

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`.

Możliwe jest uzyskanie 1 dodatkowego punktu za oddanie projektu w formie repozytorium na GitHubie, łączna maksymalna liczba punktów to dalej 10.

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. |
| Oddanie w postaci repozytorium na GitHubie | 1 p. |
| łączna maksymalna liczba pkt | 10 p. |

1. Statek
2. Księgarnia
3. Komputer
4. Uczelnia
5. Szpital
6. System alarmowy
7. Szkoła
8. Dom
9. Miasto
10. Kraj
11. Kosmos
12. Wypożyczalnia
13. Pociąg
14. Warsztat
15. Sklep
16. Rower
17. Poczta
18. Firma budowlana
19. Kontynent
20. Planeta
21. Fabryka
22. Klub
23. Restauracja
24. Samochód
25. Samolot