

# LABORATORIUM PROBI, PROJEKT 3

## SZABLONY, WZORCE, WYJĄTKI

Należy napisać klasę Kontener, która będzie służyła do przechowywania obiektów dowolnego, ale z góry nieokreślonego typu.

## ZAŁOŻENIA:

1. Kontener zaprojektowany jest z wykorzystaniem szablonów.
2. Kontener pozwala na dodawanie i usuwanie dowolnego obiektu.
3. Kontener pozwala na zamianę kolejności dwóch obiektów.
4. Kontener zrealizowany jest na liście dwukierunkowej (dowolnie - cyklicznej bądź nie).
5. Kontener nie może mieć z góry określonej maksymalnej liczby zawartych w nim obiektów.
6. Kontener powinien mieć możliwość zapisu wszystkich obiektów do pliku i odczytu z pliku (zakładając, że w danych obiektach jest zaimplementowany operator strumieniowy).
7. W klasie węzła należy tak zaimplementować destruktora, aby sam "naprawiał" powiązania listy powstałe po jego usunięciu (posiada przecież wskaźniki na sąsiednie elementy), ale tylko przy założeniu, że został wywołany dla elementu wewnątrz listy (czyli nie pierwszego i nie ostatniego).
8. Dla kontenera należy przeciążyć operator przypisania `Kontener& Kontener::operator= (const Kontener &)`
9. Dla kontenera należy przeciążyć operator porównania `bool Kontener::operator==(const Kontener &)`
10. Kontener ma zawierać dodatkowo metody (dwie wybrane przez prowadzącego)
  - a) `T& Kontener::operator[](int i)` - operator indeksowy – pobieranie obiektu T przechowywanego w węźle o indeksie i
  - b) `+` (dodanie dwóch kontenerów),
  - c) `+=` (dodanie dwóch kontenerów),
  - d) `int indexOf(const T &)` – zwraca indeks obiektu T; rzuca wyjątek, jeśli obiektu nie ma w kontenerze
  - e) `bool contains(T &t)`; – prawda, czy obiekt t jest w kontenerze
  - f) `void moveToFront(int numer)`; – przenosi i-ty obiekt na początek kontenera
  - g) `void moveToEnd(int numer)`; – przenosi i-ty obiekt na koniec kontenera
11. Ewentualne błędy powinny być obsługiwane z wykorzystaniem wyjątków.
12. Nie korzystamy z gotowych elementów kontenera z biblioteki STL (takich jak vector itp.).

Do kontenera ma zostać przygotowana aplikacja testująca posiadająca proste menu. Ma ona pozwalać na dodawanie, usuwanie, zamianę kolejności, zapisywanie do pliku, odczyt z pliku obiektów z pierwszego projektu.

Do projektu nie jest wymagana dokumentacja.

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

**PUNKTACJA:**

Poprawna implementacja listy	
Lista dwukierunkowa (dowolna liczba węzłów)	1
Operator =	1
Operator ==	
Szablony	1
Dodawanie i usuwanie dowolnego elementu listy	2
Zamiana kolejności dwóch obiektów	2
Zapis do pliku	2
Odczyt z pliku	4
Destruktor węzła	1
Inne	
Aplikacja testująca	2
Wyjątki	1
Metoda A	1
Metoda B	1

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