Министерство образования и науки Российской Федерации

Северо-Кавказский горно-металлургический институт (государственный технологический университет)

Кафедра информатики и вычислительной техники

**ОТЧЕТ**

**по лабораторной работе № 1**

**по дисциплине:**

**«лингвистическое и программное обеспечение САПР»**

**Выполнил ст.гр. ИВб-22-2**:

Степанян Константин Александрович

**Проверил**:

Преподаватель: Фомин Андрей Николаевич

Владикавказ 2025

ВАРИАНТ 1

1. Дан файл f, содержащий сведения о редукторах: КПД зубчатой передачи, число зубьев шестерни, тип редуктора (цилиндрический редуктор, конический редуктор, червячный редуктор, планетарный редуктор, комбинированный редуктор). Вывести на экран и в в типизированный файл информацию о редукторах: имеющих заданный пользователем КПД; имеющих заданный пользователем тип; имеющих не более N зубьев шестерни и КПД не менее M. Для хранения информации использовать линейные динамические структуры.

Program.cs:

**using** System**;**

**using** System**.**Collections**.**Generic**;**

**using** System**.**IO**;**

**using** System**.**Globalization**;**

**namespace** ReductorApp **{**

class Program **{**

// Структура, описывающая редуктор

public struct Reductor **{**

public float kpd**;** // КПД (коэффициент полезного действия)

public int z**;** // Количество зубьев

public string type**;** // Тип редуктора

**}**

static void Main**()** **{**

// Устанавливаем вывод в консоль в UTF-8, чтобы корректно отображались русские символы

Console**.**OutputEncoding **=** System**.**Text**.**Encoding**.**UTF8**;**

// Информация о программе

Console**.**WriteLine**(**"Программа читает файл reductors.txt и находит редукторы по заданным критериям:"**);**

Console**.**WriteLine**(**"1. Редукторы с заданным КПД"**);**

sConsole**.**WriteLine**(**"2. Редукторы с заданным типом"**);**

Console**.**WriteLine**(**"3. Редукторы с числом зубьев не более N и КПД не менее M\n"**);**

// Список для хранения всех редукторов из файла

List**<**Reductor**>** reductors **=** **new** List**<**Reductor**>();**

**try** **{**

// Чтение всех строк из файла в кодировке UTF-8

string**[]** lines **=** File**.**ReadAllLines**(**"reductors.txt"**,** System**.**Text**.**Encoding**.**UTF8**);**

// Обработка каждой строки

**for** **(**int i **=** 0**;** i **<** lines**.**Length**;** i**++)** **{**

// Разделяем строку по запятой

string**[]** parts **=** lines**[**i**].**Split**(**','**);**

**if** **(**parts**.**Length **==** 3**)** **{**

**try** **{**

// Преобразуем данные в структуру Reductor

Reductor r**;**

r**.**kpd **=** ParseFloat**(**parts**[**0**].**Trim**());** // КПД

r**.**z **=** int**.**Parse**(**parts**[**1**].**Trim**());** // Кол-во зубьев

r**.**type **=** parts**[**2**].**Trim**();** // Тип

reductors**.**Add**(**r**);** // Добавляем в список

**}**

**catch** **{** Console**.**WriteLine**(**"⚠ Ошибка при разборе строки: " **+** lines**[**i**]);** **}** **}** **}**

// --- Поиск редукторов по КПД ---

Console**.**Write**(**"\nВведите искомый КПД: "**);**

float targetKPD **=** ParseFloat**(**Console**.**ReadLine**());**

List**<**Reductor**>** foundKPD **=** **new** List**<**Reductor**>();**

// Перебор всех редукторов и сравнение КПД

**for** **(**int i **=** 0**;** i **<** reductors**.**Count**;** i**++)** **{**

**if** **(**reductors**[**i**].**kpd **==** targetKPD**)**

foundKPD**.**Add**(**reductors**[**i**]);**

**}**

Console**.**WriteLine**(**$"\nРедукторы с КПД = {targetKPD}:"**);**

PrintList**(**foundKPD**);**

// --- Поиск по типу (подстрока) ---

Console**.**Write**(**"\nВведите искомый тип редуктора: "**);**

string targetType **=** Console**.**ReadLine**().**Trim**().**ToLower**();**

List**<**Reductor**>** foundType **=** **new** List**<**Reductor**>();**

// Перебор всех редукторов и проверка, содержит ли тип нужную подстроку

**for** **(**int i **=** 0**;** i **<** reductors**.**Count**;** i**++)** **{**

**if** **(**reductors**[**i**].**type**.**ToLower**().**Contains**(**targetType**))**

foundType**.**Add**(**reductors**[**i**]);**

**}**

Console**.**WriteLine**(**$"\nРедукторы с типом \"{targetType}\":"**);**

PrintList**(**foundType**);**

// --- Поиск по числу зубьев ≤ N и КПД ≥ M ---

Console**.**Write**(**"\nВведите максимальное число зубьев (N): "**);**

int maxZ **=** int**.**Parse**(**Console**.**ReadLine**());**

Console**.**Write**(**"Введите минимальный КПД (M): "**);**

float minKPD **=** ParseFloat**(**Console**.**ReadLine**());**

List**<**Reductor**>** foundCond **=** **new** List**<**Reductor**>();**

// Перебор и проверка по обоим условиям

**for** **(**int i **=** 0**;** i **<** reductors**.**Count**;** i**++)** **{**

**if** **(**reductors**[**i**].**z **<=** maxZ **&&** reductors**[**i**].**kpd **>=** minKPD**)**

foundCond**.**Add**(**reductors**[**i**]);**

**}**

Console**.**WriteLine**(**$"\nРедукторы с числом зубьев ≤ {maxZ} и КПД ≥ {minKPD}:"**);**

PrintList**(**foundCond**);**

**}**

**catch** **(**FileNotFoundException**)** **{**

// Если файл не найден

Console**.**WriteLine**(**"❌ Файл reductors.txt не найден. Убедитесь, что он находится рядом с .exe или в каталоге проекта."**);** **}**

// Пауза перед завершением

Console**.**WriteLine**(**"\nНажмите любую клавишу для выхода..."**);**

Console**.**ReadKey**();**

**}**

// Вывод списка редукторов в консоль

static void PrintList**(**List**<**Reductor**>** list**)** **{**

**if** **(**list**.**Count **==** 0**)** **{** Console**.**WriteLine**(**"❗ Ничего не найдено."**);** **}**

**else** **{**

**for** **(**int i **=** 0**;** i **<** list**.**Count**;** i**++)** **{**

Console**.**WriteLine**(**"КПД: " **+** list**[**i**].**kpd **+** ", Зубья: " **+** list**[**i**].**z **+** ", Тип: " **+** list**[**i**].**type**);** **}** **}** **}**

// Метод для преобразования строки в float с заменой запятой на точку

static float ParseFloat**(**string input**)** **{**

input **=** input**.**Replace**(**','**,** '.'**);** // Заменяем запятую на точку

**return** float**.**Parse**(**input**,** CultureInfo**.**InvariantCulture**);** // Используем инвариантную культуру

**}** **}** **}**

Вывод:

