

# LABORATORIUM PROE.B, PROJEKT 2

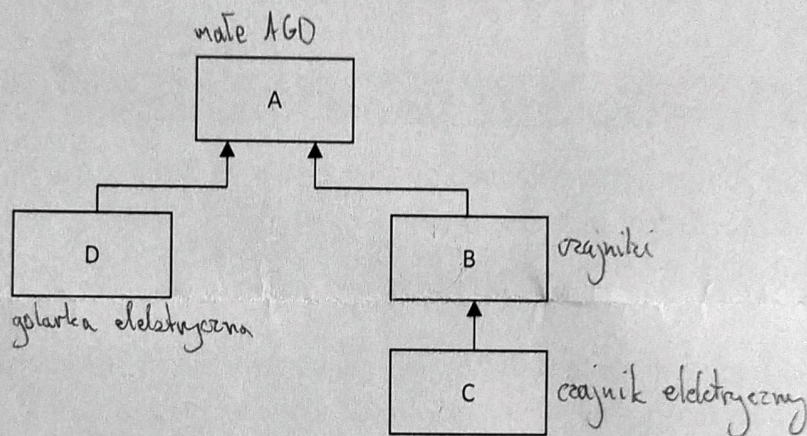
DZIEDZICZENIE, ABSTRAKCJA DANYCH, HERMETYZACJA, WEKTORY BIBLIOTEKI STL

## ZADANIE

Celem projektu jest rozszerzenie możliwości obiektu stworzonego w pierwszym projekcie z wykorzystaniem dziedziczenia.

## ZADANIA SZCZEGÓŁOWE

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



Obiekt z projektu 1 może być umieszczony w dowolnym miejscu hierarchii (jako A, B, C lub D). Klasa A powinna być abstrakcyjna. Należy zaprojektować i zdefiniować wybrane, sensowne metody wirtualne dla klasy bazowej i pochodnych.

2. Zaimplementować operatory strumieniowe dla wszystkich utworzonych klas. Działanie tych operatorów powinno być hierarchiczne, tzn. operator definiowany dla klasy potomnej powinien wykorzystywać operator dla klasy bazowej.
3. Rygorystycznie potraktować deklaratory zakresu – private, protected i public, stosując zasadę najmniejszych przywilejów.
4. Zastąpić wszystkie tablice obiektów, odpowiednimi kontenerami STL, np. vector, queue, list itp.
5. Wszystkie klasy powinny zawierać metody zapisywania stanu **całego** obiektu do pliku i odtwarzania jego stanu z pliku. Zapis i odczyt powinien być wykonany kaskadowo/hierarchicznie tzn. tak, aby metody danej klasy zajmowały się odczytem i zapisem jedynie pól tej klasy i odpowiednio wykorzystywały metody z klasy bazowej.



Mechanizm zapisywania obiektów powinien pozwalać na zapis dowolnej liczby obiektów (niekoniecznie w jednym pliku), a nie tylko jednego obiektu. Wskazówka: warto wykorzystać operatory strumieniowe zdefiniowane dla klasy.

6. Napisać aplikację posiadającą interfejs konsolowy. Interfejs powinien pozwalać na:
- dodawanie obiektów klas A, B, C oraz D do wektora wskaźników na klasę A (`vector<A*>`), ich modyfikowanie i usuwanie z wektora;
  - demonstrację działania polimorfizmu (tak, aby przy usunięciu słowa *virtual* z deklaracji funkcji wirtualnej widoczne było odmienne działanie programu);
  - zapis stanu wszystkich obiektów przechowywanych w kontenerze (jako wskaźniki na klasę A) w przestrzeni dyskowej (cała kolekcja w jednym pliku), oraz późniejsze ich odczytanie z pliku i odtworzenie kolekcji;
  - wypisanie na ekran stanu wszystkich obiektów z wektora.
- Interfejs powinien być odporny na błędy użytkownika.

7. Utworzyć dokumentację doxygen projektu.

#### UWAGA

Jeżeli jest wybór pomiędzy stosowaniem mechanizmów, funkcji, instrukcji typowych dla języka C i C++ należy stosować konstrukcje właściwe dla C++ np. `char*` - `string`, `FILE*` - `iostream`, itp. Jeden obiekt – 2 pliki: `obiekt.h`, `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.

Wszystkie projekty powinny być umieszczone w serwisie GitLab co najmniej 2 dni przed terminem obrony (np. poniedziałek do godz. 24 w tygodniu obrony dla grupy środowowej). Zaliczanie projektu na zajęciach w dniach 8-10 maja.

#### KRYTERIA OCENY

przejrzystość kodu	1 p.
implementacja dziedziczenia i pozostałe wymagania (w tym zakres aplikacji testowej)	8 p.
zapis/odczyt obiektów	2 p.
dokumentacja	2 p.
interfejs użytkownika (przejrzystość, poprawność, odporność)	2 p.