędnymi x, y. Okno graficzne wyświetla te dane. Menu Operate zawiera dwie opcji (Rys. 3). Przy naciskaniu na Input Data otwiera się dialog (Rys. 4), który umożliwia dodawanie punktu, modyfikacje współrzędnych punktu oraz usunięcie punktu. Przy naciskaniu na Graph Wind powstaje dialog, sterujący opcjami wyświetlania danych na ekranie graficznym – promieniem punktu, jego kolorem, typem i wielkością ścinaki, trybem linii łączących punkty i t.d.

Uwaga! Na Rys. 2 – 4 są podane widoki dla prostego zadania w celu pierwotnego przedstawienia, jak ta aplikacja powinna wyglądać. Szczególna specyfikacja i wymagania będą przedstawione w trakcie zajęć projektowych.



Rys. 2 Aplikacja GDI, która realizuje obsługę tablicy dynamicznej przechowującej punkty na płaszczyźnie ze współrzędnymi x, y



Rys. 3 Opcje Operate.



Rys. 4 Dialog "Input Data"

## 2. Moduł danych

 W klasie MY\_DATA stworzyć tablicę dynamiczną typu MY\_POINT. Każdy obiekt MY\_POINT przedstawia punkt na płaszczyźnie dziedziczy po klasie MY\_COORD ze współrzędnymi double x, y. MY\_POINT zawiera nazwę punktu char \*name, jego numeru int numb oraz kolor dla wyświetlania graficznego COLORREF color. Klasa MY DATA eksportuje się do Project MFC.

Dopisać metodę:

- clear\_all usunięcie wszystkich elementów tablicy
- Zrealizować zapis-odczyt danych do/z pliku binarnego przy użyciu standardowych okien Windows "Open" and "Save As".
- Przeciążyć operator [] dla pobrania i przypisania elementu tablicy dat[ind] o indeksie ind
- 2. Stworzyć funkcje szablon Find:

```
template <class T, class Key>
T * find(const T *p_begin, const T *p_end, const Key &k);
```

Tu – p\_begin – wskaźnik do pierwszego elementu tablicy, z którego się zaczyna przeszukiwanie; p\_end – pierwszy element poza ostatnim elementem tablicy; Key k – klucz przeszukiwania (dla podanego przykładu – nazwę wierzchołku). Zwraca wskaźnik do odnalezionego elementu tablicy lub NULL (w przypadku niepowodzenia).

 Układ obsługi błędów, ostrzeżeń i komunikatów (MyExcept1) przedstawić oddzielną klasą. Wszystkie błędy, ostrzeżenia i komunikaty muszą być umieszczone w jednym pliku, a nie rozrzucane po kodu. Zakończenie awaryjne aplikacji powinno być "inteligentnym" – przed zakończeniem powinni być wywołane destruktory klas, pliki pozamykane.

- 4. Dynamiczną alokacje / zwolnienie pamięci dokonywać przez operatory new / delete
- 5. Opcje powinny zawierać:
  - Dodać obiekt.
  - Pobrać ostatni obiekt i wyświetlić jego składowe.
  - Usunąć obiekt.
  - Usunąć wszystko.
  - Wybrać dowolny obiekt i zmodyfikować jego
  - Znaleźć wszystkie obiekty (po numeru wierzchołka możliwe, że kilka obiektów mogą mieć ten samy numer).
  - Zapisać dane do pliku binarnego
  - Odczytać dane z pliku binarnego
  - Wyświetlić dane
  - Importować dane do pliku csv oraz wystartować EXCEL i otworzyć ten plik.
  - Skończyć aplikacje
- 6. Każdą klasę umieścić w oddzielne pliki \*.cpp; \*.h .
- 7. Przedstawić projekt w postaci elektronicznej jako archiwum projektu razem z plikiem danych.
- 8. Zadanie powinno być wykonane w środowisku IDE VS 2019 2022 pod systemem operacyjnym Windows 10 (11).