**Практическая работа № 3.**

**Программирование алгоритмов линейной структуры на языке c#**

**Цель работы**: изучить основные принципы построения программ на языке программирования С#, изучить порядок действий при вычислении выражений; приобрести навыки в записи выражений и использовании стандартных функций; овладеть практическими навыками в программировании линейных алгоритмов и отладке программ, тем самым закрепив полученные теоретические сведения.

|  |  |  |
| --- | --- | --- |
| №  п/п | Ф.И.О. | НОМЕР ВАРИАНТА |
|  | Балицкая Анастасия |  |
|  | Вартик Максим |  |
|  | Вдовиченко Ярослав |  |
|  | Веряскин Иван |  |
|  | Гоцалюк Арзу |  |
|  | Градинар Илья |  |
|  | Гребенщиков Данил |  |
|  | Завтонев Роман |  |
|  | Захаров Станислав |  |
|  | Иванов Богдан |  |
|  | Ильинский Никита |  |
|  | Каплин Егор |  |
|  | Коваленко Михаил |  |
|  | Колак Михаил |  |
|  | Левицкая Мария |  |
|  | Милюкова Екатерина |  |
|  | Никифорова Елизавета |  |
|  | Подолян Евгений |  |
|  | Попов Максим |  |
|  | Родичев Максим |  |
|  | Рознован Роман |  |
|  | Тизул Игорь |  |
|  | Тихончук София |  |
|  | Толоченко Павел |  |
|  | Фокша Кирилл |  |
|  | Чебан Евгений | 15 |
|  | Чебан Илья | 11 |
|  | Юраш Ренат | 4 |

**Теоретическая часть**

Если в программе все операторы выполняются последовательно, один за другим, такая программа называется *линейной*.

Программа на С# состоит из функций. *Функция* — это именованная последовательность операторов. Функция состоит из заголовка и тела. Основная функция программы должна иметь имя main, указывающее, что именно с нее требуется начинать выполнение. За именем функции в скобках обычно следует список передаваемых ей параметров. Перед именем записан тип значения, возвращаемого функцией в точку ее вызова.

* + 1. Состав языка

*Алфавит* – совокупность допустимых в языке символов. Алфавит языка С# включает:

1. прописные и строчные латинские буквы и буквы национальных алфавитов (включая кириллицу);
2. арабские цифры от 0 до 9, шестнадцатеричные цифры от A до F;
3. специальные знаки: " { } , | ; [ ] ( ) + - / % \* . \ ' : ? < = > ! & ~ ^ @ \_
4. пробельные символы: пробел, символ табуляции, символ перехода на новую строку.

Из символов алфавита формируются лексемы языка: идентификаторы, ключевые (зарезервированные) слова, знаки операций, константы, разделители (скобки, точка, запятая, пробельные символы).

Границы лексем определяются другими лексемами, такими, как разделители или знаки операций. В свою очередь лексемы входят в состав выражений (выражение задает правило вычисления некоторого значения) и операторов (оператор задает законченное описание некоторого действия).

*Идентификатор*– это имя программного элемента: константы, переменной, метки, типа, класса, объекта, метода и т.д. Идентификатор может включать латинские буквы и буквы национальных алфавитов, цифры и символ подчеркивания. Прописные и строчные буквы различаются, например, myname, myName и MyName — три различных имени. Первым символом идентификатора может быть буква или знак подчеркивания, но не цифра.

Пробелы внутри имен не допускаются. Язык С# не налагает никаких ограничений на длину имен, однако для удобства чтения и записи кода не стоит делать их слишком длинными.

Для улучшения читабельности кода программным элементам следует давать осмысленные имена, составленные в соответствии с определенными правилами. Существует несколько видов нотаций – соглашений о правилах создания имен.

*Ключевые слова*– зарезервированные идентификаторы, имеющие специальное значение для компилятора, например, main, int и т.д., их можно использовать только по прямому назначению. С ключевыми словами и их назначением можно ознакомиться в справочной системе С#.

* + 1. Типы данных

С# является языком со строгой типизацией. В нем необходимо объявлять тип всех создаваемых программных элементов (например, переменных и т. д.), что позволяет компилятору предотвращать возникновение ошибок, следя за тем, чтобы объектам присваивались значения только разрешенного типа. Тип программного элемента сообщает компилятору о его размере.

В С# типы делятся на две группы: *базовые*типы, предлагаемые языком, и типы, *определяемые пользователем*. Также типы С# разбиваются на две другие категории: *размерные типы*и *ссылочные типы.*Почти все базовые типы являются размерными типами. Исключение составляют типы Object и String. Принципиальное различие между размерными и ссылочными типами состоит в способе хранения их значений в памяти. В первом случае фактическое значение хранится в стеке. Адрес переменной ссылочного типа тоже хранится в стеке, но сам объект хранится в куче.

Основные типы переменных в С#

|  |  |  |
| --- | --- | --- |
| *Тип* | *Описание* | *Диапазон значений* |
| bool | логический | true и false |
| char | символьный | 0 … 65535 |
| string | строка | последовательность символов |
| sbyte | байт | -128 … 127 |
| byte | байт | 0 … 255 |
| short | короткое целое | -32768 … 32767 |
| ushort | короткое целое | 0 … 65535 |
| Int | целое | -2147483648 …2147483647 |
| Uint | целое | 0 … 4294967295 |
| long | длинное целое |  |
| ulong | длинное целое |  |
| float | вещественное | 3,4e-38 … 3,4e+38 |
| double | с повышенной точностью | 1,7e-308 … 1,7e+308 |

*Переменная* представляет собой типизированную область памяти. Программист создает переменную, объявляя ее тип и указывая имя. При объявлении переменной ее можно инициализировать (присвоить ей начальное значение), а затем в любой момент ей можно присвоить новое значение, которое заменит собой предыдущее.

*Константа*- это переменная, значение которой нельзя изменить. Константы бывают трех видов: *литералы, символические константы*и *перечисления.*

* + 1. Организация ввода-вывода данных

Программа при вводе данных и выводе результатов взаимодействует с внешними устройствами. Совокупность стандартных устройств ввода (клавиатура) и вывода (экран) называется *консолью*. В языке С# нет операторов ввода и вывода. Вместо них для обмена данными с внешними устройствами используются специальные объекты. В частности, для работы с консолью используется стандартный класс Console, определенный в пространстве имен System.

*Ввод данных*

Для ввода данных используется метод ReadLine, реализованный в классе Console. Особенностью данного метода является то, что в качестве результата он возвращает строку (string).

*Пример*:

static void Main()

{

string s = Console.ReadLine();

Console.WriteLine(s);

}

Для того чтобы получить числовое значение необходимо воспользоваться преобразованием данных.

*Пример*:

static void Main()

{

string s = Console.ReadLine();

int x = int.Parse(s); //преобразование строки в число

Console.WriteLine(x);

}

Или сокращенный вариант:

static void Main()

{

//преобразование введенной строки в число

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

Console.WriteLine(x);

}

Преобразование типов данных осуществляется с помощью класса Convertили методаParse. Для преобразования строкового представления целого числа в типintиспользуется методint.Parse(), который реализован для всех числовых типов данных. Если требуется преобразовать строковое представление в вещественное, то используется методfloat.Parse()илиdouble.Parse(). В случае, если соответствующее преобразование выполнить невозможно, то выполнение программы прерывается и генерируется исключениеSystem.FormatExeption(входная строка имела неверный формат).

*Класс Convert*, определенный в пространстве имен System обеспечивает необходимые преобразования между различными типами. Методыкласса*Convert* поддерживают общий способ выполнения преобразований между типами. *Класс Convert* содержит 15 статических *методов* вида To <Type> (ToBoolean(),...ToUInt64()), где Type может принимать значения от Boolean до UInt64 для всех встроенных типов. Единственным *исключением* является тип object, - *метода* ToObject нет по понятным причинам, поскольку для всех типов существует *неявное преобразование* к типу object.

*Вывод данных*

Для вывода данных используется метод WriteLine, реализованный в классе Console, который позволяет организовывать вывод данных на экран. Однако существует несколько способов применения данного метода:

1. Console.WriteLine(x); //на экран выводится значение идентификатора х
2. Console.WriteLine("x=" + x +"y=" + y); /\* на экран выводится строка, образованная последовательным слиянием строки "x=", значения x, строки "у=" и значения у \*/
3. Console.WriteLine("x={0} y={1}", x, y); /\* на экран выводится строка, формат которой задан первым аргументом метода, при этом вместо параметра {0} выводится значение x, а вместо {1} – значение y\*/

*Использование управляющих последовательностей:*

*Управляющей последовательностью* называют определенный символ, предваряемый обратной косой чертой. Данная совокупность символов интерпретируется как одиночный символ и используется для представления кодов символов, не имеющих графического обозначения (например, символа перевода курсора на новую строку) или символов, имеющих специальное обозначение в символьных и строковых константах (например, апостроф).

Рассмотрим управляющие символы:

|  |  |
| --- | --- |
| *Вид* | *Наименование* |
| \a | Звуковой сигнал |
| \b | Возврат на шаг назад |
| \f | Перевод страницы |
| \n | Перевод строки |
| \r | Возврат каретки |
| \t | Горизонтальная табуляция |
| \v | Вертикальная табуляция |
| \\ | Обратная косая черта |
| \' | Апостроф |
| \" | Кавычки |

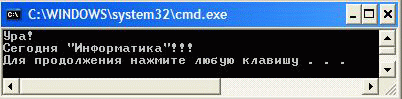
*Пример*:

static void Main()

{

Console.WriteLine("Ура!\nСегодня \"Информатика\"!!!");

}



*Управление размером поля вывода:*

Первым аргументом WriteLine указывается строка вида {n, m} – где n определяет номер идентификатора из списка аргументов метода WriteLine, а m – количество позиций (размер поля вывода), отводимых под значение данного идентификатора. При этом значение идентификатора выравнивается по правому краю. Если выделенных позиций для размещения значения идентификатора окажется недостаточно, то автоматически добавиться необходимое количество позиций.

*Пример*:

static void Main()

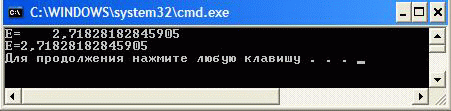
{

double x= Math.E;

Console.WriteLine("E={0,20}", x);

Console.WriteLine("E={0,10}", x);

}



*Управление размещением вещественных данных:*

Первым аргументом WriteLine указывается строка вида {n: ##.###} – где n определяет номер идентификатора из списка аргументов метода WriteLine, а ##.### определяет формат вывода вещественного числа. В данном случае под целую часть числа отводится две позиции, под дробную – три. Если выделенных позиций для размещения целой части значения идентификатора окажется недостаточно, то автоматически добавиться необходимое количество позиций.

*Пример*:

static void Main()

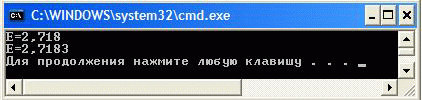
{

double x= Math.E;

Console.WriteLine("E={0:##.###}", x);

Console.WriteLine("E={0:.####}", x);

}



*Управление форматом числовых данных:*

Первым аргументом WriteLine указывается строка вида {n: <спецификатор>m} – где n определяет номер идентификатора из списка аргументов метода WriteLine, <спецификатор> - определяет формат данных, а m – количество позиций для дробной части значения идентификатора. В качестве спецификаторов могут использоваться следующие значения:

|  |  |  |
| --- | --- | --- |
| *Параметр* | *Формат* | *Значение* |
| C или c | Денежный. По умолчанию ставит знак р. Изменить его можно с помощь объекта NumberFormatInfo | Задается количество десятичных разрядов |
| D или d | Целочисленный (используется только с целыми числами) | Задается минималь-ное количество цифр. При необходимости результат дополняется начальными нулями |
| E или e | Экспоненциальное представление чисел | Задается количество символов после запятой. По умолчанию используется 6 |
| F или f | Представление чисел с фиксированной точкой | Задается количество символов после запятой |
| G или g | Общий формат (или экспоненциальный, или с фиксированной точкой) | Задается количество символов после запятой. По умолчанию выводится целая часть |
| N или n | Стандартное форматирование с использованием запятых и пробелов в качестве разделителей между разрядами | Задается количество символов после запятой. По умолчанию – 2, если число целое, то ставятся нули |
| X или x | Шестнадцатеричный формат |  |
| P или p | Процентный |  |

Порядок вычислений определяется приоритетом операций. Основные правила соответствуют принятым в математике. Операции языка С# (с учетом убывания приоритета):

**1. Инкремент (++) и декримент(--).**

Эти операции имеют две формы записи - *префиксную,*когда операция записывается перед операндом, и *постфиксную* - операция записывается после операнда. Префиксная операция инкремента (декремента) увеличивает (уменьшает) свой операнд и возвращает измененное значение как результат. Постфиксные версии инкремента и декремента возвращают первоначальное значение операнда, а затем изменяют его.

*Пример*:

|  |  |
| --- | --- |
| static void Main()  {  int i = 3, j = 4;  Console.WriteLine("{0} {1}", i, j);  Console.WriteLine("{0} {1}", ++i, --j);  Console.WriteLine("{0} {1}", i++, j--);  Console.WriteLine("{0} {1}", i, j);  } | *Результат работы программы:*  3 4  4 3  4 3  5 2 |

**2. Операция new.**Используется для создания нового объекта. С помощью ее можно создавать как объекты ссылочного типа, так и размерные, например:

object z=new object();

int i=new int(); // то же самое, что и int i =0;

3. Отрицание:

* Арифметическое отрицание (-) – меняет знак операнда на противоположный.
* Логическое отрицание (!) – определяет операцию инверсия для логического типа.

*Пример*:

|  |  |
| --- | --- |
| static void Main()  {  int i = 3, j=-4;  bool a = true, b=false;  Console.WriteLine("{0} {1}", -i, -j);  Console.WriteLine("{0} {1}", !a, !b);  } | *Результат работы программы:*  -3 4  False True |

**4. Явное преобразование типа.**Используется для явного преобразования из одного типа в другой. Формат операции:

(тип) выражение;

*Пример*:

|  |  |
| --- | --- |
| static void Main()  {  int i = -4;  byte j = 4;  int a = (int)j; //преобразование без потери точности  byte b = (byte)i; //преобразование с потерей точности  Console.WriteLine("{0} {1}", a, b);  } | *Результат работы программы:*  4 252 |

5. Умножение (\*), деление (/) и деление с остатком (%).

Операции умножения и деления применимы для целочисленных и вещественных типов данных. Для других типов эти операции применимы, если для них возможно неявное преобразование к целым или вещественным типам. При этом тип результата равен "наибольшему" из типов операндов, но не менее int. Если оба операнда при делении целочисленные, то и результат тоже целочисленный.

*Пример*:

|  |  |
| --- | --- |
| static void Main()  {  int i = 100, j = 15;  double a = 14.2, b = 3.5;  Console.WriteLine("{0} {1} {2}", i\*j, i/j, i%j);  Console.WriteLine("{0} {1} {2}", a \* b, a / b, a % b);  } | *Результат работы программы:*  1500 6 10  49.7 4.05714285714286 0.1999999999999999 |

**6. Сложение (+) и вычитание (-).** Операции сложения и вычитания применимы для целочисленных и вещественных типов данных. Для других типов эти операции применимы, если для них возможно неявное преобразование к целым или вещественным типам.

**7. Операции отношения ( <, <=, >, >=, ==, !=)**. Операции отношения сравнивают значения левого и правого операндов. Результат операции логического типа: true - если значения совпадают, false - в противном случае.

*Пример*:

|  |  |
| --- | --- |
| static void Main()  {  int i = 15, j = 15;  Console.WriteLine(i<j); //меньше  Console.WriteLine(i<=j); //меньше или равно  Console.WriteLine(i>j); //больше  Console.WriteLine(i>=j); //больше или равно  Console.WriteLine(i==j); //равно  Console.WriteLine(i!=j); //не равно  } | *Результат работы программы:*  False  True  False  True  True  False |

**8. Логические операции:** И (&&), ИЛИ (||).

Логические операции применяются к операндам логического типа. Результат логической операции И имеет значение истина тогда и только тогда, когда оба операнда принимают значение истина. Результат логической операции ИЛИ имеет значение истина тогда и только тогда, когда хотя бы один из операндов принимает значение истина.

*Пример*:

static void Main()

{

Console.WriteLine("x y x и y x или y");

Console.WriteLine("{0} {1} {2} {3}", false, false, false&&false, false||false);

Console.WriteLine("{0} {1} {2} {3}", false, true, false&&true, false||true);

Console.WriteLine("{0} {1} {2} {3}", true, false, true&&false, true||false);

Console.WriteLine("{0} {1} {2} {3}", true, true, true&&true, true||true);

}

*Результат работы программы:*

x y x и y x или y

False False False False

False True False True

True False False True

True True True True

**9. Операции присваивания:** =, +=, -= и т.д.

Формат операции *простого присваивания* (=): **операнд\_2 = операнд\_1;**

В результате выполнения этой операции вычисляется значение операнда\_1, и результат записывается в операнд\_2. Выражение вида a=b=c=100 выполняется справа налево: результатом выполнения c=100 является число 100, которое затем присваивается переменной b, результатом чего опять является 100, которое присваивается переменной a.

Кроме простой операции присваивания существуют *сложные операции присваивания*, например, умножение с присваиванием (\*=), деление с присваиванием (/=), остаток от деления с присваиванием (%=), сложение с присваиванием (+=), вычитание с присваиванием (-=) и т.д.

В сложных операциях присваивания, например, при *сложении с присваиванием*, к операнду\_2 прибавляется операнд\_1, и результат записывается в операнд\_2. То есть, выражение с += а является более компактной записью выражения с = с + а.

Если в одном выражении соседствуют операции одного приоритета, то операции присваивания и условная операции выполняются справа налево, а остальные наоборот. Если необходимо изменить порядок выполнения операций, то в выражении необходимо поставить круглые скобки.

* + 1. Выражения и преобразование типов

*Выражение* - это синтаксическая единица языка, определяющая способ вычисления некоторого значения. Выражения состоят из операндов, операций и скобок. Каждый операнд является в свою очередь выражением или одним из его частных случаев - константой, переменной или функций.

В языке С# предусмотрено вычисление различных математических функций. Для их использования необходимо обратиться к классу Math, который содержит стандартные математические функции. Этот класс содержит два статических поля, задающих константы E и PI, а также 23 статических метода.

Методы задают:

* тригонометрические функции - Sin, Cos, Tan;
* обратные тригонометрические функции - ASin, ACos, ATan, ATan2 (sinx, cosx);
* гиперболические функции - Tanh, Sinh, Cosh;
* экспоненту и логарифмические функции - Exp, Log, Log10;
* модуль, корень, знак - Abs, Sqrt, Sign;
* функции округления - Ceiling, Floor, Round;
* минимум, максимум, степень, остаток - Min, Max, Pow, IEEERemainder.

*Пример*:Написать программу для расчета функции https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-5_hJra.png.

static void Main(string[] args)

{

double x, y;

Console.WriteLine("Введите значение х:");

x = Convert.ToDouble(Console.ReadLine());

y = Math.Sqrt((x+3)/(x-3));

Console.WriteLine("Результат: {0}", y);

}

Вопросы для самоконтроля

1. Какая программа называется линейной?
2. Дайте определение термину «функция».
3. Из каких частей состоит функция?
4. Почему необходимо указывать тип используемой переменной при ее описании?
5. Как следует выбирать тип переменных?
6. Какие операции определены в С#?
7. Чем характеризуются переменные?
8. Какие функции ввода-вывода существуют в С#?
9. Каким классом необходимо воспользоваться для реализации математических выражений?
10. Каких математических функций нет в С#?
11. Как определяется порядок вычислений в С#?
12. Для чего используется оператор присваивания?

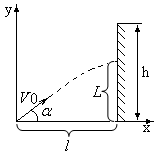
Практическая часть

**Задание №4.** Написать программу:

1. Считая, что Земля – идеальная сфера с радиусом равным 6350 км, определить расстояние до линии горизонта от точки с заданной высотой на Землей.
2. Дана длина ребра куба. Найти объем куба и площадь его боковой поверхности.
3. Даны два действительных положительных числа. Найти среднее арифметическое и среднее геометрическое этих чисел.
4. Даны два действительных числа. Найти среднее арифметическое этих чисел и среднее геометрическое их модулей.
5. Даны катеты прямоугольного треугольника. Найти его гипотенузу и площадь.
6. Определить периметр правильного *n*-угольника, описанного около окружности радиуса *r*.
7. Три сопротивления R1, R2, R3 соединены параллельно. Найти сопротивление соединения.
8. Напишите программу, запрашивающую высоту дома *h*(в метрах), ускорение свободного падения*g*и вычисляющую время падения кирпича*t*(в секундах) с крыши этого дома по формуле:https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-fSdIVE.png.
9. Расстояние до ближайшей к Земле звезды Альфа Центавра 4,3 световых года. Скорость света – 300 000 км/с. Скорость земного звездолета 100 км/с. За сколько лет звездолет долетит до звезды?
10. Пушка стреляет под углом 30° к линии горизонта. Масса снаряда 30 кг, начальная скорость 500 м/с. Найти дальность полета снаряда?
11. Определить силу притяжения *F* между телами массы *т*1 и *т*2, находящимися на расстоянии *r* друг от друга.
12. Даны гипотенуза и катет прямоугольного треугольника. Найти второй катет и радиус вписанной окружности.
13. Известна длина окружности. Найти площадь круга, ограниченного этой окружностью.
14. Известен объем и масса тела. Определить плотность материала этого тела.
15. Треугольник задан величинами своих углов и радиусом описанной окружности. Найти стороны треугольника.
16. Определить время, через которое встретятся два тела, равноускоренно движущиеся навстречу друг другу, если известны их начальные скорости, ускорения и начальное расстояние между ними.
17. Кит плавает под водой со скоростью 27 км/ч. Он развивает мощность 150 кВт. Определить силу сопротивления воды.
18. Треугольник задан длинами сторон. Найти длины высот (https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-9xUOIG.png, https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-eCC3eq.png,https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-cEoZNw.png,где https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-mkcOvE.png);
19. Треугольник задан длинами сторон. Найти длины медиан и биссектрис.
20. Пуля, летящая со скоростью 400 м/с, ударяется в земляной вал и проникает в него на глубину 36 см. Сколько времени пуля двигалась внутри вала и с каким ускорением? Какова была ее скорость на глубине 18 см? Движение считать равноускоренным. Чему будет равна скорость пули к моменту, когда она пройдет 99% своего пути?
21. Треугольник задан длинами сторон. Найти радиусы вписанной и описанной окружностей.
22. Даны основания и высота равнобедренной трапеции. Найти ее периметр.
23. Даны длины сторон прямоугольного параллелепипеда. Найти его объем и площадь боковой поверхности.
24. Даны основания и высота равнобедренной трапеции и угол при большем основании. Найти площадь трапеции.
25. Даны основания и высота равнобедренной трапеции. Найти ее площадь.

**Задание №5.** Написать программу:

1. Дано целое положительное число *п*. Присвойте переменной *т* последнюю цифру этого числа.
2. Пусть *р* и *п* - целые положительные числа и *п*=123456. Чему равно значение *р* после выполнения оператора присваивания: р:= n div 100 mod 10.
3. Пусть *р* и *п* - целые положительные числа и *п*=123456. Чему равно значение *р* после выполнения оператора присваивания: р: = n div 100 mod 10 + n mod 10.
4. Целой переменной *х* присвойте значение суммы цифр заданного трехзначного числа.
5. Определите число, полученное выписыванием в обратном порядке цифр заданного целого трехзначного числа *х*. Присвойте это число переменной *т*.
6. На заданном расстоянии *l* от пушки находится стена. Известен угол наклона пушки https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-IvD_Yi.pngи начальная скорость снаряда*V*0. Попадет ли снаряд в стену? Соотношение https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-fUp2Ul.png.



1. Поменяйте местами значения целых переменных *т* и *n*, не используя дополнительные переменные.
2. Вычислите дробную часть среднего арифметического и дробную часть среднего геометрического 3 заданных чисел.
3. Найдите сумму *п* членов арифметической прогрессии, первый член которой равен *n*, а разность равна d.
4. Смешали *v1* литров воды с температурой *t1* градусов Цельсия с *v2* литрами воды с температурой *t2* градусов Цельсия. Вычислить объём и температуру образовавшейся смеси.
5. Пусть даны четыре целых числа (hour, min, sec, time). Первые три из них (hour, min, sec) - это время запуска ракеты в часах, минутах и секундах, четвертое (time) определяет время полета в секундах. Найдите и напечатайте время возвращения ракеты на землю.
6. Пусть дано натуральное число *п*, состоящее из шести цифр. Определите число сотен и тысяч в нем.
7. Написать программу, определяющую объём и площадь боковой поверхности цилиндра с заданными радиусом основания *R* и высотой *H*.
8. Самолет летит равномерно со скоростью *V*. Рассмотреть движение свободнопадающего груза, который до падения находился в самолете.

https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-aEXav0.png https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-Dph5CH.pnghttps://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-LUbxCQ.pnghttps://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-n3FR97.png

1. В бригаде, работающей на уборке сена, имеется *п* косилок. Первая из них работала *т* часов, а каждая следующая на 10 мин больше, чем предыдущая. Сколько часов работала вся бригада?
2. Расположенный на берегу реки завод осуществляет сброс сточных вод, в результате чего произошло увеличение концентрации вредных веществ. С течением времени концентрация уменьшается. Нужно сообщить каков будет уровень загрязнения реки через сутки, двое, трое и до тех пор пока концентрация станет меньше допустимой. Д – предельно допустимая концентрация. С – начальная концентрация. К – коэффициент. Соотношение - https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-8Tq0rE.png.
3. Посчитайте, сколько очков набрала команда «Динамо» в первом круге чемпионата России по хоккею, если известно, что m встреч она выиграла, n встреч проиграла, k встреч закончились ничьими, полагая, что за выигрыш команда получает 2 очка, за ничью -1 очко, за проигрыш – 0 очков.
4. Сколько процентов от А+В-С приходится на A? На В? На С?
5. Составьте программу для вычисления времени *t* встречи автомобилей, движущихся равноускоренно навстречу друг другу, если известны их скорости V1 и V2, ускорения *а*1 и *а*2 и начальное расстояние S между ними. Расстояние *S1*, пройденное первым автомобилем, вычисляется по формуле https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-FwNaHw.png, расстояние*S2*, пройденное вторым автомобилем, вычисляется по формуле https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-udKG1L.png. Время*t* встречи автомобилей определяется из уравнения https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-iR8O8u.png, откудаhttps://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-7w710u.png.
6. Товар до деноминации (до 1 января 1998 года) стоил *r* рублей. Какова его цена после деноминации (в рублях и копейках) при условии, что *r*>10?
7. Составьте программу для вычисления времени *t* встречи автомобилей, движущихся равноускоренно навстречу друг другу, если известны их скорости V1 и V2, ускорения *а*1 и *а*2 и начальное расстояние S между ними. Расстояние S1, пройденное первым автомобилем, вычисляется по формуле https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-tntR_e.png, расстояние S2, пройденное вторым автомобилем, вычисляется по формуле https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-DlY2M9.png. Время*t* встречи автомобилей определяется из уравнения https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-EOsygU.png, откудаhttps://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-jdD0eF.png.
8. Определите стоимость набора, в который входят следующие конфеты (стоимость упаковки составляет *U* руб.):

|  |  |  |
| --- | --- | --- |
| Название | Вес | Стоимость |
| Петровские | 200 г | К руб. (1 кг) |
| Воронежские | 300 г | Р руб.(1 кг) |
| Чародейка | 250 г | R руб. (1 кг) |
| Факел | 150 г | В руб. (1 кг) |
| Ласточка | 200 г | L руб.(1 кг) |
| Упаковка | - | U руб.(1 кг) |

1. Розничная цена мужского костюма составляет R рублей. Торговое наложение магазина составляет Т% от оптовой цены. Составьте программу определения оптовой цены костюма.
2. Зарплата сотрудника частной фирмы составляет *r* рублей в месяц. Сколько денег он получит за полгода после вычета налогов в размере t% ежемесячно и s% за полгода?
3. Сколько времени в минутах затратит школьник на дорогу от школы до стадиона, если это расстояние составляет *S* км, а средняя скорость движения школьника – *V* km/ч.

**Задание №6**. Вычислить значение функции для заданного значения аргумента

1. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-8rxdfO.png

2. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-gtqvSW.png

3. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-4_M34X.png

4. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-CpEsiq.png

5. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-a3VzWa.png

6. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-aT7XqN.png

7. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-w5nJyU.png

8. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-AGF9Zk.png

9. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-2MErmM.png

10. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-QZ2gPs.png

11. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-Q5Y7Sj.png

12. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-z5xMgk.png

13. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-5RX1OX.png

14. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-YHcRF4.png

15. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-5TSkn0.png

16. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-tLjwY8.png

17. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-O2XSOZ.png

18. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-mNs57n.png

19. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-3hs5Cr.png

20. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-rtCeL4.png

21. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-ucxXxN.png

22. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-ARTUYi.png

23. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-sLuYIf.png

24. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-NA0ZR8.png

25. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-gMLY_y.png

**Задание №7.** Написать программу, вычисляющую:

1. *а*4 за две операции, *а* – действительное число, при этом не пользуясь никакими другими арифметическими операциями, кроме умножения.
2. *а*6 за три операции, *а* – действительное число, при этом не пользуясь никакими другими арифметическими операциями, кроме умножения.
3. *а*8 за три операции, *а* – действительное число, при этом не пользуясь никакими другими арифметическими операциями, кроме умножения.
4. *а*7 за четыре операции, *а* – действительное число, при этом не пользуясь никакими другими арифметическими операциями, кроме умножения.
5. *а*9 за четыре операции, *а* – действительное число, при этом не пользуясь никакими другими арифметическими операциями, кроме умножения.
6. *а*10 за четыре операции, *а* – действительное число, при этом, не пользуясь никакими другими арифметическими операциями, кроме умножения.
7. *а*13 за пять операции, *а* – действительное число, при этом, не пользуясь никакими другими арифметическими операциями, кроме умножения.
8. *а*15 за пять операции, *а* – действительное число, при этом, не пользуясь никакими другими арифметическими операциями, кроме умножения.
9. *а*21 за шесть операции, *а* – действительное число, при этом, не пользуясь никакими другими арифметическими операциями, кроме умножения.
10. *а*28 за шесть операции, *а* – действительное число, при этом, не пользуясь никакими другими арифметическими операциями, кроме умножения.
11. *а*64 за шесть операции, *а* – действительное число, при этом, не пользуясь никакими другими арифметическими операциями, кроме умножения.
12. *а*3 и *а*10 за четыре операции, *а* – действительное число, при этом не пользуясь никакими другими арифметическими операциями, кроме умножения.
13. *а*4 и *а*20 за пять операций, *а* – действительное число, при этом не пользуясь никакими другими арифметическими операциями, кроме умножения.
14. *а*5 и *а*13 за пять операций, *а* – действительное число, при этом не пользуясь никакими другими арифметическими операциями, кроме умножения.
15. *а*5 и *а*19 за пять операций, *а* – действительное число, при этом не пользуясь никакими другими арифметическими операциями, кроме умножения.
16. *а*2, *а*5, *а*17 за шесть операций, *а* – действительное число, при этом не пользуясь никакими другими арифметическими операциями, кроме умножения.
17. *а*4, *а*12, *а*28 за шесть операций, *а* – действительное число, при этом не пользуясь никакими другими арифметическими операциями, кроме умножения.
18. *а*20 за пять операции, *а* – действительное число, при этом, не пользуясь никакими другими арифметическими операциями, кроме умножения.
19. *а*19 за пять операции, *а* – действительное число, при этом, не пользуясь никакими другими арифметическими операциями, кроме умножения.
20. *а*17 за шесть операции, *а* – действительное число, при этом, не пользуясь никакими другими арифметическими операциями, кроме умножения.
21. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-99eajE.png, если дано действительное число *х*. Разрешается использовать не более четырех умножений и четырех сложений и вычитаний, но при этом, не пользуясь никакими другими арифметическими операциями, кроме умножения, сложения и вычитания.
22. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-yaPK9b.png, если даны действительные число *х,* *у*. Разрешается использовать не более восьми умножений и восьми сложений и вычитаний, но при этом, не пользуясь никакими операциями, кроме умножения, сложения и вычитания.
23. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-FBNdAn.png и https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-Yazbxo.png, если дано действительное число *х*. Разрешается использовать не более восьми операций, но при этом, не пользуясь никакими другими арифметическими операциями, кроме умножения, сложения и вычитания.
24. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-vNRhqt.png, если даны действительные число *х,* *у*. Разрешается использовать не более восьми умножений и восьми сложений и вычитаний, но при этом, не пользуясь никакими операциями, кроме умножения, сложения и вычитания.
25. https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-Vup_UE.png и https://studfile.net/html/2706/27/html_qEXQ_J0uL0.ErTg/img-Y9kuyj.png, если дано действительное число*х*. Разрешается использовать не более восьми операций, но при этом, не пользуясь никакими другими арифметическими операциями, кроме умножения, сложения и вычитания.

# Контрольные вопросы

1. Какова общая структура программы на С#.
2. Существуют ли ограничения на множество целых чисел, используемых в языке С#?
3. Какие формы записи вещественных чисел используются в языке С#?
4. Из каких символов может состоять идентификатор?
5. Чем вызвана необходимость использования комментариев в программе?
6. Каково назначение методов write и writeline?
7. Перечислите способы вывода данных на дисплей?
8. Перечислите основные управляющие последовательности языка С#?
9. Назовите основные методы класса Math языка С#?
10. Какой метод класса Console используется для того, чтобы вводимые данные читались с новой строки?
11. Что такое выражения в языке С#?
12. Понятие преобразования типов в языке С#?
13. Назовите формы записи операцииинкремента и декремента языка С#?