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–№ 23 – ПЗ |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Студента** | | Спицкой Марины Александровны | | | | | |
|  |  |  |  | | 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](#_Toc187613221)

[Постановка задачи «Упаковки молока» 4](#_Toc187613222)

[Ход решения задачи «Упаковки молока» 5](#_Toc187613223)

[Постановка задачи «Угадай число» 9](#_Toc187613224)

[Ход решения задачи «Угадай число» 10](#_Toc187613225)

[Постановка задачи «Тау-Кита» 13](#_Toc187613226)

[Ход решения задачи «Тау-Кита» 15](#_Toc187613227)

[Постановка задачи «Зельеварение» 18](#_Toc187613228)

[Ход решения задачи «Зельеварение» 20](#_Toc187613229)

[Заключение 23](#_Toc187613230)

[Список используемой литературы 24](#_Toc187613231)

**Введение**

C# (читается как «Си шарп») — это язык программирования от компании Microsoft. Изначально его создавали для проектов под Windows, но теперь это по-настоящему универсальный язык: на нём пишут игры, десктопные приложения, веб-сервисы, нейросети и даже графику для метавселенных.

К основным преимуществам языка относят строгую типизацию, сохранение концепций ООП, мощный инструментарий, компактный и читаемый код. Из минусов выделяют сложный синтаксис и невозможность выполнения некоторых специфических задач с помощью библиотек.

Язык также поддерживает современные подходы к разработке, такие как использование LINQ для работы с данными и асинхронные методы для повышения производительности. C# является популярным выбором для разработчиков благодаря своей простоте, мощным возможностям и активному сообществу.

В данной расчетно-графической работе язык C# будет использован с целью решения четырех предложенных задач.

## **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»**

В первую очередь для решения задачи нам необходимо вычислить объемы двух видов упаковок молока. Это возможно сделать по формуле V1 = X1

В первую очередь для решения задачи нам необходимо вычислить объемы двух видов упаковок молока. Это можно сделать по формулам: V1 = X1 \* Y1 \* Z1 и V2 = X2 \* Y2 \* Z2. Также необходимо вычислить площади поверхности двух упаковок. Эти данные вычисляется по формулам: S1 = 2 \* (X1 \* Y1 + X1 \* Z1 + Y1 \* Z1) и S2 = 2 \* (X2 \* Y2 + X2 \* Z2 + Y2 \* Z2). Далее составим систему уравнений для вычисления стоимости молока: . q – стоимость сантиметра упаковки, а p – стоимость одного литра молока. Получим итоговую формулу с помощью выражения значения p из системы уравнений:

Так как исходные размеры упаковок подаются в сантиметрах, а объем молока в литрах, то полученное значение необходимо умножать на 1000.   
цикл повторяется столько раз, сколько разных фирм вводит пользователь. В результате будет выведена минимальная стоимость молока и номер фирмы, которая его производит.

Составим блок-схему алгоритма:

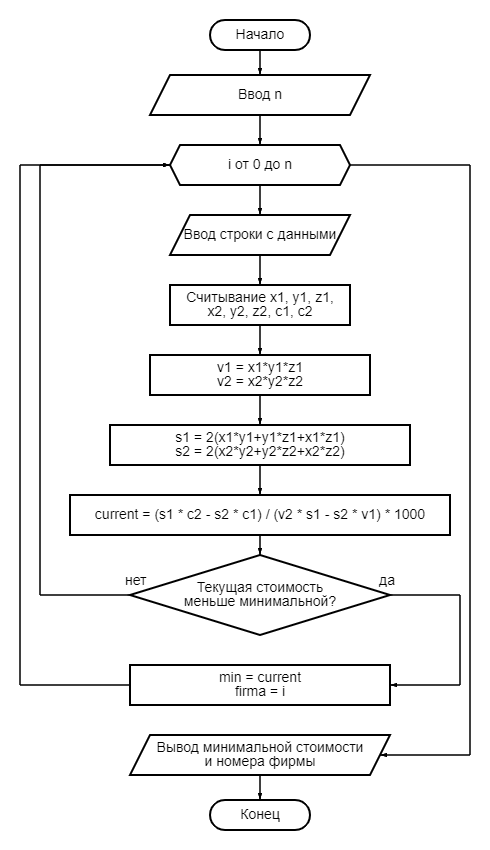


Рисунок 1 – Блок-схема алгоритма задачи 1

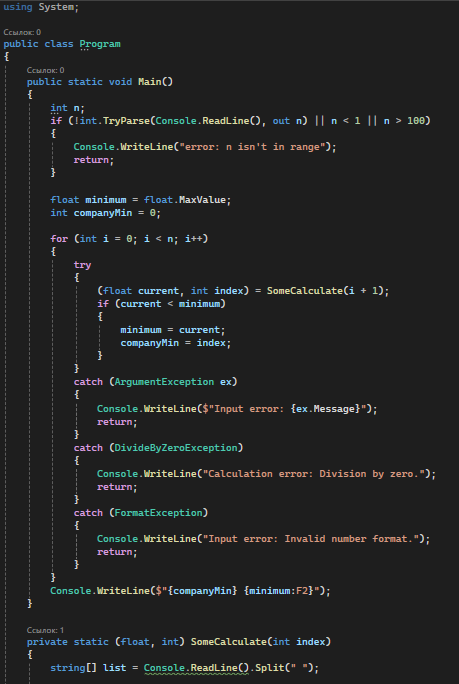


Рисунок 2 – Код решения задачи 1

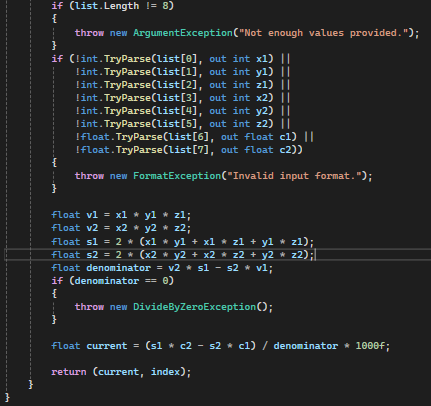


Рисунок 3 – Код решения задачи 1

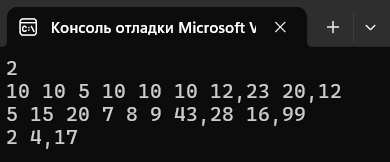


Рисунок 4 – Тест задачи 1

## **Постановка задачи «Угадай 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**.

## **Ход решения задачи «Угадай число»**

Для решения это задачи необходимо представить решение уравнения относительно задуманного числа. Зная, что при решении, все операции попарно меняются на противоположные, а именно «+» меняется на «-» и наоборот, а также «\*» на «/» и наоборот. Таким образом мы можем сразу решать уравнение без его составления.

Код программы:

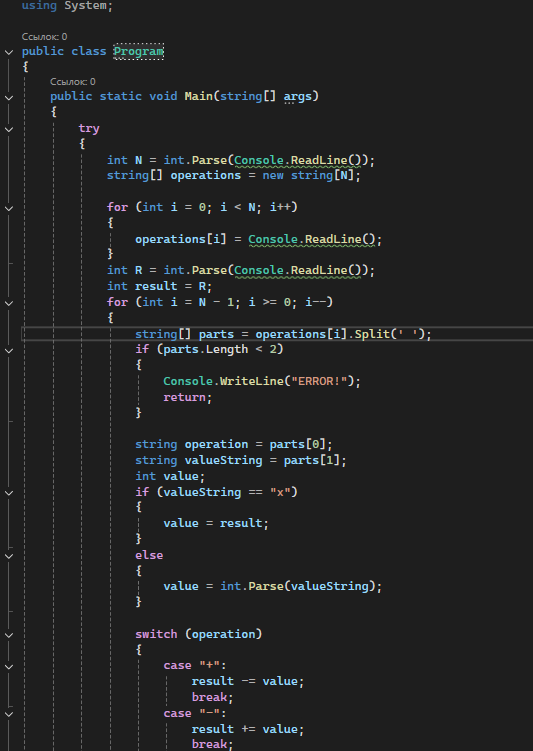


Рисунок 5 – Код решения задачи 2



Рисунок 6 – Код решения задачи 2

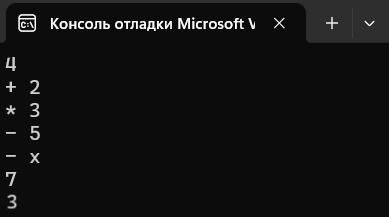


Рисунок 7 – Тест задачи 2

## **Постановка задачи «Тау-Кита»**

В 3573 году Межгалактическая Ассоциация Языков (МАЯ) приняла решение о введении единого межгалактического языка, в котором фраза, после ее прочтения, всегда выглядит одинаково.

Однако, учитывая особенности физиологии аборигенов различных галактик, методы записи одной и той же фразы могут быть различными. Так, аборигены Тау-Кита имеют удлиненное лицо, с каждой из сторон которого расположено по два глаза (т.е. всего 4). При этом, каждый из четырех глаз может независимо считывать информацию. Поэтому алгоритм записи фразы для аборигенов Тау-Кита следующий:

1. в строку записывается первое слово фразы;

2. второе слово дописывается в начало строки;

3. третье слово дописывается в конец строки;

4. четвертое слово снова дописывается в начало строки;

5. и т.д.

Кроме того, в каждом слове буквы располагаются таким же образом, т.е.:

1. записывается первая буква;

2. вторая буква дописывается в начало слова;

3. третья буква дописывается в конец слова;

4. четвертая буква снова дописывается в начало слова;

5. и т.д.

Таким образом, фраза

Eto frasa pervogo primera

будет записана как

rmrpiea srfaa tEo gveproo

Требуется для заданной фразы в формате аборигенов Тау-Кита получить фразу в человеческой интерпретации.

**Ограничения:**

1. Слова в исходной фразе разделены строго одним пробелом.

2. Слова в получаемой фразе должны быть разделены строго одним пробелом.

3. В качестве букв используются только английские строчные и прописные буквы.

4. В начале и конце исходной фразы нет пробелов.

5. В начале и конце получаемой фразы не должно быть пробелов.

6. Длина исходной фразы лежит в диапазоне от 1 до 255 символов.

**Входной файл** содержит исходную фразу в формате аборигенов Тау-Кита.

**Выходной файл** должен содержать человеческую интерпретацию исходной фразы.

## **Ход решения задачи «Тау-Кита»**

В первую очередь для решения данной задачи необходимо переделать алгоритм перевода с человеческого языка на язык аборигенов Тау-Кита. Для этого нам необходимо как в предыдущей задаче выполнять обратные действия тем, что указаны в условии. Обработка фразы начинается с центра и с помощью чередования слов из левой и правой части фразы восстанавливается порядок слов. Далее каждое слово преобразуется аналогичным образом. Начиная с центра каждого слова происходит чередование символов, расположенных справа и слева. После всех преобразований фраза станет понятной для человеческого языка.

Код программы:

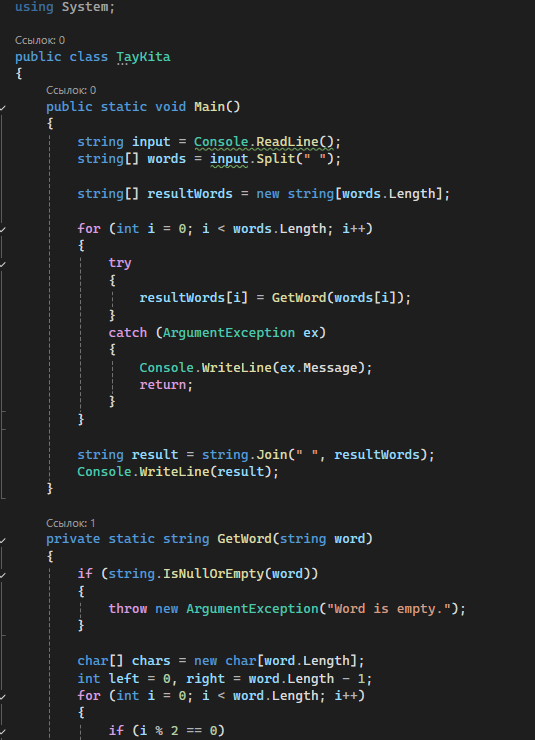


Рисунок 8 – Код решения задачи 3

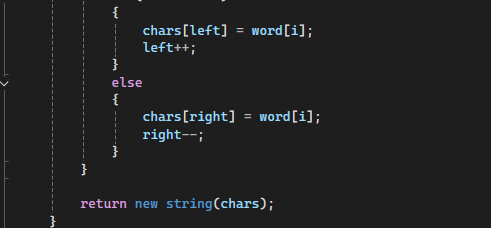


Рисунок 9 – Код решения задачи 3 (продолжение)

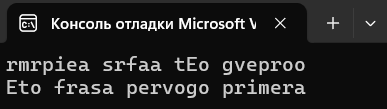


Рисунок 10 – Тест задачи 3

## **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.

## **Ход решения задачи «Зельеварение»**

Алгоритм решения задачи заключается в том, что программа проходит по каждому слову и проверяет, удовлетворяет ли оно условиям (начинается на одну из допустимых букв и заканчивается на другую). Если слово соответствует условиям, итоговый счетчик увеличивается, и количество доступных слов для данной буквы уменьшается. Цикл повторяется, пока остаются непроверенные слова.

Код программы:

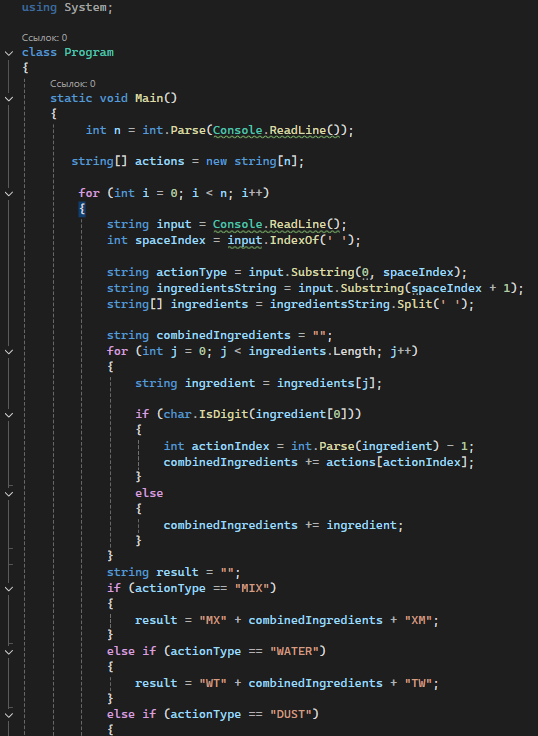


Рисунок 11 – Код решения задачи 4

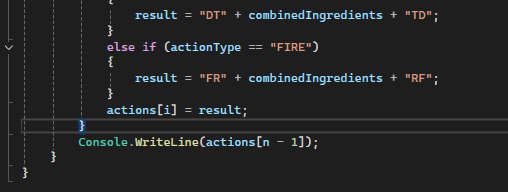


Рисунок 12 – Код решения задачи 4 (продолжение)

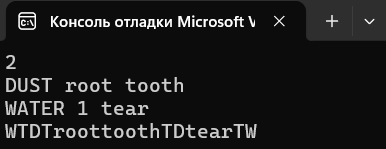


Рисунок 13 – Тест задачи 4

## **Заключение**

При выполнении расчетно-графической работы я изучила основные правила и способы написания программ на языке С#. Был изучен синтаксис, основные функции, и основные концепции языка. С помощью данной работы я изучила и поняла базовые принципы программирования на языке С#.

Работа принесла множество возможностей для дальнейшего развития полученных знаний и опыта. Например, в виде мотивации к более углубленному изучению языка для сложных профессиональных задач. Также расчетно-графическая работа более широко показала возможности языка программирования С#.

## **Список используемой литературы**

1. Документация Microsoft C# https://learn.microsoft.com/ru-ru/dotnet/csharp/ дата обращения 10.01.2025.
2. [Online C# Compiler (Editor) - Programiz](https://www.programiz.com/csharp-programming/online-compiler/) https://www.programiz.com дата обращения 11.01.2025
3. <https://dotnet.microsoft.com/> дата обращения 11.01.2025
4. https://otus.ru/ дата обращения 12.01.2025
5. https://kpfu.ru/ дата обращения 12.01.2025