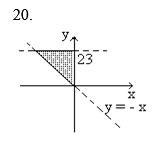
Отчет по практическим занятиям по дисциплине “Структуры и алгоритмы компьютерной обработки данных”

Акимов А.А. 2 курс, 241 группа

Практикум №3

I. Дана точка на плоскости с координатами (х, у). Составить программу, которая выдает одно из сообщений «Да», «Нет», «На границе» в зависимости от того, лежит ли точка внутри заштрихованной области, вне заштрихованной области или на ее границе. Области задаются графически следующим образом:



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Example

{

class Program

{

static void Main()

{

Console.Write("x=");

double x = double.Parse(Console.ReadLine());

Console.Write("y=");

double y = double.Parse(Console.ReadLine());

if ((y > -x) && (x < 0) && (y < 23)) Console.WriteLine("Точка лежит внутри области");

if ((y < -x) || (x > 0) || (y > 23)) Console.WriteLine("Точка не лежит внутри области");

else Console.WriteLine("Точка лежит на границе");

}

}

}

Входные данные: x=1 y=1

Выходные данные: Точка не лежит внутри области

II. Составить программу:

8.Проведен тест, оцениваемый в целочисленный баллах от нуля до ста. Вывести на экран оценку тестируемого в зависимости от набранного количества баллов: от 90 до 100 – «отлично», от 70 до 89 – «хорошо», от 50 до 69 – «удовлетворительно», менее 50 – «неудовлетворительно».

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Example

{

class Program

{

static void Main()

{

Console.Write("Введите баллы ");

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

if ((m < 0) || (m > 100))

{

Console.WriteLine("Данные введены неверно");

}

else

{

if (m >= 90) Console.WriteLine("Оценка: отлично ");

else if (m >= 70) Console.WriteLine("Оценка: хорошо ");

else if (m >= 50) Console.WriteLine("Оценка: удовлетворительно ");

else Console.WriteLine("Оценка: неудовлетворительно ");

}

}

}

}

Входные данные: Введите баллы 75

Выходные данные: Оценка: хорошо

III. Вывести на экран:

20) все трехзначные числа, которые начинаются и заканчиваются на одну и ту же цифру;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Example

{

class Program

{

static void Main()

{

int m = 100;

while (m <= 999)

{

if ((m % 10 == m / 100) || ((m / 10) % 10 == m / 100) || ((m / 10) % 10 == m % 10))

Console.WriteLine(m);

m++;

}

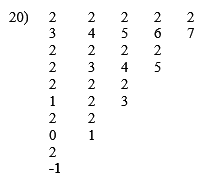
}

}

}

Выходные данные: 101 … 999

IV. Вывести на экран числа следующим образом:



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int y = 3;

int z = 5;

for (int i = 0; i < 5; i++)

{

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

Console.Write("2 ");

Console.WriteLine();

for (int t = 0; t < z; t++)

Console.Write("{0} ", y + t);

Console.WriteLine();

y--;

z--;

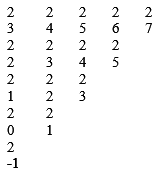
}

}

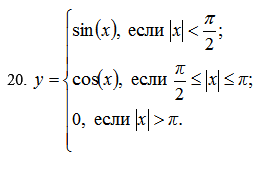
}

}

Выходные данные:



V. Постройте таблицу значений функции y=f(x) для х∈[a, b] с шагом h.



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

Console.Write("введите начальную точку отрезка ");

double a = double.Parse(Console.ReadLine());

Console.Write("введите конечную точку отрезка ");

double b = double.Parse(Console.ReadLine());

Console.Write("введите шаг ");

double h = double.Parse(Console.ReadLine());

double pi = 3.14;

for (double x = a; x <= b; x = x + h)

{

if (Math.Abs(x) < pi / 2)

{

Console.WriteLine("y({0:f2})={1:f2}", x, Math.Sin(x));

}

else if ((Math.Abs(x) >= pi / 2) && (Math.Abs(x) <= pi))

{

Console.WriteLine("y({0:f2})={1:f2}", x, Math.Cos(x));

}

else

{

Console.WriteLine("y({0})={1:f2}", x, 0);

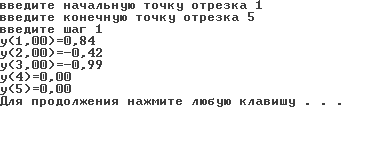
}

}

}

}

}



Практикум №5

II Разработать функцию, которая для заданных натуральных чисел N и N возвращает их

20) найти наибольший общий делитель для n натуральных чисел.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Практикум\_5\_3\_20

{

class Program

{

static int F(int a, int b)

{

if (b == 0)

return a;

else

return F(b, a % b);

}

static void Main()

{

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

int[] a = new int[n];

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

a[i]=int.Parse(Console.ReadLine());

int ans = a[0];

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

{

ans = F(ans, a[i]);

}

Console.WriteLine("{0}", ans);

}

}

}

Входные данные:

5

10

15

20

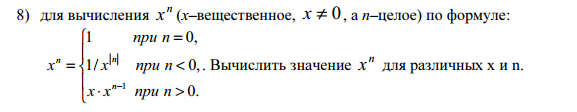
25

30

Выходные данные:

5

IV. Разработать рекурсивный метод, возвращающий значение:



using System;

namespace \_5.\_4.\_9

{

class Program

{

static double recurcy(int n, int x)

{

if (n == 0)

return 1;

else if (n < 0)

return (1/(double)Math.Pow(x, Math.Abs(n)));

else

return (x \* recurcy(n - 1, x));

}

static void Main()

{

Console.Write("n=");

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

Console.Write("x=");

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

double z = recurcy(n,x);

Console.WriteLine("{0}", z);

}

}

}

Входные данные: n=5 x=2

Выходные данные: 32

V. Разработать рекурсивный метод, не возвращающий значений:

8) Разработать рекурсивный метод для вывода на экран цифр натурального числа в прямом порядке. Применить этот метод ко всем числам из интервала от А до В.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication2

{

class Program

{

static void extent(int a)

{

int q = a % 10;

a = a / 10;

if (a != 0)

{

extent(a);

}

Console.Write("{0} ", q);

}

static void Main()

{

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

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

while(a<=n)

{

extent(a);

Console.WriteLine();

a++;

}

}

}

}

Входные данные: 101 110

Выходные данные:



Практикум №6

III. Дан массив размером n×n, элементы которого целые числа.

20. В каждом столбце найти минимум и заменить его нулем.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace \_6.\_3.\_20

{

class Program

{

static void Print(int[,] a)

{

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

{

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

{

Console.Write("{0,5} ", a[i, j]);

}

Console.WriteLine();

}

}

static void Input(out int[,] a)

{

Console.Write("n= ");

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

a = new int[n, n];

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

{

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

{

Console.Write("a[{0},{1}]= ", i, j);

a[i, j] = int.Parse(Console.ReadLine());

}

}

}

static void F(int[,] a)

{

int min = 0;

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

{

min = a[0,j];

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

{

if (a[i, j] < min)

min = a[i, j];

}

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

{

if (a[i, j] == min)

{

a[i, j] = 0;

}

}

}

}

static void Main()

{

int[,] a;

Input(out a);

F(a);

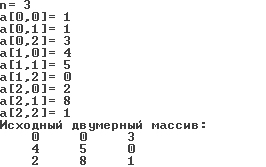
Console.WriteLine("Исходный двумерный массив:");

Print(a);

}

}

}



IV. Дан массив размером n×n, элементы которого целые числа.

20. Для каждого столбца найти номер первой пары одинаковых элементов. Данные записать в новый массив.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace \_6.\_4.\_20

{

class Program

{

static void Print(double[] a)

{

foreach (double elem in a)

{

Console.Write("{0} ", elem);

}

Console.WriteLine();

}

static void Print(int[,] a)

{

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

{

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

{

Console.Write("{0,5} ", a[i, j]);

}

Console.WriteLine();

}

}

static void Input(out int[,] a)

{

Console.Write("n= ");

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

a = new int[n, n];

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

{

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

{

Console.Write("a[{0},{1}]= ", i, j);

a[i, j] = int.Parse(Console.ReadLine());

}

}

}

static double[] F(int[,] a)

{

double[] b = new double[a.GetLength(1)];

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

{

b[j] = -1;

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

{

if (a[i, j] == a[i+1, j])

{

b[j] = i;

break;

}

}

}

return b;

}

static void Main()

{

int[,] a;

Input(out a);

Console.WriteLine("Исходный двумерный массив:");

Print(a);

double[] b = F(a);

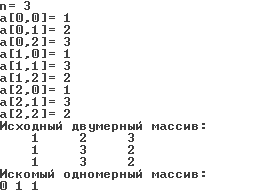
Console.WriteLine("Искомый одномерный массив:");

Print(b);

}

}

}



V. В одномерном массиве, элементы которого – целые числа, произвести следующие действия:

8. Вставить новый элемент перед всеми четными элементами.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace \_6.\_5.\_8

{

class Program

{

static void Main(string[] args)

{

int[] a;

Console.Write("n= ");

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

Console.Write("k= ");

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

a = new int[n];

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

{

Console.Write("A[{0}]= ", i);

a[i] = int.Parse(Console.ReadLine());

}

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

{

if (a[i] % 2 == 0)

{

Console.Write("{0} {1} ", k, a[i]);

}

else

{

Console.Write("{0} ", a[i]);

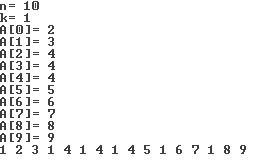
}

}

}

}

}



Практикум №8

III. Дана строка, в которой содержится осмысленное текстовое сообщение. Слова сообщения разделяются пробелами и знаками препинания. 12. Найти все самые длинные слова сообщения.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

namespace \_8.\_3.\_12

{

class Program

{

static void Main()

{

Console.WriteLine("Введите сообщение");

string[]vvod = Console.ReadLine().Split(' ','.',',','!','?');

int max=0;

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

{

if (vvod[i].Length > max)

{

max = vvod[i].Length;

}

}

Console.WriteLine("Самое(-ые) длинное(-ые) слово(-а): ");

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

{

if (vvod[i].Length == max)

{

Console.WriteLine(" {0} ", vvod[i]);

}

}

}

}

}

C:\Users\1\Desktop\Снимок.PNG

Практикум №9

I. Работа с символьными потокам.

7. Дан текстовый файл. Выяснить, имеется ли в нем строка, которая начинается с данной буквы. Если да, то напечатать ее.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

namespace \_9.\_1.\_7

{

class Program

{

static void Main()

{

Console.WriteLine("Введите первый символ");

Console.WriteLine();

char a = char.Parse(Console.ReadLine());

Console.WriteLine("Результат");

foreach (string s in File.ReadAllLines("text.txt"))

{

if (s[0]==a)

Console.WriteLine(s);

}

}

}

}

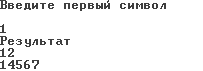
Входной файл:

12

2345

14567

789



II. Работа с символьными потоками

7. Даны два файла с числами. Получить новый файл, записав в него сначала все положительные числа из первого файла, потом все отрицательные числа из второго. using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace ans

{

class Program

{

static void Main(string[] args)

{

using (StreamReader fileIn = new StreamReader("f.txt", Encoding.GetEncoding(1251)))

{

using (StreamReader fileInn = new StreamReader("g.txt", Encoding.GetEncoding(1251)))

{

using (StreamWriter fileOut = new StreamWriter("h.txt", false))

{

string line1;

int n1 = 0;

while ((line1 = fileIn.ReadLine()) != null)

{

char[] mas1 = { ' ', '\n' };

string[] lines1 = line1.Split(mas1, StringSplitOptions.RemoveEmptyEntries);

for (int i = 0; i < lines1.Length; i++)

{

string p2 = lines1[i];

n1 = int.Parse(p2);

if (n1 > 0)

fileOut.WriteLine(n1);

}

}

string line2;

int n2 = 0;

while ((line2 = fileInn.ReadLine()) != null)

{

char[] mas2 = { ' ', '\n' };

string[] lines2 = line2.Split(mas2, StringSplitOptions.RemoveEmptyEntries);

for (int i = 0; i < lines2.Length; i++)

{

string p2 = lines2[i];

n2 = int.Parse(p2);

if (n2 < 0)

fileOut.WriteLine(n2);

}

}

}

}

}

}

}

}

Входной файл 1:

12 -1 -2 3 -4

2345

14567

789

Входной файл 2:

-1

-2

4

-4

Выходной файл:

12

3

2345

14567

789

-1

-2

-4

Практикум №10

Задание 2. Создать класс Triangle, содержащий следующие члены класса:

1. Поля:

• int a, b, c;

2. Конструктор, позволяющий создать экземпляр класса с заданными длинами сторон.

3. Методы, позволяющие:

• вывести длины сторон треугольника на экран;

• расчитать периметр треугольника;

• расчитать площадь треугольника.

4. Свойство:

• позволяющее получить-установить длины сторон треугольника (доступное для чтения

и записи);

• позволяющее установить, существует ли треугольник с данными длинами сторон

(доступное только для чтения).

5. Индексатор, позволяющий по индексу 0 обращаться к полю a, по индексу 1 – к полю b,

по индексу 2 – к полю c, при других значениях индекса выдается сообщение об ошибке.

6. Перегрузку:

• операции ++ (--): одновременно увеличивает (уменьшает) значение полей a, b и c на 1;

• констант true и false: обращение к экземпляру класса дает значение true, если

треугольник с заданными длинами сторон существует, иначе false;

• операции \*: одновременно домножает поля a, b и c на скаляр.

Продемонстрировать работу класса.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Pr\_10

{

class Line

{

double a;

double b;

double c;

Line(double a, double b, double c)

{

this.a = a;

this.b = b;

this.c = c;

if ((a >= b + c) || (b >= a + c) || (c >= a + b))

{

this.a = 1;

this.b = 1;

this.c = 1;

}

}

Line(Line T)

{

this.a = T.a;

this.b = T.b;

this.c = T.c;

}

public double Perimetr()

{

double perimetr = a + b + c;

return perimetr;

}

public double Square()

{

double half = (a + b + c) / 2;

double square = Math.Sqrt(half \* (half - a) \* (half - b) \* (half - c));

return square;

}

public void Change(double x, double y, double z)

{

if ((a < y + z) && (y < a + z) && (z < y + a))

{

this.b = y;

this.c = z;

}

else if ((x < b + z) && (b < x + z) && (z < b + x))

{

this.a = x;

this.c = z;

}

else if ((x < y + c) && (y < x + c) && (c < y + x))

{

this.a = x;

this.b = y;

}

else if ((x < y + z) && (y < x + z) && (z < y + x))

{

this.a = x;

this.b = y;

this.c = z;

}

}

public void Show()

{

Console.WriteLine("a={0} b={1} c={2}", a, b, c);

}

public double A

{

get

{

return a;

}

set

{

if ((value < b + c) && (b < value + c) && (c < value + b))

a = value;

}

}

public double B

{

get

{

return b;

}

set

{

if ((a < value + c) && (value < a + c) && (c < a + value))

b = value;

}

}

public double C

{

get

{

return c;

}

set

{

if ((a < b + value) && (b < a + value) && (value < a + b))

c = value;

}

}

public double Existance

{

get

{

if ((a < b + c) && (b < a + c) && (c < a + b))

return 1;

else

return 0;

}

}

public double this[int i]

{

get

{

if (i == 0)

{

return a;

}

else

{

if (i == 1)

{

return b;

}

else

{

if (i == 2)

{

return c;

}

else

{

Console.WriteLine("Недопустимые длины сторон");

return 0;

}

}

}

}

}

public static Line operator ++(Line pnt)

{

Line tmp = new Line(pnt);

tmp.a += 1;

tmp.b += 1;

tmp.c += 1;

return tmp;

}

public static Line operator --(Line pnt)

{

Line tmp = new Line(pnt);

tmp.a -= 1;

tmp.b -= 1;

tmp.c -= 1;

return tmp;

}

public static Line operator \*(Line t, double scal)

{

Line tmp = new Line(t);

tmp.a \*= scal;

tmp.b \*= scal;

tmp.c \*= scal;

return tmp;

}

public static bool operator true(Line t)

{

if ((t.a < t.b + t.c) && (t.b < t.a + t.c) && (t.c < t.a + t.b))

{

return true;

}

else

{

return false;

}

}

public static bool operator false(Line t)

{

if ((t.a >= t.b + t.c) || (t.b >= t.a + t.c) || (t.c >= t.a + t.b))

{

return false;

}

else

{

return true;

}

}

static void Main()

{

Console.WriteLine("Ввод сторон");

string[] line = Console.ReadLine().Split(' ');

double k = int.Parse(line[0]);

double m = int.Parse(line[1]);

double l = int.Parse(line[2]);

Line ln = new Line(k, m, l);

Console.WriteLine("Конструктор с заданными длинами сторон");

ln.Show();

Console.WriteLine("Установление длин сторон через свойства");

ln.A = 3;

ln.B = 3;

ln.C = 3;

ln.Change(ln.A, ln.B, ln.C);

ln.Show();

ln.Change(7, 7, ln.C);

ln.Show();

ln.A = 2;

ln.B = 2;

ln.C = 2;

Console.WriteLine("Установление длин сторон через свойства");

ln.Show();

Console.WriteLine("Инкремент");

ln++;

ln.Show();

Console.WriteLine("Дикремент");

ln--;

ln.Show();

Console.WriteLine("индексатор, сторона по индексу");

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

Console.WriteLine(ln[y]);

Console.WriteLine("Периметр {0}", ln.Perimetr());

Console.WriteLine("Площадь {0}", ln.Square());

Console.WriteLine("умножение на скаляр");

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

ln = ln \* x;

ln.Show();

Console.WriteLine("Проверка на существование");

if (ln)

{

Console.WriteLine("Существвует");

}

else

{

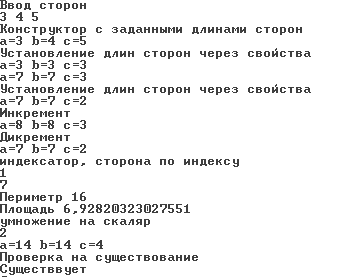
Console.WriteLine("Не существует");

}

}

}

}



Практикум №12, 13

Задание 1

1. Создать абстрактный класс Figure с методами вычисления площади и периметра, а также

методом, выводящим информацию о фигуре на экран.

2. Создать производные классы: Rectangle (прямоугольник), Circle (круг), Triangle

(треугольник) со своими методами вычисления площади и периметра.

3. Создать массив n фигур и вывести полную информацию о фигурах на экран. 4. В абстрактном классе Figure реализовать метод CompareTo так, чтобы можно было отсортировать объекты по их площадям.

FILE.txt

circle 5

rectangle 2 2

circle 2

triangle 7 8 9

Triangle.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Figure

{

class Triangle : Figure

{

int a, b, c;

public Triangle(int a, int b, int c)

{

this.a = a;

this.c = c;

this.b = b;

}

public override double Perimeter()

{

if (a < b + c && b < a + c && c < b + a)

{

return a + b + c;

}

else return 0;

}

public override double Area()

{

double p = Perimeter() / 2;

if (a < b + c && b < a + c && c < b + a)

{

return Math.Sqrt(p \* (p - a) \* (p - b) \* (p - c));

}

else return 0;

}

public override void ShowInfo()

{

Console.WriteLine("Треугольник со сторонами {0}, {1}, {2}", a, b, c);

}

}

}

Rectangle.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Figure

{

class Rectangle : Figure

{

int a, b;

public Rectangle(int a, int b)

{

this.a = a;

this.b = b;

}

public override double Area()

{

return a \* b;

}

public override double Perimeter()

{

return (a + b) \* 2;

}

public override void ShowInfo()

{

Console.WriteLine("Прямоугольник со сторонами {0} и {1}", a, b);

}

}

}

Circle.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Figure

{

class Circle : Figure

{

const double pi = 3.14;

int r;

public Circle(int r)

{

this.r = r;

}

public override double Area()

{

return pi \* r \* r;

}

public override double Perimeter()

{

return 2 \* pi \* r;

}

public override void ShowInfo()

{

Console.WriteLine("Круг радиусом {0}.", r);

}

}

}

Figure.cs

using System;

using System.IO;

namespace Figure

{

abstract class Figure : IComparable

{

public abstract double Area();

public abstract double Perimeter();

public abstract void ShowInfo();

public int CompareTo(object figure)

{

Figure Expample = (Figure)figure;

if (this.Area() == Expample.Area())

{

return 0;

}

else

{

if (this.Area() > Expample.Area())

{

return 1;

}

else

{

return -1;

}

}

}

}

}

Program.cs

using System;

using System.IO;

namespace Figure

{

class Program

{

static void Main()

{

Figure[] figures = GetFigures();

Array.Sort(figures);

foreach (Figure figure in figures)

{

Console.WriteLine();

figure.ShowInfo();

Console.WriteLine("S: {0}", figure.Area());

Console.WriteLine("P: {0}", figure.Perimeter());

}

Console.ReadKey();

}

public static Figure[] GetFigures()

{

string[] s = ReadFile().Split('\n');

Figure[] figures = new Figure[s.Length];

int a, b, c, r, i = 0, n = 0;

while (i < s.Length)

{

string[] str = s[i].Split(' ');

if (str[0] == "rectangle")

{

a = Convert.ToInt32(str[1]);

b = Convert.ToInt32(str[2]);

figures[n] = new Rectangle(a, b);

n++;

}

if (str[0] == "circle")

{

r = Convert.ToInt32(str[1]);

figures[n] = new Circle(r);

n++;

}

if (str[0] == "triangle")

{

a = Convert.ToInt32(str[1]);

b = Convert.ToInt32(str[2]);

c = Convert.ToInt32(str[3]);

figures[n] = new Triangle(a, b, c);

n++;

}

i++;

}

return figures;

}

public static string ReadFile()

{

string str = string.Empty;

StreamReader fig = new StreamReader(@"C:\Users\1\Documents\Projects\file.txt");

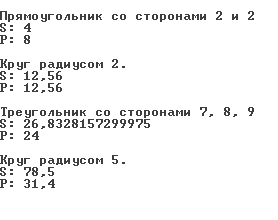
str = fig.ReadToEnd();

fig.Close();

return str;

}

}}



Практикум №18

9. В сообщении может содержаться дата в формате дд.мм.гггг. В заданном формате дд –

целое число из диапазона от 1 до 31, мм – целое число из диапазона от 1 до 12, а гггг –

целое число из диапазона от 1900 до 2010 (если какая-то часть формата нарушена, то

данная подстрока в качестве даты не рассматривается). Выведите на экран все даты,

которые относятся к текущему году.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.IO;

namespace \_18.\_9

{

class Program

{

static void Main()

{

string p = @"((^|[\s])((((0[1-9]|[1-2][0-9])\.([0][2]))|((0[1-9]|[1-2][0-9]|[3][0])\.(0[468]|([1][1])))|(0[1-9]|[1-2][0-9]|3[0-1])\.(0[13579]|([1][0]])|([1][2])))\.((19[0-9][0-9])|(2010)|(200[0-9])))($|[\s]))";

Regex r = new Regex(p, RegexOptions.IgnoreCase);

string line;

using (StreamReader inStream = new StreamReader(@"C:\Users\1\Documents\ФЛЕШКА/f.txt", Encoding.GetEncoding(1251)))

{

while ((line = inStream.ReadLine()) != null)

{

Match tel = r.Match(line);

while (tel.Success)

{

string s = tel.Value;

if ((char.IsWhiteSpace(s[0])) && (!char.IsDigit(s[s.Length - 1])))

{

s = s.Remove(0, 1);

s = s.Remove(s.Length - 1, 1);

Console.Write(s);

Console.WriteLine();

}

else if ((!char.IsDigit(s[s.Length - 1])))

{

s = s.Remove(s.Length - 1, 1);

Console.Write(s);

Console.WriteLine();

}

else if (char.IsWhiteSpace(s[0]))

{

s = s.Remove(0, 1);

Console.Write(s);

Console.WriteLine();

}

else

{

Console.Write(s);

Console.WriteLine();

}

tel = tel.NextMatch();

}

}

}

}

}

}

Входные данные:

дата рождения 22212.12.1999

20.12.2010

Ntv&&& thr20.02.2010,srggrg 13.12.2010,

12.12.2011

Выходные данные:

20.12.2010

10. В сообщении могут содержаться IP-адреса компьютеров в формате d.d.d.d, где

d – целое число из диапазона от 0 до 255. Вывести все IP-адреса содержащиеся в тексте.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.IO;

namespace \_18.\_10

{

class Program

{

static void Main()

{

string p = @"(^|\s)(([0-9]|[1-9]\d|1\d\d|2[0-4]\d|25[0-5])\.){3}" + @"([0-9]|[1-9]\d|1\d\d|2[0-4]\d|25[0-5])($|\s)";

Regex r = new Regex(p, RegexOptions.IgnoreCase);

string line;

using (StreamReader inStream = new StreamReader(@"C:\Users\1\Documents\ФЛЕШКА/h.txt", Encoding.GetEncoding(1251)))

{

while ((line = inStream.ReadLine()) != null)

{

Match tel = r.Match(line);

while (tel.Success)

{

string s = tel.Value;

if ((char.IsWhiteSpace(s[0])) && (!char.IsDigit(s[s.Length - 1])))

{

s = s.Remove(0,1);

s = s.Remove(s.Length-1, 1);

Console.Write(s);

Console.WriteLine();

}

else if ((!char.IsDigit(s[s.Length - 1])))

{

s = s.Remove(s.Length - 1, 1);

Console.Write(s);

Console.WriteLine();

}

else if (char.IsWhiteSpace(s[0]))

{

s = s.Remove(0, 1);

Console.Write(s);

Console.WriteLine();

}

else

{

Console.Write(s);

Console.WriteLine();

}

tel = tel.NextMatch();

}

}

}

}

}

}

Входные данные:

упа 00.0.0.0

257.2.2.2

255.2.2.2

1111.2.2.2

1.1.1.11111

nklnlkjnl .11.11.11.11

Выходные данные:

255.2.2.2