Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С.Галущака»

|  |  |  |
| --- | --- | --- |
| УТВЕРЖДАЮ  председатель цикловой  комиссии УГС 09.00.00  \_\_\_\_\_\_\_\_\_\_\_\_ /О.О.Чекушкина/  Протокол № \_ от «\_\_\_» \_\_\_ 20\_\_г. | УТВЕРЖДАЮ  председатель цикловой  комиссии УГС 09.00.00  \_\_\_\_\_\_\_\_\_\_\_\_ /О.О.Чекушкина/  Протокол № \_ от «\_\_\_» \_\_\_ 20\_\_г. | УТВЕРЖДАЮ  председатель цикловой  комиссии УГС 09.00.00  \_\_\_\_\_\_\_\_\_\_ /О.О.Чекушкина/  Протокол № \_ от «\_\_\_» \_\_\_ 20\_\_г. |

**Разрабатывать программные модули в соответствии с заданием.**

**Отладка и тестирование программных модулей**

Методические указания к практическим занятиям 12-17

Учебная дисциплина: Основы алгоритмизации и программирование

Специальность: 09.02.07 Информационные системы и программирование

Разработал:

Мельникова Т.Ф.

2020

1. **Цели**
   1. В ходе выполнения работы студенты осваивают:
      1. Общие компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности

ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста

ОК 09. Использовать информационные технологии в профессиональной деятельности

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.

1.1.2 Профессиональные компетенции:

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием

ПК.1.3. Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4. Выполнять тестирование программных модулей.

1.2.2 Осваивают умения:

* работать в среде программирования.
* реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.
* оформлять код программы в соответствии со стандартом кодирования.
* выполнять проверку, отладку кода программы.

1.2.3 Усваивают знания:

- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.

- подпрограммы, составление библиотек подпрограмм

- объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения

1. **Оборудование, дидактическое обеспечение**

2.1 Исходные данные согласно варианта задания;

2.2 Конспект лекций по МДК;

2.3 Рабочее место за компьютером

**3 Форма организации –** индивидуальная

**4 Инструктаж**

4.1 Вариант задания выбирается по порядковому номеру в журнале

4.2 Работа выполняется в компьютерной лаборатории и сдается на проверку преподавателю с предоставлением разработанных тестов

4.3 Отчет оформляется во время проведения практического занятия или как внеаудиторная самостоятельная работа студента

**5 Порядок выполнения**

5.1Ознакомиться с постановкой задачи

5.2 Разработать математическую, логическую модель решения задачи

5.3 Разработать контрольный пример, который будет предусматривать проход по каждой ветви алгоритма.

5.4 Составить программный код на языке C#

5.5 Войти в среду Visual Studio С#. Создать проект новый ( консольное приложение )

5.6 Выполнить работу по набору текста программы

5.7Сохранить файл в своей папке.

5.8 Запустить компиляцию программы. Добиться успешной отладки.

5.9 Запустить программу на выполнение. Получить результаты. Провести анализ результатов в соответствии с тестами.

**6 Методические рекомендации**

**Составить отчет для защиты заданий по форме**

1. Постановка задачи.
2. Описание объектов программы

Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование объекта | Условное обозначение | Тип | Назначение |
|  |  |  |  |

1. Тесты ( в зависимости от условия задачи)
2. Алгоритм решения задачи
3. Текст программы
4. Результаты работы ( скриншоты)
5. Выводы

**Практическое занятие 12-13**

**Организация процедур. Организация функций.**

**Цель работы:**

1. Научиться составлять прогаммы и тесты, отлаживать модульные программы с использованием функций.

2. Изучить механизм передачи параметров в функциях.

**Краткие сведения из теории**

Функция – это последовательность операторов, оформленная таким образом, что ее можно вызвать по имени из любого места программы. Функция описывается следующим образом:

*[модификатор] тип\_ возвращаемого\_ значения* ***имя\_ функции*** ([тип параметра имя параметра][,…])

{

*тело функции*

}

Первая строка описания называется *заголовком функции*. Тип возвращаемого значения может быть любым. Допустимо не возвращать никакого значения (тип **void**). В С# не допускается вложение функций друг в друга.

Выход из функции осуществляется следующими способами:

1. Если нет необходимости возвращать вычисленное значение, то выход осуществляется по достижении закрывающей скобки или при выполнении оператора **return**.

2. Если необходимо вернуть определенное значение, то выход осуществляется оператором

**return** *выражение*;

Передача параметров

При работе важно соблюдать следующее правило: при объявлении и вызове функции параметры должны соответствовать по количеству, порядку следования и типам. Функция может не иметь параметров, в этом случае после имени функции обязательно ставятся круглые скобки. Существует следующие способы передачи параметров: передача по значению, ссылке.

*Передача параметров по значению*

В момент обращения к функции в памяти создаются временные переменные с именами, указанными в списке формальных параметров. В эти временные переменные копируются значения фактических параметров. Пример

static int Max(int a, int b) // метод выбора максимального значения

{

if ( a > b ) return a;

else return b;

}

static void Main()

{

int a = 2, b = 4;

int x = Max( a, b ); // вызов метода Max

Console.WriteLine( x ); // результат: 4

short t1 = 3, t2 = 4;

int y = Max( t1, t2 ); // вызов метода Max

Console.WriteLine( y ); // результат: 4

int z = Max( a + t1, t1 / 2 \* b ); // вызов метода Max

Console.WriteLine( z ); // результат: 5

}

**Порядок работы**

1. Разработать алгоритм и тесты, написать и отладить модульную программу для решения математических задач (табл.11). Решение выполнить двумя способами: с использованием метода-функции и метода-процедуры. Предусмотреть перегрузку метода. Выполнить программу, проанализировать результаты в соответствии с тестами.

**Таблица 11**

|  |  |
| --- | --- |
| Номер варианта | Условие |
| 1 | Вычислить число сочетаний из n по m(n>m) по формуле: |
| 2 | Вычислить:  , где min – это минимальное из трех  значений |
| 3 | Вычислить: |
| 4 | Найти все трехзначные числа, равные сумме факториалов своих цифр. |
| 5 | Два треугольника заданы координатами своих вершин. Вычислить площади треугольников с помощью формулы Герона и определить, какой треугольник имеет большую площадь. |
| 6 | Решить уравнение ax+b=0, где . Значения k,l,m вводятся.  мax-это максимальное из трех чисел |
| 7 | Разработать метод , который вычисляет длину отрезка по координатам вершин  (x1, y1) и (x2,y2), и метод  max(a,b) , который вычисляет максимальное из чисел a и b. С помощью данных методов определить, какая из трех точек на плоскости наиболее удалена от начала координат. |
| 8 | Вычислить число сочетаний с повторениями по формуле: |
| 9 | Разработать метод , который возвращает вторую справа цифру натурального числа x. Вычислить с помощью него значение выражения z=f(a)+f(b)-f(c). |
| 10 | Разработать метод , который по длинам сторон треугольника x , y, z  вычисляет его площадь. С помощью данного метода по заданным вещественным числам a, b, c, d, e, f, g найти площадь пятиугольника, изображенного на рисунке. |
| 11 | Решить уравнение  , где , r,m,n вводятся.  где min – это минимальное из трех значений |
| 12 | Разработать метод, который для заданного натурального числа n и вещественного х находит значение выражения \cfrac{x^n}{n}. Вычислить с помощью данного метода значение выражения \cfrac{x^2}{2}+\cfrac{x^4}{4}+\cfrac{x^6}{6}. |
| 13 | Разработать метод , который в трехзначном числе меняет местами первую с последней цифрой, а остальные числа оставляет без изменения. |
| 14 | Вычислить формулу.    где min – это минимальное из двух значений |

2. Разработать алгоритм и тесты, написать и отладить модульную программу с использованием подпрограммы для организации обработки массивов (табл.12). Выполнить программу, проанализировать результат в соответствии с тестами.

**Таблица 12**

|  |  |
| --- | --- |
| Номер варианта | Условие |
| 1 | В массиве A(n) и B(n) все элементы, следующие за максимальным (за первым по порядку, если их несколько), заменить на 1. |
| 2 | Дана матрица A(n, m). Упорядочить столбцы по убыванию элементов. |
| 3 | Дана матрица X(n, m). Упорядочить строки по возрастанию элементов. |
| 4 | Дана матрица У(k, l). Получить из нее вектор, содержащий суммы элементов строк. Отсортировать его в порядке убывания элементов. |
| 5 | Заданы три матрицы Q(m, m), Z(n, n), C(k, k). Найти среднее геометрическое значение из всех минимальных по модулю элементов главных диагоналей матриц. |
| 6 | Вычислить:  ,где Xmax - максимальный элемент вектора X(n)  Уmax - максимальный элемент вектора У(m). |
| 7 | Вычислить суммы элементов векторов У(k) и X(L), находящиеся в интервале от а до b. |
| 8 | Вычислить: |
| 9 | Решить уравнение a\*x + b=0, где а - длина вектора У(n), b - длина вектора Z(m). |
| 10 | Решить уравнение, где а - произведение элементов целочисленного вектора Т(к), b - произведение элементов целочисленного вектора D(L), с - произведение элементов целочисленного вектора R(n). |
| 11 | Дана матрица У(к, к). Сформировать вектор, содержащий диагональные элементы матрицы У. Отсортировать его в порядке убывания элементов. |
| 12 | Решить уравнение  где р - минимальный элемент матрицы A(3, 3), q - минимальный элемент матрицы B(4, 4), r - минимальный элемент матрицы С(5, 5). |
| 13 | Заданы три матрицы A(m, m), B(n, n), C(k, k). Найти минимальный из всех максимальных элементов главных диагоналей матриц. |
| 14 | Вычислить: |
|  |  |

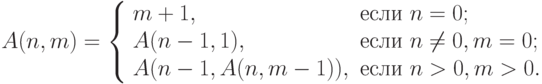
**Практическое занятие 14**

**Применение рекурсивных функций**

1. **Разработать рекурсивный метод (возвращающий значение)**:
   1. для вычисления n -го члена следующей последовательности b_{1} = -10, b_{2} = 2, b_{n+2} = |b_{n}| – 6 b_{n+1}.
   2. для вычисления n -го члена следующей последовательности b_{1} = 5, b_{n+1} = b_{n}/(n^{2}+n+1)
   3. для нахождения наибольшего общего делителя методом Евклида:

НОД(a,b)=\left\{\begin{array}{ll}
a, & \text{если } a=b;\\
НОД(a-b,b), &  \text{если } a > b;\\
НОД(a,b-a), &  \text{если } b > a;\end{array}

* 1. для вычисления значения функции Аккермана для неотрицательных чисел n и m. Функция Аккермана определяется следующим образом:



* 1. для вычисления числа сочетаний C(n, m)  , где 0 \le m \le n, используя следующие свойства

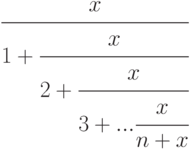
C_n^0 = C_n^n =1; C_n^m = C_{n-1}^{m} + C_{n-1}^{m-1} =1; при 0<m<n.

* 1. вычисляющий число *а*, для которого выполняется неравенство 2^{a-1} \le n \le 2^{a}, где n  – натуральное число. Для подсчета числа а использовать формулу: a(n)=\left\{\begin{array}{ll}
     1, & \text{если } n=1;\\
     a(n/2), &  \text{если } n >1;
     \end{array}
  2. для вычисления x^{n} ( x –вещественное, x \ne 0, а n –целое) по формуле:

x^n=\left\{\begin{array}{ll}
1, & \text{если } n=0;\\
\frac{1}{x^{|n|}}, &  \text{если } n <0;\\
x \cdot{x^{n-1}}, &  \text{если } n >0;
\end{array}. Вычислить значение x^{n} для различных x и n.

* 1. для вычисления \sum_{i=1}^n{i}, где n – натуральное число. Для заданных натуральных чисел m и k вычислить с помощью разработанного метода значение выражения \sum_{i=1}^n{i}+\sum_{i=1}^{2k}{i}.
  2. для вычисления значения функции F(N)=\frac{N}{\sqrt{1+\sqrt{2+\sqrt{3+...\sqrt{N}}}}}.

Найти ее значение при заданном натуральном N.

* 1. для вычисления цепной дроби: .

Найти значение данной дроби при заданном натуральном n.

11 Вычислить значение арифметического выражения:



**Задание 2. Разработка рекурсивных методов ( не возвращающих значений)**:

1. Даны первый член и разность арифметической прогрессии. Написать *рекурсивный метод* для нахождения n -го члена и суммы n  первых членов прогрессии.
2. Даны первый член и знаменатель геометрической прогрессии. Написать *рекурсивный метод* для нахождения n-го члена и суммы n  первых членов прогрессии.
3. Разработать *рекурсивный метод*, который по заданному натуральному числу N ( N>=1000) выведет на экран все натуральные числа не больше N  в порядке возрастания. Например, для N=8, на экран выводится 12345678.
4. Разработать *рекурсивный метод*, который по заданному натуральному числу N ( N>=1000 ) выведет на экран все натуральные числа не больше N в порядке убывания. Например, для N=8, на экран выводится 87654321.
5. Разработать *рекурсивный метод* для вывода на экран стихотворения:
   * 1. 10 лунатиков жили на луне
     2. 10 лунатиков ворочались во сне
     3. Один из лунатиков упал с луны во сне
     4. 9 лунатиков осталось на луне
     5. 9 лунатиков жили на луне
     6. 9 лунатиков ворочались во сне
     7. Один из лунатиков упал с луны во сне
     8. 8 лунатиков осталось на луне
     9. ......
     10. И больше лунатиков не стало на луне
6. Дано натуральное число n. Разработать *рекурсивный метод* для вывода на экран следующей последовательности чисел:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. 1 |  |  |  |  |
| 1. 2 | 1. 2 |  |  |  |
| 1. 3 | 1. 3 | 1. 3 |  |  |
| 1. … |  |  |  |  |
| 1. n | 1. n | 1. n | 1. n… | 1. n |

7 Дано натуральное число n. Разработать *рекурсивный метод* для вывода на экран следующей последовательности чисел:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 |  |  |  |  |
| 2 | 1 |  |  |  |
| 3 | 2 | 1 |  |  |
| … |  |  |  |  |
| n | n-1 | n-2 | … | 1 |

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

9 Разработать *рекурсивный метод* для перевода числа из десятичной системы счисления в двоичную.

10 Разработать *рекурсивный метод* для перевода числа из двоичной системы счисления в десятичную.

11 Разработать *рекурсивный метод* для вывода на экран всех делителей заданного натурального числа n.

12 Дано натуральное четное число n. Разработать *рекурсивный метод* для вывода на экран следующей картинки:

|  |  |
| --- | --- |
| \*\*\*\*\*\*\*\*\* | (0 пробелов, n звездочек) |
| \*\*\*\*\*\*\*\* | (1 пробел, n-1 звездочка) |
| \*\*\*\*\*\*\* | (2 пробела, n-2 звездочки) |
| … |  |
| \* | (n-1 пробел, 1 звездочка) |

13 Дано натуральное четное число n. Разработать *рекурсивный метод* для вывода на экран следующей картинки:

|  |  |
| --- | --- |
| \* \* | (n пробелов между звездочками) |
| \*\* \*\* | (n-2 пробела) |
| \*\*\* \*\*\* | (n-4 пробела) |
| … | … |
| \*\*\*\*\* \*\*\*\*\* | (2 пробела) |
| \*\*\*\*\*\*\*\*\*\* | (0 пробелов) |
| \*\*\*\*\* \*\*\*\*\* | (2 пробела) |
| … | … |
| \*\*\* \*\*\* | (n-4 пробела) |
| \*\* \*\* | (n-2 пробела) |
| \* \* | (n пробелов |

14 Дано натуральное число n. Разработать *рекурсивный метод* для вывода на экран следующей картинки:

|  |  |
| --- | --- |
| 1 | (1 раз) |
| 222 | (3 раза) |
| 33333 | (5 раз) |
| … | (n раз) |
| 33333 | (5 раз) |
| 222 | (3 раза) |
| 1 | (1 раз) |

15 Разработать *рекурсивный метод* для вывода на экран следующей картинки:

|  |  |
| --- | --- |
| AAAAAAAAAA…AAAAAAAAAA | (80 раз) |
| BBBBBBBBB…BBBBBBBB | (78 раз) |
| СССССССС …СССССССС | (76 раз) |
| … | … |
| YYY…YYY | (32 раза) |
| ZZ...ZZ | (30 раз) |
| YYY…YYY | (32 раза) |
| … | … |
| СССССССС …СССССССС | (76 раз) |
| BBBBBBBBB…BBBBBBBB | (78 раз) |
| AAAAAAAAAA…AAAAAAAAAA | (80 раз) |

**Практическое занятие 15**

**Работа со строками**

**Цель работы:**

1. Овладеть навыками составления алгоритмов и программ на языке С# по обработке символьных строк.

2. Научиться использовать в программе стандартные функции для обработки строк.

**Краткие сведения из теории**

Обработка текстовой информации является одной из самых распространенных задач современного программировании. С# предоставляет для ее решения широкий набор средств: символы char, неизменяемые строки string, изменяемые строки String Builder и регулярные выражения Regex.

**1. Символы char**

Символьный тип char предназначен для хранения символа в кодировке Unicode. Символьный тип относится к встроенным типам данных С# и соответствует стандартному классу Сhar библиотеки .Net из пространства имен System. В этом классе определены статические методы, позволяющие задавать вид и категорию символа, а также преобразовывать символ в верхний или нижний регистр, в число.

char b = 'B', c = '\x64', d = '\uffff';

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

Console.WriteLine("{0}, {1}, {2}", char.ToLower(b), char.ToUpper(c), char.GetNumericValue(d));

Используя символьный тип можно организовать массив символов и работать с ним на основе базового класса Array:

**static void Main()**

**{**

**char[] a ={ 'm', 'a', 'Х', 'i', 'M', 'u', 'S' , '!', '!', '!' };**

**char [] b="кол около колокола".ToCharArray();** //преобразование строки в массив символов

**PrintArray("Исходный массив а:", a);**

**for (int x=0;x<a.Length; x++)**

**if (char.IsLower(a[x])) a[x]=char.ToUpper(a[x]);**

**PrintArray("Измененный массив а:", a);**

**PrintArray("Исходный массив b:", b);**

**Array.Reverse(b);**

**PrintArray("Измененный массив b:", b);**

**}**

**static void PrintArray(string line, Array a)**

**{**

**Console.WriteLine(line);**

**foreach( object x in a) Console.Write(x);**

**Console.WriteLine('\n');**

**}**

**2. Неизменяемые строки string**

Тип string, предназначенный для работы со строками символов в кодировке Unicode, является встроенным типом С#. Ему соответствует базовый тип класса System.String библиотеки .Net. Каждый объект string - это неизменяемая последовательность символов Unicode, т.е. методы, предназначенные для изменения строк, возвращают измененные копии, исходные же строки остаются неизменными.

Создать строку можно несколькими способами:

1. string s; // инициализация отложена

2. string s=''кол около колокола''; //инициализация строковым литералом

3. string s=@'Привет!'

4. string s=new string (' ', 20); //конструктор создает строку из 20 пробелов

5. int x = 12344556; //инициализировали целочисленную переменную

6. string s = x.ToString(); //преобразовали ее к типу string

7. char [] a={'a', 'b', 'c', 'd', 'e'}; //создали массив символов

8. string v=new string (a); // создание строки из массива символов

9. char [] a={'a', 'b', 'c', 'd', 'e'}; // создание строки из части массива символов, при этом:

10. string v=new string (a, 0, 2) // 0 показывает с какого символа, 2 - сколько символов

// использовать для инициализации

11string s4 = new String(new char[]{'w', 'o', 'r', 'l', 'd'});

***Функции для работы со строками***

|  |  |
| --- | --- |
| Имя метода | Описание |
| **Compare** | сравнивает две строки с учетом текущей культуры (локали) пользователя |
| **CompareOrdinal** | сравнивает две строки без учета локали |
| **Contains** | определяет, содержится ли подстрока в строке |
| **Concat** | соединяет строки |
| **CopyTo** | копирует часть строки или всю строку в другую строку |
| **EndsWith** | определяет, совпадает ли конец строки с подстрокой |
| **Format** | форматирует строку |
| **IndexOf** | находит индекс первого вхождения символа или подстроки в строке |
| **Insert** | вставляет в строку подстроку |
| **Join** | соединяет элементы массива строк |
| **LastIndexOf** | находит индекс последнего вхождения символа или подстроки в строке |
| **Replace:** | замещает в строке символ или подстроку другим символом или подстрокой |
| **Split** | разделяет одну строку на массив строк |
| **Substring** | извлекает из строки подстроку, начиная с указанной позиции |
| **ToLower** | переводит все символы строки в нижний регистр |
| **ToUpper** | переводит все символы строки в верхний регистр |
| **Trim** | удаляет начальные и конечные пробелы из строки |

Стандартный класс string позволяет выполнять над строками различные операции, в том числе поиск, замену, вставку и удаление подстрок. Тем не менее, есть классы задач по обработке символьной информации, где стандартных возможностей явно не хватает. Чтобы облегчить решение подобных задач, в NetFramework встроен более мощный аппарат работы со строками, основанный на регулярных выражениях.

Регулярные выражения предназначены для обработки текстовой информации и обеспечивают:

1. Эффективный поиск в тексте по заданному шаблону;
2. Редактирование текста;
3. Формирование итоговых отчетов по результатам работы с текстом.

Регулярное выражение - это шаблон, по которому выполняется поиск соответствующего фрагмента текста. Язык описания регулярных выражений состоит из символов двух видов: обычных символов и метасимволов. Обычный символ представляет в выражении сам себя, а метасимвол - некоторый класс символов.

Регулярное выражение записывается в виде строкового литерала, причем перед строкой необходимо ставить символ @, который говорит о том, что строку нужно будет рассматривать и в том случае, если она будет занимать несколько строчек на экране. Однако символ @ можно не ставить, если в качестве шаблона используется шаблон без метасимволов.

Примеры регулярных выражений:

1. слово rus - @"rus" или "rus"
2. номер телефона в формате xxx-xx-xx - @"\d\d\d-\d\d-\d\d" или @"\d{3}(-\d\d){2}"
3. номер автомобиля - @"[A-Z]\d{3}[A-Z]{2}\d{2,3}RUS"

Порядок работы

Задание 1. Разработать алгоритм и тесты, написать и отладить программу обработки строк (табл.13), используя ***стандартные методы***. Выполнить программу, проанализировать результат в соответствии с тестами.

Таблица 13

|  |  |
| --- | --- |
| Номер варианта | Условие |
| 1 | Дана строка-предложение на русском языке. Зашифровать ее, через циклическую замену каждой буквы на следующую за ней в алфавите и сохраняя при этом регистр букв ("А" перейдет в "Б", "а" — в "б", "Б" — в "В", "я" — в "а" и т.д.). Букву "ё" в алфавите не учитывать ("е" должна переходить в "ж"). Знаки препинания и пробелы не изменять. |
| 2 | В заданной строке, не содержащей букв русского алфавита, найти процент гласных букв  (a, e, i, o, u, y).Определите, содержится ли в сообщении заданное слово. |
| 3 | Дана строка, состоящая из русских слов, разделенных пробелами (одним или несколькими). Определить количество слов, которые содержат ровно три буквы "А" и вывести их. |
| 4 | Из заданной строки, не содержащей цифр, получить "сжатую" строку, заменив самую  длинную подстроку вида "XXX…XX" на подстроку вида "nX", гдеX - повторяющийся символ,n - число его повторений Удалите из сообщения все знаки препинания. |
| 5 | В заданном предложении найти самое короткое и самое длинное слово. Удалите из сообщения только те слова, которые начинаются на гласную букву. |
| 6 | Для каждого символа заданного текста указать, сколько раз он встречается в тексте. Сообщение об одном символе должно печататься не более одного раза. |
| 7 | Строка содержит дату в формате "mm-dd-yyyy",гдеmm - месяц,dd - день,yyyy - год.  Проверить корректность задания даты и получить строку с датой в формате "dd.mm.yy", где yy - две последние цифры года. |
| 8 | В заданной строке, состоящей из слов, разделенных одним или более пробелами,  определить количество слов, начинающихся и заканчивающихся на одну и ту же букву. Удалите из сообщения только те русские слова, которые начинаются на гласную букву. |
| 9 | В заданной строке, состоящей из слов, разделенных одним или более пробелами,  подсчитать количество симметричных слов, т.е. читающихся одинаково слева направо и справа налево. |
| 10 | Подсчитать количество слов в предложении. Удалите из сообщения все однобуквенные слова. |
| 11 | В заданной строке определить наибольшее количество цифр, идущих подряд. Для каждой найденной подстроки выполнить проверку: если она представляет собой запись целого числа, то вывести 1; если вещественного (с дробной частью), то вывести 2; если строку нельзя преобразовать в число, то вывести 0. |
| 12 | Удалить часть символьной строки, заключённой в скобки (вместе со скобками). В строке имеется не менее одной пары правильно расставленных скобок.. |
| 13 | Из заданной строки удалить все символы, входящие в нее более одного раза |
| 14 | Выведите все слова заданной длины.Определить, можно ли из символов заданной строки составить вашу фамилию. |
| \*\* | Требуется определить подходит ли заданное слово под заданный шаблон.  Шаблон задается большими латинскими буквами, знаками "?" - любой символ, "\*" - любая последовательность символов (даже пустая).  Примеры: слово – ABBCDA, шаблон - A\*CDA ответ YES  слово – AADAAVA, шаблон - A\*DA\*AA\* ответ NO |

Задание 2. Составить программу, используя регулярное выражение (табл.14). Выполнить программу, проанализировать результат в соответствии с тестами.

Таблица 14

|  |  |
| --- | --- |
| Номер варианта | Условие |
| 1 | Получить все подстроки между круглыми скобками |
| 2 | Шаблоны регулярных выражений для групп time, ip и site записаны в упрощенном виде. Преобразуйте их к такому виду, чтобы они соответствовали ограничениям, накладываемым на формат времени, ip-адреса и адреса web-сайта. |
| 3 | Получить тег <imgsrc="..." /> и заменить его содержимое |
| 4 | Шаблоном называется строка, состоящая из английских букв (a,...,z, A,...,Z) и символов ? и \*. Каждый из символов ? разрешается заменить на одну произвольную букву, а каждый из символов \* - на произвольную (возможно пустую) последовательность букв.  Про любую строку из букв, которую можно получить из шаблона такими заменами, будем говорить, что она удовлетворяет этому шаблону.  Имеются два шаблона. Требуется найти строку минимальной длины, которая удовлетворяет обоим шаблона, либо выдать сообщение, что такой строки не существует. |
| 5 | В сообщении могут встречаться номера телефонов, записанные в формате xx-xx-xx, xxx-xxx или xxx-xx-xx. Вывести все номера телефонов, которые содержатся в сообщении. |
| 6 | Удаление чисел (цифр) из текста |
| 7 | 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). Замените каждую дату в сообщении на дату предыдущего дня. |
| 8 | В сообщении может содержаться время в формате чч:мм:сс. В заданном формате чч - целое число из диапазона от 00 до 24, мм и сс - целые числа из диапазона от 00 до 60 (если какая-то часть формата нарушена, то данная подстрока в качестве даты не рассматривается). Преобразуйте каждое время к формату чч:мм, применив правило округления до целого числа минут. |
| 9 | В сообщении может содержаться время в формате чч:мм. В заданном формате чч - целое число из диапазона от 00 до 24, мм - целое число из диапазона от 00 до 60 (если какая-то часть формата нарушена, то данная подстрока в качестве даты не рассматривается). Увеличите время на n минут. |
| 10 | В сообщении может содержаться время в формате чч:мм. В заданном формате чч - целое число из диапазона от 00 до 24, мм - целое число из диапазона от 00 до 60 (если какая-то часть формата нарушена, то данная подстрока в качестве даты не рассматривается). Уменьшите время на n часов. |
| 11 | В сообщении могут содержаться IP-адреса компьютеров в формате d.d.d.d, где d - целое число из диапазона от 0 до 255. Удалить из сообщения IP-адреса, в которых последнее число d начинается с заданной цифры (данная цифра вводится с клавиатуры). |
| 12 | Выведите на экран все адреса web-сайтов, содержащиеся в сообщении. |
| 13 | 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. |
| 14 | 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. |

**Практическое занятие 16**

**Работа с типом дата . Обработка исключительных ситуаций.**

**Цель работы:**

1. Изучить правила описания и использования переменных типа дата в языке С#.

2. Приобретение практических навыков обработки исключительных ситуаций.

**Краткие сведения из теории**

**Структура DateTime** предназначена для хранения и обработки переменных в формате даты или времени.

Структура ***DateTime*** настолько универсальна, что ее используют и при работе со строками, и при работе с числами, и при работе с базами данных.

Структура DateTime имеет отдельные свойства для каждой категории времени (год, месяц,число, час, минута, секунда, миллисекунда), а также имеет набор методов для обработки временных интервалов.

Например,можно сложить два временных значения, вычесть, конвертировать в другой формат...

Основные операции со структурой DateTime связаны со сложением или вычитанием дат.

Для добавления дат используется ряд методов:

**Add(DateTime date) :** добавляет дату date

**AddDays(double value)** : добавляет к текущей дате несколько дней

**AddHours(double value)** : добавляет к текущей дате несколько часов

**AddMinutes(double value)** : добавляет к текущей дате несколько минут

AddMonths(int value) : добавляет к текущей дате несколько месяцев

**AddYears(int value)** : добавляет к текущей дате несколько лет

Например, добавим к некоторой дате 3 часа:

DateTime date1 = new DateTime(2017, 7, 20, 18, 30, 25); // 20.07.2017 18:30:25

Console.WriteLine(date1.AddHours(3)); // 20.07.2017 21:30:25

Исключительная ситуация – это нарушение условий выполнения программы, вызывающее прерывание или полное прекращение ее работы. Обработка исключительных ситуаций состоит в нейтрализации динамической ошибки, вызвавшей ее.

Исключения обнаруживаются и обрабатываются в операторе **try.** Оператор **try** содержит три части:

*-* ***контролируемый блок*** — составной оператор, предваряемый ключевым словом **try.** В контролируемый блок включаются потенциально опасные операторы программы. Все функции, прямо или косвенно вызываемые из блока, также считаются ему принадлежащими;

- один или несколько ***обработчиков исключений*** — **блоков catch**, в которых описывается, как обрабатываются ошибки различных типов;

*-* ***блок завершения* finally** выполняется независимо от того, возникла ошибка в контролируемом блоке или нет.

*Синтаксис оператора* ***try****:*

**try блок [ блоки catch ] [ блок finally ]**

Отсутствовать могут либо блоки **catch**, либо блок **finally,** но

не оба одновременно.

Реализация обработки исключительных ситуаций.

1 Обработка исключения начинается с появления ошибки. Функция или операция, в которой возникла ошибка, генерирует исключение.

2 Выполнение текущего блока прекращается, отыскивается соответствующий обработчик исключения, и ему передается управление.

3 Выполняется блок **finally**, если он присутствует.

4 Если обработчик не найден, вызывается стандартный обработчик исключения. Обычно он выводит на экран окно с информацией об исключении и завершает текущий процесс.

Пример:

try {

... // Контролируемый блок

}

catch ( OverflowException e )

{// Обработка исключений класса

... OverflowException (переполнение)

}

catch ( DivideByZeroException )

{// Обработка исключений класса

... DivideByZeroException (деление на 0)

}

catch {

... // Обработка всех остальных исключений

}

Для генерации исключения используется оператор **throw** с параметром, определяющим вид исключения.

***Параметр должен быть объектом, порожденным от стандартного класса System.Exception.***

Этот объект используется для передачи информации об исключении его обработчику.

*Синтаксис оператора* ***throw****:*

**throw [ выражение ];**

**Порядок работы.**

**Задание 1** Создать Windows Forms приложение для работы с типом DateTime. Оно должно запрашивать дату и время рождения студента, выводить текущие дату/время, запрашивать желаемое время дальнейшего обучения, выводить дату/время и день недели конца обучения.

**Пример**. Активизируем ИСР, выбираем создание Windows приложения. В форму заносим компоненты:

 2 компонента dateTimePicker, один для отображения текущей даты, второй для редпктирования даты рождения.

 5 компонентов label для размещения поясняющих текстов.

 2 компонента TextBox для ввода данных.

 Компонент listBox для вывода данных

Компонент button (кнопка) для запуска обработчика события.

Компонентам назначаем свойства, используя окно свойств ИСР. Итог:

Двойным щелчком по кнопке инициируем в редакторе кода шаблон

обработчика события, в который заносим программу.

**Листинг программы**

using System;

using System.Windows.Forms;

namespace WindowsFormsDateTime

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{ tring f = textBox1.Text;

int c = Convert.ToInt32(textBox2.Text);

int c1 = Convert.ToInt32(textBox3.Text);

dateTimePicker2.MaxDate = DateTime.Now;

DateTime dt1 = dateTimePicker2.Value;

DateTime dt2 = DateTime.Now;

int d = (dt2.Year - dt1.Year)\*365;

string s = "Вы " + f;

listBox1.Items.Add(s);

s = Convert.ToString(d);

s = "Вы прожили " + s + " дней";

listBox1.Items.Add(s);

s = Convert.ToString(c1);

s = "Вы хотите учиться еще " + s + " дней";

listBox1.Items.Add(s);

dt1 = dt1.AddDays(c1);

s = "Вы станете очень умным " + dt1;

listBox1.Items.Add(s);

s = Convert.ToString(dt1.DayOfWeek);

s = "Это будет в " + s;

listBox1.Items.Add(s);

s = "Если не все поняли, подучите английский";

listBox1.Items.Add(s);

}

}

}

Далее окна формы на этапах прогона:

 Сначала стартовое Окно.

 Затем окно при редактировании компонента DateTimePicker2. В нем нужно установить год, Затем из списка вызвать коллекцию календарей месяцев, в ней выбрать календарь нужного месяца, в котором выбрать день.

 Затем заполнить поля формы. Час рождения должен быть от 0 до 24.

 Нажатие кнопки Принять приводит к обработке данных и формирование итогов в компоненте LictBox1.

**Дополнить приложение, используя различные методы работы с датой.**

* Преобразование дня года в дату типа месяц– день (, вводим числом 40, получаем результат “February 9”.)
* Учет високосных годов . Конечный вариант программы будет запрашивать у пользователя не только день года, но и сам год. Программа будет определять, является ли год високосным. Если да, то будет проверяться, попадает ли значение дня года в диапазон от 1 до 366. Если год не является високосным, то проверяется попадание значения дня года в диапазон от 1 до 365.

|  |  |
| --- | --- |
| Год День | Месяц и день Года |
| 1999 32 | February 1 |
| 2000 32 | February 1 |
| 1999 60 | March 1 |
| 2000 60 | February 29 |
| 1999 91 | April 1 |
| 2000 91 | March 31 |
| 1999 186 | July 5 |
| 2000 186 | July 4 |
| 1999 327 | November 23 |
| 2000 327 | November 22 |

Задание 2. Разработать тесты, написать и отладить приложение, в котором необходимо проводить обработку возможных исключений. (табл.14). Выполнить программу, проанализировать результат в соответствии с тестами.

Таблица 14

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вариант | Задание | | | |
| 1 | Создать программу, позволяющую пользователю ввести два числа, которые программа разделит. Необходимо поместить на форму три объекта класса TextBox - два для операндов, один – для результата и кнопку (объект класса Button), нажимая на которую пользователь выполняет деление. Исключить попытку деления на ноль а так же введения символов вместо цифр. Выдать сообщение о типе возникшей ошибки. | | | |
| 2 | Создать программу, вычисляющую корни квадратного уравнения (ax2+bx+c=0). Необходимо поместить на форму четыре объекта класса TextBox - три для коэффициентов квадратного уравнения, один – для результата и кнопку (объект класса button), нажимая на которую пользователь выполняет нахождение корней. Исключить ввод символов вместо цифр, получение отрицательного дискриминанта и ввод а = 0. Вывести при всех типах ошибок одно и то же сообщение. | | | |
| 3 | Создать программу с “бесконечным” циклом типа while. В цикле увеличивать переменную I до значения, заданного пользователем. При достижении этого значения выходить из цикла с помощью возбуждения исключения ***Exception*** Выдать сообщение о выходе из цикла в блоке cath. Необходимо поместить на форму кнопку , которая запускает цикл; сообщение можно выдать с помощью функции ShowMessage, или поместить на форму метку ( label), в которую помещается сообщение. | | | |
| 4 | Создать программу, вычисляющую тангенс угла. Необходимо поместить в форму два компонента TextBox для ввода значения и результата и кнопку button для вычисления значения тангенса. Исключить ввод символов вместо цифр и получение значения тангенса угла 90 градусов. Предусмотреть возможность ввода значений в радианах. | | | |
| 5 | Создать программу, вычисляющую логарифм числа. Для этого необходимо поместить в форму два компонента TextBox для ввода значения и результата и кнопку button для вычисления значения логарифма. Исключить ввод символов вместо цифр и получение значения логарифма 0. | | | |
| 6 | Создать программу обработки исключения при обращении к несуществующему элементу массива. В форму поместите поля редактирования для ввода – вывода значений и номеров элементов массива и кнопку для обработки события. | | | |
| 8 | Проверка вводимого пользователем значения дня года. Программа будет проверять день года, вводимого пользователем. Если он будет меньше 1 или больше 365, то будет выбрасываться исключениеInvalidArgument (“Dayoutofrange”). Программа будет перехватывать это исключение в блоке catch и выводить на консоль сообщение об ошибке. | | | |
| 8 | Постройте таблицу значений функции y=f(x) для х \in [a, b] с шагом h. Если в некоторой точке x *функция* не определена, то выведите на экран сообщение об этом. | | | |
| 8.1-8.4 | y=\cfrac{x}{\sqrt{2x-1}} | y=\cfrac{3x+4}{\sqrt{x^2+2x+1}} | y=\cfrac{1}{x-1}+\cfrac{2}{1-4x} | y=\ln{\cfrac{x}{x-2}} |
| 8.5-8.7 | y=\cfrac{\sqrt{x^2-2x+1}}{\ln{(4-2x)}} | y=\cfrac{\sqrt{x^3-1}}{\sqrt{x^2-1}} | 7y=\cfrac{\ln{(x-2)}}{\sqrt{5x+1}} |  |

**Практическое занятие 17**

**Файлы последовательного доступа. Типизированные файлы**

**Цель работы:**

1. Научиться создавать текстовый файл данных.

2. Научиться читать из файла данные и использовать их в обработке информации.

3. Формировать файл результатов.

**Краткие сведения из теории**

Для работы с файлами предназначены классы **File** и **FileInfo**. С их помощью мы можем создавать, удалять, перемещать файлы, получать их свойства и многое другое.

Некоторые полезные методы и свойства класса FileInfo:

* **CopyTo(path)**: копирует файл в новое место по указанному пути path
* **Create()**: создает файл
* **Delete()**: удаляет файл
* **MoveTo(destFileName)**: перемещает файл в новое место
* Свойство **Directory**: получает родительский каталог в виде объекта DirectoryInfo
* Свойство **DirectoryName**: получает полный путь к родительскому каталогу
* Свойство **Exists**: указывает, существует ли файл
* Свойство **Length**: получает размер файла
* Свойство **Extension**: получает расширение файла
* Свойство **Name**: получает имя файла
* Свойство **FullName**: получает полное имя файла

Класс File реализует похожую функциональность с помощью статических методов:

* **Copy()**: копирует файл в новое место
* **Create()**: создает файл
* **Delete()**: удаляет файл
* **Move**: перемещает файл в новое место
* **Exists(file)**: определяет, существует ли файл

Класс **FileStream** представляет возможности по считыванию из файла и записи в файл. Он позволяет работать как с текстовыми файлами, так и с бинарными.

Рассмотрим наиболее важные его свойства и методы:

* Свойство **Length**: возвращает длину потока в байтах
* Свойство **Position**: возвращает текущую позицию в потоке
* Метод **Read**: считывает данные из файла в массив байтов. Принимает три параметра: intRead(byte[] array, intoffset, intcount) и возвращает количество успешно считанных байтов. Здесь используются следующие параметры:
  + array - массив байтов, куда будут помещены считываемые из файла данные
  + offset представляет смещение в байтах в массиве array, в который считанные байты будут помещены
  + count - максимальное число байтов, предназначенных для чтения. Если в файле находится меньшее количество байтов, то все они будут считаны.
* Метод **longSeek(longoffset, SeekOriginorigin)**: устанавливает позицию в потоке со смещением на количество байт, указанных в параметре offset.
* Метод **Write**: записывает в файл данные из массива байтов. Принимаеттрипараметра: Write(byte[] array, int offset, int count)
  + array - массив байтов, откуда данные будут записываться в файла
  + offset - смещение в байтах в массиве array, откуда начинается запись байтов в поток
  + count - максимальное число байтов, предназначенных для записи

Объект **fstream** создается двумя разными способами: через конструктор и через один из статических методов класса File.

Здесь вконструктор передается два параметра: путь к файлу и перечисление FileMode. Данное перечисление указывает на режим доступа к файлу и может принимать следующие значения:

•Append: если файл существует, то текст добавляется в конец файл. Если файла нет, то он создается. Файл открывается только для записи.

•Create: создается новый файл. Если такой файл уже существует, то он перезаписывается

•CreateNew: создается новый файл. Если такой файл уже существует, то он приложение выбрасывает ошибку

•Open: открывает файл. Если файл не существует, выбрасывается исключение

•OpenOrCreate: если файл существует, он открывается, если нет - создается новый

•Truncate: если файл существует, то он перезаписывается. Файл открывается только для записи.

**Чтение и запись текстовых файлов. StreamReader и StreamWriter**

Класс StreamReader позволяет нам легко считывать весь текст или отдельные строки из текстового файла. Среди его методов можно выделить следующие:

* **Close**: закрывает считываемый файл и освобождает все ресурсы
* **Peek**: возвращает следующий доступный символ, если символов больше нет, то возвращает -1
* **Read**: считывает и возвращает следующий символ в численном представлении. Имеет перегруженную версию:Read(char[] array, intindex, intcount), где array - массив, куда считываются символы, index - индекс в массиве array, начиная с которого записываются считываемые символы, и count - максимальное количество считываемых символов
* **ReadLine**: считывает одну строку в файле
* **ReadToEnd**: считывает весь текст из файла

Для записи в текстовый файл используется класс StreamWriter. Свою функциональность он реализует через следующие методы:

•Close: закрывает записываемый файл и освобождает все ресурсы

•Flush: записывает в файл оставшиеся в буфере данные и очищает буфер.

•Write: записывает в файл данные простейших типов, как int, double, char, string и т.д.

•WriteLine: также записывает данные, только после записи добавляет в файл символ окончания строки

**Порядок работы**

Разработать алгоритм и тесты, написать и отладить программу формирования и поиска в файле информации (табл.15). Выполнить программу, проанализировать результат в соответствии с тестами.

***Указание.*** Создать файл данных ( 10-15 записей , согласно варианта). В программе считываем данные из файла, обрабатываем и формируем файл результатов.

**Таблица 15**

|  |  |
| --- | --- |
| Номер варианта | Условие |
| 1 | В магазине сформирован список постоянных клиентов, который включает ФИО, домашний адрес покупателя и размер предоставляемой скидки. Удалить из этого списка все повторяющиеся записи, проверив ФИО и домашний адрес покупателя. |
| 2 | Список товаров, имеющихся на складе, включает в себя наименование товара, количество единиц товара, цену единицы и дату поступления товара на склад. Вывести в алфавитном порядке список товаров, хранящихся больше месяца, стоимость которых превышает 100 000 р. |
| 3 | Для получения места в общежитии формируется список студентов, который включает ФИО студента, группу, средний балл, доход на члена семьи. Общежитие в первую очередь предоставляется тем, у кого доход на члена семьи меньше двух минимальных зарплат, затем остальным в порядке уменьшения среднего балла. Вывести список очередности предоставления мест в общежитии. |
| 4 | В справочной автовокзала хранится расписание движения автобусов. Для каждого рейса указаны его номер, тип автобуса, пункт назначения, время отправления и прибытия. Вывести информацию о рейсах, которыми можно воспользоваться для прибытия в пункт назначения раньше заданного времени. |
| 5 | Разработать программу ("электронную сваху") для поиска женихов и невест. Создать два файла: список женихов и список невест. В каждом списке кандидат (жених или невеста) характеризуется записью содержащей порядковый номер кандидата; данные о кандидате (возраст, рост, вес); требования к партнеру (минимальное и максимальное значение соответствующего параметра). Сформировать списки пар с учетом требований кандидата. |
| 6 | На междугородной АТС информация о разговорах содержит дату разговора, код и название города, время разговора, тариф, номер телефона в этом городе и номер телефона абонента. Вывести для заданного города общее время разговоров с ним и сумму. |
| 7 | Информация об участниках спортивных соревнований содержит наименование страны, название команды, ФИО игрока, игровой номер, возраст, рост и вес. Вывести информацию о самой молодой команде. |
| 8 | В радиоателье хранятся квитанции о сданной в ремонт радиоаппаратуре. Каждая квитанция содержит следующую информацию: наименование группы изделий (телевизор, радиоприемник и т.п.), марка изделия, дата приемки в ремонт, состояние готовности заказа (выполнен, не выполнен). Вывести информацию о состоянии заказов на текущие сутки по группам изделий. |
| 9 | Разработать программу формирования ведомости об успеваемости студентов. Каждая запись этой ведомости должна содержать: номер группы, Ф.И.О. студента, оценки за последнюю сессию. Вывести списки студентов по группам. В каждой группе Ф.И.О. студентов должны быть расположены в порядке убывания среднего балла. |
| 10 | В справочной аэропорта хранится расписание вылета самолетов на следующие сутки. Для каждого рейса указаны номер рейса, тип самолета, пункт назначения, время вылета. Вывести все номера рейсов, типы самолетов и время вылета в заданный пункт назначения в порядке возрастания времени вылета. |
| 11 | Информация об участниках спортивных соревнований содержит: наименование страны, название команды, Ф.И.О. игрока, игровой номер, возраст, рост, вес. Вывести информацию о самой молодой, рослой и легкой команде. |
| 12 | Ведомость абитуриентов, сдавших вступительные экзамены в университет, содержит ФИО., адрес, оценки. Определить количество абитуриентов, проживающих в г.Новосибирске и сдавших экзамены со средним баллом не ниже 4.5, вывести их фамилии в алфавитном порядке. |
| 13 | В справочной аэропорта хранится расписание вылета самолетов на следующие сутки. Для каждого рейса указаны номер рейса, тип самолета, пункт назначения, время вылета. Вывести все номера рейсов, типы самолетов и время вылета в заданный пункт назначения в порядке возрастания времени вылета. |
| 14 | В магазине имеется список поступивших в продажу автомобилей. Каждая запись этого списка содержит: марку автомобиля, стоимость, расход топлива на 100 км, надежность (число лет безотказной работы), комфортность (отличная, хорошая, удовлетворительная). Вывести перечень автомобилей, удовлетворяющих требованиям покупателя, которые вводятся с клавиатуры в виде некоторого интервала допустимых значений. |
| 15 | Каждая запись списка вакантных рабочих мест содержит: наименование организации, должность, квалификация (разряд или образование), стаж работы по специальности, заработная плата, наличие социального страхования (да/нет), продолжительность ежегодного оплачиваемого отпуска. Вывести список рабочих мест в соответствии с требованиями клиента. |
| 16 | Для участия а конкурсе на замещение вакантной должности сотрудника фирмы желающие подают следующую информацию: Ф.И.О., год рождения, образование (среднее, специальное, высшее), знание иностранных языков (английский, немецкий, французский, владею свободно, читаю и перевожу со словарем), владение компьютером (MSDOS, Windows), стаж работы, наличие рекомендаций. Вывести список претендентов в соответствии с требованиями руководства фирмы |