Отчет по лабораторному занятию 9

Тема: «Линейные однонаправленные 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 LR9

{

public class Product<T>

{

public Product(T name, T number, T id, T dateOfManufacture)

{

Name = name;

Number = number;

ID = id;

DateOfManufacture = dateOfManufacture;

}

public T Name { get; set; }

public T Number { get; set; }

public T ID { get; set; }

public T DateOfManufacture { get; set; }

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

}

public class LinkedList<T>

{

Product<T> head;

Product<T> tail;

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

{

Product<T> product = new Product<T>(name, number, id, dateOfManufacture);

if (head == null)

{

head = product;

}

else

{

tail.Next = product;

}

tail = product;

}

public void Remove(T name)

{

Product<T> current = head;

Product<T> previous = null;

while (current != null)

{

if (current.Name.Equals(name))

{

if (previous != null)

{

previous.Next = current.Next;

if (current.Next == null)

{

tail = previous;

}

}

else

{

head = head.Next;

if (head == null)

{

tail = null;

}

}

}

previous = current;

current = current.Next;

}

}

public bool EmptyList()

{

if (head == null)

return true;

else

return false;

}

public void Clear()

{

head = null;

tail = null;

}

public void IsPresentName(T name)

{

Product<T> current = head;

int countItem = 0;

while (current != null)

{

if (current.Name.Equals(name))

{

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

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

$" Кол-во: {current.Number}," +

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

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

countItem++;

}

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

{

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

return;

}

current = current.Next;

}

}

public void IsPresentID(T id)

{

Product<T> current = head;

int countItem = 0;

while (current != null)

{

if (current.ID.Equals(id))

{

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

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

$" Кол-во: {current.Number}," +

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

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

countItem++;

}

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

{

Console.WriteLine($"ID: {id} не принадлежит ни одному из товаров списка.");

return;

}

current = current.Next;

}

}

public void Output()

{

Product<T> current = head;

while (current != null)

{

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

$" Кол-во: {current.Number}," +

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

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

if (current.Next == null)

break;

current = current.Next;

}

}

class Program

{

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

static void ViewList()

{

Console.Clear();

if (linkedList.EmptyList() == true)

{

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

return;

}

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

linkedList.Output();

Console.WriteLine();

}

static void AddToList()

{

Console.Clear();

Random rand = new Random();

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

Console.Write("Что Вы хотите добавить: ");

string productName = Console.ReadLine();

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

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

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

string dateOfManufacture = Console.ReadLine();

linkedList.Add(productName.ToLower(), number, id, dateOfManufacture);

Console.Clear();

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

}

static void RemoveProduct()

{

Console.Clear();

ViewList();

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

string deleteProduct = Console.ReadLine();

linkedList.Remove(deleteProduct.ToLower());

Console.Clear();

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

}

static void FindName()

{

Console.Clear();

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

string product = Console.ReadLine();

linkedList.IsPresentName(product.ToLower());

}

static void FindID()

{

Console.Clear();

Console.Write("Введите ID продукта, который желаете найти: ");

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

linkedList.IsPresentID(id);

}

static void Main(string[] args)

{

while (true)

{

Console.WriteLine("Что Вы хотите сделать?\n" +

"1) Добавить товар.\n" +

"2) Удалить товар.\n" +

"3) Очистить список товаров.\n" +

"4) Посмотреть товары.\n" +

"5) Проверить список на пустоту.\n" +

"6) Поиск продукта по названию.\n" +

"7) Поиск продукта по уникальному id.\n");

Console.Write("Ваш выбор: ");

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

switch (choice)

{

case 1:

AddToList();

break;

case 2:

RemoveProduct();

break;

case 3:

linkedList.Clear();

Console.Clear();

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

break;

case 4:

ViewList();

break;

case 5:

Console.Clear();

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

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

break;

case 6:

FindName();

break;

case 7:

FindID();

break;

}

}

}

}

}

}

















 

