|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 **Информатика и вычислительная техника**

**Отчет**

**по лабораторной работе № 10**

**Дисциплина:** Разработка приложений на языке C#

**Название работы:** Интерфейсы, сортировка с компаратором



Студент гр. ИУ6-72Б **\_\_**16.12.2022**\_\_\_\_\_\_\_** И.С.Марчук

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** А.М. Минитаева

(Подпись, дата) (И.О. Фамилия)

Москва, 2022

**Задание:**

Дан текстовый файл, содержащий данные о продуктах на складе и их

описания, например:

3кг Апельсины

10л Квас

100л Вода

3780г Шоколад

10т Бананы

13кг Мангал

1. Определите класс с тремя закрытыми полями:

* Количество в кг (вещественное число);
* Исходное представление количества (строка);
* Название (строка).

2. Реализуйте конструктор, принимающий на вход

два строковых значения: количество и название товара. Конструктор должен

генерировать исключение, если количество является некорректным (меньше

нуля). Добавьте свойства для преобразования количества в кг. В основной

программе загрузите все температурные данные из исходного файла в

список List< > .

3. Попытайтесь вызвать метод Sort для загруженного ранее списка

температур. Возникающее при этом исключение свидетельствует о

невозможности выполнять сравнение объектов произвольного класса.

Чтобы это стало возможным, необходимо, например, реализовать в

классе интерфейс IComparable<T> . Для этого:

* измените заголовок класса на следующий

class ‘Название’: IComparable<’Название’>

* Необходимости реализовать метод сравнения. Метод сравнения

должен возвращать отрицательное число, если объект, для которого вызывается метод, меньше объекта, переданного в качестве параметра, 0 — если оба объекта равны, и положительное число — если исходный объект больше — реализуйте этот метод;

4. Убедитесь, что метод Sort работает и сортирует список.

**Выполнение задания:**

Был разработан класс для создания объектов «Продуктовый магазин», хранящий в себе наименование продукта и его вес:

internal class Product : IComparable<Product>

{

private string \_name;

private string \_count\_str;

private double \_count;

public Product(string name, string count\_str)

{

\_name = name;

\_count\_str = count\_str;

try

{

\_count = MyConvert(count\_str);

}

catch(ArgumentException f)

{

\_name = "Err ";

\_count = 0;

Console.WriteLine($"{f}");

}

}

Функция конвертации строки в число:

private double MyConvert(string str)

{

if (str.Length < 2)

throw new ArgumentException("not correct");

if (str[0] < '1' || str[0] > '9')

throw new ArgumentException("not correct number");

if (str[0] == '-')

throw new ArgumentException("only positive!!!");

double count = 0, e = 1;

int offset = 1;

if (!char.IsDigit(str[str.Length - 2]))

{

switch ((char)str[str.Length - 2])

{

case 'к':

e = 1000;

break;

case 'м':

e = 0.001;

break;

}

offset = 2;

}

str = str.Remove(str.Length - offset, offset);

count = double.Parse(str);

return count \* e;

}

}

Основная программа:

static void Main(string[] args)

{

List<Product> pr = new List<Product>();

var lines = File.ReadLines("data.txt");

foreach (var line in lines)

{

var buf = line.Split(" ");

if (buf.Length != 2)

continue;

string count = buf[0];

string name = buf[1];

pr.Add(new Product(name, count));

}

Console.WriteLine(" Вывод списка:\n");

foreach (var p in pr)

{

Console.WriteLine($"{p.getName()} {p.getCount()}");

}

Console.WriteLine();

Console.WriteLine(" Проверка сортировки:\n");

pr.Sort();

foreach (var p in pr)

{

Console.WriteLine($"{p.getName()} {p.getCount()}");

}

}

Работа программы приведена на рисунке 1.

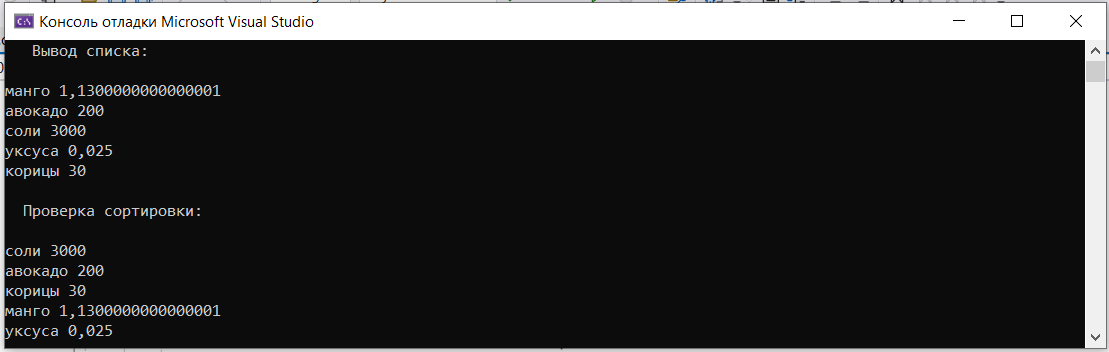


Рисунок 1 – Выполнение программы

**Вывод**: в процессе выполнения лабораторной работы был создан класс, унаследованный от IComparable и с определенной функцией сравнения двух объектов. Также в программе используется построчное чтение из файла.