

« (. .) . .»

« () »
8
«28» _____ 20 .

« _____ . . »
«28» _____ 20 .

« **01.01.** »
09.02.07 «
»

«

. »

. .

3 ,

09.02.07 «

»

СОДЕРЖАНИЕ

.....	3
.....	4
.....	4
.....	5
.....	8

•

01.

02.

03.

04.

05.

06.

07.

08.

09.

10.

11.

1.1

1.2

1.3

1.4

1.5

1.6

—

,

;

—

,

,

—

,

;

,

,

,

.



Структура лабораторного занятия

1

_____:

_____ -

_____ -

_____ -

- 1.
- 2.
3. ()
- 4.

1. : .. / .. , .. / ..
- : « »: - , 2019. - 512 : - ()
)
2. ,
: / .. - :) , - , 2020. -
336 : 60x90 1/16. - () (7) ISBN 978-5-
906818-41-6
3. ..
.. - : 2020.
4. .. :
.. - 8- .. - :
« », 2019. - 384 .
5. / .. - : , - , 2020. -384 : 60x90 1/16. - ()
)

(- ,) .

,

:

- c ;

- ;

- ;

- ;

- (, ...);

-

-

System.IO,

using System.IO.

using System.Text;

using System;

using System.Collections.Generic;

using System.Linq;

```

using System.Text;
using System.IO;
    ?
        , File Create(). new_file.txt D: -
static void Main(string[] args)
{
    File.Create("D:\\new_file.txt");
}
    WriteAllText() ( ),
static void Main(string[] args)
{
    File.WriteAllText("D:\\new_file.txt", " ");
}
    AppendAllText() , WriteAllText() ,
static void Main(string[] args)
{
    File.AppendAllText("D:\\new_file.txt", " AppendAllText ("); //
}
    ?
    Delete() :
static void Main(string[] args)
{
    File.Delete("d:\\test.txt"); //
}
    /
    ( ),
    ,
    Stream
    FileStream ( ).
    FileStream - , /
static void Main(string[] args)
{
    FileStream file = new FileStream("d:\\test.txt", FileMode.Open
, FileAccess.Read); //
}

```

FileMode:

- Append - (, FileAccess.Write)
- Create - (-)
- CreateNew - (-)
- Open - (-)
- OpenOrCreate - , , ()
- Truncate - , ()

```
static void Main(string[] args)
{
    FileStream file1 = new FileStream("d:\\file1.txt", FileMode.CreateNew); //
    FileStream file2 = new FileStream("d:\\file2.txt", FileMode.Open); //

    FileStream file3 = new FileStream("d:\\file3.txt", FileMode.Append); //
}
```

FileAccess:

- Read - .
- Write - .
- ReadWrite - .

StreamReader.

```
static void Main(string[] args)
{
    FileStream file1 = new FileStream("d:\\test.txt", FileMode.Open); //
    StreamReader reader = new StreamReader(file1); // « »

    Console.WriteLine(reader.ReadToEnd()); //
    reader.Close(); //
    Console.ReadLine();
}
```

ReadToEnd()

```
( ).
```

. ReadLine() -

EndOfStream

```
( ), true false.
```

StreamWriter.

```

static void Main(string[] args)
{
    FileStream file1 = new FileStream("d:\\test.txt", FileMode.Create); //
    StreamWriter writer = new StreamWriter(file1); //          «          »

    writer.WriteLine(" "); //
    writer.Close(); //          .          ,
}

WriteLine()          (          ,
    Write(),          ).

,          ,          (          ),
Close().

,          /

```

StreamReader/StreamWriter:

```

static void Main(string[] args)
{
    FileStream file1 = new FileStream("d:\\test.txt", FileMode.Open);
    StreamReader reader = new StreamReader(file1, Encoding.Unicode);
    StreamWriter writer = new StreamWriter(file1, Encoding.UTF8);
}

,          StreamReader  StreamWriter
    FileStream,          StreamReader/StreamWriter:

```

```

static void Main(string[] args)
{
    StreamWriter writer = new StreamWriter("d:\\test.txt"); //          ,
    writer.WriteLine(" ");
    writer.Close();
}

```

?

CreateDirectory() Directory:

```

static void Main(string[] args)
{
    Directory.CreateDirectory("d:\\new_folder");
}

```

?

Delete():

```

static void Main(string[] args)
{
    Directory.Delete("d:\\new_folder"); //
}

```

,

- true:

```

static void Main(string[] args)
{
    Directory.Delete("d:\\new_folder", true); //          ,          ,
}

```

Номер варианта	Задание
1	Дан файл f , компоненты которого являются целыми числами. Записать в файл g , компоненты файла f , исключив повторные вхождения чисел.
2	Дан файл f , компоненты которого являются действительными числами. Найти: 1. наибольшее из значений компонентов f ; 2. наименьшее из значений компонентов с четными номерами; 3. наибольшее из значений модулей компонентов с нечетными номерами; 4. сумму наибольшего и наименьшего из значений компонентов файла f ; 5. разность первой и последней компоненты файла f .
3	Дан символьный файл f . Подсчитать число вхождений в файл каждой из букв a, b, c, d, e, f . Результат вывести в файл g в виде таблицы с комментариями.
4	Дан файл f , компоненты которого являются целыми числами. Записать в файл g все четные числа исходного файла, в файл h – все нечетные. Порядок следования чисел сохраняется. Записать в файл g и h комментарий.
5	Дан текстовый файл, содержащий программу на языке C. Проверить эту программу на соответствие числа открывающих и закрывающих фигурных скобок.
6	Дан символьный файл f . Найти и записать в файл g самое длинное слово файла f , снабдив его комментарием.
7	Дан файл f , компоненты которого являются целыми числами. Получить в файле g все компоненты файла f : 1. являющиеся четными числами; 2. делящиеся на 3 и не делящиеся на 7; 3. являющиеся точными квадратами. Записать в файл g комментарий.
8	Дан файл f . Создать два файла, записав в первый из них все четные числа, расположив их в порядке возрастания, а во второй – все нечетные, расположив их в порядке убывания.
9	Дан текстовый файл f . Переформатировать исходный файл, разделяя его на строки так, чтобы каждая строка содержала столько символов, сколько содержит самая короткая строка исходного файла.
10	Дан файл f . Создать два файла, записав в первый из них среднее геометрическое всех четных чисел, а во второй – среднее арифметическое всех нечетных чисел.
11	Дан числовой файл f . Выбрать все значения, которые делятся нацело на 2 и 4, но не делятся на 6. Записать эти значения в файл g , а все остальные – в файл h .
12	Дан текстовый файл f . Определить, являются ли первые два символа цифрами и если да, то четно ли это число. Записать его в файл g , если оно четно и в h , если оно нечетно.
13	Дан текстовый файл f . Создать новый файл g и переписать в него исходный файл в обратном порядке, разделив пробелами.
14	Дан текстовый файл f . Создать новый файл, включив в него только те строки исходного файла, которые содержат самый часто используемый символ исходного файла.
15	Дан текстовый файл, содержащий программу на языке C. Проверить эту программу на соответствие открывающих и закрывающих скобок разных типов - $()$, $\{\}$, $[\]$.

Задание второе: реализовать свой класс и перегрузить в нём операторы \ll и \gg для работы со стандартными потоковыми классами. Использовать этот класс для решения задания вместе со стандартными потоковыми классами.

Номер варианта	Задание
1	Сформировать массив на диске, содержащий сведения о пациентах глазной клиники. Класс содержит поля: фамилия пациента, пол, возраст, место проживания (город), диагноз. Написать программу, которая выбирает необходимую информацию с диска и выводит на экран: - количество иногородних, прибывших в поликлинику; - список пациентов старше X лет с диагнозом J .

2	<p>Сформировать массив на диске, содержащий сведения о сотрудниках института. Класс содержит поля: фамилия работающего, название отдела, год рождения, стаж работы, должность, оклад.</p> <p>Написать программу, которая выбирает необходимую информацию с диска и выводит на экран:</p> <ul style="list-style-type: none"> - список сотрудников пенсионного возраста на сегодняшний день с указанием стажа работы; - средний стаж, работающих в отделе X.
3	<p>Сформировать массив на диске, содержащий сведения об отправлении поездов дальнего следования с жд вокзала. Класс содержит поля: номер поезда, станция назначения, время отправления, время в пути, наличие билетов.</p> <p>Написать программу, которая выбирает необходимую информацию с диска и выводит на экран:</p> <ul style="list-style-type: none"> - время отправления поездов в город X во временном интервале от A до B часов; - наличие билетов на поезд с номером XXX.
4	<p>Сформировать массив на диске, содержащий сведения о том, какие из пяти предлагаемых дисциплин по выбору желает изучать студент. Класс содержит поля: фамилия студента, индекс группы, пять дисциплин, средний балл успеваемости. Выбираемая дисциплина отмечается символом 1, иначе – пробелом.</p> <p>Написать программу, которая выбирает необходимую информацию с диска и выводит на экран список студентов, желающих прослушать дисциплину X. Если число желающих превышает 4 человека, то отобразить студентов, имеющих более высокий средний балл успеваемости.</p>
5	<p>Сформировать массив на диске, содержащий сведения об игроках футбольной команды. Класс содержит поля: имена нападающих, число заброшенных ими шайб, число сделанных голевых передач, заработанное штрафное время.</p> <p>Написать программу, которая выбирает необходимую информацию с диска и выводит на экран четырех лучших игроков по сумме очков (голы + передачи).</p>
6	<p>Сформировать массив на диске, содержащий сведения об ассортименте обуви в магазине фирмы. Класс содержит поля: артикул, наименование, количество, стоимость одной пары. Артикул начинается с буквы Д – для дамской обуви, М – для мужской, П – для детской.</p> <p>Написать программу, которая выбирает необходимую информацию с диска и выводит на экран:</p> <ul style="list-style-type: none"> - сведения о наличии и стоимости обуви артикула X; - ассортиментный список дамской обуви с указанием наименования и имеющегося в наличии числа пар каждой модели.
7	<p>Сформировать массив на диске, содержащий сведения о наличии билетов на авиарейсы. Класс содержит поля: номер рейса, пункт назначения, время вылета, время прибытия, количество свободных мест в салоне.</p> <p>Написать программу, которая выбирает необходимую информацию с диска и выводит на экран:</p> <ul style="list-style-type: none"> - время вылета самолетов в город X; - наличие свободных мест на рейс в город X с временем отправления Y.
8	<p>Сформировать массив на диске, содержащий сведения о личной коллекции книголюбца. Класс содержит поля: шифр книги, автор, название, год издания, местоположение (номер стеллажа).</p> <p>Написать программу, которая выбирает необходимую информацию с диска и выводит на экран:</p> <ul style="list-style-type: none"> - местоположение книги, автора X названия Y; - список книг автора Z, находящихся в коллекции; - число книг издания XX года, имеющихся в библиотеке.

9	<p>Сформировать массив на диске, содержащий сведения о сдаче студентами сессии. Класс содержит поля: индекс группы, фамилия студента, оценки по пяти экзаменам.</p> <p>Написать программу, которая выбирает необходимую информацию с диска и выводит на экран:</p> <ul style="list-style-type: none"> - фамилии неуспевающих студентов с указанием индексов групп и количества задолженностей; - средний балл, полученный каждым студентом группы X, и всей группы в целом.
10	<p>Сформировать массив на диске, содержащий сведения об ассортименте игрушек в магазине. Класс содержит поля: название игрушки, цена, количество, возрастные границы (2 - 5).</p> <p>Написать программу, которая выбирает необходимую информацию с диска и выводит на экран:</p> <ul style="list-style-type: none"> - название игрушки, которые подходят детям от 1 до 3 лет; - стоимость самой дорогой игрушки и ее название; - название игрушки, которая по стоимости не превышает X руб. и подходит ребенку в возрасте от A до B лет. <p>Значения A, B, X вводятся с клавиатуры.</p>
11	<p>Сформировать массив на диске, содержащий сведения о телефонах абонентов. Класс содержит поля: фамилия абонента, место жительства (название улицы, номер дома), год установки телефона.</p> <p>Написать программу, которая выбирает необходимую информацию с диска и выводит на экран:</p> <ul style="list-style-type: none"> - номер телефона по вводимой с клавиатуры фамилии абонента; - количество установленных телефонов с XXXX года; - список номеров телефонов, принадлежащих жильцам определенного дома и улицы. <p>Номер года, название улицы и номер дома вводятся с клавиатуры.</p>
12	<p>Сформировать массив на диске, содержащий сведения о количестве изделий категорий A, B, C, собранных рабочим за месяц. Класс содержит поля: фамилия сборщика, наименование цеха, количество изделий по категориям, собранных рабочим за месяц.</p> <p>Считая заданными значения расценок SA, SB, SC за выполненную работу по сборке единицы изделия категорий A, B, C, выбрать необходимую информацию с диска и вывести на экран:</p> <ul style="list-style-type: none"> - общее количество изделий категорий A, B, C, собранных рабочим цеха; - средний размер заработной платы рабочих цеха X.
13	<p>Сформировать массив на диске, содержащий сведения о количестве изделий, собранных сборщиками цеха за неделю. Класс содержит поля: фамилия сборщика, количество изделий, собранных им ежедневно в течение шестидневной недели, т.е. отдельно в понедельник, вторник и т.д. Написать программу, которая выбирает необходимую информацию с диска и выводит на экран:</p> <ul style="list-style-type: none"> - фамилию сборщика и общее количество деталей, собранных им за неделю; - фамилию сборщика собравшего наибольшее количество изделий, и день, когда он достиг наивысшей производительности труда.
14	<p>Сформировать массив на диске, содержащий сведения о личной видеотеке. Класс содержит поля: шифр по IMDb, название студии, название фильма, год издания, категория, жанр.</p> <p>Написать программу, которая выбирает необходимую информацию с диска и выводит на экран:</p> <ul style="list-style-type: none"> - статистика по жанрам для студии Y; - список фильмов категории Z, находящихся в коллекции; - число фильмов XX года, имеющихся в коллекции.
15	<p>Сформировать массив на диске, содержащий сведения о клиентах предприятия. Класс содержит поля: фамилия, имя, отчество, дата рождения, число покупок, потраченная сумма.</p> <p>Написать программу, которая выбирает необходимую информацию с диска и выводит на экран:</p> <ul style="list-style-type: none"> - список клиентов у которых на этой неделе день рождения; - список клиентов у которых покупок более чем на X; - средний размер покупки для клиентов пенсионного возраста (старше 60).


```

Program.cs
using System;

namespace HelloApp
{
    class Person
    {
    }

    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}

```

```

Person
),
,
,
.
,
(
Person
:
using System;

namespace HelloApp
{
    class Person
    {
        public string name; //
        public int age = 18; //

        public void GetInfo()
        {
            Console.WriteLine($" : {name} : {age}");
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Person tom;
        }
    }
}

```

```

        Person
        GetInfo
        Person
        age
        18.
        tom,
        Person.
        null.
        Person.
        Person:

```

```

class Person
{
    public string name; //
    public int age; //

    public void GetInfo()
    {
        Console.WriteLine($"    : {name}    : {age}");
    }
}
class Program
{
    static void Main(string[] args)
    {
        Person tom = new Person();
        tom.GetInfo(); //    :    : 0

        tom.name = "Tom";
        tom.age = 34;
        tom.GetInfo(); //    : Tom    : 34
        Console.ReadKey();
    }
}

```

```

new Person()
new Person()

```

```

        ,
        tom
        -
        null (
        tom
        :
    
```

```

        : 0
        : Tom    : 34
    
```

```

class Person
{
    public string name;
    public int age;

    public Person() { name = " "; age = 18; } // 1
    public Person(string n) { name = n; age = 18; } // 2
    public Person(string n, int a) { name = n; age = a; } // 3

    public void GetInfo()
    {
        Console.WriteLine($" : {name} : {age}");
    }
}
    
```

```

static void Main(string[] args)
{
    Person tom = new Person(); // 1-
    Person bob = new Person("Bob"); // 2-
    Person sam = new Person("Sam", 25); // 3-

    bob.GetInfo(); // : Bob : 18
    tom.GetInfo(); // : : 18
    sam.GetInfo(); // : Sam : 25
}
    
```

```

:           : 18
: Bob       : 18
: Sam       : 25

```

C# 9.0

```

Person tom = new (); // new Person();
Person bob = new ("Bob"); // new Person("Bob");
Person sam = new ("Sam", 25); // new Person("Sam", 25);

```

```

    this
    this
    ?
    name
age.

```

```

class Person
{
    public string name;
    public int age;

    public Person() : this(" ")
    {
    }
    public Person(string name) : this(name, 18)
    {
    }
    public Person(string name, int age)
    {
        this.name = name;
        this.age = age;
    }
    public void GetInfo()
    {
        Console.WriteLine($" : {name} : {age}");
    }
}

```

```

public Person(string name) : this(name, 18)
{
}

```

```
public Person(string name, int age)
{
    this.name = name;
    this.age = age;
}
```

```
public Person(string name, int age)
{
    this.name = name;
    this.age = age;
}
```

```
name - this.name = name; this.name
- name.
this
:
```

```
Person tom = new Person { name = "Tom", age=31 };
tom.GetInfo(); // : Tom : 31
```

```
name age public,
```

1.

2.

3.

13.

14.

Figure

: Rectangle (), Circle (), Triangle ()
n

15.

1.

3

_____ :
_____ -
_____ -
_____ -
_____ -

- 1.
- 2.
3. ()
- 4.

1.

.1. : / , /
.. - : « »: - , 2019. - 512 : - ()
).

2.

: / - : , - , 2020. -
336 : 60x90 1/16. - () (7) ISBN 978-5-
906818-41-6

3.

.. , - : 2020.

4.

.. - 8- :
« », 2019. - 384 .

5. / . . - .: , - , 2020. -384 .: 60x90 1/16. - ().

```
public string Describe()
```

```
public Car()
```

```
public Car()
```

```
public Car(string color)
```

Download sample code

```
public Car()
```

```
public Car(string color) : this()
```

Download sample code

```
public Car(string color) : this()
```

```
    this.color = color;
```

```

    Console.WriteLine("Constructor with color parameter called!");
}

public Car(string param1, string param2) : this(param1)
{
}

C#

~Car()
{
    Console.WriteLine("Out..");
}

```

4

-
1. _____ :
 2. _____ -
 3. _____ -
 4. _____ -
-
1. _____ : _____ / _____ , _____ / _____ .
 2. _____ : _____ / _____ - _____ , _____ , 2020. - 336 : 60x90 1/16. - (_____) (7) ISBN 978-5-906818-41-6
 3. _____ : _____ 2020.

4. :
 - 8- :
 « », 2019. - 384
 5. / : , , 2020. -384 . : 60x90 1/16. - (.).

(inheritance)

Person,

```

1 class Person
2 {
3     private string _name;
4
5     public string Name
6     {
7         get { return _name; }
8         set { _name = value; }
9     }
10    public void Display()
11    {
12        Console.WriteLine(Name);
13    }
14 }

```

```

, Employee.
, Person,
- Employee
( , ) Person, ,
( ):

```

```

1 class Employee : Person
2 {
3
4 }

```

```

Person, Employee
, Person. ,
,
Person: is-a ( ), Employee

```

```

1  static void Main(string[] args)
2  {
3      Person p = new Person { Name = "Tom" };
4      p.Display();
5      p = new Employee { Name = "Sam" };
6      p.Display();
7      Console.Read();
8  }

```

```

Employee
: Person p = new Employee().
Person,
Object,
Person
Object:
Employee
ToString(), Equals(), GetHashCode() GetType().

```

- internal, private, public.

```

( ),
public.

```

- sealed,

```

1  sealed class Admin
2  {
3  }

```

-


```

using System.IO;
using System.Dynamic;
8.
    public enum StringSearchOption
    {
        StartsWith,
        Contains,
        EndsWith
    }
9.
    class ReadOnlyFile : DynamicObject
    {
        private string p_filePath;

        // Public constructor. Verify that file exists and store the path in
        // the private variable.
        public ReadOnlyFile(string filePath)
        {
            if (!File.Exists(filePath))
            {
                throw new Exception("File path does not exist.");
            }

            p_filePath = filePath;
        }

        public override object GetPropertyValue(string propertyName,
            DynamicObject rootObject,
            object[] arguments)
        {
            return GetPropertyValue(propertyName, rootObject, arguments);
        }

        public List<string> GetPropertyValue(string propertyName,
            StringSearchOption StringSearchOption =
            StringSearchOption.StartsWith,
            bool trimSpaces = true)
        {
            StreamReader sr = null;
            List<string> results = new List<string>();
            string line = "";
            string testLine = "";

            try
            {
                sr = new StreamReader(p_filePath);
            }
        }
    }
10.
11.

```

```

while (!sr.EndOfStream)
{
    line = sr.ReadLine();

    // Perform a case-insensitive search by using the specified search options.
    testLine = line.ToUpper();
    if (trimSpaces) { testLine = testLine.Trim(); }

    switch (StringSearchOption)
    {
        case StringSearchOption.StartsWith:
            if (testLine.StartsWith(propertyName.ToUpper())) { results.Add(line); }
            break;
        case StringSearchOption.Contains:
            if (testLine.Contains(propertyName.ToUpper())) { results.Add(line); }
            break;
        case StringSearchOption.EndsWith:
            if (testLine.EndsWith(propertyName.ToUpper())) { results.Add(line); }
            break;
    }
}
}
catch
{
    // Trap any exception that occurs in reading the file and return null.
    results = null;
}
finally
{
    if (sr != null) { sr.Close(); }
}

return results;
}
12.      GetPropertyValue
         TryGetMember      DynamicObject.      TryGetMember
                                     binder
                                     result
                                     TryGetMember
                                     true,
                                     false.
// Implement the TryGetMember method of the DynamicObject class for dynamic member
calls.
public override bool TryGetMember(GetMemberBinder binder,
                                out object result)
{
    result = GetPropertyValue(binder.Name);
    return result == null ? false : true;
}

```

13. TryGetMember
 TryInvokeMember DynamicObject. TryInvokeMember
 binder
 result
 args
 TryInvokeMember true,
 false.
 TryInvokeMember
 StringSearchOption,
 TryInvokeMember
 GetPropertyValue

```
C#
// Implement the TryInvokeMember method of the DynamicObject class for
// dynamic member calls that have arguments.
public override bool TryInvokeMember(InvokeMemberBinder binder,
    object[] args,
    out object result)
{
    StringSearchOption StringSearchOption = StringSearchOption.StartsWith;
    bool trimSpaces = true;

    try
    {
        if (args.Length > 0) { StringSearchOption = (StringSearchOption)args[0]; }
    }
    catch
    {
        throw new ArgumentException("StringSearchOption argument must be a
StringSearchOption enum value.");
    }

    try
    {
        if (args.Length > 1) { trimSpaces = (bool)args[1]; }
    }
    catch
    {
        throw new ArgumentException("trimSpaces argument must be a Boolean value.");
    }

    result = GetPropertyValue(binder.Name, StringSearchOption, trimSpaces);

    return result == null ? false : true;
}
```

14. DynamicSample

1. >

TextFile1.txt

2. *TextFile1.txt*

text

List of customers and suppliers

Supplier: Lucerne Publishing (<https://www.lucernepublishing.com/>)

Customer: Preston, Chris

Customer: Hines, Patrick

Customer: Cameron, Maria

Supplier: Graphic Design Institute (<https://www.graphicdesigninstitute.com/>)

Supplier: Fabrikam, Inc. (<https://www.fabrikam.com/>)

Customer: Seubert, Roxanne

Supplier: Proseware, Inc. (<http://www.proseware.com/>)

Customer: Adolphi, Stephan

Customer: Koch, Paul

3.

1. Visual Basic, *Program.cs*, *Program.vb*,
Visual C#.

2. `ReadOnlyFile` *TextFile1.txt*.
`Main`,

`"Customer".`

`dynamic` rFile = `new` `ReadOnlyFile`(@"..\..\TextFile1.txt");

`foreach` (`string` line `in` rFile.Customer)

{

`Console.WriteLine`(line);

}

`Console.WriteLine`("-----");

`foreach` (`string` line `in` rFile.Customer(`StringSearchOption.Contains`, `true`))

{

`Console.WriteLine`(line);

}

3.

CTRL+F5

IronPython.

1. Visual Studio

>

>

2.

C#

Visual Basic,

3.

DynamicIronPythonSample

4.

()

.NET 5.0

5.

NuGet [IronPython](#).

4.

1.
: /
. - .: « »: - , 2019. - 512 .: . - (/)).

2. ,
: . / . . - .:) , - , 2020. -
336 .: 60x90 1/16. - () (7) ISBN 978-5-906818-41-6

3.
. - .: 2020.

4.
. - 8- . . - .: .
« » , 2019. - 384 .

5. / . . - .: , - , 2020. -384 .: 60x90 1/16. - ().

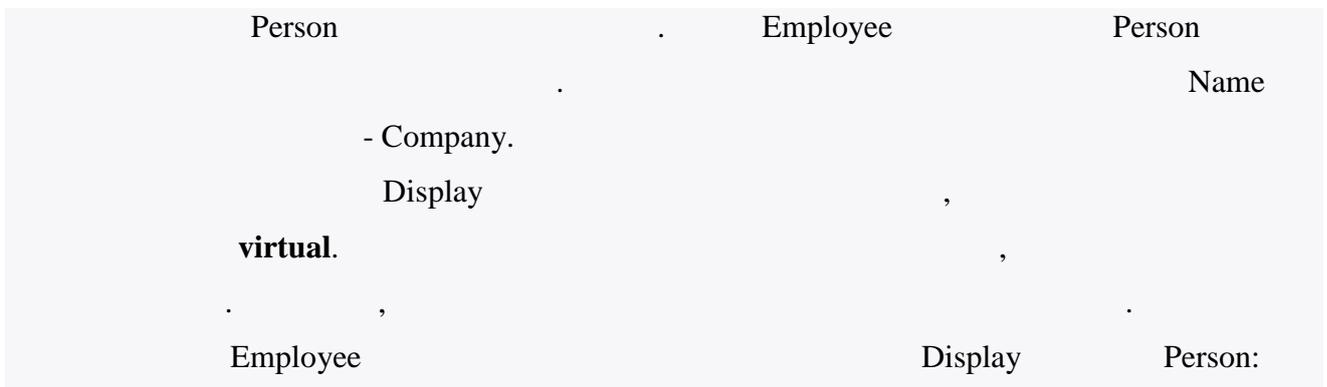


```
class Person
{
    public string Name { get; set; }
    public Person(string name)
    {
        Name = name;
    }
    public virtual void Display()
    {
        Console.WriteLine(Name);
    }
}
```

```

    }
}
class Employee : Person
{
    public string Company { get; set; }
    public Employee(string name, string company) : base(name)
    {
        Company = company;
    }
}

```



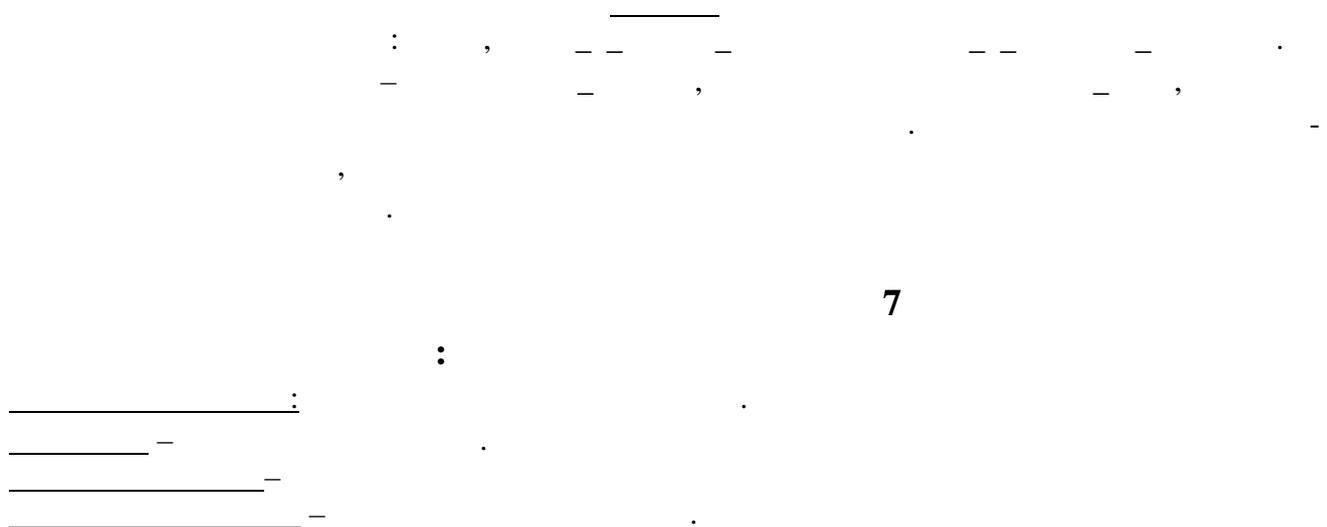
```

static void Main(string[] args)
{
    Person p1 = new Person("Bill");
    p1.Display(); // Display Person

    Employee p2 = new Employee("Tom", "Microsoft");
    p2.Display(); // Display Person

    Console.ReadKey();}

```



-
- 1.
 - 2.
 3. ()
 - 4.
-
1. 1. : „ / „ / « »: - , 2019. – 512 .: .- ().
 2. , : . / . . - .: , - , 2020. - 336 .: 60x90 1/16. - () (7) ISBN 978-5-906818-41-6
 3. . „ . „ .: 2020.
 4. . . . - 8- . . - .: « », 2019. - 384 .
 5. / . . - .: , - , 2020. - 384 .: 60x90 1/16. - ().

1. MS SQL Server

- 2.
3. . WPF, .NET, *app.config* :

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6" />
  </startup>
</configuration>
```

4. , :

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6" />
  </startup>
  <connectionStrings>
    <add name="DefaultConnection"
      connectionString="Data Source=.\SQLEXPRESS;Initial Catalog=mobiledb;Integrated Security=True;
      providerName="System.Data.SqlClient"/>
  </connectionStrings>
</configuration>
```

```

        <connectionStrings>.
    <add>.
    name. "DefaultConnection".

connectionString . :

```

- Data Source=.\SQLEXPRESS: MS
SQL Server Express ".\SQLEXPRESS"
- Initial Catalog=mobiledb:
mobiledb,
- Integrated Security=True:

MS SQL Server,
SQL Server,
System.Data.SqlClient.

xaml:

```

<Window x:Class="DbApp.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:DbApp"
    mc:Ignorable="d"
    Title="MainWindow" Height="250" Width="350" Loaded="Window_Loaded">
<Window.Resources>
    <Style TargetType="Button">
        <Setter Property="Margin" Value="20 8 20 8" />
        <Setter Property="Width" Value="100" />
        <Setter Property="Height" Value="30" />
    </Style>
</Window.Resources>
<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="*" />
        <RowDefinition Height="Auto" />
    </Grid.RowDefinitions>
    <DataGrid AutoGenerateColumns="False" x:Name="phonesGrid">
        <DataGrid.Columns>
            <DataGridTextColumn Binding="{Binding Title}" Header=" " Width="120"/>
            <DataGridTextColumn Binding="{Binding Company}" Header=" " Width="120"/>
            <DataGridTextColumn Binding="{Binding Price}" Header=" " Width="80"/>
        </DataGrid.Columns>

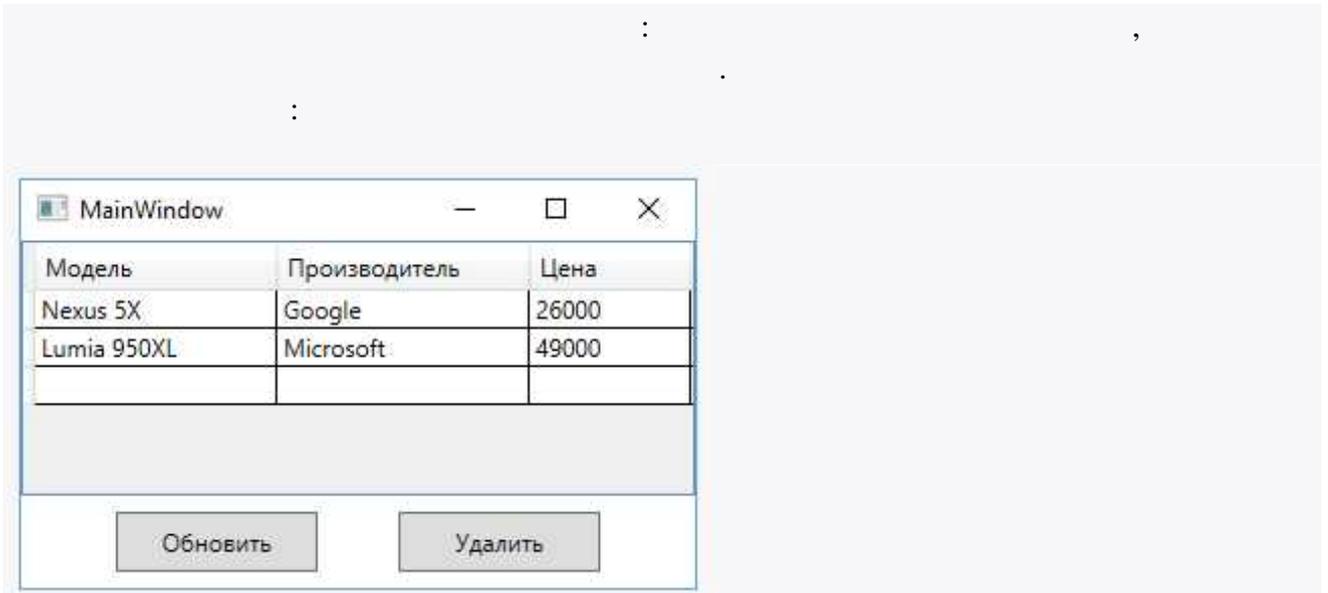
```

```

</DataGrid>

<StackPanel HorizontalAlignment="Center" Grid.Row="1" Orientation="Horizontal">
  <Button x:Name="updateButton" Content="Обновить" Click="updateButton_Click" />
  <Button x:Name="deleteButton" Content="Удалить" Click="deleteButton_Click" />
</StackPanel>
</Grid>
</Window>

```



```

5. c#:
System;
System.Windows;
System.Windows.Controls;
System.Data.SqlClient;
System.Data;
System.Configuration;

namespace DbApp
{
    public partial class MainWindow : Window
    {
        private string connectionString;
        private SqlDataAdapter adapter;
        private DataTable phonesTable;

        public MainWindow()
        {
            InitializeComponent();
            // app.config
            connectionString =

```

```
gurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;
```

```
private void Window_Loaded(object sender, RoutedEventArgs e)
```

```
    string sql = "SELECT * FROM Phones";  
    phonesTable = new DataTable();  
    SqlConnection connection=null;  
    try  
    {  
        connection = new SqlConnection(connectionString);  
        SqlCommand command = new SqlCommand(sql, connection);  
        adapter = new SqlDataAdapter(command);  
  
        //  
        adapter.InsertCommand = new SqlCommand("sp_InsertPhone", connection);  
        adapter.InsertCommand.CommandType = CommandType.StoredProcedure;  
        adapter.InsertCommand.Parameters.Add(new SqlParameter("@title", SqlDbType.NVarChar,  
title));  
        adapter.InsertCommand.Parameters.Add(new SqlParameter("@company",  
SqlDbType.NVarChar, 50, "Company"));  
        adapter.InsertCommand.Parameters.Add(new SqlParameter("@price", SqlDbType.Int, 0,  
price));  
        SqlParameter parameter = adapter.InsertCommand.Parameters.Add("@Id", SqlDbType.Int, 0,  
id);  
        parameter.Direction = ParameterDirection.Output;  
  
        connection.Open();  
        adapter.Fill(phonesTable);  
        phonesGrid.ItemsSource = phonesTable.DefaultView;  
    }  
    catch(Exception ex)  
    {  
        MessageBox.Show(ex.Message);  
    }  
    finally  
    {  
        if(connection!=null)  
            connection.Close();  
    }  
}
```

```
private void UpdateDB()
```

```
    SqlCommandBuilder comandbuilder = new SqlCommandBuilder(adapter);  
    adapter.Update(phonesTable);
```

```

private void updateButton_Click(object sender, RoutedEventArgs e)
{
    UpdateDB();
}
private void deleteButton_Click(object sender, RoutedEventArgs e)
{
    if (phonesGrid.SelectedItems != null)
    {
        for (int i = 0; i < phonesGrid.SelectedItems.Count; i++)
        {
            DataRowView datarowView = phonesGrid.SelectedItems[i] as DataRowView;
            if (datarowView != null)
            {
                DataRow dataRow = (DataRow)datarowView.Row;
                dataRow.Delete();
            }
        }
    }
    UpdateDB();
}
}

```

5. ADO.NET
SqlDataAdapter.
, *app.config:*

connectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;

6. ,
System.Configuration.dll.

Window_Loaded SqlDataAdapter:

adapter = new SqlDataAdapter(command);

7. :

adapter.InsertCommand = new SqlCommand("sp_InsertPhone", connection);

:

adapter.Fill(phonesTable);
phonesGrid.ItemsSource = phonesTable.DefaultView;

UpdatedB():

```
private void UpdateDB()
{
    SqlCommandBuilder comandbuilder = new SqlCommandBuilder(adapter);
    adapter.Update(phonesTable);
}
```

```
8.          SqlDataAdapter,
           SqlCommandBuilder.
           adapter.Update().
```

```
adapter.Update
    DataTable
    adapter.Update()
SQL,
```

: DataSet

UpdateDB,

DataGridView

datagrid id.

```
9.
    RowEditEnding
    DataGridView:
```

```
public MainWindow()
{
    InitializeComponent();

    connectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;
    phonesGrid.RowEditEnding += PhonesGrid_RowEditEnding;
}
```

```
private void PhonesGrid_RowEditEnding(object sender, DataGridViewRowEditEndingEventArgs e)
{
    UpdateDB();
}
```


3. ()
4.

1. 1. : „ / „ / .
.. - : « »: - , 2019. - 512 : . - ()
).

2. , . / . . - : , - , 2020. -
336 : 60x90 1/16. - () (7) ISBN 978-5-
906818-41-6

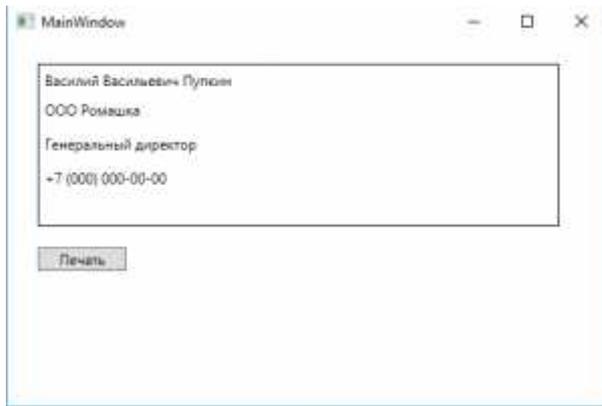
3. . „ . „ 2020.
. „ . „ - : 2020.

4. . . . - 8- . . - :
« », 2019. - 384 .

5. / . . - : , - , 2020. - 384 : 60x90 1/16. - ()
).

. , .NET

WPF. WPF. XAML



XAML:

```
<Window
  xmlns=http://schemas.microsoft.com/winfx/2006/xaml/presentation
  xmlns:x=http://schemas.microsoft.com/winfx/2006/xaml
  xmlns:d=http://schemas.microsoft.com/expression/blend/2008
  xmlns:mc=http://schemas.openxmlformats.org/markup-compatibility/2006
  xmlns:local="clr-namespace:WpfApp1"
  xmlns:Forms="clr-namespace:System.Windows.Forms;assembly=System.Windows.Forms"
  x:Class="WpfApp1.MainWindow"
  mc:Ignorable="d"
  Title="MainWindow" Height="350" Width="525">
  <Grid>
    <Border x:Name="card" BorderBrush="Black" BorderThickness="1"
  HorizontalAlignment="Left" Height="138" Margin="26,20,0,0" VerticalAlignment="Top"
  Width="441">
      <Grid>
        <Label x:Name="FIO" Content="Василий Васильевич Пупкин"
          Margin="0,25,0,82"/>
        <Label x:Name="Company" Content="ООО Ромашка"
          Margin="0,54,0,53"/>
        <Label x:Name="Position" Content="Генеральный директор"
          Margin="0,83,0,24"/>
        <Label x:Name="Phone" Content="+7 (000) 000-00-00" Margin="0,83,0,24"/>
      </Grid>
    </Border>
    <Button x:Name="printBtn" Content="Печать" HorizontalAlignment="Left"
  Margin="26,176,0,0" VerticalAlignment="Top" Width="75" Click="printBtn_Click"/>
  </Grid>
</Window>
```

```
WPF
,
.
Border
,
.
PrintDialog,
PrintVisual.
:
,
Windows
(
,
).
```

```
PrintDialog dialog = new PrintDialog();
```

```
dialog.PrintVisual(this.card, " ");
PrintVisual
.ShowDialog,
```

```
PrintDialog dialog = new PrintDialog();
if (dialog.ShowDialog() == true)
{
    dialog.PrintVisual(this.card, " ");
}
```

PDF.



XAML

10

CMS.
 cms.

- 1.
- 2.
- 3.
- 4.

1. 1. : « » / / - , 2019. - 512 : . - ().
2. : / - , 2020. - 336 : 60x90 1/16. - () (7) ISBN 978-5-906818-41-6
3. 2020.
4. - 8- : « », 2019. - 384 .
5. / - , 2020. - 384 : 60x90 1/16. - ().

() - , (, , ,).

12 500 ().

Oracle VM VirtualBox

, VirtualBox
- Ethernet .

« »

Network Address Translation (NAT)
Network Address Translation (NAT)

NAT
Internet VirtualBox,
NAT,

(internet);

NAT,
NAT - - .

VirtualBox

«net filter». VirtualBox

Wireshark)

VirtualBox

VirtualBox

Ethernet

promiscuous mode.

, VirtualBox

()

web

web

share the same generic network interface),
VirtualBox

(Rarely used modes
(extension pack).

GPL 2.

- PHP version 5.6 or greater
- MySQL version 5.5 or greater

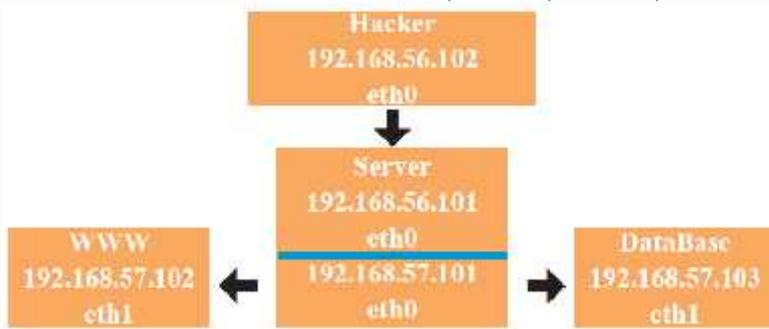
PHP (PHP: Hypertext Preprocessor — «PHP: Personal Home Page Tools — «

»;

MySQL — MySQL Oracle,

MySQL AB. Sun Microsystems, GNU General Public License,

- 4 Ubuntu Server.
: Hacker, Server, WWW, DataBase.



- (2 (-> 1 -> ,).
- IP- 192.168.*.* , * - 0
- 255 (192.168.56.1, 192.168.57.1).
- DHCP , , DHCP . DHCP- IP- DHCP

VirtualBox

- Hacker 2 NAT. (
- Server 3 ip- 192.168.56.*).
- Hacker. - , ip- 192.168.57.*).
- WWW DataBase 2 - . (, 192.168.57.*).

- VirtualBox
- IP- (DHCP).
- /etc/network/interfaces.

nano /etc/network/interfaces

- :
- auto eth1
- iface eth1 inet static
- address 192.168.58.103
- netmask 255.255.255.0
- gateway 192.168.58.102

```
address — ip , netmask — , gateway — , gateway
ip- gateway
```

```
ip- eth1 eth2, - Server
eth0, eth0,
```

- ifconfig
- ping
- Hacker – « »,

– Nmap:

- sudo apt-get install nmap
- Server – Hacker.

CMS WordPress.

- WWW – web- Server, web- DataBase. PHP.

- sudo apt-get install php5 libapache2-mod-php5 php5-mysql /var/www/index.php, :

```
<?php
$link = mysql_connect('192.168.57.101', 'sit', 'sit');
if (!$link) {
die('Error: '. mysql_error());
}
echo 'Ok';

mysql_close($link);
?>
```

```

WWW DataBase,
ip- WWW /index.php,
« ».
• DataBase Mysql –
• sudo apt-get install mysql-server
• MySQL:
• sudo mysql -p
• , MySql
• ( sit)
• “GRANT ALL PRIVILEGES ON *.* TO sit@localhost IDENTIFIED BY ‘sit’ WITH GRANT OPTION
• “GRANT ALL PRIVILEGES ON *.* TO sit@%" IDENTIFIED BY ‘sit’ WITH GRANT OPTION
• /etc/mysql/my.cnf (
• #) bind-address = 127.0.0.1.
2
• CMS Server
• CMS WordPress, Server
• Server
• LAMP (Linux Apache MySQL PHP)
• sudo apt-get update
•
• sudo apt-get install tasksel
•
• sudo tasksel
• LAMP. Enter.
• root ( sit)
• phpmyadmin <sudo apt-get install phpmyadmin>. Apache,
Yes,
• ip- /phpmyadmin (
192.168.1.113/phpmyadmin)
• . ( - root. -
phpmyadmin)
• Databases.
• Create Database (wordpressdb01)
• Create.
• Server. WordPress
• <wget http://wordpress.org/latest.tar.gz>
• /var/www/html <sudo tar -xvpzf latest.tar.gz -C
/var/www/html/>
• 192.168.1.113/wordpress ,
• - wordpressdb01, - root - sit.
• Submit.
• , wp-config.php wordpress.
• Server.
• /var/www/html/wordpress

```

