

LABORATORIUM PROBI, PROJEKT 1

KLASA, KONSTRUKTORY I DESTRUKTORY, PRZECIĄŻANIE FUNKCJI I OPERATORÓW

ZADANIE

Napisać program wykorzystujący klasę

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

INFORMACJE SZCZEGÓŁOWE

Obiekt klasy głównej (będącej przedmiotem projektu) ma być złożony z 2 podobiektów: jednego tworzonego **dynamicznie** i jednego tworzonego **automatycznie**. Podobiekt dynamiczny oznacza, że polem obiektu głównego jest wskaźnik na podobiekt (do celów tego projektu w konstruktorze domyślnym **ma być inicjalizowany na zero**).

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 i przypisywania**.

Należy zaprojektować i zaimplementować dla klas **sześć sensownych** operatorów: jednoargumentowy, dwuargumentowy, konwersji, przypisania, indeksowy, wypisywania na strumień.

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

Projekt należy udostępnić prowadzącemu zajęcia w Gitlabie (<https://gitlab-stud.elka.pw.edu.pl/>) najpóźniej dzień przed terminem obrony.

sensowność konstrukcji klas	2 p.
brak wycieków pamięci (poprawny konstruktor kopiujący i operator przypisania)	2 p.
statyczne pole i metoda	1 p.
6 operatorów	3 p.
program testujący	1 p.
czytelność kodu	1 p.

Materiały opracowane w ramach zadania 15 „Modyfikacja międzywydziałowych studiów I stopnia na kierunku Inżynieria Biomedyczna” projektu „NERW PW. Nauka – Edukacja – Rozwój – Współpraca”, współfinansowanego jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego