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–№ 1 – ПЗ |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Студента** | | Байбулов Ильяс Ерланович | | | | | |
|  |  |  |  | | фамилия, имя, отчество полностью | | | | | |
|  |  |  | Курс | 1 |  | Группа | | ФИТ-242 | | |
|  |  |  |  |  |  | |  |  | |  |
|  | | | **Направление (специальность)** | | | | | ***02.03.02*** | | |
|  | | | Фундаментальная информатика и информационные технологии | | | | | | | |
|  |  |  | 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

**Содержание**

[Введение 3](#_Toc185878851)

[Постановка задачи «Упаковки молока» 4](#_Toc185878852)

[Ход решения задачи «Упаковки молока» 5](#_Toc185878853)

[Постановка задачи «Отбор в разведку» 9](#_Toc185878854)

[Ход решения задачи «Отбор в разведку» 10](#_Toc185878855)

[Постановка задачи «Интенсификация производства» 12](#_Toc185878856)

[Ход решения задачи «Интенсификация производства» 13](#_Toc185878857)

[Постановка задачи «Крестьянин и черт» 15](#_Toc185878858)

[Ход решения задачи «Крестьянин и черт» 16](#_Toc185878859)

[Заключение 18](#_Toc185878860)

[Список используемой литературы 19](#_Toc185878861)

**Введение**

C# представляет собой объектно-ориентированный язык программирования, разработанный командой инженеров Microsoft в период с 1998 по 2002 год. Он принадлежит к группе языков, схожих с C, и обладает синтаксисом, похожим на Java и C++. Среди его ключевых характеристик можно выделить статическую типизацию, полиморфизм, возможность перегрузки операторов, а также поддержку делегатов, атрибутов, событий, обобщённых типов и анонимных функций. Язык заимствовал некоторые элементы из Delphi, Smalltalk и Java, при этом избавившись от ряда проблемных практик.

C# находит применение в разработке разнообразного программного обеспечения, включая бизнес-приложения, видеоигры, современные веб-приложения, а также программное обеспечение для Windows, macOS и мобильных устройств на iOS и Android. Благодаря своей универсальности, его можно считать подходящим для решения множества задач.

Среди основных достоинств C# по сравнению с другими языками программирования можно отметить простоту, широкую популярность, ясный синтаксис и невысокий порог вхождения. Это позволяет разработчикам быстро приступать к созданию полноценных приложений, эффективно разрабатывать функциональные и отзывчивые программы, а также легко читать и понимать код, написанный другими программистами.

## **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.

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

Для определения минимальной стоимости литра молока у некоторой фирмы, нам нужно вычислить стоимость молока каждой фирмы.

Значения размеров двух видов упаковок X1, Y1, Z1 и X2, Y2, Z2, а также их стоимости C1 и C2 мы получим из консольного ввода.

Далее вычислим объемы первой и второй упаковки, а также их площади поверхности: V1 = X1 \* Y1 \* Z1 и V2 = X2 \* Y2 \* Z2, S1 = 2(X1 \* Y1 + Y1 \* Z1 + X1 \* Z1), S2 = 2(X2 \* Y2 + Y2 \* Z2 + X2 \* Z2).

Зная, что стоимость упаковки молока состоит из цены самого молока и цены тары, составим систему уравнений: , где а p — стоимость одного литра молока, а k — стоимость одного квадратного сантиметра упаковки. Далее, с помощью элементарных операций выражаем переменную (цену одного литра молока). Получив итоговую формулу, вычислим ее значение. Данные об этом будут сохраняться лишь до того момента, пока с помощью все тех же операций, не будет найдена другая, более дешевая цена. В ином случае, если более дешевой цены не найдется, то установленная раннее цена и являлась минимальной.

Составим блок-схему алгоритма:

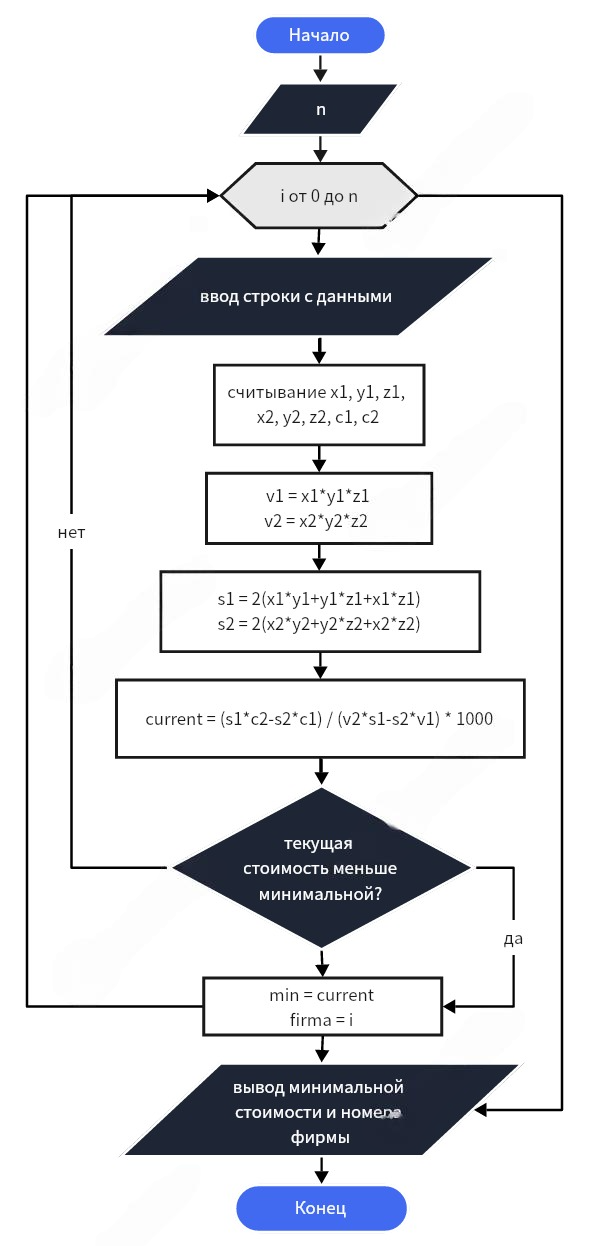


Рисунок 1 – Блок-схема алгоритма задачи 1

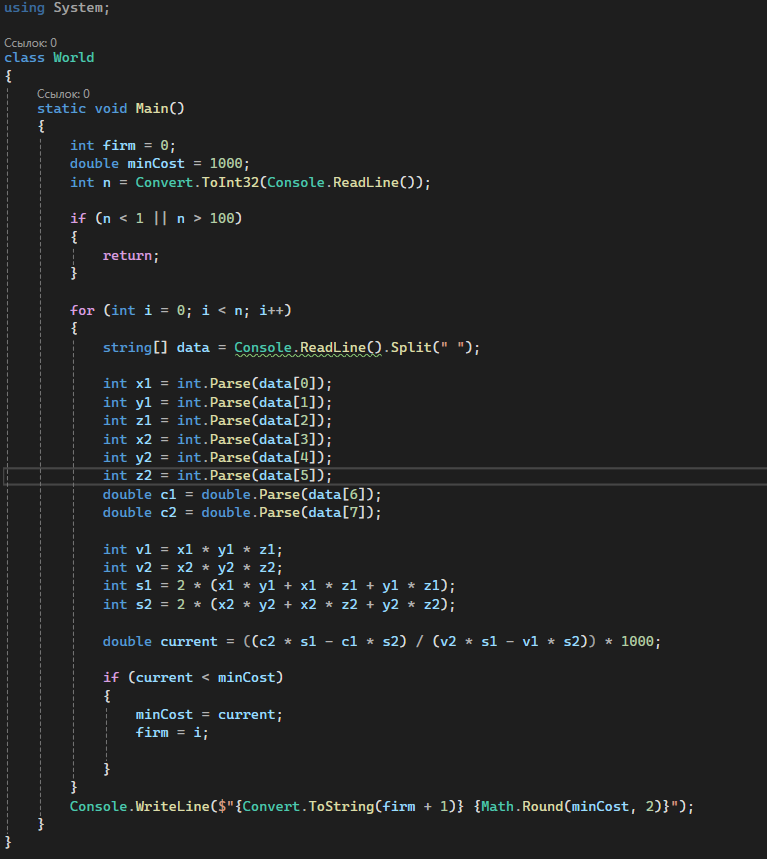


Рисунок 2 – Код решения задачи 1

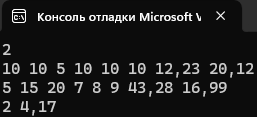


Рисунок 3 – Первый тест

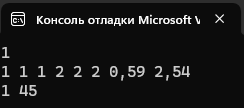
****

Рисунок 4 – Второй тест

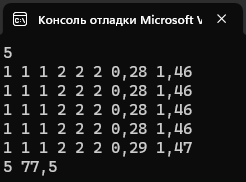


Рисунок 5 – Третий тест

## **Постановка задачи «Отбор в разведку»**

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 «Отбор в разведку»**

Для решения данной задачи будем использовать рекурсивный подход. Шеренга разбивается на две подгруппы (солдаты на четных и нечетных позициях) до тех пор, пока в каждой из подгрупп окажется не более 3 солдат. Далее подсчитываем группы по три солдата.

Код программы:

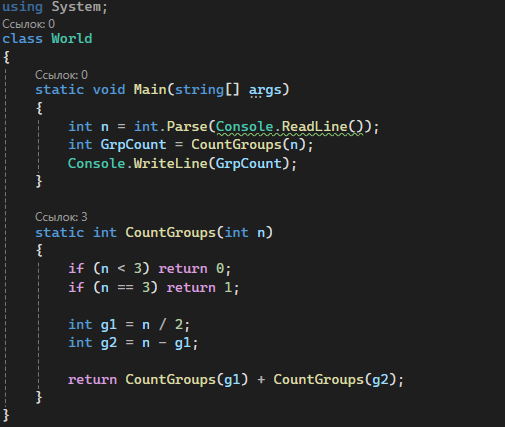


Рисунок 6 – Код решения задачи 2

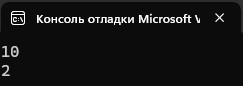


Рисунок 7 – Первый тест

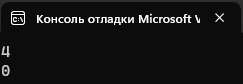


Рисунок 8 – Второй тест

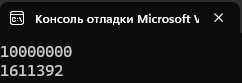


Рисунок 9 – Третий тест

## **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.

## **Ход решения задачи «Интенсификация производства»**

Логика решения поставленной задачи заключается в вычисление общего объема производства за заданный период. Для расчета общего объема продукции, используется формула, которая учитывает следующие составляющие: начальный объем производства, умноженный на количество дней в заданном периоде, а также увеличение производства ежедневно. Ежедневное увеличение производства составляет арифметическую прогрессию. Сумма этой прогрессии будет складываться с общим производством. В итоге программа выводит на консоль суммарное количество произведенной продукции, рассчитанное с учетом заданных условий. Также стоит сказать, что входные данные преобразуются в необходимые форматы: даты разбиваются на день, месяц и год, а начальный объем продукции преобразуется в целое число.

Код программы:

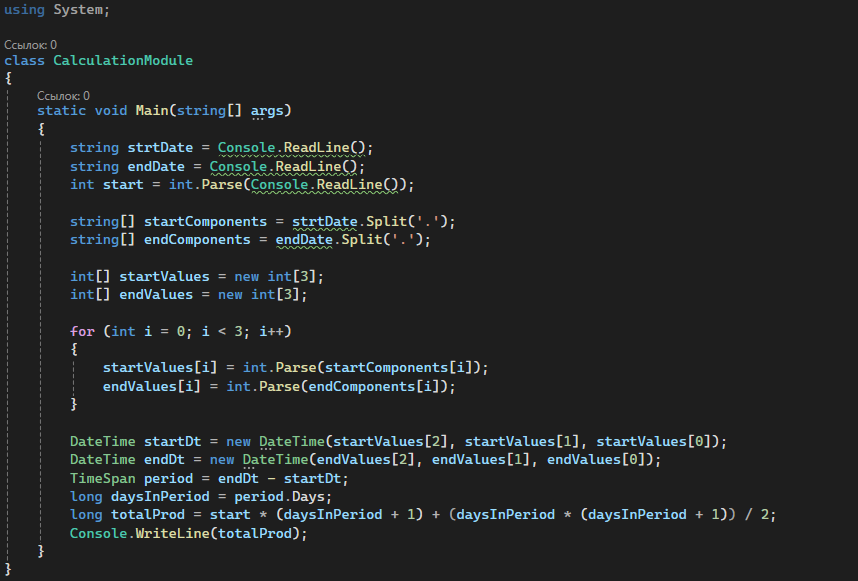


Рисунок 10 – Код решения задачи 3

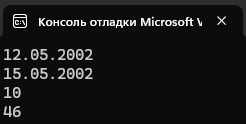


Рисунок 11 – Первый тест

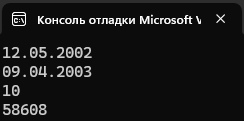


Рисунок 12 – Второй тест

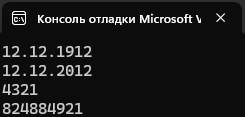


Рисунок 13 – Третий тест

## **Постановка задачи «Крестьянин и черт»**

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 «Крестьянин и черт»**

Алгоритм решения задачи сводится к нахождению количества возможных комбинаций условий перехода через мост, используя начальное количество монет, сумму, которую крестьянин отдает черту после каждого перехода, и количества переходов через мост. Значение максимального количества монет считывается с консоли, затем запускается цикл где постепенно увеличивается значение K по заданной формуле. В каждой итерации цикла вычисляется сколько переходов крестьянин может осуществить, основываясь на текущем K, и это значение добавляется к общему количеству комбинаций.

Код программы:

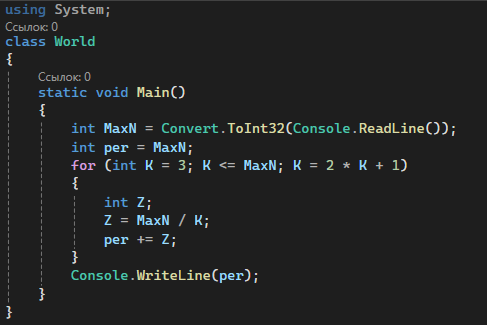


Рисунок 14 – Код решения задачи 4

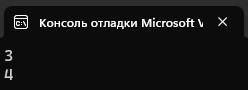


Рисунок 15 – Первый тест

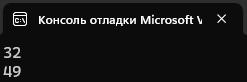


Рисунок 16 – Второй тест

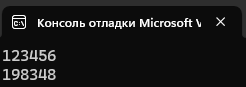


Рисунок 17 – Третий тест

## **Заключение**

В ходе выполнения расчетно-графической работы были использованы основные конструкции C#, такие как переменные, типы данных, операторы, условные конструкции и циклы. Особенное внимание было уделено разработке алгоритмов, обеспечивающих эффективность и точность в решение поставленных задач.

Выполнение работы позволило не только закрепить теоретические знания, но и приобрести практические навыки программирования на языке C#. Полученные знания и опыт будут полезны при дальнейшем изучении программирования и разработке более сложных проектов.

.

## **Список используемой литературы**

1. Документация Microsoft C# <https://learn.microsoft.com/ru-ru/dotnet/csharp/> дата обращения 18.12.2024.
2. Stack Overflow <https://stackoverflow.com> дата обращения 19.12.2024.
3. Васильев А.Н. Программирование на C# для начинающих. Основные сведения. – Эксмо, Москва, 2018. – 592 с.
4. LearnCS <https://learncs.org/> дата обращения 20.12.2024
5. Д.Албахари, Б.Албахари. C# 9.0. Карманный справочник – Диалектика, 2021. – 253 с.