МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Системний аналіз та інформаційно-аналітичні технології»

Лабораторна робота № 1  
з дисципліни «Алгоритми та структури даних»

(Варіант III)

Виконав:

Олексій ДРАБЧАК

студент групи КН-320Г

Перевірила:

Наталя МАРЧЕНКО

Харків 2022

**Тема лабораторної роботи.** Двозв’язні та кільцеві списки та робота з ними.

**Мета:** ознайомитися із основними способами організації списків та особливостями їх програмної реалізації. Набути практичних навичок по роботі зі двозв’язними та кільцевими списками.

# Порядок виконання роботи:

1. Для організації двозв’язного списку визначити структурний тип, що містить покажчик на свій тип і поля. Варіант завдання вибирається за номером студента в списку групи.



1. Написати програму, яка демонструє основні дії по роботі із двозв’язним списком і пропонує користувачу вибір дії:
   1. сформувати список;
   2. додати елемент до списку (в кінець);
   3. вилучити вказаний елемент зі списку;
   4. визначити кількість елементів у списку;
   5. поміняти два сусідні елементи місцями;
   6. об’єднати два списки в один з утворенням нового списку шляхом дописування другого списку в кінець першого списку;
   7. очистити список.
2. Перетворити лінійний список в кільцевої. Внести необхідні зміни в операції роботи зі списком.
3. Написати програму, яка демонструє основні дії по роботі із двозв’язним списком і пропонує користувачу вибір дії:
   1. сформувати список;
   2. додати елемент до списку;
   3. вилучити вказаний елемент зі списку;
   4. визначити кількість елементів у списку;
   5. поміняти два сусідні елементи місцями;
   6. очистити список

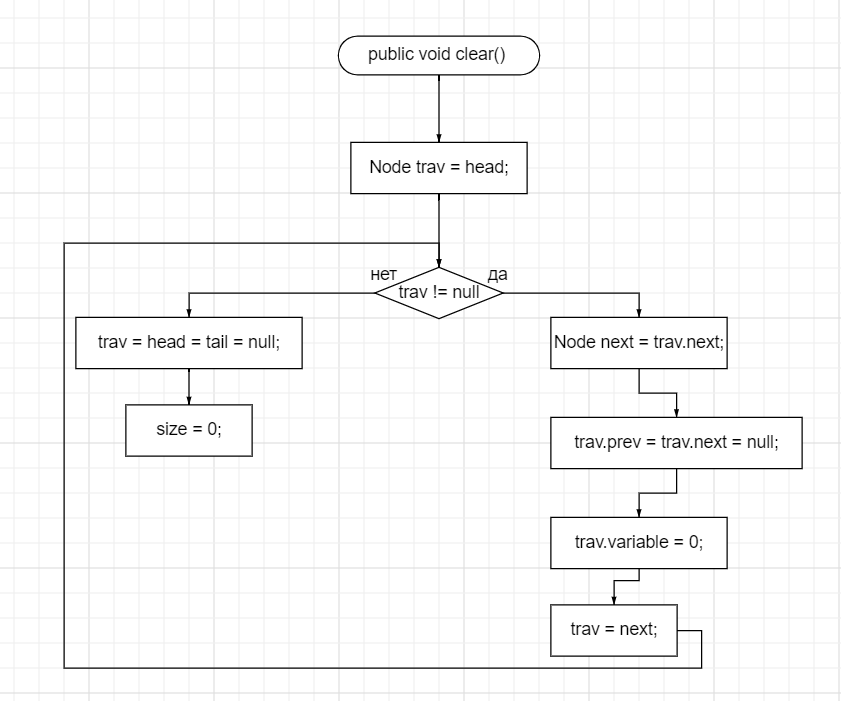
Перевірити роботу операцій. Зробити висновки. Зберегти файл з тестом програми для подальших робіт.

**Структура програми:**

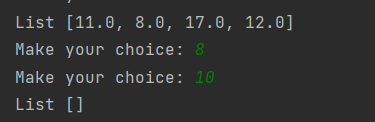
В програмі описаний клас Node який реалізовує вузол як одиницю двохзв’язного спуску. Також є два класи відповідно DoublyLinkedList та CycleDoublyLinkedList які реалізовують інтерфейс Collection. Для виконання програми було створено клас-драйвер для виконання циклу виконання програми. В програмі були використані основні парадигми ООП: інкапсуляція, наслідування яке описане зв’язком «IS A» та поліморфізм а також механізм композиції який описаний зв’язком «HAS A».

**Блок-схеми для методів:**

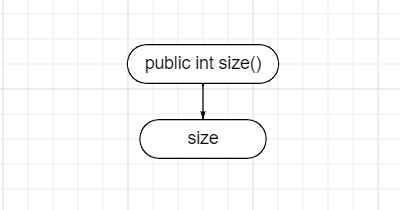
1. Метод clear() – очищення списку.



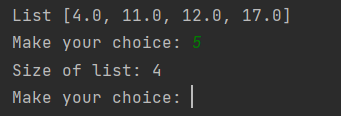
Очищення спуску:



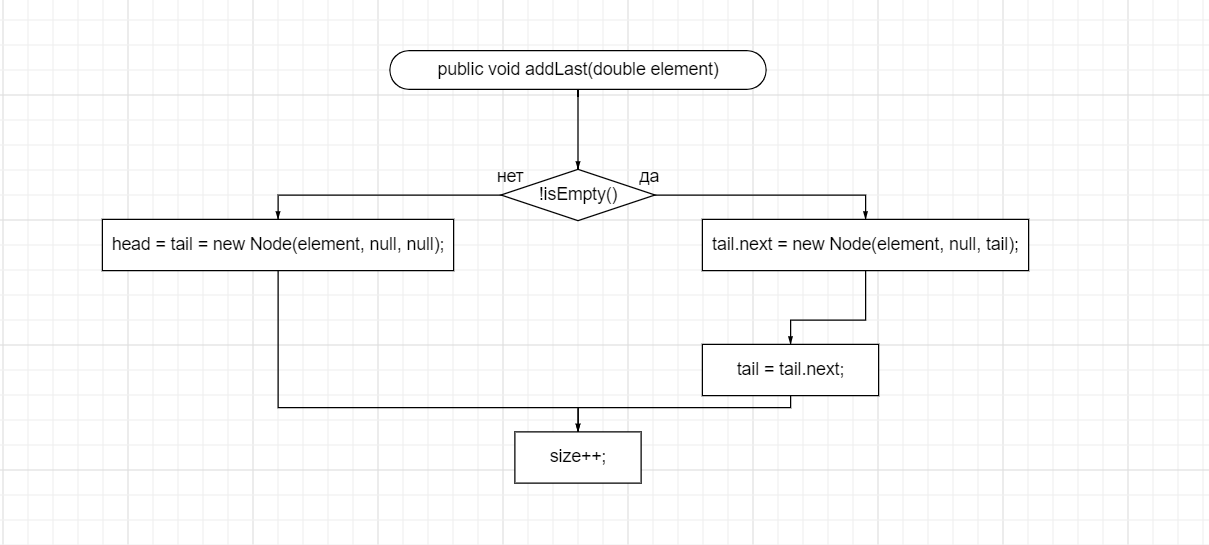
1. Метод size() – отримання розміру списку.



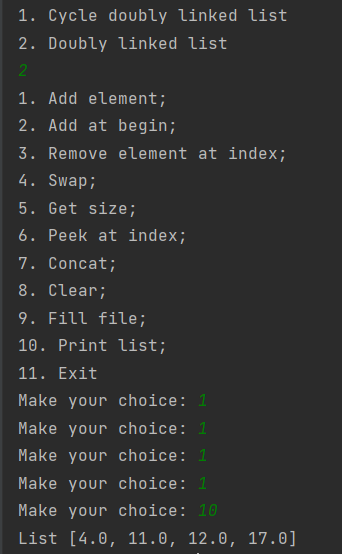
Отримання розміру списку:



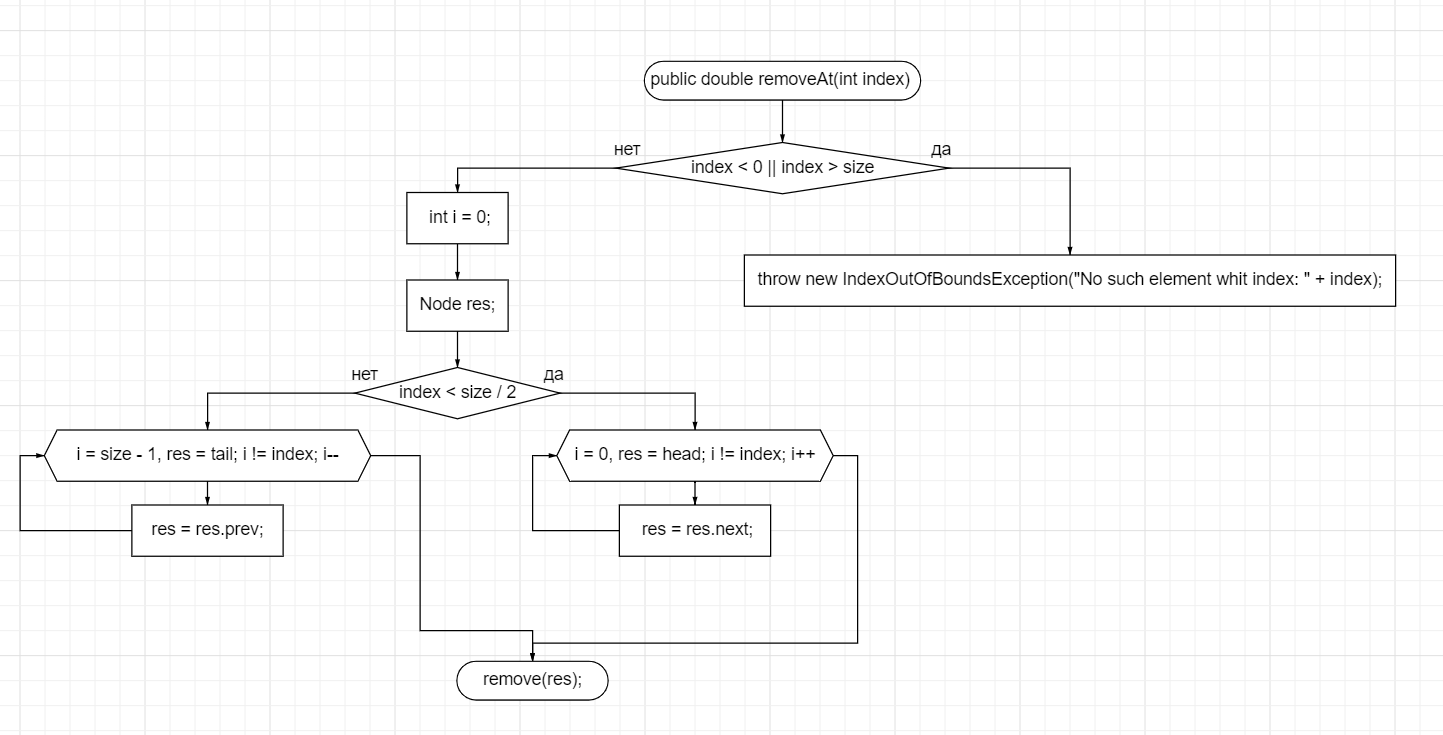
1. Метод addLast() – додавання нового вузла в кінець списку.

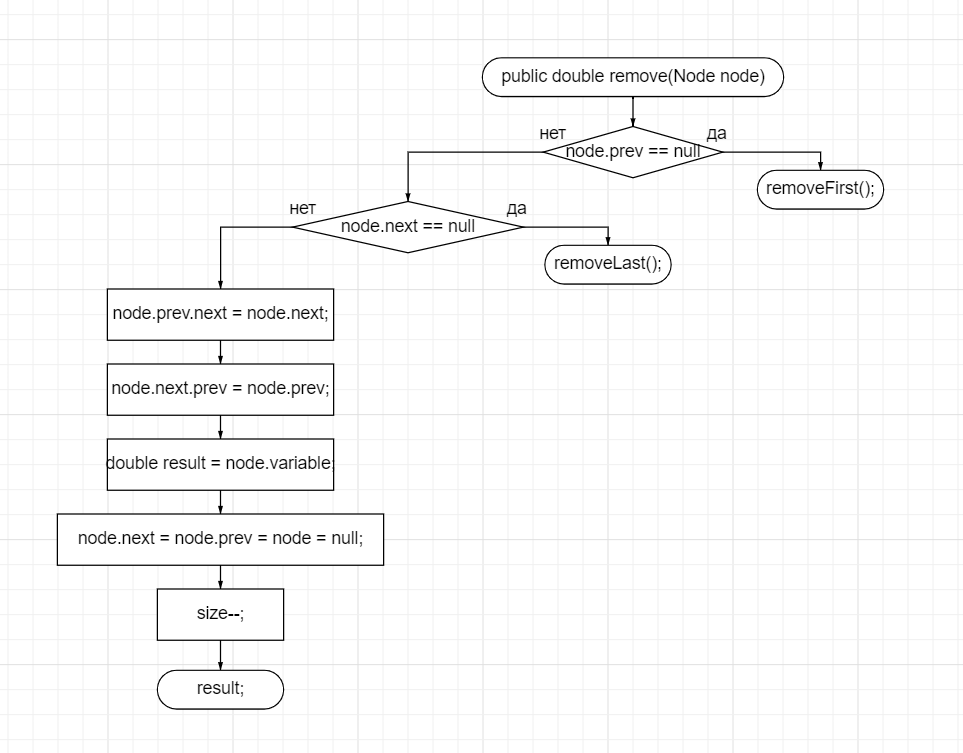


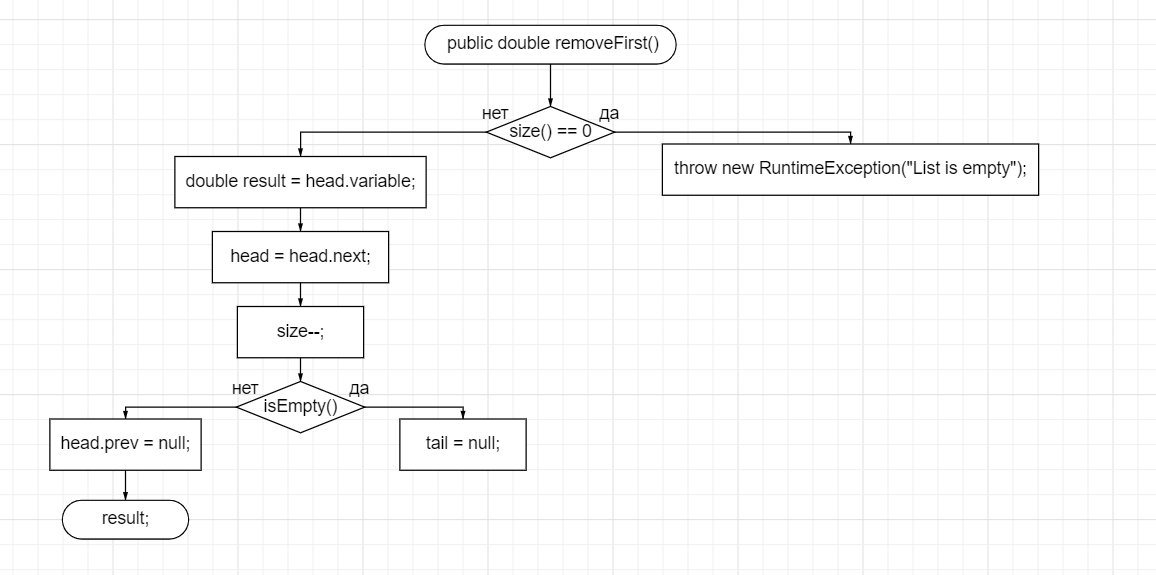
Демонстрація роботи методу додавання елемента в список:

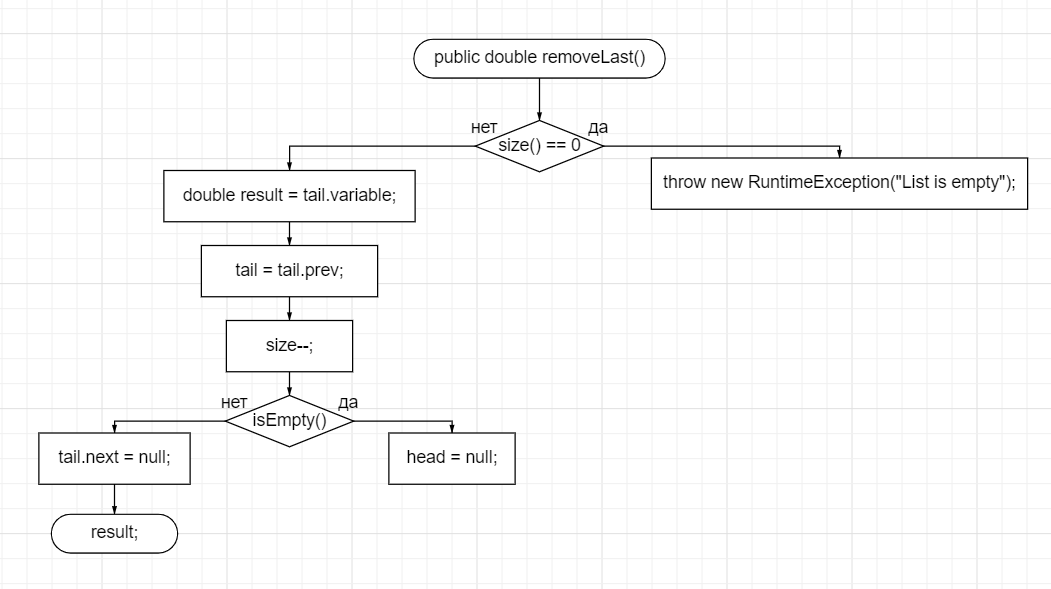


1. Метод removeAt() – видалення вузла із вказаної позиції.



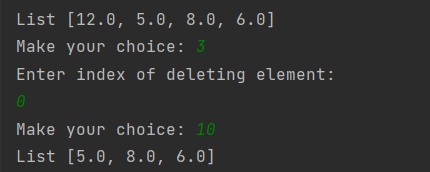




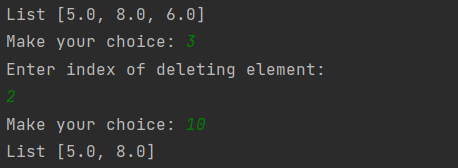


Демонстрація видалення елемента зі списку:

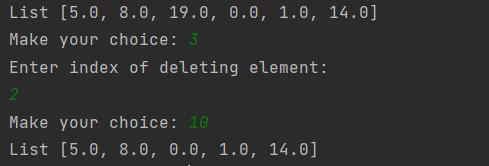
1. Видалення першого елемента:



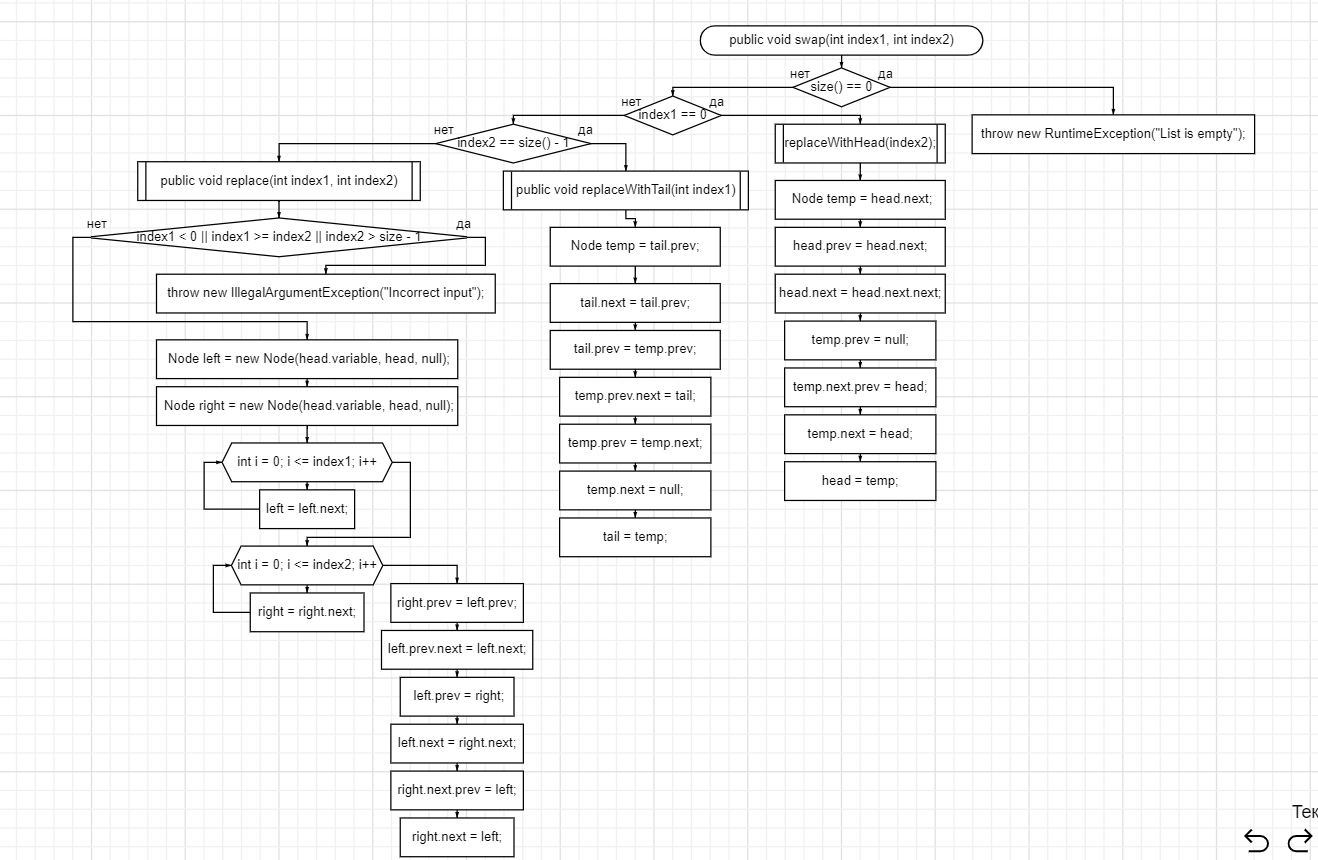
1. Видалення останнього елемента:



1. Видалення довільного елемента:

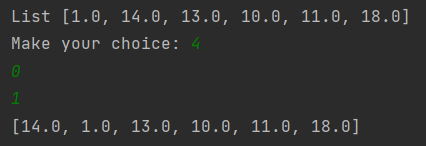


1. Метод swap() – поміняти два елементи місцями.

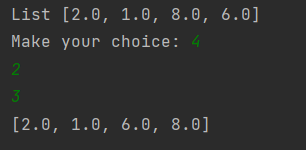


Демонстрація зміни елементів місцями:

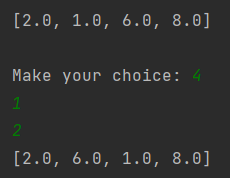
1. Заміна елемента з першим:



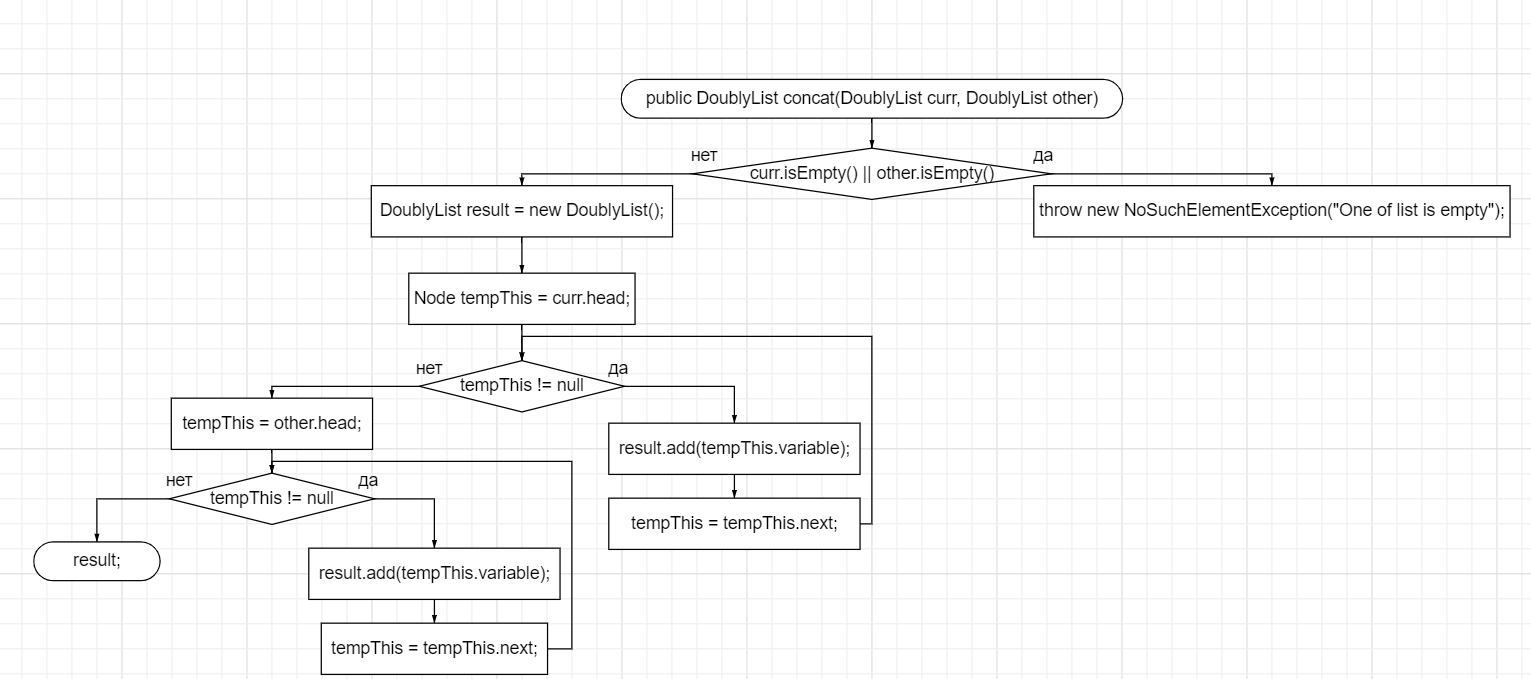
1. Заміна елемента з останнім:



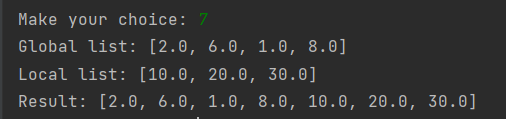
1. Заміна довільних двох сусідніх елементів:



1. Метод concat() – зв’язування двох списків з утворенням нового.



Демонстрація методу приєднання одного списку в кінець другого і запису результату в новий список:



**Висновки:**

На цій лабораторній роботі, я ознайомився із основними способами організації списків та особливостями їх програмної реалізації. Набув практичних навичок по роботі зі двозхв’язними та кільцевими списками.