

Celem ćwiczenia jest nabycie praktycznych umiejętności zarządzania dynamiczną pamięcią komputera. W trakcie ćwiczenia należy zaimplementować klasę listy wskaźnikowej oraz porównać wydajność jej działania z kolekcją standardową. Na zajęciach należy wykonać następujące zadania:

1. Stworzyć klasę listy wskaźnikowej CLista, której elementy są strukturą punktów 3D(double x,y,z),
2. Klasa listy musi posiadać metody pozwalające na dodawanie, wstawianie, usuwanie i wyszukiwanie (po kursorze) elementów z listy,
3. Należy napisać program, który w nieskończonej pętli dodaje i usuwa np. 1000 elementów. Następnie należy uruchomić Manager zadań systemu Windows i odszukać swój program w szczegółowej liście procesów. Jeżeli lista poprawnie zarządza pamięcią ilość pamięci używana przez nasz program powinna być niezmienna.
4. Następnie w celu zaobserwowania efektu wycieku pamięci należy usunąć, lub zakomentować wszystkie instrukcje *delete* odpowiedzialną za zwalnianie pamięci, po czym w Menagerze zadań będzie można zauważyć nieustanny przyrost pamięci programu,
5. Stworzyć kolekcję standardową (np. `std::vector<>`) z elementami tego samego typu co w klasie CLista
6. Należy zaimplementować dwie pętle (z dużą ilością powtórzeń), w których są wykonywane identyczne operacje dla klasy CLista w pętli pierwszej oraz dla kolekcji standardowej w pętli drugiej. Należy obliczyć czas jaki minął podczas wykonywania każdej z pętli i czasy te porównać.

Wskazówki:

1. Konieczne jest zapoznanie się z listą wskaźnikową,
2. Należy zapoznać się z przykładowym programem ilustrującym wykorzystanie kolekcji standardowych (`std::vector`) (kod 2.1).

Uwagi:

- W metodach klas nie wolno używać funkcji typu : `printf`, `scanf`, `std::cout`, `std::cin`. Warstwa prezentacji musi zostać zaimplementowana w pętli głównej (poza klasą),
- W plikach nagłówkowych (\*.h) nie wolno zawierać źródeł/rozwojów funkcji i metod klasowych (dotyczy to także konstruktorów i destruktorów),
- Powyższe uwagi obowiązują na wszystkich laboratoriach.