**Содержание**

[Введение 3](#_Toc185609878)

[Постановка задачи 1 4](#_Toc185609879)

[Ход решения задачи 1 5](#_Toc185609880)

[Постановка задачи 2 7](#_Toc185609881)

[Ход решения задачи 2 8](#_Toc185609882)

[Постановка задачи 3 10](#_Toc185609883)

[Ход решения задачи 3 11](#_Toc185609884)

[Постановка задачи 4 12](#_Toc185609885)

[Ход решения задачи 4 13](#_Toc185609886)

[Заключение 15](#_Toc185609887)

[Cписок используемой литературы 16](#_Toc185609888)

# Введение

В настоящее время язык программирования C# широко используется для решения разнообразных алгоритмических задач, разработки приложений, игр и других программных продуктов. Он легко усваивается как начинающими, так и опытными специалистами благодаря удобному синтаксису и множеству инструментов.

C# поддерживает основные принципы современного программирования, такие как работа с объектами и классами, что упрощает создание сложных проектов. Это способствует написанию понятного и структурированного кода, который легко модифицировать и улучшать. Кроме того, язык автоматически управляет ресурсами, освобождая программиста от необходимости заботиться о технических деталях.

Одним из основных преимуществ C# является его универсальность. С его помощью можно разрабатывать программное обеспечение для различных устройств, таких как компьютеры, телефоны и веб-сайты. Кроме того, язык обеспечивает эффективную работу с данными и создание удобных пользовательских интерфейсов.

Этот материал направлен на изучение основных возможностей языка программирования C# и их применение для решения задач. Мы рассмотрим основные конструкции языка, создадим несколько программ и приобретем практические навыки, которые будут полезны при работе с этим языком в реальных проектах.

# Постановка задачи 1

Из N солдат, выстроенных в шеренгу, требуется отобрать троих в разведку. Для того чтобы сделать это, выполняется следующая операция: если солдат в шеренге больше 3, то шеренга разбивается на две, одна из которых состоит из солдат, стоящие на четных позициях, а вторая – стоящих на нечетных позициях. Эта процедура повторяется для всех полученных шеренг до тех пор, пока в каждой из них не останется 3 или менее солдат. Если солдат осталось трое, то данную группу можно послать в разведку. Требуется определить, сколько групп по 3 человека может быть сформировано из исходной шеренги.

Входной файл содержит число N – количество солдат в исходной шеренге. (0 < N ≤ 10000000).

Выходной файл должен содержать количество вариантов формирования групп разведки.

# Ход решения задачи 1.

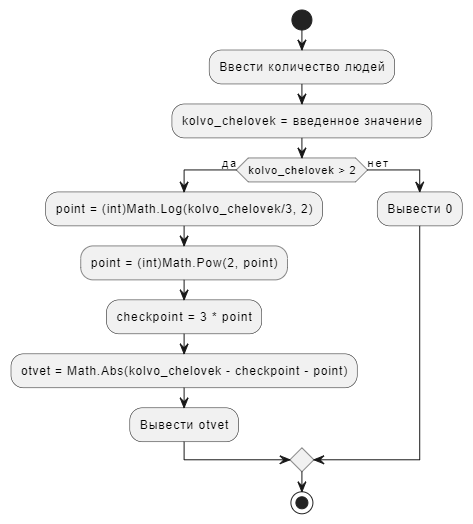


Рисунок 1 – Блок-схема решения задачи 1. Использовались сложные операторы присваивания, консольные средства ввода/вывода данных.

Код программы представлен на рисунке ниже.

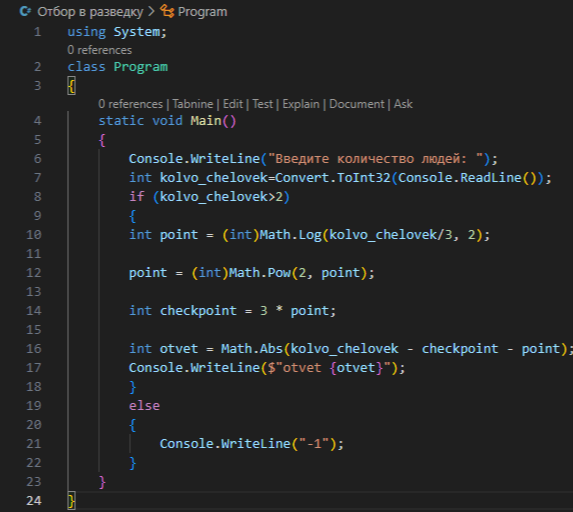


Рисунок 2 – Код к задаче 1



Рисунок 3 – Пример работы программы на входных данных 1



Рисунок 4 – Пример работы программы на входных данных 2



Рисунок 5 – Пример работы программы на входных данных 3

# Постановка задачи 2

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.

# Ход решения задачи 2

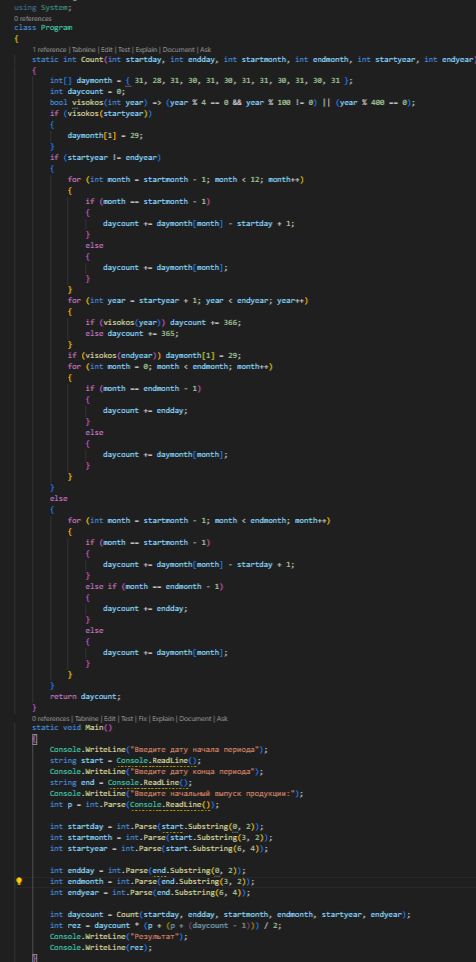
В ходе решения задачи использовались методы преобразования в различные типы данных, массивы, методы работы со строками, циклы, условные операторы, сложные операторы присваивания, консольные средства ввода/вывода данных. Код программы представлен на рисунке ниже.

Рисунок 6 – код к задаче 2

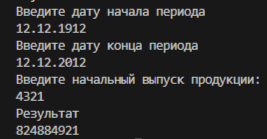


Рисунок 7 – Пример работы программы на входных данных 1

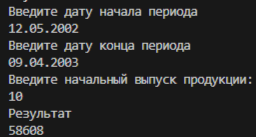


Рисунок 8 – Пример работы программы на входных данных 2

# Постановка задачи 3

Идет крестьянин и плачется: "Эхма! Жизнь моя горькая! Заела нужда совсем! Вот в кармане только несколько монет, да и те сейчас нужно отдать. И как это у других бывает, что на всякие свои деньги они еще деньги получают? Хоть бы кто помочь мне захотел".

Только успел это сказать, как глядь, а перед ним черт стоит и говорит: "Вот видишь этот мост через реку. Стоит тебе перейти через мост, и у тебя будет вдвое больше денег, чем есть. Перейдешь опять, и снова станет вдвое больше. Но за то, что я у тебя деньги удваиваю, после каждого перехода ты мне должен отдавать по K монет".

"Ой ли," - сказал крестьянин -"ну-ка, попробуем". Перешел мост, и деньги у него удвоились. Отдал он черту K монет, перешел мост еще раз, и опять деньги удвоились. Снова отдал крестьянин черту K монет.

Однако после Z переходов и отдач черту по K монет оказалось, что у крестьянина не осталось ни одной монеты.

Требуется определить, сколько комбинаций условий перехода через мост может быть, если известно, что у крестьянина изначально было не более MaxN монет. Комбинацией условий перехода является тройка чисел N, K, Z, где N - начальное количество монет у крестьянина, K - количество монет, отдаваемых черту после каждого перехода, Z - количество переходов. Естественно, что для этой тройки должно выполняться условие, что после Z циклов у крестьянина не должно остаться монет.

Входной файл содержит целое число MaxN - максимальное количество, которое может быть изначально у крестьянина (1 ≤ MaxN ≤ 2000000000).

Выходной файл должен содержать одно целое число - количество комбинаций условий перехода через мост.

# Ход решения задачи 3

В ходе решения задачи использовались сложные операторы присваивания, условные операторы, методы преобразования в различные типы данных, метод возведения в степень, консольные средства ввода/вывода данных. Код программы представлен на рисунке ниже.

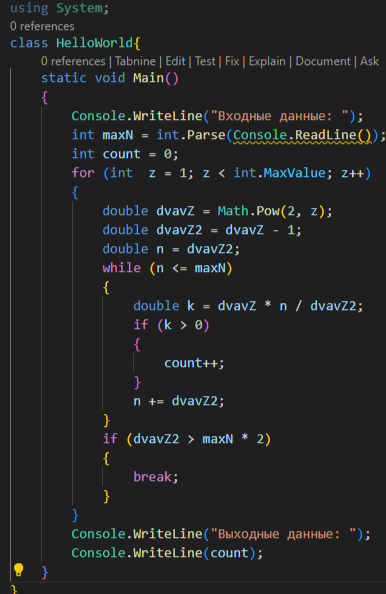


Рисунок 9 – код к задаче 3

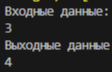


Рисунок 10 – Пример работы программы на входных данных 1

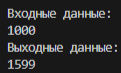


Рисунок 11 – Пример работы программы на входных данных 2

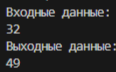


Рисунок 12 – Пример работы программы на входных данных 3

# Постановка задачи 4

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.

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.

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.

# Ход решения задачи 4

Ход решения задачи “Молоко”

Создадим цикл for, в котором будем вычислять стоимость одного литра молока. Для каждого числа вводимого с консоли сделаем отдельную переменную. Далее вычислим объем и площадь каждой упаковки. После этого вычислим стоимость за литр в зависимости от того, какая цена больше. Делаем проверку минимальной цены, и выходим из цикла с дальнейшим выводом результата.

Код программы представлен на рисунке ниже.

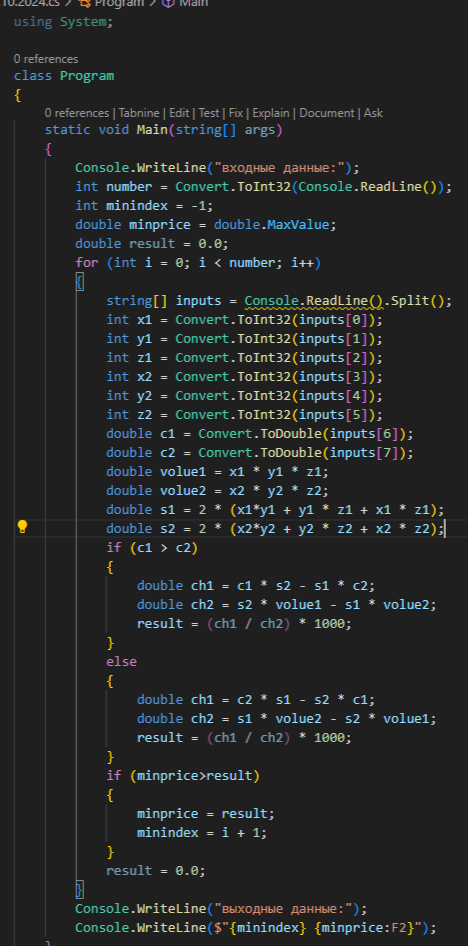


Рисунок 13 к задаче 4

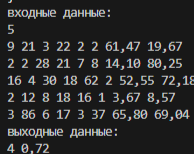


Рисунок 14 – Пример работы программы на входных данных 1

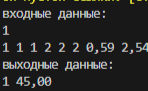


Рисунок 15 – Пример работы программы на входных данных 2

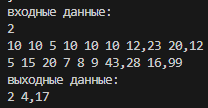


Рисунок 16 – Пример работы программы на входных данных 3

# Заключение

В процессе выполнения расчетно-графической работы были изучены и применены основные возможности языка программирования C# для решения алгоритмических задач.

Основное внимание было уделено практическому использованию базовых конструкций языка, таких как циклы, условные операторы и методы. Эта работа позволила приобрести навыки структурирования и оптимизации кода, использования основных элементов программирования и разработки простых консольных приложений.

Были решены задачи, где требовалось работать со строками, сравнивать данные и массивы, что способствовало закреплению теоретических знаний на практике. Особое внимание было уделено логике работы программ, шаг за шагом алгоритму их создания и проверке решений на правильность.

В результате стало очевидно, что язык C# является мощным инструментом для разработки программ разной сложности. Эта выполненная работа не только углубила понимание принципов программирования на C#, но и продемонстрировала, как с помощью четко структурированного кода можно эффективно решать разнообразные задачи.

Полученные знания и навыки могут быть использованы в будущем для создания более сложных проектов и разработки программных решений в профессиональной области.

# Cписок используемой литературы

1. Microsoft Learn: офиц. cайт. – URL: <https://learn.microsoft.com/ru-ru/dotnet/api/system.array.convertall?view=net-9.0> (дата обращения: 21.12.2024)
2. Microsoft Learn: офиц. cайт. – URL: <https://learn.microsoft.com/ru-ru/dotnet/api/system.math.round?view=net-8.0> (дата обращения: 20.12.2024)
3. Microsoft Learn: офиц. cайт. – URL: <https://learn.microsoft.com/ru-ru/dotnet/api/system.int32.parse?view=net-8.0> (дата обращения: 27.12.2024)
4. Microsoft Learn: офиц. cайт. – URL: <https://learn.microsoft.com/ru-ru/dotnet/api/system.string.split?view=net-8.0> (дата обращения: 28.12.2024)
5. Преобразование и приведение типов в C#: сайт. – URL: <https://www.youtube.com/watch?v=cfoec_hRfPU> (дата обращения: 21.12.2024)