**Отчёт  
по практической работе №17**

**по дисциплине «Основы алгоритмизации и программирования»**

**Тема: «Использование указателей для организации связанных**

**списков»**

|  |  |  |
| --- | --- | --- |
| **Студенты группы 13ИТ** | **\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Липинский К.С.**  **Кондратьев М.М.** |
| **Преподаватель** | **\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Счастливцев А.К.** |

**Цель работы:**

Получение навыков работы с указателями для организации связанных списков.

**Выполнение работы:**

using System;

using System.Collections.Generic;

using System.IO.Pipes;

using System.Linq;

using System.Security.Cryptography.X509Certificates;

using System.Text;

namespace consoleApp17

{

class Program

{

static void Main(string[] args)

{

unsafe

{

Person person = new Person();

person.age = 28;

person.height = 178;

fixed (int\* p = &person.age)

{

if (\*p < 30)

{

\*p = 30;

}

}

Console.WriteLine(person.age);

}

Console.ReadLine();

unsafe

{

int[] nums = { 0, 1, 2, 3, 7, 88 };

string str = "Hello";

fixed (int\* p = nums)

{

int third = \*(p + 2);

Console.WriteLine(third);

}

fixed (char\* p = str)

{

char forth = \*(p + 3);

Console.WriteLine(forth);

}

}

Console.ReadKey();

}

public class Person

{

public int age;

public int height;

}

}

}



**Операция \* и &**

unsafe

{

int\* x;

int y = 10;

x = &y;

Console.WriteLine(\*x);

y = y + 20;

Console.WriteLine(\*x);

\*x = 50;

Console.WriteLine(y);

{

Console.ReadLine();

}



**Указатель на другой указатель**

unsafe

{

int\* x;

int y = 10;

x = &y;

x = &y;

int\*\* z = &x;

\*\*z = \*\*z + 40;

Console.WriteLine(y);

Console.WriteLine(\*\*z);

}

Console.ReadLine();

}

}

}

****

**Указатели на типы и операция**

unsafe

{

Person person;

person.age = 29;

person.height = 176;

Person\* p = &person;

p->age = 30;

Console.WriteLine(p->age);

(\*p).height = 180;

Console.WriteLine((\*p).height);

}

}

}

public struct Person

{

public int age;

public int height;

}

}



**Задание 12:**

public class RandomQueue

{

private Queue<int> numbers;

public RandomQueue()

{

numbers = new Queue<int>();

}

public void FillQueueWithRandomNumbers(int count, int min, int max)

{

Random random = new Random();

for (int i = 0; i < count; i++)

{

int randomNumber = random.Next(min, max + 1);

numbers.Enqueue(randomNumber);

}

}

public int CountPositiveNumbers()

{

int positiveCount = 0;

foreach (int number in numbers)

{

if (number > 0)

{

positiveCount++;

}

}

return positiveCount;

}

public void AddResultToQueue(int result)

{

numbers.Enqueue(result);

}

public void PrintQueue()

{

foreach (int number in numbers)

{

Console.Write(number + " ");

}

}

}

class Program

{

static void Main()

{

RandomQueue randomQueue = new RandomQueue();

randomQueue.FillQueueWithRandomNumbers(10, -40, 50);

int positiveNumbersCount = randomQueue.CountPositiveNumbers();

Console.WriteLine("Число положительных чисел: " + positiveNumbersCount);

randomQueue.AddResultToQueue(positiveNumbersCount);

randomQueue.PrintQueue();

}

}

****

**Вопросы:**

1. Указатели в C# — это переменные, которые хранят адрес памяти другой переменной. В контексте связных списков, указатели используются для хранения ссылок на следующие (и иногда предыдущие) элементы списка, позволяя таким образом формировать цепочку узлов.
2. Принцип работы указателей в связных списках заключается в следующем: каждый элемент списка (узел) содержит данные и указатель на следующий узел в списке. Это позволяет элементам списка быть физически разбросанными по различным местам памяти, при этом оставаясь логически упорядоченными. Если список двусвязный, то каждый узел будет содержать также указатель на предыдущий элемент.
3. Основные операции над указателями в связных списках включают:

- Инициализация списка (создание первого узла)

- Вставка узла (в начало, конец или после/до определенного узла)

- Удаление узла (из начала, конца или определенного узла)

- Поиск узла по значению

- Обход списка (для выполнения операций с каждым элементом или для поиска)

- Очистка списка (удаление всех узлов)