

# LABORATORIUM PROE.B, PROJEKT 1

KLASA, KONSTRUKTORY I DESTRUKTORY, PRZECIĄŻANIE FUNKCJI I OPERATORÓW

## ZADANIE

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

Aplikacja będzie rozwijana w trakcie trzech kolejnych projektów. Do każdego z nich należy przygotować interfejs umożliwiający przetestowanie zaimplementowanych możliwości.

## INFORMACJE SZCZEGÓŁOWE

Aplikacja ma być oparta na zestawie klas, z których głównym obiektem będzie element, którym zarządzamy. Obiekt ten ma być złożony z minimum 3 podobiektów, w tym co najmniej jednego tworzego **dynamicznie** i jednego tworzego **automatycznie**. Odzworowanie powinno być możliwie realistyczne - dla skomplikowanych obiektów odpowiednio uproszczone.

We wszystkich konstruktorach i destruktorach należy wstawić kod drukujący na ekran informację o ich wywołaniu. Wyświetlenie to ma być warunkowe – jedynie w momencie zdefiniowania zmiennej kompilacji **\_DEBUG**. Wydruki te będą pomocne w czasie śledzenia sekwencji wywołania konstruktorów i destruktorów.

Klasa główna ma zawierać mechanizm określania liczby stworzonych obiektów tego typu (**statyczne pole** klasy), oraz **statyczną metodę** zwracającą to statyczne pole klasy.

Każda klasa powinna prawidłowo zachowywać się w przypadku **kopiowania**. Należy rozważyć realizację konstruktora kopiującego lub użycie standardowego konstruktora kopiującego. Podobnie rozważyć operator **przypisania** dla klas.

Proszę zaprojektować i zaimplementować dla klas kilka **sensownych, różnorodnych** operatorów (minimum 10), w tym: jednoargumentowe, dwuargumentowe, konwersji, przypisania, indeksowe. Należy zastosować wybrane operatory jako metody klas oraz jako funkcje zaprzyjaźnione z klasami – ale tylko tam gdzie jest to niezbędne. Zastanów się w jaki sposób można **zablokować** możliwość wywołania dowolnego operatora (np. a+b).

Napisać program główny testujący klasę główną i jej podklasy (oddzielny moduł/plik). Dla testów należy stworzyć obiekty **automatyczne, dynamiczne i statyczne (lokalne, globalne)** w **funkcji**

**testowej** wywoływanej z funkcji main. Celem powyższych testów jest między innymi obserwowanie **czasu życia obiektów** oraz zachowanie się funkcji i operatorów przeciążonych.

W osobnej funkcji, wywoływanej w funkcji main jedynie przy ustawionej zmiennej kompilacji `_DEBUG`, należy przetestować wszystkie zaimplementowane operatory.

Na każdą klasę powinny przypadać 2 pliki - plik nagłówkowy .h i plik definicji .cpp.

#### 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.cpp`.

Proszę przysyłać projekty 2 dni przed terminem obrony (tj. wtorek do godz. 24 w tygodniu obrony dla grupy czwartkowej) na adres mailowy prowadzącego zajęcia.

#### KRYTERIA OCENY

czytelność kodu	1 p.
sensowność konstrukcji klas	3 p.
brak wycieków pamięci	1 p.
spełnienie pozostałych założeń	5 p.

1. Robot
2. Statek
3. Boisko
4. Księgarnia
5. Monitor
6. Komputer
7. Uczelnia
8. Szpital
9. System alarmowy
10. Szkoła
11. Dom
12. Telefon
13. Miasto
14. Kraj
15. Kosmos
16. Drukarka
17. Park
18. Wypożyczalnia
19. Pociąg
20. Warsztat
21. Sklep
22. Rower
23. Poczta
24. Firma budowlana
25. Łódź
26. Kontynent
27. Planeta
28. Fabryka
29. Klub
30. Restauracja
31. Samochód
32. Pociąg
33. Samolot
34. Cukiernia