



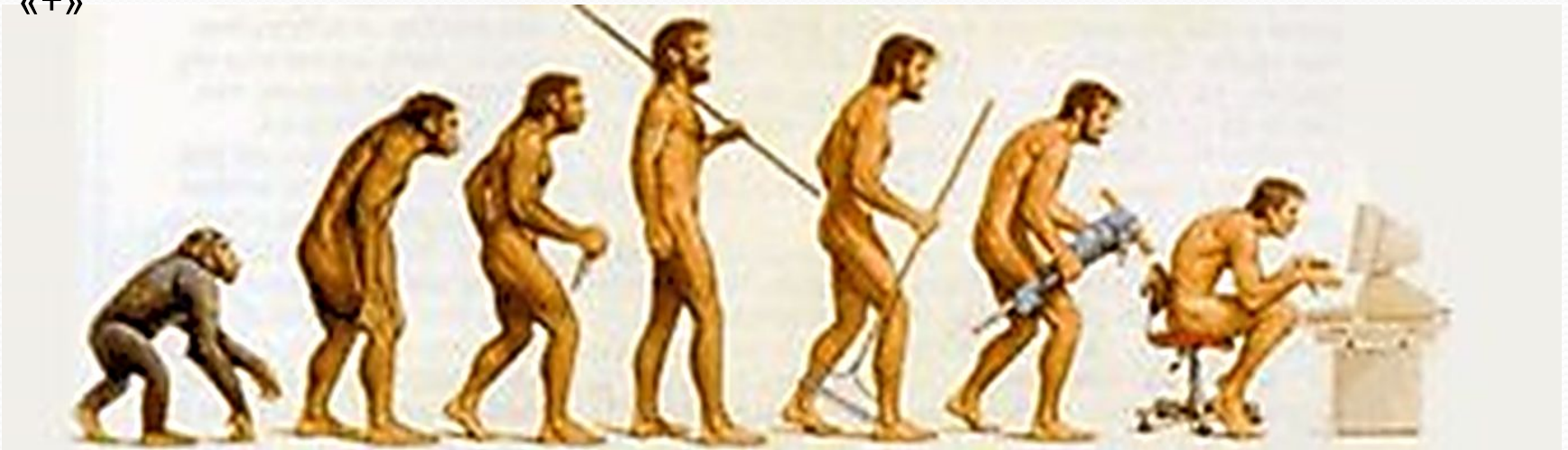
Основы языка программирования

Название «Си шарп» (от англ. sharp — диез) происходит от музыкальной нотации, где знак диез, означает повышение соответствующего ноте звука на полутон, что аналогично названию языка C++, где «++» обозначает инкремент переменной.

Название так же является игрой с цепочкой C → C++ → C++++(C#), так как символ «#» можно составить из 4х знаков «+»

Worldwide, Feb 2018 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Java	22.55 %	-1.1 %
2		Python	21.3 %	+5.6 %
3		PHP	8.53 %	-1.8 %
4	↑	Javascript	8.49 %	+0.4 %
5	↓	C#	8.06 %	-0.6 %
6		C	6.51 %	-1.4 %
7	↑	R	4.23 %	+0.5 %
8	↓	Objective-C	3.86 %	-1.2 %
9		Swift	3.09 %	-0.4 %
10		Matlab	2.34 %	-0.5 %



Среда программирования

The screenshot shows the Microsoft Visual C# 2010 Express IDE. The main window displays the source code for a program named "Program.cs" within the "HelloWorld.Program" namespace. The code is as follows:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello, World!");
            Console.ReadKey();
        }
    }
}
```

The "Обозреватель решений" (Solution Explorer) on the right shows the project structure:

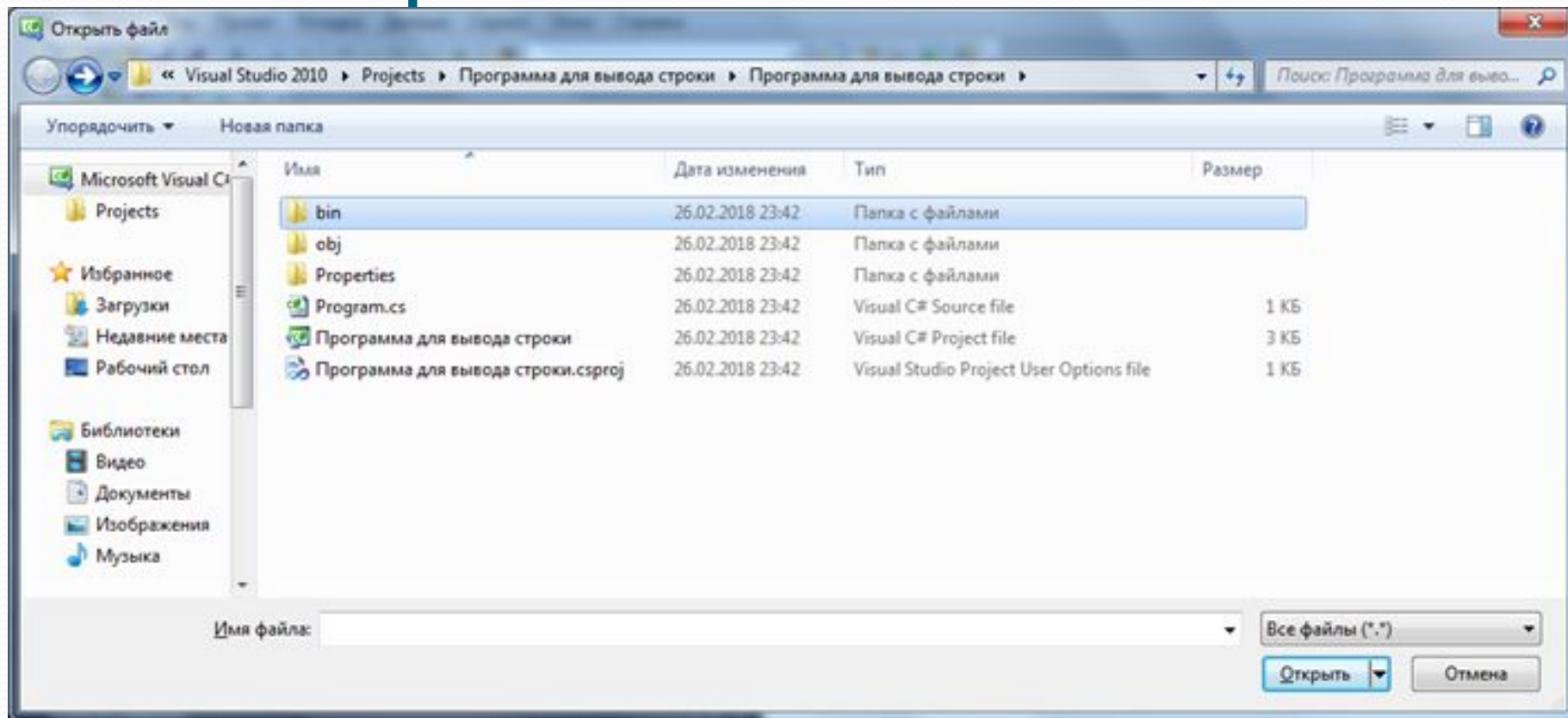
- Решение "Программа для вывода строки" (проектов: 1)
 - Программа для вывода строки
 - Properties
 - Ссылки
 - Program.cs

The "Список ошибок" (Error List) at the bottom shows one error:

Опис...	Файл	Строка	Столбец	Проект
1 Требуется закрывающая скобка "}"	Program.cs	15	6	Программа для вывода строки

The status bar at the bottom indicates "Готово" (Ready) and shows the current cursor position: "Строка 1" (Line 1), "Столбец 1" (Column 1), "Знак 1" (Character 1).

Хранение проекта



Знакомство с C#

```
/* =====  
 * Подключение необходимых пространств имен  
 * Здесь подключаем System, т.к. в нем объявлены все классы библиотеки .NET  
=====*/  
using System;  
  
namespace HelloWorld  
{  
    // программа оформляется в виде объекта класса (т.к. C# - это объектно-ориентированный язык)  
  
    class Program  
    {  
        // код основной программы здесь (главная процедура)  
  
        static void Main(string[] args)  
        {  
            Console.WriteLine("Hello world!");    // вывод на экран сообщения "Hello world!"  
            Console.ReadKey();                    // ожидание нажатия любой клавиши пользователем  
        }  
    }  
}
```

Состав языка

● Символы:

- буквы: A-Z, a-z, _, буквы нац. алфавитов
- цифры: 0-9, A-F
- спец. символы: +, *, {, ...
- пробельные символы

■ Лексемы:

- константы 2 0.11 "Вася"
- имена Vasia a _11
- ключевые слова double do if
- знаки операций + - =
- разделители ; [] ,

■ Выражения

- выражение - правило вычисления значения: $a + b$

■ Операторы

- исполняемые: $c = a + b;$
- описания: `double a, b;`

C#

Вид	Примеры
<u>Булевские</u>	<u>true</u> <u>false</u>
<u>Целые</u> дес.	8 199226 0Lu
<u>шестн.</u>	<u>0xA</u> <u>0x1B8</u> <u>0X00FFL</u>
<u>Веществ.</u> с тчк	5.7 .001f 35m
<u>с порядком</u>	<u>0.2E6</u> <u>.11e-3</u> <u>5E10</u>
<u>Символьные</u>	<u>'A'</u> <u>'\x74'</u> <u>'\o'</u> <u>'\uA81B'</u> <u>Строковые</u>
	"Здесь был Vasia"
	"\tЗначение r=\xF5\n"
	"Здесь был \u0056\u0061"
	<u>@\"C:\temp\file1.txt"</u>
<u>Константа null</u>	null

Имена (идентификаторы)

- имя должно начинаться с буквы или _;
- имя должно содержать только буквы, знак подчеркивания и цифры;
- прописные и строчные буквы различаются;
- длина имени практически не ограничена.
- имена не должны совпадать с ключевыми словами, однако допускается: @if, @float...
- в именах можно использовать управляющие последовательности Unicode

Примеры правильных имен:

Vasia, Вася, _13, \u00F2\u01DD, @while.

Примеры неправильных имен:

2late, Big gig, Б#г

разделители

- *Ключевые слова* — идентификаторы, имеющие специальное значение для компилятора. Их можно использовать только в том смысле, в котором они определены.
 - Например, для оператора перехода определено слово **goto**.
- *Знак операции* — один или более символов, определяющих действие над операндами. Внутри знака операции пробелы не допускаются.
 - Например, сложение +, деление /, сложное присваивание %=.
- Операции делятся на *унарные* (с одним операндом), *бинарные* (с двумя) и *тернарную* (с тремя).
- *Разделители* используются для разделения или, наоборот, группирования элементов. Примеры разделителей: скобки, точка, запятая.

Ключевые слова C#

<code>abstract</code>	<code>as</code>	<code>base</code>	<code>bool</code>	<code>break</code>
<code>byte</code>	<code>case</code>	<code>catch</code>	<code>char</code>	<code>checked</code>
<code>class</code>	<code>const</code>	<code>continue</code>	<code>decimal</code>	<code>default</code>
<code>delegate</code>	<code>do</code>	<code>double</code>	<code>else</code>	<code>enum</code>
<code>event</code>	<code>explicit</code>	<code>extern</code>	<code>false</code>	<code>finally</code>
<code>fixed</code>	<code>float</code>	<code>for</code>	<code>foreach</code>	<code>goto</code>
<code>if</code>	<code>implicit</code>	<code>in</code>	<code>int</code>	<code>interface</code>
<code>internal</code>	<code>is</code>	<code>lock</code>	<code>long</code>	<code>namespace</code>
<code>new</code>	<code>null</code>	<code>object</code>	<code>operator</code>	<code>out</code>
<code>override</code>	<code>params</code>	<code>private</code>	<code>protected</code>	<code>public</code>
<code>readonly</code>	<code>ref</code>	<code>return</code>	<code>sbyte</code>	<code>sealed</code>
<code>short</code>	<code>sizeof</code>	<code>stackalloc</code>	<code>static</code>	<code>string</code>
<code>struct</code>	<code>switch</code>	<code>this</code>	<code>throw</code>	<code>true</code>
<code>try</code>	<code>typeof</code>	<code>uint</code>	<code>ulong</code>	<code>unchecked</code>
<code>unsafe</code>	<code>ushort</code>	<code>using</code>	<code>virtual</code>	<code>void</code>
<code>volatile</code>	<code>while</code>			

ЦЕЛЫЕ

Название	Ключевое слово	Тип .NET	Диапазон значений	Описание	Размер, бит
Булевский	bool	Boolean	true, false		
Целые	sbyte	SByte	-128 — 127	знаковое	8
	byte	Byte	0 — 255	беззнаковое	8
	short	Int16	-32768 — 32767	знаковое	16
	ushort	UInt16	0 — 65535	беззнаковое	16
	int	Int32	$\approx(-210^9 — 210^9)$	знаковое	32
	uint	UInt32	$\approx(0 — 410^9)$	беззнаковое	32
	long	Int64	$\approx(-910^{18} — 910^{18})$	знаковое	64
	ulong	UInt64	$\approx(0 — 1810^{18})$	беззнаковое	64

Остальные

Название	Ключевое слово	Тип .NET	Диапазон значений	Описание	Размер бит
Символьный	<code>char</code>	<code>Char</code>	U+0000 — U+ffff	символ Unicode	16
Вещественные	<code>float</code>	<code>Single</code>	$1.5 \cdot 10^{-45}$ — $3.4 \cdot 10^{38}$	7 цифр	32
	<code>double</code>	<code>Double</code>	$5.0 \cdot 10^{-324}$ — $1.7 \cdot 10^{308}$	15-16 цифр	64
Финансовый	<code>decimal</code>	<code>Decimal</code>	$1.0 \cdot 10^{-28}$ — $7.9 \cdot 10^{28}$	28-29 цифр	128
Строковый	<code>string</code>	<code>String</code>	длина ограничена объемом доступной памяти	строка из символов Unicode	
<code>object</code>	<code>object</code>	<code>Object</code>	можно хранить все, что угодно	всеобщий предок	

Арифметические операторы

Оператор	Действие
+	Сложение
-	Вычитание, унарный минус
*	Умножение
/	Деление
%	Деление по модулю
--	Декремент
++	Инкремент

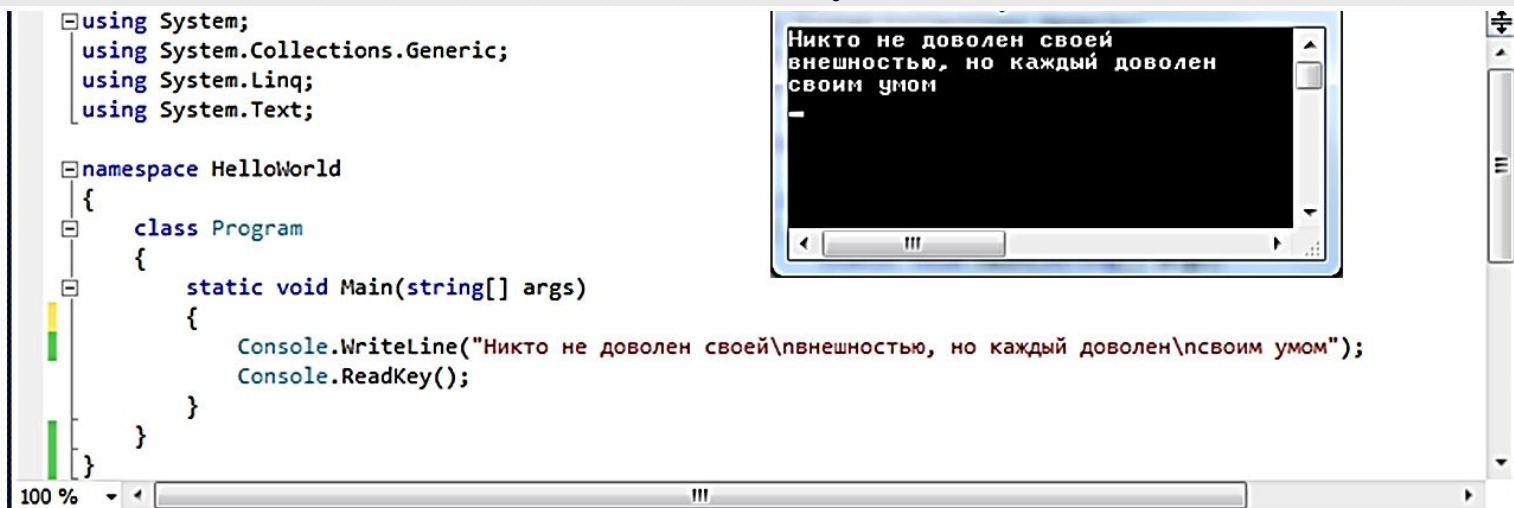
```
static void Main(string[] args)
{
    int x = 18, y = 12;

    int sum = x + y;
    int diff = x - y;
    int mul = x * y;
    int div = x / y;    // целая часть деления (частное)
    int mod = x % y;   // остаток от деления

    Console.WriteLine("Sum = \t\t{0}\nDifference = \t{1}\nProduct = \t{2}\nQuotient = \t{3}\nRemainder = \t{4}\n",
        sum, diff, mul, div, mod);
}
```

Управляющие последовательности в C#

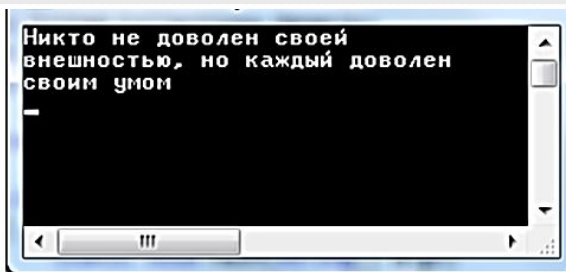
Вид	Наименование
\a	Звуковой сигнал
\b	Возврат на шаг
\f	Перевод страницы (формата)
\n	Перевод строки
\r	Возврат каретки
\t	Горизонтальная табуляция
\v	Вертикальная табуляция
\\	Обратная косая черта
\'	Апостроф
\"	Кавычка
\0	Нуль-символ



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Никто не доволен своей\nвнешностью, но каждый доволен\nсвоим умом");
            Console.ReadKey();
        }
    }
}
```

100 %



Инкременты и декременты

```
// инкремент справа (сначала идет присваивание, а потом sum наращивается на 1)  
int newSum = sum++;
```

```
Console.WriteLine( "NOT incremented sum = " + newSum );    // что здесь выведется на экран?
```

```
// инкремент слева (сначала sum наращивается на 1, а потом идет присваивание newSum)  
newSum = ++sum;
```

```
Console.WriteLine("Incremented sum = " + newSum);          // и что здесь выведется на экран?
```

```
// это эквивалентно записи sum = sum + 1  
sum++;
```

```
Sum = 30  
Difference = 6  
Product = 216  
Quotient = 1  
Remainder = 6  
  
NOT incremented sum = 30  
Incremented sum = 32
```

Составной оператор

```
int number = 30;
```

```
number *= 10;    // то же самое, что number = number * 10
```

```
number += 17;   // то же самое, что number = number + 17
```

Класс Math из пространства имен System

```
static void Main(string[] args)
{
    double radius = 2.0;
    double area = Math.PI * Math.Pow( radius, 2 );
    Console.WriteLine( "2 * pi * r^2 = " + area );

    int maxValue = Math.Max( 5, 17 );
    Console.WriteLine( "max { 5, 17 } = " + maxValue );

    double res1 = Math.Log( Math.E );
    Console.WriteLine( "ln(e) = " + res1 );

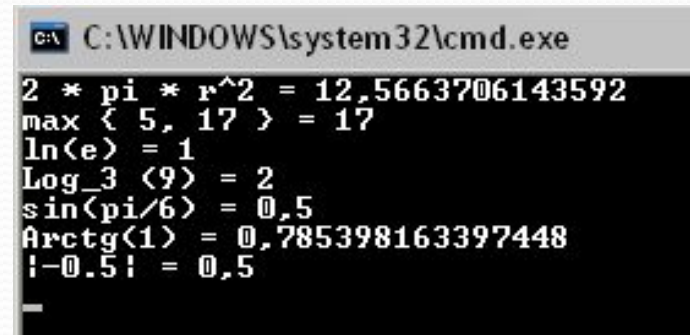
    double res2 = Math.Log( 9, 3 );
    Console.WriteLine( "Log_3 (9) = " + res2 );

    double res3 = Math.Sin(Math.PI / 6);
    Console.WriteLine( "sin(pi/6) = " + res3 );

    double res4 = Math.Atan( 1.0 );
    Console.WriteLine( "Arctg(1) = " + res4 );

    double res5 = Math.Abs( -0.5 );
    Console.WriteLine( "|-0.5| = " + res5 );

    Console.ReadKey();
}
```



```
C:\WINDOWS\system32\cmd.exe
2 * pi * r^2 = 12,5663706143592
max { 5, 17 } = 17
ln(e) = 1
Log_3 (9) = 2
sin(pi/6) = 0,5
Arctg(1) = 0,785398163397448
|-0.5| = 0,5
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
namespace A
{
    class Class1
    {
        static void Main()
        {
            string s = Console.ReadLine(); // ввод строки

            char c = (char)Console.Read(); // ввод символа
            Console.ReadLine();

            string buf; // буфер для ввода чисел
            buf = Console.ReadLine();
            int i = Convert.ToInt32( buf ); // преобразование в целое

            buf = Console.ReadLine();
            double x = Convert.ToDouble( buf ); // преобразование в вещ.

            buf = Console.ReadLine();
            double y = double.Parse( buf ); // преобразование в вещ.
            Console.WriteLine("строка="+s+"\nsимвол="+c+"\nцелое="+i+"\nвещественные=\n"+x+"\n"+y);

            Console.ReadKey();
        }
    }
}
```

ВВОД И ВЫВОД С КОНСОЛИ

задание

Вариант 1

$$z_1 = 2 \sin^2 (3\pi - 2\alpha) \cos^2 (5\pi + 2\alpha)$$

$$z_2 = \frac{1}{4} - \frac{1}{4} \sin \left(\frac{5}{2}\pi - 8\alpha \right)$$

Вариант 2

$$z_1 = \cos \alpha + \sin \alpha + \cos 3\alpha + \sin 3\alpha$$

$$z_2 = 2\sqrt{2} \cos \alpha \cdot \sin \left(\frac{\pi}{4} + 2\alpha \right)$$

Вариант 3

$$z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha + 1 - 2 \sin^2 2\alpha} \quad z_2 = 2 \sin \alpha$$

Вариант 4

$$z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha - \cos 3\alpha + \cos 5\alpha}$$

$$z_2 = \operatorname{tg} 3\alpha$$

Вариант 5

$$z_1 = 1 - \frac{1}{4} \sin^2 2\alpha + \cos 2\alpha \quad z_2 = \cos^2 \alpha + \cos^4 \alpha$$

Вариант 6

$$z_1 = \cos \alpha + \cos 2\alpha + \cos 6\alpha + \cos 7\alpha$$

$$z_2 = 4 \cos \frac{\alpha}{2} \cdot \cos \frac{5}{2}\alpha \cdot \cos 4\alpha$$

Вариант 7

$$z_1 = \cos^2 \left(\frac{3}{8}\pi - \frac{\alpha}{4} \right) - \cos^2 \left(\frac{11}{8}\pi + \frac{\alpha}{4} \right)$$

$$z_2 = \frac{\sqrt{n}}{2} \sin \frac{\alpha}{2}$$

Вариант 8

$$z_1 = \cos^4 x + \sin^2 y + \frac{1}{4} \sin^2 2x - 1$$

$$z_2 = \sin(y+x) \cdot \sin(y-x)$$

Вариант 9

$$z_1 = (\cos \alpha - \cos \beta)^2 - (\sin \alpha - \sin \beta)^2$$

$$z_2 = -4 \sin^2 \frac{\alpha - \beta}{2} \cdot \cos(\alpha + \beta)$$

Вариант 10

$$z_1 = \frac{\sin \left(\frac{\pi}{2} + 3\alpha \right)}{1 - \sin(3\alpha - \pi)}$$

$$z_2 = \operatorname{ctg} \left(\frac{5}{4}\pi + \frac{3}{2}\alpha \right)$$

Вариант 11

$$z_1 = \frac{1 - 2 \sin^2 \alpha}{1 + \sin 2\alpha} \quad z_2 = \frac{1 - \operatorname{tg} \alpha}{1 + \operatorname{tg} \alpha}$$

Вариант 12

$$z_1 = \frac{\sin 4\alpha}{1 + \cos 4\alpha} \cdot \frac{\cos 2\alpha}{1 + \cos 2\alpha}$$

$$z_2 = \operatorname{ctg} \left(\frac{3}{2}\pi - \alpha \right)$$

Вариант 13

$$z_1 = \frac{\sin \alpha + \cos(2\beta - \alpha)}{\cos \alpha - \sin(2\beta - \alpha)} \quad z_2 = \frac{1 + \sin 2\beta}{\cos 2\beta}$$

Вариант 14

$$z_1 = \frac{\cos \alpha + \sin \alpha}{\cos \alpha - \sin \alpha} \quad z_2 = \operatorname{tg} 2\alpha + \sec 2\alpha$$

Вариант 15

$$z_1 = \frac{\sqrt{2b+2}\sqrt{b^2-4}}{\sqrt{b^2-4}+b+2} \quad z_2 = \frac{1}{\sqrt{b+2}}$$

Вариант 16

$$z_1 = \frac{x^2 + 2x - 3 + (x+1)\sqrt{x^2-9}}{x^2 - 2x - 3 + (x-1)\sqrt{x^2-9}}$$

$$z_2 = \sqrt{\frac{x+3}{x-3}}$$

Вариант 17

$$z_1 = \frac{\sqrt{(3m+2)^2 - 24m}}{3\sqrt{m} - \frac{2}{\sqrt{m}}} \quad z_2 = -\sqrt{m}$$

Вариант 18

$$z_1 = \left(\frac{a+2}{\sqrt{2a}} - \frac{a}{\sqrt{2a+2}} + \frac{2}{a-\sqrt{2a}} \right) \cdot \frac{\sqrt{a}-\sqrt{2}}{a+2}$$

$$z_2 = \frac{1}{\sqrt{a+\sqrt{2}}}$$

Вариант 19

$$z_1 = \left(\frac{1+a+a^2}{2a+a^2} + 2 - \frac{1-a+a^2}{2a-a^2} \right)^{-1} (5-2a^2)$$

$$z_2 = \frac{4-a^2}{2}$$

Вариант 20

$$z_1 = \frac{(m-1)\sqrt{m} - (n-1)\sqrt{n}}{\sqrt{m^2n} + nm + m^2 - m}$$

$$z_2 = \frac{\sqrt{m} - \sqrt{n}}{m}$$

Ссылки на учебные материалы

Курс лекций по языку программирования С#

Автор: Татьяна Павловская

<https://www.intuit.ru/studies/courses/629/485/info>

Курс лекций по языку программирования С#

http://pta-ipm.narod.ru/cs_present.html

Ссылка на дистрибутив

Microsoft_Visual_C#_2010_10.0_Express_Web_Installer.exe

<https://yadi.sk/d/bazX67oq9qijv>