

# LABORATORIUM PROE 18Z, PROJEKT 1

KLASA, KONSTRUKTORY I DESTRUKTORY, PRZECIĄŻANIE FUNKCJI I OPERATORÓW

Prowadzący: Grzegorz Krawczyk ([G.Krawczyk@elka.pw.edu.pl](mailto:G.Krawczyk@elka.pw.edu.pl))

## INFORMACJE SZCZEGÓŁOWE

Aplikacja ma być oparta na zestawie klas (min. 3). Obiekt główny ma być złożony z minimum 2 podobiektów (lub obiektów będących z nim w relacji), w tym co najmniej jednego tworzonego **dynamicznie** i jednego tworzonego **automatycznie**. Odwzorowanie powinno być możliwie realistyczne - dla skomplikowanych obiektów odpowiednio uproszczone.

Każda klasa powinna prawidłowo zachowywać się w przypadku **kopiowania oraz przypisania**. Należy rozważyć realizację konstruktora kopiującego lub użycie standardowego konstruktora kopiującego, przenoszącego oraz operatora przypisania.

Proszę zaprojektować i zaimplementować dla klas kilka (3) **sensownych, różnorodnych** operatorów lub funkcjonalności. Każda klasa ma posiadać **dodatkowo** operator wypisania na ekran (ostream) oraz porównania.

Napisać program testujący klasę główną i jej funkcjonalności. Należy przetestować wszystkie zaimplementowane operatory i metody

## WYMAGANIA

1. Projekt ma się kompilować i uruchamiać na komputerach w lab09.
2. *Koncepcja wstępna (pdf na adres prowadzącego) zawierająca **bardzo krótki** opis podziału problemu na klasy oraz zależności między nimi (0.5 - 1 str)*
3. Kod źródłowy programów powinien być napisany zgodnie z regułami dobrego stylu programowania. Na każdą klasę powinny przypadać 2 pliki - plik nagłówkowy .h i plik definicji .cpp. Należy stosować identyfikatory ułatwiające czytającemu zrozumienie kodu).
4. Moduł testowania (podstawowe operacje + przypadek brzegowy)
5. Kompilacja z włączonymi flagami -Wall -Werror (dla Window analogicznie)

## UWAGI DODATKOWE

6. Program powinien być napisany w zgodzie z paradygmatem obiektowym.
7. Jeżeli jest wybór pomiędzy stosowaniem mechanizmów, funkcji, instrukcji typowych dla języka C i C++ proszę stosować odpowiednie konstrukcje właściwe dla C++ np. char\* - string,
8. Wycieki pamięci, zakleszczenia, niespodziewane zakończenia programu to **poważne błędy**.
9. Kończącą wersję projektu należy wysłać w czasie laboratorium na którym nastąpił termin obrony (warunek konieczny do otrzymania punktów).

## KRYTERIA OCENY

Koncepcja p. obiektowego	1 pkt
Realizacja p. obiektowego	1 pkt
Konstruktory/Destrukcyjne/op przypisania	1 pkt
Dynamiczne zarządzanie pamięcią	2 pkt
Operatory podstawowe(wypisanie,porównanie)	1 pkt
Operatory dodatkowe / funkcjonalność	2 pkt
Styl programowania (czytelność)	1 pkt
Testowanie	1 pkt

### Punkty Karne:

Problemy z kompilacją	< -3:0 > pkt
Problemy z działaniem	< -3:0 > pkt
Brak podziału na pliki/brak Makefile	< -3:0 > pkt
Brak sensownych testów	< -3:0 > pkt
Brak zrozumienia własnego kodu	< -10:0 > pkt

Minimalna liczba punktów możliwych do uzyskania wynosi **0**.