

# LABORATORIUM PROE 16Z, PROJEKT 2

DZIEDZICZENIE, ABSTRAKCJA DANYCH, KLASY SZABLONOWE, WYJĄTKI

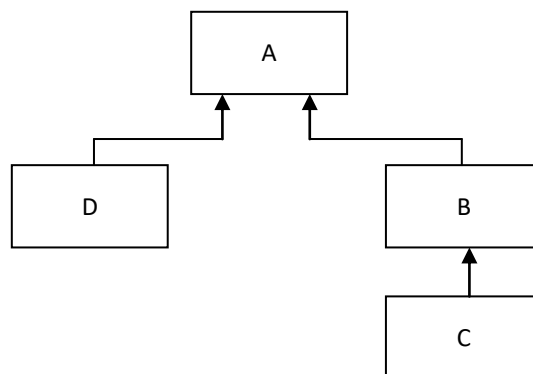
## ZADANIE

Zaprojektować aplikację zarządzania ..... (tu wpisz swój obiekt)

Celem projektu jest stworzenie aplikacji z konsolowym interfejsem użytkownika. Interfejs ten ma umożliwiać przetestowanie zaimplementowanych możliwości.

## ZADANIA SZCZEGÓŁOWE

1. Zaprojektować i zrealizować dla klasy głównej następujący schemat dziedziczenia:



Wybrany obiekt (wpisany w puste miejsce w *Zadaniu*) może być dowolnym obiektem z powyższych (A, B, C lub D). Klasa A powinna być abstrakcyjna. Proszę zaprojektować i zdefiniować wybrane, sensowne metody wirtualne dla klasy bazowej i pochodnych.

2. Zaimplementować operatory strumieniowe dla wszystkich utworzonych klas
3. Rygorystycznie potraktować deklaratory zakresu - private, protected i public stosując zasadę najmniejszych przywilejów.
4. Wszystkie klasy powinny zawierać metody zapamiętywania stanu **całego** obiektu w pliku i odtwarzania jego stanu z dysku. Zapis i odczyt powinien być wykonany kaskadowo/hierarchicznie tzn. tak, aby dana metoda danej klasy zajmowała się odczytem i zapisem jedynie "swoich" pól.

5. Wykorzystać właściwości polimorfizmu do zademonstrowania działania metod wirtualnych (należy użyć wskaźników na obiekty klasy podstawowej).
6. 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.
7. Napisać aplikację z konsolowym interfejsem. Ma ona umożliwiać zarządzanie (dodawanie, usuwanie, przeglądanie) obiektami w liście, stworzonej w ramach punktu 6. Program główny powinien umożliwić testowanie wybranych działań obiektów. Dostęp do nich powinien odbywać się za pomocą metod wirtualnych. Dla klas pochodnych powinny działać metody rzutu i odczytu stanu obiektu do/z pliku dyskowego z wykorzystaniem przeciążonych operatorów strumieniowych. Interfejs powinien być niewrażliwy na błędy użytkownika. Niewrażliwość interfejsu powinna zostać uzyskana przez wykorzystanie wyjątków.

#### UWAGA

Jeżeli jest wybór pomiędzy stosowaniem mechanizmów, funkcji, instrukcji typowych dla języka C i C++ proszę stosować odpowiednie konstrukcje właściwe dla C++ np. char\* - string, FILE\* - iostream, itp. Jeden obiekt – 2 pliki: obiekt.h, obiekt.cc (lub obiekt.cpp).

Schemat dziedziczenia (nawet narysowany na kartce) należy jak najszybciej skonsultować z prowadzącym laboratorium aby uniknąć sytuacji, w której przystąpią Państwo do implementacji z błędnymi założeniami.

Proszę przesyłać projekty 2 dni przed terminem obrony (tj. niedziela do godz. 24 w tygodniu obrony dla grupy wtorkowej) na adres mailowy prowadzącego zajęcia.

#### KRYTERIA OCENY

przejrzystość kodu	1 p.
implementacja dziedziczenia i pozostałe wymagania	5 p.
zapis/odczyt obiektów	2 p.
implementacja listy	4 p.
interfejs i obsługa wyjątków	3 p.