

Celem ćwiczenia jest zapoznanie się z jednym z podstawowych filarów programowania obiektowego (polimorfizmu) oraz z przeciążaniem operatorów w języku C++. Do wykonania ćwiczenia konieczne jest zastosowanie rozwiązania zaimplementowanego w ramach Ćwiczenia 2 – lista wskaźnikowa.

1. Klasa listy wskaźnikowej powinna zostać rozszerzona o następujące elementy:

- a. Konstruktor kopiujący, przyjmujący jako argument istniejącą listę wskaźnikową (z wykorzystaniem **referencji**). **UWAGA:** Błędne opisanie tego konstruktora może skutkować nieoczekiwanym działaniem operatora przypisania (=). Poprawnie napisany konstruktor kopiujący powinien być zadeklarowany następująco:
NazwaKlasy(const NazwaKlasy &);
- b. Konstruktor inicjalizujący, przyjmujący jako argument referencję do kolekcji standardowej `std::vector<>`, zawierającej dowolną liczbę punktów 3d, o strukturze zgodnej z wykorzystaną w ramach ćwiczenia 2.
- c. Metoda pozwalająca na konkatencję dwóch list: listy wskaźnikowej, w której jest wywołana oraz listy wskaźnikowej, która jest podawana jako argument, z wykorzystaniem referencji. Lista, do której referencja jest tworzona podczas wywołania ma zostać dodana do listy, w ramach której została wywołana.
- d. Metoda pozwalająca na mnożenie listy wskaźnikowej przez skalar, przekazany jako argument. Efektem działania metody ma być powielenie listy. Metoda nie może zmieniać obiektu, w którym została wywołana.

2. Przeciążanie operatorów

- a. Należy przeciążyć podane operatory w taki sposób, aby realizowały implementację następujących funkcjonalności:

operator	opis działania
--	usunięcie ostatniego elementu listy
+=	konkatenacja dwóch list wskaźnikowych
+	konkatenacja dwóch list wskaźnikowych [operator powinien zwracać obiekt, żeby wywołany został konstruktor kopiujący] UWAGA: łączenie ma działać poprawnie także dla łączenia listy samej ze sobą!
*	Zwrócenie nowej, powielonej listy wskaźnikowej
[]	wydobycie referencji elementu listy wyszukanego po kursorze
++	usunięcie całej listy
<<	wyświetlanie całej listy

Uwagi:

- W metodach klas nie wolno używać funkcji typu : printf, scanf, std::cout, std::cin.
- W plikach nagłówkowych (*.h) nie wolno zawierać źródła/rozwinienia funkcji i metod klasowych (dotyczy to także konstruktorów i destruktorów),
- Powyższe uwagi obowiązują na wszystkich laboratoriach.
- Przed rozpoczęciem ćwiczenia warto zapoznać się z materiałami dotyczącymi przeciążania operatorów, dostępnymi między innymi na stronach Internetowych:
<https://en.cppreference.com/w/cpp/language/operators>
https://pl.wikibooks.org/wiki/C%2B%2B/Przeci%C4%85%C5%BCanie_operator%C3%B3w