1. Даны натуральные числа n и k. Составить программу вычисления выражения 1k  
   + 2k +...+ nk.  
   Разработать меню программы. Пункты меню: ввод данных, обработка, просмотр  
   результатов, выход

using System;

namespace SumOfPowers

{

class Program

{

static void Main(string[] args)

{

int n = 0;

int k = 0;

bool inputComplete = false;

do

{

Console.WriteLine("Меню:");

Console.WriteLine("1. Ввод данных");

Console.WriteLine("2. Обработка");

Console.WriteLine("3. Просмотр результатов");

Console.WriteLine("4. Выход");

Console.Write("Выберите пункт меню (1-4): ");

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

switch (menuChoice)

{

case 1:

Console.Write("Введите количество чисел (n): ");

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

Console.Write("Введите степень (k): ");

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

inputComplete = true;

break;

case 2:

if (!inputComplete)

{

Console.WriteLine("Сначала введите данные!");

break;

}

int sum = 0;

for (int i = 1; i <= n; i++)

{

int poweredNumber = (int)Math.Pow(i, k);

sum += poweredNumber;

}

Console.WriteLine("Сумма степеней чисел от 1 до {0} в степени {1} равна {2}", n, k, sum);

break;

case 3:

if (!inputComplete)

{

Console.WriteLine("Сначала введите данные!");

break;

}

Console.WriteLine("n = {0}, k = {1}", n, k);

break;

case 4:

Console.WriteLine("До свидания!");

return;

default:

Console.WriteLine("Выберите пункт меню от 1 до 4");

break;

}

} while (true);

}

}

}

2 Одномерные массивы.

Пример задания:

Дан целочисленный массив размера N. Проверить, чередуются ли в нем четные

и нечетные числа. Если чередуются, то вывести 0, если нет, то вывести

порядковый номер первого элемента, нарушающего закономерность.

int[] mass = { 1, 2, 4, 4, 5, 6, 7, 8 };

for (int i = 1; i < mass.Length; i++)

{

if ((mass[i] % 2 == 0 && mass[i - 1] % 2 == 0) || (mass[i] % 2 != 0 && mass[i - 1] % 2 != 0))

{

Console.WriteLine(i);

break;

}

if (i == (mass.Length - 1))

{

Console.WriteLine(0);

}

}

Random random = new Random();

int n = 10; // размер массива

int[] arr = new int[n]; // инициализация массива

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

{

arr[i] = random.Next(1, 10); // заполнение массива случайными числами от 1 до 10

Console.Write(arr[i] + " ");

}

int firstViolationIndex = -1; // индекс первого нарушения закономерности

for (int i = 1; i < n; i++)

{

if ((arr[i - 1] % 2 == 0 && arr[i] % 2 == 0) || (arr[i - 1] % 2 != 0 && arr[i] % 2 != 0))

{

firstViolationIndex = i;

break;

}

}

if (firstViolationIndex == -1)

{

Console.WriteLine("\nЧередуются четные и нечетные числа");

}

else

{

Console.WriteLine($"\nЗакономерность нарушается на элементе с индексом {firstViolationIndex}");

}

3. Методы сортировки  
Пример задания: Создать матрицу вещественных элементов А(n,n) и выполнить сортировку элементов, расположенных под главной диагональю методом «пузырька».  
Отсортированную последовательность вывести на экран

using System;

class Program

{

static void Main(string[] args)

{

int n = 5; // размер матрицы

double[,] A = new double[n, n];

// заполнение матрицы случайными числами

Random rand = new Random();

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

{

for (int j = 0; j < n; j++)

{

A[i, j] = rand.NextDouble() \* 10;

}

}

// вывод неотсортированной матрицы

Console.WriteLine("Исходная матрица:");

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

{

for (int j = 0; j < n; j++)

{

Console.Write("{0:F2}\t", A[i, j]);

}

Console.WriteLine();

}

// выполнение сортировки пузырьком для нижнего треугольника матрицы

for (int k = 0; k < n - 1; k++)

{

for (int i = 0; i < n - k - 1; i++)

{

for (int j = 0; j < n; j++)

{

if (j < k + 1)

{

continue; // элементы выше диагонали игнорируем

}

if (A[i, j] > A[i + 1, j])

{

// обмен элементов местами

double temp = A[i, j];

A[i, j] = A[i + 1, j];

A[i + 1, j] = temp;

}

}

}

}

// вывод отсортированной матрицы

Console.WriteLine("\nОтсортированная матрица:");

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

{

for (int j = 0; j < n; j++)

{

Console.Write("{0:F2}\t", A[i, j]);

}

Console.WriteLine();

}

Console.ReadKey();

}

}

4. Двумерные массивы.  
Пример задания:  
Дана квадратная матрица A[N, N]. Записать на место отрицательных элементов  
матрицы нули, а на место положительных — единицы. Вывести на печать  
нижнюю треугольную матрицу в общепринятом виде.

using System;

class Program

{

static void Main(string[] args)

{

int N = 5; // размер матрицы

int[,] A = new int[N, N];

// заполнение матрицы случайными значениями

Random rnd = new Random();

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

for (int j = 0; j < N; j++)

A[i, j] = rnd.Next(-10, 10);

// замена отрицательных на 0 и положительных на 1

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

for (int j = 0; j < N; j++)

A[i, j] = A[i, j] < 0 ? 0 : 1;

// вывод на печать нижней треугольной матрицы

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

{

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

Console.Write(A[i, j] + " ");

Console.WriteLine();

}

}

}

5 Строки.  
Пример задания:  
С клавиатуры вводятся строки S, S1, S2. Заменить в строке S все вхождения  
строки S1 на строку S2.

string S = Console.ReadLine();

string S1 = Console.ReadLine();

string S2 = Console.ReadLine();

string result = S.Replace(S1, S2);

Console.WriteLine(result);

6 Работа с файлами

Исходная информация содержится в файле. Определить, сколько слов текста имеют длину 1,2,3,…, 10 и более 10 символов. Вывести эти слова в

последовательности возрастания их длины. Слова очередной длины вывести с новой строки.

using System;

using System.IO;

using System.Linq;

class Program

{

static void Main(string[] args)

{

string text = File.ReadAllText($"C:\\Users\\krist\\OneDrive\\Рабочий стол\\1.txt");

// Разбиваем текст на слова, используя пробелы и знаки препинания как разделители

string[] words = text.Split(new char[] { ' ', ',', '.', '!', '?', ':', ';', '"', '(', ')' }, StringSplitOptions.RemoveEmptyEntries);

int[] lengthCount = new int[11]; // Для каждой длины слова от 1 до 10 и более 10 символов храним количество слов

// Подсчитываем количество слов каждой длины

foreach (string word in words)

{

int length = word.Length;

if (length >= 1 && length <= 10) // Длина от 1 до 10 символов

lengthCount[length]++;

else if (length > 10) // Длина более 10 символов

lengthCount[10]++;

}

// Выводим найденные слова для каждой длины на экран, сортируя их по возрастанию длины

for (int i = 1; i <= 10; i++)

{

if (i <= 9)

Console.WriteLine("Слова длины " + i + ": ");

else

Console.WriteLine("Слова длины более 10 символов: ");

var wordsOfLength = from word in words

where word.Length == i

orderby word

select word;

foreach (string word in wordsOfLength)

{

Console.WriteLine(word);

}

Console.WriteLine("Количество слов длины " + i + ": " + lengthCount[i]);

Console.WriteLine();

}

Console.ReadLine();

}

}

7Методы

Пример задания:

В программе обработку данных оформить в качестве метода. Дана символьная строка и символ. Слово - последовательность символов между пробелами, не содержащая пробелы внутри себя. Определить количество слов в строке, оканчивающихся на заданный символ.

static int CountWordsEndingWith(char endingSymbol, string sentence)

{

int count = 0;

string[] words = sentence.Split(' '); // разбиваем предложение на слова по пробелам

foreach (string word in words)

{

if (word.EndsWith($"{endingSymbol}"))

count++;

}

return count;

}

string sentence = "Сегодня мы гуляли в парке и наблюдали за птицами";

char endingSymbol = 'C';

int wordCount = CountWordsEndingWith(endingSymbol, sentence);

Console.WriteLine($"Количество слов, заканчивающихся на букву {endingSymbol}: {wordCount}");

8Структуры

Пример задания:

Разработать программу, реализующую работу со структурой Аптека. В

программе необходимо создать базу данных (массив структур) из N записей (N

– определяется при работе программы), выполнить просмотр и поиск записи по

заданному критерию (вводится при работе программы). Поля структуры:

название лекарства, дата изготовления, срок годности.

struct Drug

{

public string name;

public DateTime manufactureDate;

public DateTime expiryDate;

}

class Pharmacy

{

static void Main(string[] args)

{

Console.Write("Введите количество записей в базе данных: ");

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

Drug[] drugs = new Drug[n];

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

{

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

drugs[i].name = Console.ReadLine();

Console.Write("Введите дату изготовления (в формате dd/mm/yyyy): ");

drugs[i].manufactureDate = DateTime.Parse(Console.ReadLine());

Console.Write("Введите срок годности (в месяцах): ");

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

drugs[i].expiryDate = drugs[i].manufactureDate.AddMonths(expiryMonths);

}

Console.WriteLine("Список всех записей:");

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

{

Console.WriteLine("Лекарство: {0}", drugs[i].name);

Console.WriteLine("Дата изготовления: {0}", drugs[i].manufactureDate.ToShortDateString());

Console.WriteLine("Срок годности: {0}", drugs[i].expiryDate.ToShortDateString());

Console.WriteLine();

}

Console.Write("Введите критерий поиска (название лекарства): ");

string searchTerm = Console.ReadLine().ToLowerInvariant();

bool found = false;

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

{

if (drugs[i].name.ToLowerInvariant().Contains(searchTerm))

{

Console.WriteLine("Лекарство: {0}", drugs[i].name);

Console.WriteLine("Дата изготовления: {0}", drugs[i].manufactureDate.ToShortDateString());

Console.WriteLine("Срок годности: {0}", drugs[i].expiryDate.ToShortDateString());

Console.WriteLine();

found = true;

}

}

if (!found)

{

Console.WriteLine("Запись с таким критерием не найдена.");

}

Console.ReadLine();

}

}

9

Динамические списки

Пример задания:

Разработать программу, реализующую работу с линейным списком. В

программе необходимо создать базу данных (список) из N записей (N –

определяется при работе программы), выполнить просмотр, поиск записи по

заданному критерию. Картинная галерея. Ведётся учёт экспонатов галереи:

наименование картины, художник, цена.

class Painting

{

public string Name { get; set; }

public string Artist { get; set; }

public decimal Price { get; set; }

}

class Program

{

static void Main(string[] args)

{

LinkedList<Painting> gallery = new LinkedList<Painting>();

int n;

// запрашиваем количество записей в базе данных

Console.WriteLine("Введите количество записей в базе данных:");

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

// заполняем базу данных случайными записями

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

{

Painting painting = new Painting();

Console.WriteLine("Заполните информацию о картинах:");

Console.WriteLine("Наименование картины:");

painting.Name = Console.ReadLine();

Console.WriteLine("Художник:");

painting.Artist = Console.ReadLine();

Console.WriteLine("Цена:");

painting.Price = decimal.Parse(Console.ReadLine());

gallery.AddLast(painting);

}

// выводим все записи в базе данных

Console.WriteLine("База данных картин:");

foreach (Painting painting in gallery)

{

Console.WriteLine("{0}, {1}, {2}", painting.Name, painting.Artist, painting.Price);

}

// ищем запись по заданному критерию (например, по художнику)

Console.WriteLine("Поиск картины по художнику:");

string artist = Console.ReadLine();

LinkedListNode<Painting> node = gallery.First;

while (node != null)

{

if (node.Value.Artist == artist)

{

Console.WriteLine("Найдена картина: {0}, {1}, {2}", node.Value.Name, node.Value.Artist, node.Value.Price);

break;

}

node = node.Next;

}

}

}

Классы

Пример задания:

Разработать класс для определения одномерных массивов строк фиксированной длины. В классе определить методы создания массива, просмотра и поиска.

using System;

FixedLengthStringArray myArray = new FixedLengthStringArray(3);

myArray.CreateArray(new string[] { "one", "two", "three" });

myArray.ViewArray(); // Выводит "one, two, three"

Console.WriteLine(myArray.Search("two")); // Выводит "True"

Console.WriteLine(myArray.Search("four")); // Выводит "False"

class FixedLengthStringArray

{

private string[] array;

// Конструктор класса

public FixedLengthStringArray(int length)

{

array = new string[length];

}

// Метод создания нового массива

public void CreateArray(string[] values)

{

if (values.Length != array.Length)

{

throw new ArgumentException("Неверная длина массива");

}

values.CopyTo(array, 0);

}

// Метод просмотра массива

public void ViewArray()

{

Console.WriteLine(string.Join(", ", array));

}

// Метод поиска значения в массиве

public bool Search(string value)

{

return Array.IndexOf(array, value) != -1;

}

}

11.Динамические структуры Пример задания: Гостиница. Содержится следующая информация о проживающих в гостинице: ФИО клиента, номер комнаты, дата въезда, количество дней проживания, стоимость суточного проживания (зависит от категории номера).

using System;

using System.Collections.Generic;

namespace Hotel {

// Класс, описывающий информацию о проживающем

class Resident {

public string FIO { get; set; }

public int RoomNumber { get; set; }

public DateTime CheckInDate { get; set; }

public TimeSpan StayDuration { get; set; }

public decimal DailyCost { get; set; }

// Конструктор класса

public Resident(string fio, int roomNumber, DateTime checkInDate, TimeSpan stayDuration, decimal dailyCost) {

FIO = fio;

RoomNumber = roomNumber;

CheckInDate = checkInDate;

StayDuration = stayDuration;

DailyCost = dailyCost;

}

// Метод для расчета полной стоимости проживания

public decimal GetTotalCost() {

return DailyCost \* (decimal)StayDuration.TotalDays;

}

}

// Класс, описывающий гостиницу и ее функционал

class Hotel {

// Список жителей гостиницы

private List<Resident> residents = new List<Resident>();

// Метод для добавления нового жителя в гостиницу

public void AddResident(Resident resident) {

residents.Add(resident);

}

// Метод для поиска жителей по номеру комнаты

public List<Resident> FindResidentByRoomNumber(int roomNumber) {

List<Resident> foundResidents = new List<Resident>();

foreach (Resident resident in residents) {

if (resident.RoomNumber == roomNumber) {

foundResidents.Add(resident);

}

}

return foundResidents;

}

// Метод для поиска жителей по ФИО

public List<Resident> FindResidentByFIO(string fio) {

List<Resident> foundResidents = new List<Resident>();

foreach (Resident resident in residents) {

if (resident.FIO == fio) {

foundResidents.Add(resident);

}

}

return foundResidents;

}

// Метод для получения общей выручки гостиницы

public decimal GetTotalIncome() {

decimal totalIncome = 0;

foreach (Resident resident in residents) {

totalIncome += resident.GetTotalCost();

}

return totalIncome;

}

}

// Пример использования

class Program {

static void Main(string[] args) {

Hotel hotel = new Hotel();

// Добавляем жителей гостиницы

hotel.AddResident(new Resident("Иванов Иван Иванович", 101, new DateTime(2020, 1, 1), TimeSpan.FromDays(10), 1000));

hotel.AddResident(new Resident("Петров Петр Петрович", 101, new DateTime(2020, 1, 5), TimeSpan.FromDays(7), 1000));

hotel.AddResident(new Resident("Сидоров Сидор Сидорович", 201, new DateTime(2020, 1, 2), TimeSpan.FromDays(5), 1500));

hotel.AddResident(new Resident("Кузнецова Екатерина Александровна", 202, new DateTime(2020, 1, 3), TimeSpan.FromDays(3), 2000));

// Поиск жителей по номеру комнаты

List<Resident> residentsInRoom101 = hotel.FindResidentByRoomNumber(101);

Console.WriteLine("Жители комнаты 101:");

foreach (Resident resident in residentsInRoom101) {

Console.WriteLine("{0}, проживает {1} дней, общая стоимость проживания: {2}",

resident.FIO, resident.StayDuration.TotalDays, resident.GetTotalCost());

}

// Поиск жителей по ФИО

List<Resident> residentsWithNameIvanov = hotel.FindResidentByFIO("Иванов Иван Иванович");

Console.WriteLine("Жители с ФИО Иванов Иван Иванович:");

foreach (Resident resident in residentsWithNameIvanov) {

Console.WriteLine("{0}, проживает {1} дней, общая стоимость проживания: {2}",

resident.FIO, resident.StayDuration.TotalDays,

}

}

12.Многомерные массивы Пример задания: Дана квадратная матрица A[N, N]. Записать на место отрицательных элементов матрицы нули, а на место положительных — единицы. Вывести на печать нижнюю треугольную матрицу в общепринятом виде.

using System;

class Program

{

static void Main(string[] args)

{

const int N = 4;

int[,] a = new int[N, N] { { 1, -2, 3, -4 }, { -5, 6, -7, 8 }, { 9, -10, 11, -12 }, { -13, 14, -15, 16 } };

Console.WriteLine("Матрица до замены отрицательных на 0 и положительных на 1:");

PrintMatrix(a);

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

{

for (int j = 0; j < N; j++)

{

if (a[i, j] < 0)

{

a[i, j] = 0;

}

else

{

a[i, j] = 1;

}

}

}

Console.WriteLine("Матрица после замены:");

PrintMatrix(a);

Console.WriteLine("Нижняя треугольная матрица:");

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

{

for (int j = 0; j < N; j++)

{

if (i >= j)

{

Console.Write(a[i, j] + " ");

}

else

{

Console.Write(" ");

}

}

Console.WriteLine();

}

}

static void PrintMatrix(int[,] a)

{

for (int i = 0; i < a.GetLength(0); i++)

{

for (int j = 0; j < a.GetLength(1); j++)

{

Console.Write(a[i, j] + " ");

}

Console.WriteLine();

}

}

}

13 Дана символьная строка. Получить новую строку, взяв из данной все символы, находящиеся между первой открывающейся скобкой и последней закрывающейся (если какие-либо скобки отсутствует, то вывести соответствующее сообщение).

string str = "Это (текст) для";

int startIndex = str.IndexOf("(");

int endIndex = str.LastIndexOf(")");

if (startIndex == -1 || endIndex == -1)

{

Console.WriteLine("Скобки отсутствуют");

}

else if (startIndex >= endIndex)

{

Console.WriteLine("Некорректные скобки");

}

else

{

string result = str.Substring(startIndex + 1, endIndex - startIndex - 1);

Console.WriteLine(result);

}

14 Блок-схемы Пример задания: Дана последовательность Х1, Х2, …, Х20. Заменить в ней отрицательные значения элементов на их квадраты

