LABORATORIUM PROE.B, PROJEKT 3

WŁASNE WZORCE I KONTENERY. GUI.

ZADANIE

Celem projektu jest stworzenie aplikacji z graficznym interfejsem użytkownika, będącej encyklopedią obiektów stworzonych w ramach projektu 2. Program ma umożliwiać, m.in. dodawanie nowych obiektów, usuwanie istniejących, przeglądanie zasobów.

ZADANIA SZCZEGÓŁOWE

W projekcie proszę:

1. Zaimplementować listę jednokierunkową, acykliczną, gdzie każdy węzeł, poza obiektem i wskaźnikiem na kolejny węzeł, będzie zawierać czas utworzenia węzła liczony w milisekundach od rozpoczęcia działania programu. Lista powinna być zrealizowana w postaci szablonu. Funkcje realizowane przez listę to, m.in.: dostęp do dowolnego węzła przez operator indeksowania, dodanie i usunięcie węzła w dowolnej pozycji, kopiowanie listy (konstruktor kopiujący), operator przypisania, zwrócenie czasu w którym został stworzony dany węzeł.

2. Stworzyć aplikację pozwalającą na zarządzanie (dodawanie, usuwanie, przeglądanie) minimum trzema rodzajami obiektów (jeśli schemat dziedziczenia zawierał np. 5 obiektów, wybieramy z nich minimum 3).

3. Zaprojektować i wykonać graficzny interfejs użytkownika (w dowolnym środowisku, np. WinAPI, Qt, MFC itp.) pozwalający na zarządzanie obiektami stworzonymi w projekcie drugim. Każdy obiekt powinien zawierać obraz (np. zdjęcie), które można wyświetlić. Interfejs powinien być odporny na błędy użytkownika.

4. Wykorzystać mechanizmy dziedziczenia, m.in. wszystkie obiekty encyklopedii powinny znajdować się pod jednym kontenerem wskaźników na obiekty klasy bazowej. Wyświetlenie informacji o danym obiekcie w oknie powinno odbywać się poprzez metody wirtualne. Do przechowywania wskaźników należy wykorzystać listę stworzoną w punkcie 1.

5. Tam gdzie jest to wskazane wykorzystywać możliwości standardu C++11.

Proszę o przesyłanie projektów 2 dni przed terminem obrony (tj. środa do godz. 24 w tygodniu obrony dla grupy piątkowej) na adres mailowy prowadzącego zajęcia.

Dokumentacja tej części projektu **nie jest** wymagana.

KRYTERIA OCENY

przejrzystość kodu	2 p.
implementacja listy	4 p.
graficzny interfejs użytkownika	5 p.
pozostałe wymagania	4 p.