

**ОТЧЕТ О ВЫПОЛНЕНИИ**

**практической работы № 14**

Выполнил: ст.гр.

2ИСП9-23

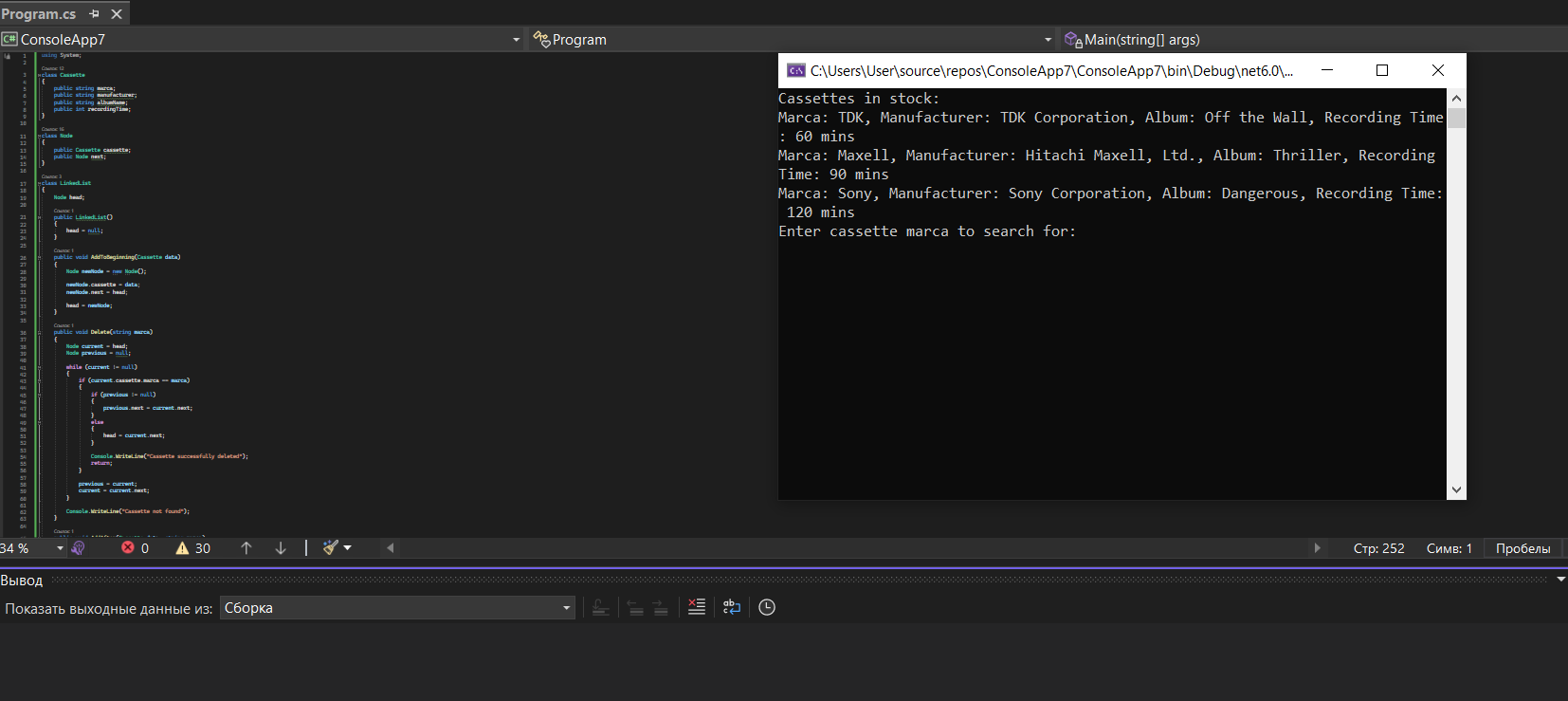
Быховцева Дарья

Специальность: 09.02.07

Информационные системы и программирование

Проверил: Боклач Б.И.

Москва 2022



ing System;

class Cassette

{

public string marca;

public string manufacturer;

public string albumName;

public int recordingTime;

}

class Node

{

public Cassette cassette;

public Node next;

}

class LinkedList

{

Node head;

public LinkedList()

{

head = null;

}

public void AddToBeginning(Cassette data)

{

Node newNode = new Node();

newNode.cassette = data;

newNode.next = head;

head = newNode;

}

public void Delete(string marca)

{

Node current = head;

Node previous = null;

while (current != null)

{

if (current.cassette.marca == marca)

{

if (previous != null)

{

previous.next = current.next;

}

else

{

head = current.next;

}

Console.WriteLine("Cassette successfully deleted");

return;

}

previous = current;

current = current.next;

}

Console.WriteLine("Cassette not found");

}

public void AddAfter(Cassette data, string marca)

{

Node newNode = new Node();

newNode.cassette = data;

Node current = head;

while (current != null)

{

if (current.cassette.marca == marca)

{

newNode.next = current.next;

current.next = newNode;

Console.WriteLine("Cassette successfully added after " + marca);

return;

}

current = current.next;

}

Console.WriteLine("Cassette not found");

}

public void AddBefore(Cassette data, string marca)

{

Node newNode = new Node();

newNode.cassette = data;

Node current = head;

Node previous = null;

while (current != null)

{

if (current.cassette.marca == marca)

{

if (previous != null)

{

previous.next = newNode;

}

else

{

head = newNode;

}

newNode.next = current;

Console.WriteLine("Cassette successfully added before " + marca);

return;

}

previous = current;

current = current.next;

}

Console.WriteLine("Cassette not found");

}

public void Display()

{

if (head == null)

{

Console.WriteLine("No cassettes to display");

return;

}

Node current = head;

Console.WriteLine("Cassettes in stock:");

while (current != null)

{

Console.WriteLine("Marca: " + current.cassette.marca + ", Manufacturer: " +

current.cassette.manufacturer + ", Album: " +

current.cassette.albumName + ", Recording Time: " +

current.cassette.recordingTime + " mins");

current = current.next;

}

}

internal Node Search(string? searchMarca)

{

throw new NotImplementedException();

}

}

class Program

{

static void Main(string[] args)

{

LinkedList list = new LinkedList();

Cassette[] cassettes = new Cassette[] {

new Cassette {marca = "Sony", manufacturer = "Sony Corporation", albumName = "Dangerous", recordingTime = 120},

new Cassette {marca = "Maxell", manufacturer = "Hitachi Maxell, Ltd.", albumName = "Thriller", recordingTime = 90},

new Cassette {marca = "TDK", manufacturer = "TDK Corporation", albumName = "Off the Wall", recordingTime = 60}

};

for (int i = 0; i < cassettes.Length; i++)

{

list.AddToBeginning(cassettes[i]);

}

list.Display();

// Поиск записи по марке

Console.Write("Enter cassette marca to search for: ");

string searchMarca = Console.ReadLine();

Node cassette = list.Search(searchMarca);

if (cassette != null)

{

Console.WriteLine("Cassette found: Marca: " + cassette.cassette.marca + ", Manufacturer: " +

cassette.cassette.manufacturer + ", Album: " +

cassette.cassette.albumName + ", Recording Time: " +

cassette.cassette.recordingTime + " mins");

}

Console.Write("Enter marca of existing cassette to add after: ");

string existingMarca = Console.ReadLine();

Console.Write("Enter marca of new cassette: ");

string newMarca = Console.ReadLine();

Console.Write("Enter manufacturer of new cassette: ");

string newManufacturer = Console.ReadLine();

Console.Write("Enter album name of the new cassette: ");

string newAlbumName = Console.ReadLine();

Console.Write("Enter recording time of the new cassette (in minutes): ");

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

Cassette newCassette = new Cassette

{

marca = newMarca,

manufacturer = newManufacturer,

albumName = newAlbumName,

recordingTime = newRecordingTime

}

list.AddAfter(newCassette, existingMarca);

Console.Write("Enter marca of existing cassette to add before: ");

existingMarca = Console.ReadLine();

Console.Write("Enter marca of new cassette: ");

newMarca = Console.ReadLine();

Console.Write("Enter manufacturer of new cassette: ");

newManufacturer = Console.ReadLine();

Console.Write("Enter album name of the new cassette: ");

newAlbumName = Console.ReadLine();

Console.Write("Enter recording time of the new cassette (in minutes): ");

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

newCassette = new Cassette

{

marca = newMarca,

manufacturer = newManufacturer,

albumName = newAlbumName,

recordingTime = newRecordingTime

};

list.AddBefore(newCassette, existingMarca);

Console.Write("Enter cassette marca to delete: ");

string deleteMarca = Console.ReadLine();

list.Delete(deleteMarca);

list.Display();

}

}

Контрольные вопросы

1. Описание структуры. Конструкторы.

Для определения структуры применяется ключевое слово struct:

struct имя\_структуры

{

// элементы структуры

}

1) Переменные структуры хранят не ссылку на объект, а сам объект. То есть это значимый тип, а не ссылочный.

2) Значит если одной структуре присвоить другую, то Скопируются Все Поля одной структуры в другую, а не ссылка, как было бы с классами.

3) Т.к. переменная хранит сами данные, а не ссылку на них, то доступ к полям осуществляется быстрее и есть некоторая экономия памяти.

4) Структуры не поддерживают наследование.

2. Обращение к элементам структуры.

Наименование структуры. Наименование элементов.

1. Чем отличается класс от структуры?

Различия Структуры и классы отличаются следующими особенности: Структуры — это типы значений; классы являются ссылочными типами. Переменная типа структуры содержит данные структуры, а не ссылку на данные в качестве типа класса.

1. Как объявить массив структур

Имя\_Структуры[]

Имя\_Переменной\_массива = New Имя\_Структуры[N]

N – количество элементов массива