**ДНІПРОВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ОЛЕСЯ ГОНЧАРА**

**Факультет прикладної математики**

**Кафедра математичного забезпечення ЕОМ**

**Звіт**

з лабораторної роботи № 1

з курсу «Дискретні структури»

на тему: «Дек. Операції над деком»

Виконав:

студент 3-го курсу

групи ПЗ-21у-1

Войцехов М.О.

м. Дніпро

2022 р.

**Постановка задачі**

1Реалізувати дек. Для виконання операцій з елементами заданої структури створити функції , які реалізують:

-додавання елемента на початок деку;

-додавання елемента до кінця деку;

-видалення першого елемента деку;

-видалення останнього елемента деку;

-читання першого елемента деку;

-читання останнього елемента деку;

-змінити місцями перший та останній елемент деку;

-перевірка дека на порожнечу;

-отримання розміру деку;

-розвернути дек (зробити «дно» стеку вершиною, а вершину –«дном»);

-операція «чи належить заданий елемент» деку;

-очистити дек.

**Програмний код**

using System;

using System.Collections;

using System.Collections.Generic;

namespace Lab\_1

{

public class Deque

{

public Deque(string data)

{

Data = data;

}

public string Data { get; set; }

public Deque Previous { get; set; }

public Deque Next { get; set; }

}

public class Program

{

static Deque head;

static Deque tail;

static int count;

public static bool IsEmpty { get { return count == 0; } }

public static void AddLast(string data)

{

Deque node = new Deque(data);

if (IsEmpty)

head = node;

else

{

tail.Next = node;

node.Previous = tail;

}

tail = node;

count++;

}

public static void AddFirst(string data)

{

Deque node = new Deque(data);

Deque temp = head;

node.Next = temp;

head = node;

if (IsEmpty)

tail = head;

else

temp.Previous = node;

count++;

}

public static string RemoveFirst()

{

if (IsEmpty)

throw new InvalidOperationException("Deque is Empty");

string output = head.Data;

if (count == 1)

{

head = tail = null;

}

else

{

head = head.Next;

head.Previous = null;

}

count--;

return "Remove: " + output;

}

public static string RemoveLast()

{

if (IsEmpty)

throw new InvalidOperationException("Deque is Empty");

string output = tail.Data;

if (count == 1)

{

head = tail = null;

}

else

{

tail = tail.Previous;

tail.Next = null;

}

count--;

return "Remove: " + output;

}

public static string First

{

get

{

if (IsEmpty)

throw new InvalidOperationException("Deque is Empty");

return head.Data;

}

}

public static string Last

{

get

{

if (IsEmpty)

throw new InvalidOperationException("Deque is Empty");

return tail.Data;

}

}

public static void Clear()

{

head = null;

tail = null;

count = 0;

}

public static string Contains(string data)

{

Deque current = head;

if (IsEmpty)

{

throw new InvalidOperationException("Deque is Empty");

return "";

}

while (current != null)

{

if (current.Data.Equals(data))

return "Data is contains";

current = current.Next;

}

return "Data is not contains";

}

static void Main()

{

while (true)

{

Console.WriteLine("1. Add to head;");

Console.WriteLine("2. Add to tail;");

Console.WriteLine("3. Remove first;");

Console.WriteLine("4. Remove last;");

Console.WriteLine("5. First element;");

Console.WriteLine("6. Last element;");

Console.WriteLine("7. Last to First;");

Console.WriteLine("8. isEmpty;");

Console.WriteLine("9. Count;");

Console.WriteLine("10. Constains;");

Console.WriteLine("11. Clear;");

switch (Console.ReadLine())

{

case "1":

{

Console.WriteLine("Write data: ");

AddFirst(Console.ReadLine());

break;

}

case "2":

{

Console.WriteLine("Write data: ");

AddLast(Console.ReadLine());

break;

}

case "3":

{

Console.WriteLine(RemoveFirst());

break;

}

case "4":

{

Console.WriteLine(RemoveLast());

break;

}

case "5":

{

Console.WriteLine(First);

break;

}

case "6":

{

Console.WriteLine(Last);

break;

}

case "8":

{

Console.WriteLine("isEmpty - " + IsEmpty);

break;

}

case "9":

{

Console.WriteLine("Count - " + count);

break;

}

case "10":

{

Console.WriteLine("Write data to find: ");

Console.WriteLine(Contains(Console.ReadLine()));

break;

}

case "11":

{

Clear();

Console.WriteLine("Clear");

break;

}

}

}

}

}

}

**Отримані результати**

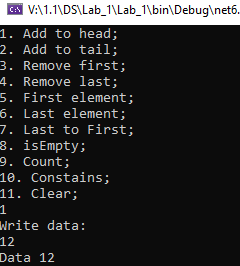


Рисунок 1 ­­– Додавання в початок

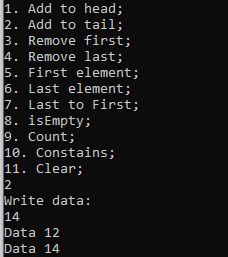


Рисунок 2 ­­– Додавання в кінець

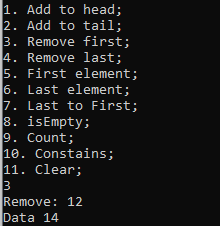


Рисунок 3 ­­– Видалення першого

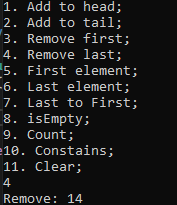


Рисунок 4­­– Видалення останнього

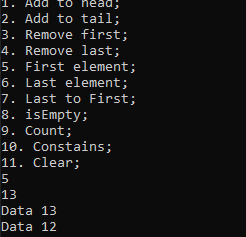


Рисунок 5 ­­– Перший елемент

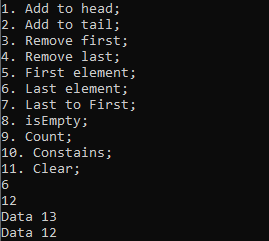


Рисунок 6 ­­– Останній елемент

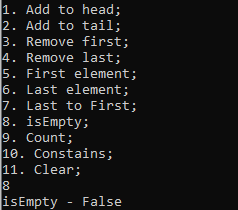


Рисунок 7 ­­– Порожній дек

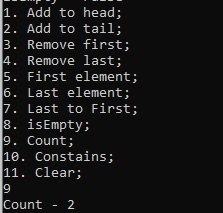


Рисунок 8 ­­– Кількість елементів

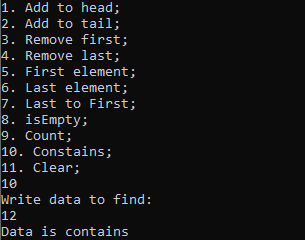


Рисунок 9 ­­– Чи є такий елемент в деку

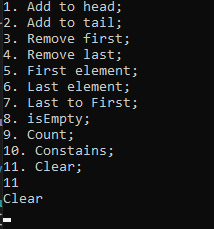


Рисунок 10 ­­– Видалення деку