Министерство образования и науки РФ

Федеральное государственное автономное образовательное учреждение высшего образования

«Омский государственный технический университет»

|  |  |
| --- | --- |
| Факультет (институт) | *Информационных технологий и компьютерных систем* |
|  |  |
| Кафедра | *Прикладная математика и фундаментальная информатика* |
|  |  |

**Расчетно–графическая работа**

|  |  |
| --- | --- |
| по дисциплине | ***Алгоритмизация и программирование*** |
|  |  |
| на тему | Программная реализация задач |

Пояснительная записка

|  |  |
| --- | --- |
| **Шифр проекта** | 020–РГР–02.03.02–№ 08 – ПЗ |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Студента** | | Киснер Анастасии Евгеньевны | | | | | |
|  |  |  |  | | фамилия, имя, отчество полностью | | | | | |
|  |  |  | Курс | 1 |  | Группа | | МО–241 | | |
|  |  |  |  |  |  | |  |  | |  |
|  | | | **Направление (специальность)** | | | | | ***02.03.03*** | | |
|  | | | Математическое обеспечение и администрирование информационных систем | | | | | | | |
|  |  |  | код, наименование | | | | | | | |
|  |  |  | Руководитель | | ***ст. преподаватель*** | | | | | |
|  |  |  | ученая степень, звание | | | | | |
|  |  |  | ***Федотова И.В.*** | | | | | | | |
|  |  |  | фамилия, инициалы | | | | | | | |
|  |  |  | Выполнил | |  | | | | | |
|  |  |  | дата, подпись студента | | | | | |
|  |  |  |  | | | | | | | |
|  |  |  | **Работа защищена с количеством баллов** | | | | | |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | дата, подпись руководителя |  |  |  |

Омск 2024

**Оглавление**

[**Введение** 3](#_Toc186238135)

[**Постановка задачи «Упаковки молока»** 4](#_Toc186238136)

[**Ход решения задачи «Упаковки молока»** 6](#_Toc186238137)

[**Постановка задачи «Интенсификация производства»** 9](#_Toc186238138)

[**Ход программы «Интенсификация производства»** 10](#_Toc186238139)

[**Постановка задачи «Отгадай число»** 12](#_Toc186238140)

[**Ход задачи «Отгадай число»** 14](#_Toc186238141)

[**Постановка задачи «Зельеварение»** 17](#_Toc186238142)

[**Ход задачи «Зельеварение»** 19](#_Toc186238143)

[**Заключение** 22](#_Toc186238144)

[**Список используемой литературы** 23](#_Toc186238145)

# **Введение**

C# – это мощный и универсальный объектно-ориентированный язык программирования от Microsoft, тесно интегрированный с платформой .NET. Его компилируемый характер обеспечивает высокую производительность, а использование промежуточного языка (IL) и среды выполнения .NET (CLR) гарантирует переносимость на различные платформы. Автоматическое управление памятью (сборщик мусора) упрощает разработку и повышает надежность, минимизируя риск утечек памяти. Строгая типизация способствует предотвращению ошибок на этапе компиляции. C# эффективно поддерживает многопоточность, что позволяет создавать высокопроизводительные приложения, а обширная стандартная библиотека .NET предоставляет широкий спектр готовых компонентов для решения разнообразных задач. Он поддерживает различные парадигмы программирования, включая объектно-ориентированное, функциональное (с помощью LINQ) и аспектно-ориентированное программирование. C# широко используется для создания веб-приложений (ASP.NET Core), десктопных приложений (WinForms, WPF, .NET MAUI), мобильных приложений (Xamarin), игр (Unity) и бэкенд-систем, что делает его одним из наиболее востребованных языков программирования в мире. Его популярность обусловлена сочетанием высокой производительности, относительно простого синтаксиса, обширной поддержки со стороны Microsoft и активного сообщества разработчиков, постоянно расширяющего его возможности и обеспечивающего обилие ресурсов для обучения и поддержки. Постоянное развитие языка, добавление новых функций и улучшений гарантирует его актуальность и востребованность в будущем.

# **Постановка задачи «Упаковки молока»**

На оптовой базе имеется молоко, выпущенное несколькими фирмами.

Молоко каждой фирмы расфасовано в два вида упаковок, представляющих собой параллелепипеды. Для каждого вида упаковки каждой из фирм известна стоимость, которая включает как стоимость материала тары, так и стоимость собственно молока. Требуется определить фирму, у которой стоимость одного литра собственно молока минимальна, а также эту стоимость.

Примечание

Считать, что материал тары абсолютно тонкий и все плоскости параллелепипеда состоят из одного слоя материала.

Считать, что у двух упаковок одной фирмы стоимость единицы площади материала одинакова.

Считать, что у двух упаковок одной фирмы стоимость одного литра собственно молока одинакова.

Входной файл

Первая строка содержит целое число N - количество фирм (1 ≤ N ≤ 100).

Следующие N строк содержат шесть целых чисел Xi1, Yi1, Zi1, Xi2, Yi2, Zi2 - размеры двух видов упаковок i-ой фирмы в сантиметрах (0 < Xi1, Yi1, Zi1, Xi2, Yi2, Zi2 ≤ 100; 1 ≤ i ≤ N), а также два вещественных числа Ci1 и Ci2 - стоимости первой и второй упаковок соответственно у i-ой фирмы в рублях (0 < Ci1, Ci2 ≤ 1000.0). В стоимости упаковок включаются как стоимость материала тары, так и стоимость собственно молока.

Выходной файл

Должен содержать одну строку, состоящую из целого и вещественного чисел, разделенных пробелом - номер фирмы, у которой стоимость одного литра собственно молока минимальна, а также эту стоимость в рублях (стоимость выводить с двумя знаками после запятой).

Если имеется несколько фирм с одинаковой минимальной стоимостью собственно молока, то вывести ту из них, номер которой минимален.

Пример:

Input.txt

2

10 10 5 10 10 10 12.23 20.12

5 15 20 7 8 9 43.28 16.99

Output.txt

2 4.17

# **Ход решения задачи «Упаковки молока»**

Программа предназначена для обработки данных о нескольких "фирмах", каждая из которых имеет характеристики, связанные с размерами и ценами на упаковку. Программа вычисляет некоторую "цену" для каждой фирмы и в итоге выводит номер фирмы с минимальной "ценой" и саму эту "минимальную цену".

1. Ввод данных: программа запрашивает количество фирм-производителей молока, а также параметры двух упаковок с их стоимостью.
2. Вычисление объема молока: расчет площади поверхности и объема для каждой упаковки
3. Вычисление цены за единицу объема по формуле для каждой из фирм
4. Поиск минимальной цены: сравнить цены с помощью «Min»
5. Вывод результата: программа выводит номер фирмы с минимальной ценой за литр молока и саму цену.

Код программы представлен на рисунке 1:

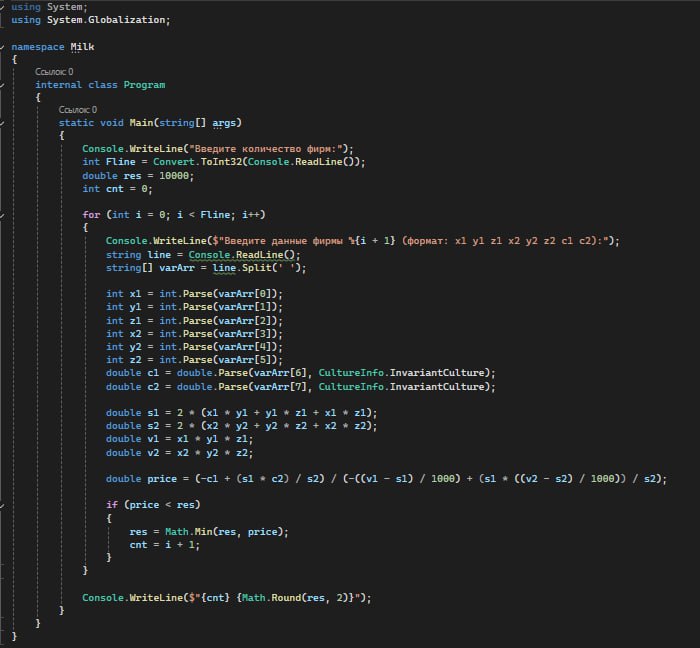


Рисунок 1 – код программы «Упаковки молока»

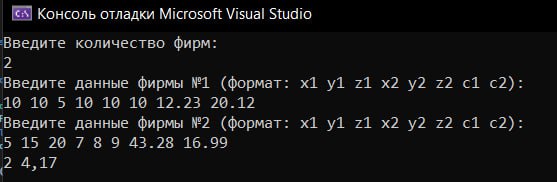


Рисунок 2 – пример вывода программы «Упаковки молока»

Рисунок 3 ­ блок-схема

Ввод N

Цикл I

от 1 до N

Ввод x1, y1, z1, x2, y2, z2, c1, c2

Вычислить v1, v2

Вычисление current\_min

currenr\_min >

min\_cost ?

Увеличить i на 1

Вывод best\_firm и min\_cost

# **Постановка задачи «Интенсификация производства»**

Перед коллективом предприятия “Ни шагу назад” была поставлена задача наращивать каждый день производство продукции на 1.

Требуется определить, какой суммарный объем продукции будет выпущен предприятием за заданный период, если в первый день периода предприятие выпускало P единиц продукции.

Примечания:

* период задается в виде двух календарных дат;
* длительность периода лежит в диапазоне от 1 до 60000;
* високосные годы учитываются по упрощенному правилу: високосным считается год, делящийся нацело на 4;
* день начала периода и день его окончания учитываются при подсчете суммарного объема продукции и длительности периода;
* все даты заданы корректно.

Входной файл содержит:

* в первой строке – дата начала периода в формате ДД.ММ.ГГГГ;
* во второй строке – дата окончания периода в формате ДД.ММ.ГГГГ;
* в третьей строке целое число – начальный выпуск продукции P (0 ≤ P ≤ 5000).

Выходной файл должен содержать суммарный объем продукции.

Пример:

Input.txt

12.05.2002

15.05.2002

10

Output.txt

46

# **Ход программы «Интенсификация производства»**

Программа предназначена для расчета периода, в которой произведено некоторое количество продукции. Начнем с расчета периода времени, а затем с использованием объема продукции вычислим суммарный объем за эти даты.

1. Ввод данных: программа получает от пользователя начальную и конечную даты и объем продукции.
2. Расчет количества дней: считаем разницу между двумя датами и получаем количество дней.
3. Проверка находится ли число в диапазоне от 1 до 60000.
4. Расчет объема продукции: вычисляем объем продукции, используя формулу арифметической прогрессии и каждый день прибавляем 1 к результату.
5. Вывод результата: программа выводит итоговый объем продукции за полученное количество дней.

Код программы представлен на рисунке 4:

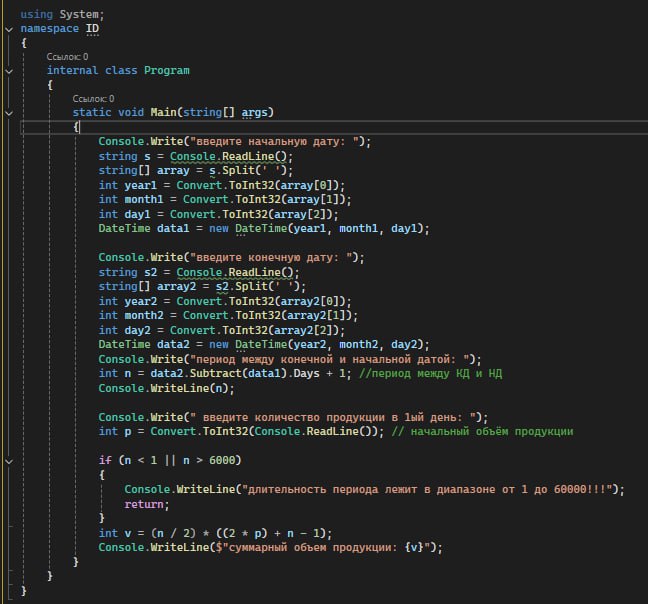


Рисунок 4 – код программы «Интенсификация производства»

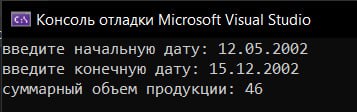


Рисунок 5 – пример вывода программы «Интенсификация производства»

# **Постановка задачи «Отгадай число»**

Известен следующий фокус. Фокусник предлагает выполнить действия следующего характера: задумайте число, прибавьте 2, умножьте на 3, отнимите 5, отнимите задуманное число и т.д. После этого по названному полученному результату фокусник определяет задуманное число.

Пусть задумано некоторое целое число X. Требуется после выполнения ряда действий по известному результату R определить это число.

Примечание:

* гарантируется, что имеется только один ответ;
* гарантируется, что во время выполнения действий какие-либо промежуточные результаты не превышают по модулю 2 000 000 000.

Входной файл

Первая строка содержит количество действий N (0  N  100).

Следующие N строк содержат описания действий в последовательности их выполнения, причем в каждой строке указывается одно действие в формате S V, где:

* S - тип действия, состоящий из одного символа: "\*" - умножить; "-" - отнять; "+" - прибавить;
* V - аргумент действия. Может быть целым числом (|V|  100) либо символом "x". Символ "x" может применяться только в действиях "-" и "+" и обозначает, что нужно отнять или прибавить задуманное число, соответственно.

Последняя строка содержит результат R (|R|  2 000 000 000).

Выходной файл должен содержать одно целое число - задуманное число X.

Пример:

Input.txt

4

+ 2

\* 3

- 5

- x

7

Output.txt

3

# **Ход задачи «Отгадай число»**

Программа имитирует выполнение последовательности математических операций над переменными. После выполнения всех операций программа вычисляет число, которое было «загадано»

1. Ввод данных: пользователь вводит количество операций, сами операции и результат вычислений.
2. Обработка операций с числом: проверка, если операция содержит «x», применение операций
3. Проверка является ли операция делением на 0
4. Выводит текущие значения a и b после каждой операции, чтобы пользователь мог отслеживать их изменения.
5. Вывод данных: программа выводи число, которое «загадал» пользователь.

Код программы «Отгадай число» представлен на Рисунке 6.1–6.2:

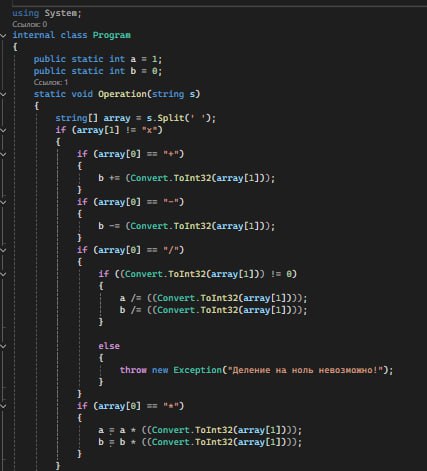


Рисунок 6.1 – первая часть кода программы «Отгадай число»

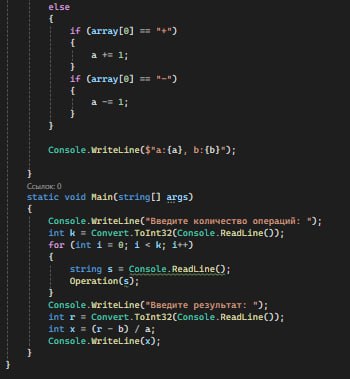


Рисунок 6.2 – вторая часть кода программы «Отгадай число»

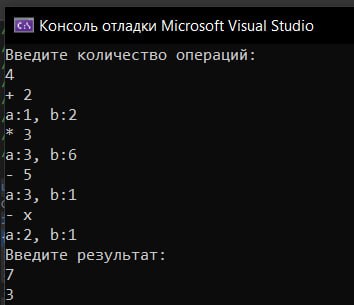


Рисунок 7 – пример вывода программы «Отгадай число»

# **Постановка задачи «Зельеварение»**

Одним из нелюбимых предметов Невилла Долгопупса, товарища Гарри Поттера с факультета Гриффиндор школы чародейства и волшебства “Хогвартс”, было зельеварение.

Чтобы помочь Невиллу в совершенствовании навыков зельеварения, Гарри придумал зельеварочный комбайн, изготавливающий зелья с помощью определенных заклинаний. Каждое заклинание представляет собой одно слово, формируемое в зависимости от последовательности и методов приготовления зелья.

Последовательность приготовления зелья описывается в виде набора действий, каждое из которых указывает на метод обработки заданного списка ингредиентов:

* смешивание описывается MIX < ингредиент1 ингредиент2, …>;
* растворение в воде описывается WATER < ингредиент1 ингредиент2, …>;
* измельчение описывается DUST < ингредиент1 ингредиент2, …>;
* обжиг описывается FIRE < ингредиент1 ингредиент2, …>,

при этом в качестве любого ингредиента может выступать либо некоторое вещество, задаваемое строковой константой, либо результат выполнения любого из предыдущих действий, задаваемый с помощью номера действия. Название действия и названия ингредиентов разделяются пробелами. В действии участвует как минимум один ингредиент.

Каждое действие переводится в слово по следующему правилу:

* смешивание задается в формируемом заклинании словом MX<список ингредиентов>XM;
* растворение в воде - слово WT<список ингредиентов>TW;
* измельчение - слово DT<список ингредиентов>TD;
* обжиг - слово FR<список ингредиентов>RF,
* где <список ингредиентов> - единое слово, сформированное путем сложения названий ингредиентов или слов, описывающих предыдущие действия.

Последнее действие явно или неявно использует результаты выполнения всех предыдущих действий и является основой для заклинания.

Помогите Невиллу по заданной последовательности действий сформировать заклинание.

Примечание:

* при формировании заклинания учитывается регистр названий действий и ингредиентов;
* порядок названий ингредиентов в заклинании должен соответствовать их порядку в действии, т.е. для действия “DUST root tooth” в заклинании правильным считается слово “DTroottoothTD”, а слово “DTtoothrootTD” считается неправильным;
* в названии веществ нет цифр, а используются только английские буквы;
* каждое действие может быть несколько раз использовано в последующих действиях;
* гарантируется, что длина строки, содержащей сформированное заклинание, не превышает 50000 символов.

Входной файл содержит набор строк, каждая из которых описывает отдельное действие. Строки расположены в порядке выполнения действий. Длина каждой строки не превышает 255 символов. Количество строк не превышает 100.

Выходной файл должен содержать строку, содержащую сформированное заклинание.

Пример:

Input.txt

DUST root tooth

WATER 1 tear

Output.txt

WTDTroottoothTDtearTW

# **Ход задачи «Зельеварение»**

Программа принимает от пользователя команды и ингредиенты для создания "заклинания". Каждая команда (MIX, WATER, DUST, FIRE) добавляет определенный суффикс и префикс к финальному "заклинанию", используя введенные ингредиенты. В конце выводится итоговое заклинание.

1. Ввод данных: пользователь вводит заклинание с ингредиентами до тех пор, пока не будет введено «END».
2. Проверка ввода: программа проверяет правильность ввода команд.
3. Сохранение данных: данные, введенные через пробел разбиваются в отдельные списки, входящие в зубчатый массив.
4. Сбор заклинания в одну строку: программа проверяет, какое заклинание было введено, а затем собирает его и ингредиенты в одну строку.
5. Вывод данных: программа выводит строку, содержащую все заклинания и ингредиенты.

Код программы «Зельеварение» представлен на Рисунке 8.1–8.2

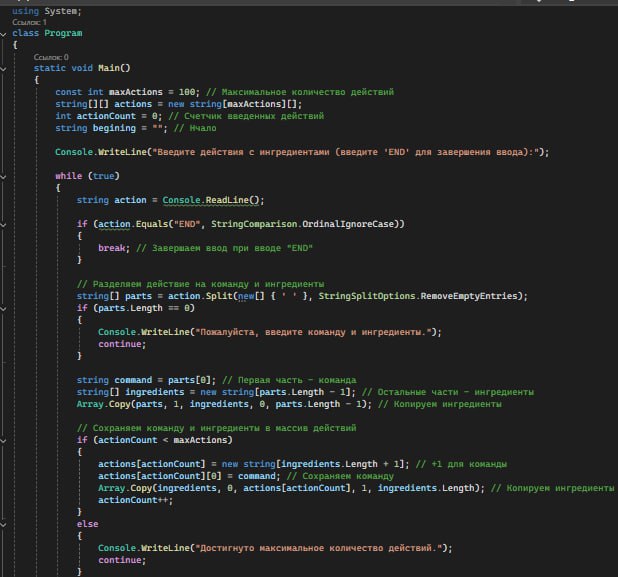


Рисунок 8.1 – первая часть кода программы «Отгадай число»

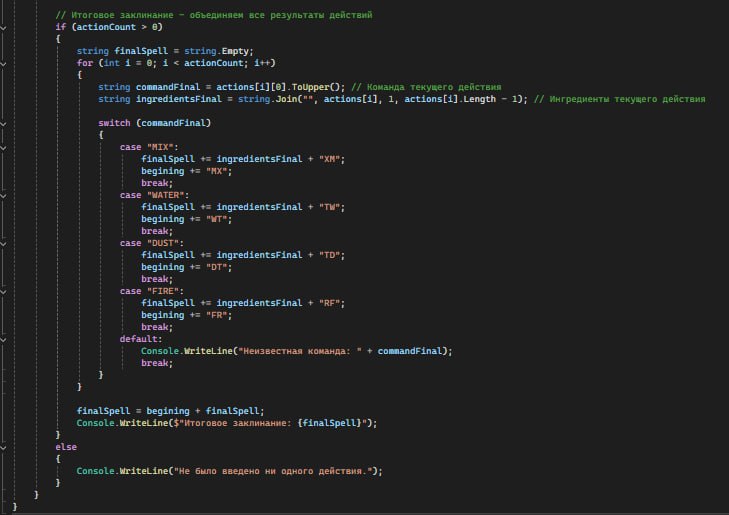


Рисунок 8.2 – вторая часть кода программы «Отгадай число»

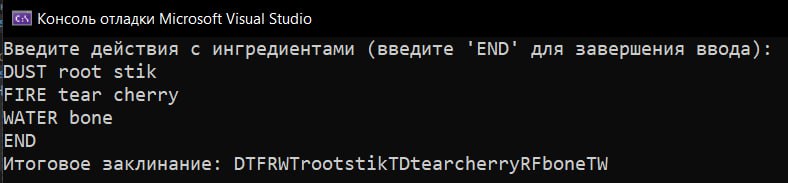


Рисунок 9 – пример вывода программы «Зельеварение»

# **Заключение**

C# – это объектно-ориентированный, строго типизированный язык программирования, разработанный Microsoft, являющийся ключевым для платформы .NET. Он компилируется в промежуточный язык (IL), который затем выполняется .NET CLR, обеспечивая кроссплатформенность на Windows, Linux и macOS. C# широко используется для разработки настольных Windows-приложений, веб-приложений ([ASP.NET](https://asp.net/) Core), игр (Unity), мобильных приложений (.NET MAUI), облачных сервисов (Azure) и в области AI. Одной из важных особенностей C# является автоматическое управление памятью через сборщик мусора, освобождающий разработчиков от ручного управления памятью. C# интегрирует LINQ (Language Integrated Query) для удобной работы с данными, позволяя выполнять запросы непосредственно в коде. Для обеспечения отзывчивости приложений C# поддерживает асинхронное программирование с async и await. Язык имеет простой синтаксис, что делает его сравнительно легким в изучении. Он предоставляет обширную библиотеку классов .NET для решения разнообразных задач. C# обеспечивает высокую производительность благодаря JIT-компиляции. Существует большое и активное сообщество разработчиков, обеспечивающее поддержку и ресурсы. C# хорошо интегрирован с продуктами Microsoft, что является преимуществом для разработки Windows-ориентированных решений. Он подходит для создания как простых, так и сложных проектов. Разнообразие областей применения делает его привлекательным для многих разработчиков. C# продолжает развиваться, становясь еще более актуальным и удобным для работы. Он остается одним из ключевых языков программирования в современной разработке программного обеспечения.

# **Список используемой литературы**

1. Васильев А. Н. Программирование на C# для начинающих. - Москва: Эксмо, 2019. - 525 с.
2. Джон Пол Мюллер, Билл Семпф, Чак Сфер C# для чайников. - СБП: Диалектика, 2019. - 596 с.
3. Стиллимен Эндрю, Грин Дженнифер Head First. Изучаем C#. - СБП: Питер, 2022. - 769 с.
4. C# и .NET // METANIT.COM URL: https://metanit.com/sharp/tutorial/2.7.php (дата обращения: 03.11.2024).
5. Документация по языку C# // Mcrosoft.com URL: https://learn.microsoft.com/ru-ru/dotnet/csharp/ (дата обращения: 03.11.2024).