Отчет по лабораторному занятию 10

Тема: «Линейные двунаправленные списки»

Выполнил: студент группы ИСП-6 Алёшин Максим

Вариант 3

**Задание 1.**Создать тип Продукт с полями Название продукта; кол-во, уникальный id, дата производства.

Создайте односвязный список, в котором будут присутствовать возможности:

1. Инициализация списка

2. Добавление нового продукта

3. Удаление продукта

4. Проверка списка на пустоту

5. Поиск продукта по названию

6. Поиск продукта по уникальному id

7. Очистка списка

Код программы:

using System;

namespace LR\_10

{

public class Item<T>

{

public Item(T name, T quantity, T id, T dateOfManufacture)

{

Name = name;

Quantity = quantity;

ID = id;

DateOfManufacture = dateOfManufacture;

}

public T Name { get; set; }

public T Quantity { get; set; }

public T ID { get; set; }

public T DateOfManufacture { get; set; }

public Item<T> Previous { get; set; }

public Item<T> Next { get; set; }

}

public class LinkedList<T>

{

Item<T> head;

Item<T> tail;

public void Add(T name, T quantity, T id, T dateOfManufacture)

{

Item<T> node = new Item<T>(name, quantity, id, dateOfManufacture);

if (head == null)

{

head = node;

}

else

{

tail.Next = node;

node.Previous = tail;

}

tail = node;

}

public void Remove(T name)

{

Item<T> current = head;

while (current != null)

{

if (current.Name.Equals(name))

break;

current = current.Next;

}

if (current != null)

current.Next.Previous = current.Previous;

else

tail = current.Previous;

if (current.Previous != null)

current.Previous.Next = current.Next;

else

head = current.Next;

}

public void Clean()

{

Console.Clear();

head = null;

tail = null;

Console.WriteLine("Список очищен.\n");

}

public bool isEmpty()

{

if (head == null)

return true;

else

return false;

}

public bool CheckUniqueID(T id)

{

Item<T> current = head;

while (current != null)

{

if (current.ID.Equals(id))

{

return false;

}

current = current.Next;

}

return true;

}

public void FindNameOrID(T product)

{

Item<T> current = head;

int countItem = 0;

while (current != null)

{

if (current.Name.Equals(product) || current.ID.Equals(product))

{

Console.WriteLine("\nНайдено сходство:");

Console.WriteLine($"Название: {current.Name}, " +

$"Кол-во: {current.Quantity}, " +

$"ID: {current.ID}, " +

$"Дата производства: {current.DateOfManufacture}\n");

countItem++;

}

else if (current.Next == null && countItem == 0)

{

Console.WriteLine($"\nПо запросу '{product}' ничего не найдено.\n");

}

current = current.Next;

}

}

public void Output()

{

Item<T> current = head;

if (isEmpty() == true)

{

Console.WriteLine("Список пуст. Добавьте продукты.\n");

return;

}

Console.WriteLine("Продукты в списке: ");

while (current != null)

{

Console.WriteLine($"Название: {current.Name}, " +

$"Кол-во: {current.Quantity}, " +

$"ID: {current.ID}, " +

$"Дата производства: {current.DateOfManufacture}");

current = current.Next;

}

Console.WriteLine();

}

}

class Program

{

static LinkedList<object> linkedList = new LinkedList<object>();

static void AddToList()

{

Console.Clear();

Random rand = new Random();

int id = rand.Next(1000000, 9999999);

while (linkedList.CheckUniqueID(id) != true)

{

id += 666;

}

Console.Write("Введите название продукта: ");

string name = Console.ReadLine();

Console.Write("Введите количество продукта: ");

int quantity = int.Parse(Console.ReadLine());

Console.Write("Введите дату производства: ");

string dateOfManufacture = Console.ReadLine();

linkedList.Add(name, quantity, id, dateOfManufacture);

Console.Clear();

Console.WriteLine("Продукт добавлен.\n");

}

static void RemoveAtList()

{

Console.Clear();

if (linkedList.isEmpty() == true)

{

Console.WriteLine("Список пуст. Добавьте продукты.\n");

return;

}

linkedList.Output();

Console.Write("Введите название продукта, который хотите удалить: ");

string name = Console.ReadLine();

linkedList.Remove(name);

Console.Clear();

Console.WriteLine("Продукт удалён из списка.\n");

}

static void SearchName()

{

Console.Clear();

if (linkedList.isEmpty() == true)

{

Console.WriteLine("Список пуст. Добавьте продукты.\n");

return;

}

Console.Write("Введите название продукта, который хотите найти: ");

string name = Console.ReadLine();

linkedList.FindNameOrID(name);

}

static void SearchID()

{

Console.Clear();

if (linkedList.isEmpty() == true)

{

Console.WriteLine("Список пуст. Добавьте продукты.\n");

return;

}

Console.Write("Введите id продукта, который хотите найти: ");

int id = int.Parse(Console.ReadLine());

linkedList.FindNameOrID(id);

}

static void Main(string[] args)

{

while (true)

{

Console.WriteLine("Что вы хотите сделать:\n" +

"1) Добавить продукт.\n" +

"2) Удалить продукт.\n" +

"3) Проверка списка на пустоту.\n" +

"4) Поиск продукта по названию.\n" +

"5) Поиск продукта по id.\n" +

"6) Очистить список.\n" +

"7) Посмотреть продукты в списке.\n");

Console.Write("Ваш выбор: ");

int choice = int.Parse(Console.ReadLine());

switch (choice)

{

case 1:

AddToList();

break;

case 2:

RemoveAtList();

break;

case 3:

Console.Clear();

Console.WriteLine(linkedList.isEmpty() == true ?

"Список пуст.\n" : "Список не является пустым.\n");

break;

case 4:

SearchName();

break;

case 5:

SearchID();

break;

case 6:

linkedList.Clean();

break;

case 7:

Console.Clear();

linkedList.Output();

break;

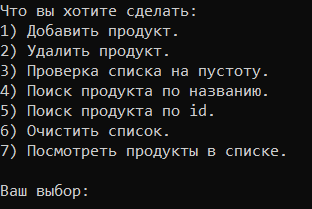
}

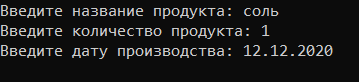
}

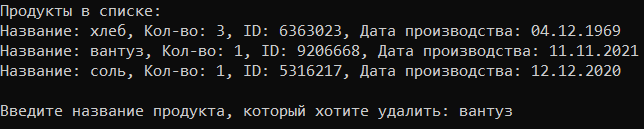
}

}

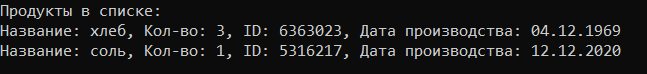
}











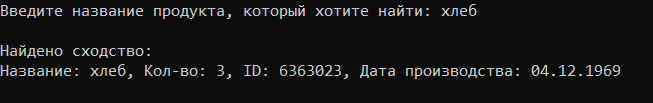


Рис. 6(Поиск продукта по названию)

