Министерство образования и молодежной политики Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по учебной практике

УП 01.02

Выполнила: Рудных Екатерина Юрьевна

Группа: ПР-21

Преподаватель: Мирошниченко Г.В.

2025

Содержание

Оглавление

[Задание №3. Мобильное приложение «Питание» 2](#_Toc200545123)

[1.1 Описание задачи 2](#_Toc200545124)

[1.2 Структура проекта 3](#_Toc200545125)

[1.3 Описание разработанных функций 10](#_Toc200545126)

[1.4 Алгоритм решения 11](#_Toc200545127)

[1.5 Используемые библиотеки 11](#_Toc200545128)

[1.6 Тестовые случаи 12](#_Toc200545129)

[1.7 Используемые инструменты 18](#_Toc200545130)

[1.8 Описание пользовательского интерфейса 18](#_Toc200545131)

[1.9 Приложение 20](#_Toc200545132)

# Задание №3. Мобильное приложение «Питание»

## 1.1 Описание задачи

Разработать Windows Forms приложение на языке C#, реализующее систему управления продуктами питания. Программа должна позволять:

1. Вводить данные о продуктах: название, количество белков, углеводов и калорий.
2. Рассчитывать «качество» продукта по заданной формуле:
   1. Для базового продукта: Q = Белки \* 4 + Углеводы \* 4
   2. Для расширенного продукта: Qp = Q \* 1.2 + Калории \* 7
3. Отображать список продуктов в таблице
4. Реализовывать операции добавления, удаления и фильтрации записей
5. Использовать коллекции и LINQ для обработки данных
6. Добавить комментарии

## 1.2 Структура проекта

**Входные данные:**

Название (Name) — строка, только буквы.

Белки (Protein) — положительное число.

Углеводы (Carbs) — положительное число.

Калории (Calories) — положительное число.

Выбор строки в DataGridView.

Число Q — положительное число для фильтрации продуктов с качеством выше указанного.

**Выходные данные:**

Таблица продуктов (dataGridView1):

Название (Name)

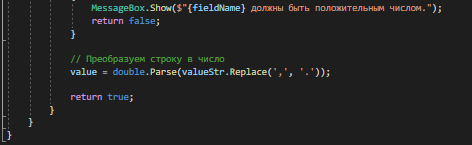
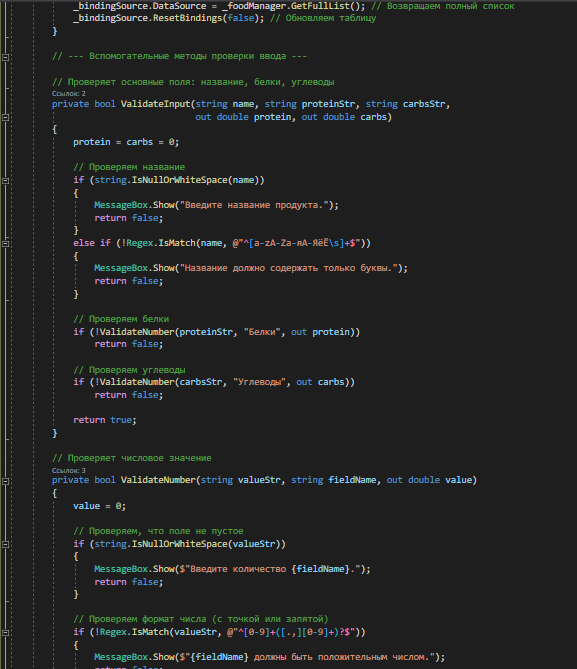
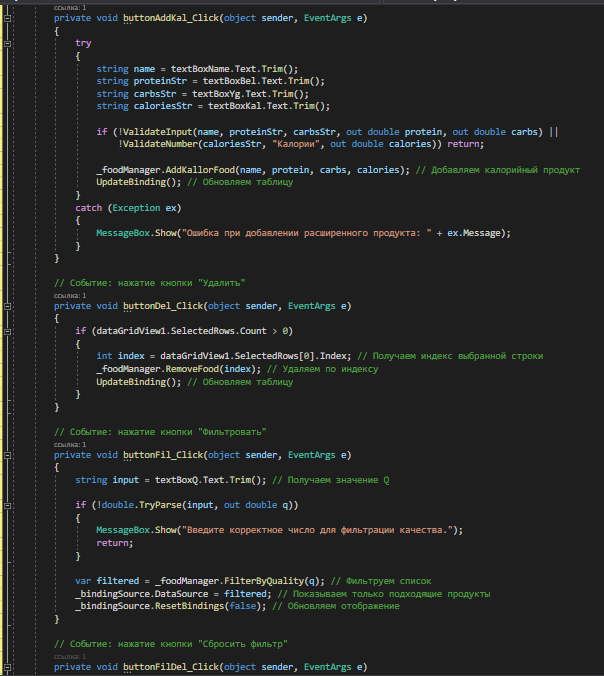
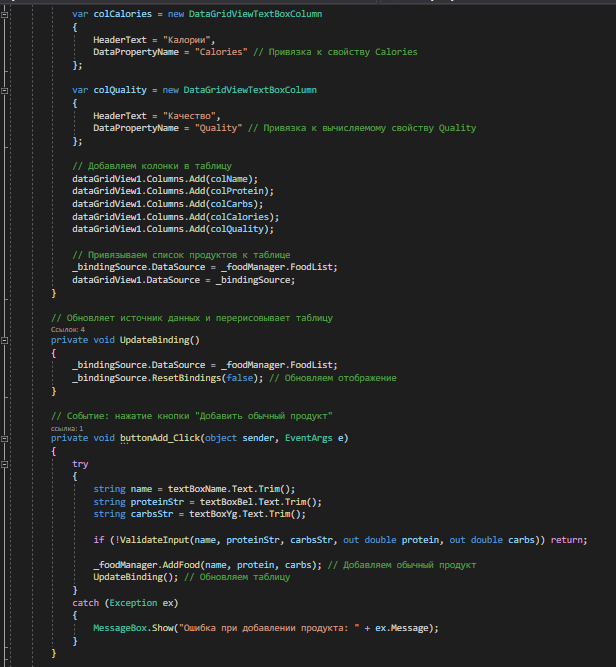
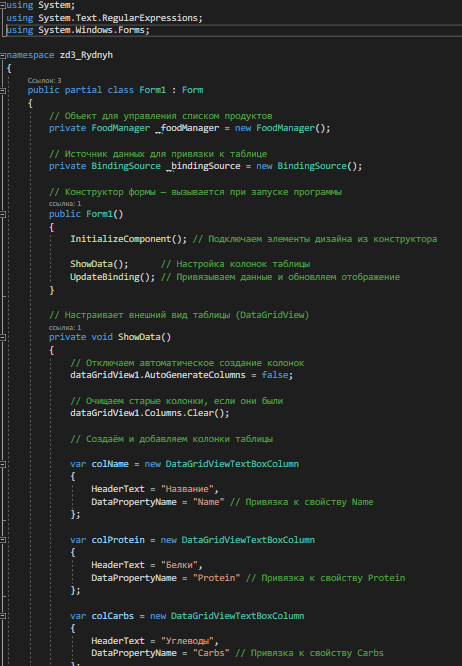
Белки (Protein)

Углеводы (Carbs)

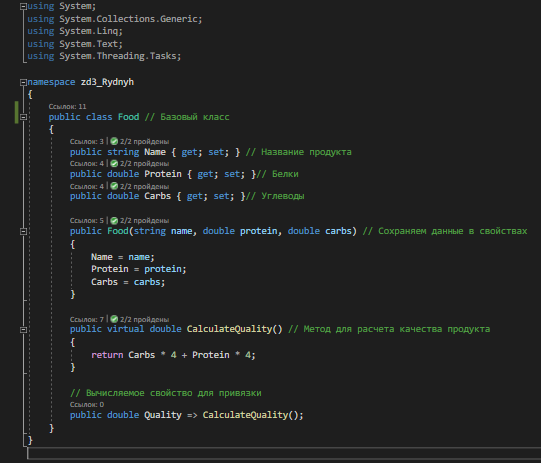
Калории (Calories)

Качество (Quality) — вычисляется по формуле

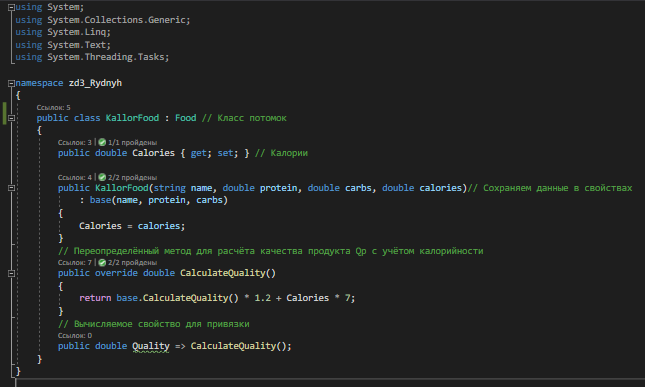
**Код формы:**

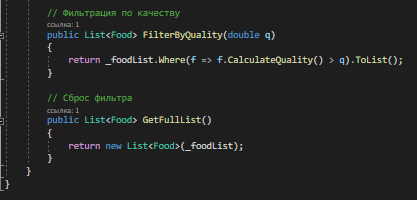
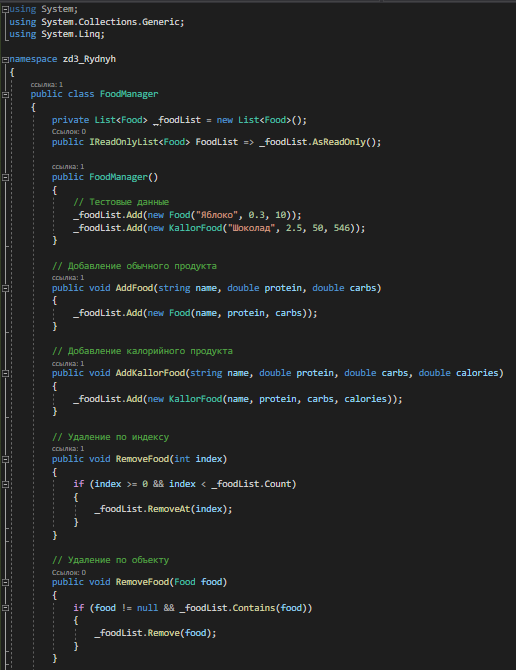


**Код базового класса:**

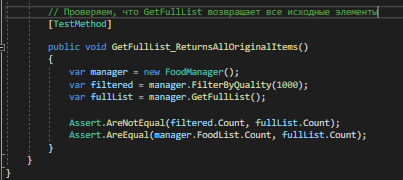
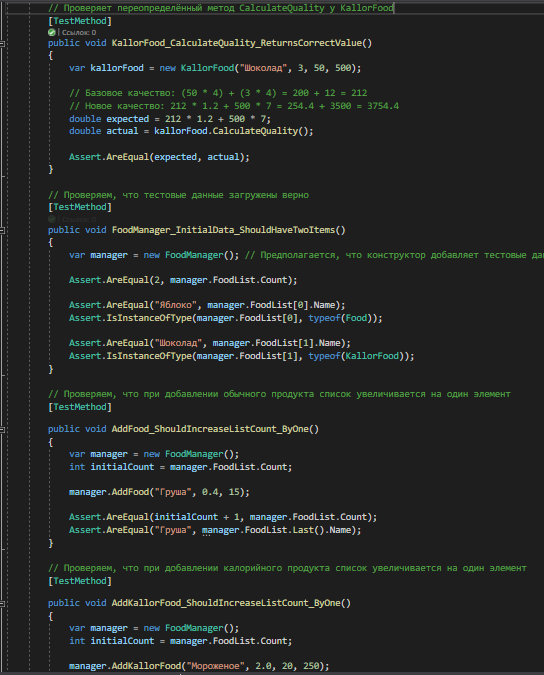
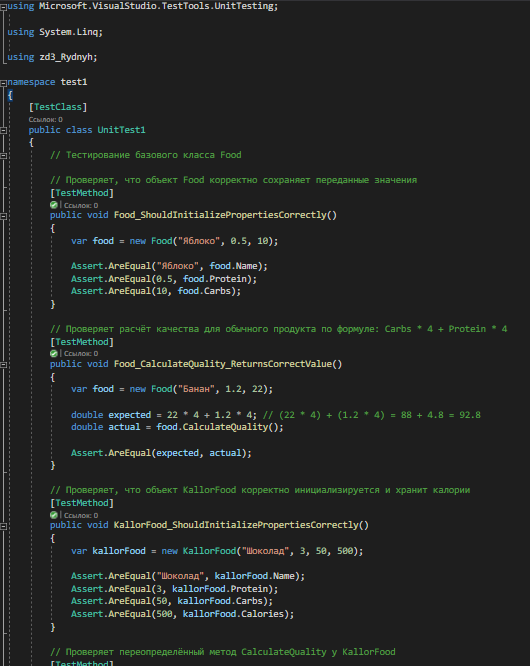


**Код класса потомка:**



**Код класса с методами:**

**Код тестов:**

****

## 1.3 Описание разработанных функций

1**. Метод AddFood**

**Назначение:** Добавление обычного продукта в список.

**Входные данные:**

name (string) – название продукта

protein (double) – количество белков

carbs (double) – количество углеводов

**Выходные данные:** изменяет \_foodList, добавляя новый объект Food.

1. **Метод AddKallorFood**

**Назначение:** Добавление калорийного продукта в список.

**Входные данные:**

name (string) – название продукта

protein (double) – количество белков

carbs (double) – количество углеводов

calories (double) – количество калорий

**Выходные данные:** изменяет \_foodList, добавляя новый объект KallorFood.

1. **Метод RemoveFood (по индексу)**

**Назначение:** Удаление продукта из списка по индексу.

**Входные данные:**

index (int) – позиция элемента в списке (начиная с 0).

**Выходные данные:** удаляет элемент из \_foodList, если индекс корректен.

1. **Метод RemoveFood (по объекту)**

**Назначение:** Удаление продукта из списка по ссылке на объект.

**Входные данные:**

food (Food) – объект для удаления.

**Выходные данные:** удаляет элемент из \_foodList, если он существует.

1. **Метод FilterByQuality**

**Назначение:** Фильтрация продуктов по качеству (значение > q).

**Входные данные:**

q (double) – пороговое значение качества.

**Выходные данные:** List<Food> – отфильтрованный список продуктов, где CalculateQuality() > q.

1. **Метод GetFullList**

**Назначение:** Получение полного списка продуктов (без фильтрации).

**Входные данные:** Нет.

**Выходные данные:** List<Food> – копия списка \_foodList.

## 1.4 Алгоритм решения

Блок схема к форме:

Блок схема к базовому классу:

Блок схема к классу потомку:

Блок схема к классу с методами:

## 1.5 Используемые библиотеки

using System;

using System.Collections.Generic;

using System.Linq;  
using Microsoft.VisualStudio.TestTools.UnitTesting;

using zd3\_Rydnyh;

using System.Text;

using System.Text.RegularExpressions;

using System.Windows.Forms;

using System.Threading.Tasks;

## 1.6 Тестовые случаи

**1. Тест: Food\_ShouldInitializePropertiesCorrectly**

Назначение теста: Проверить корректность инициализации объекта Food.

**Входные данные:**

- Название: "Яблоко"

- Белки: 0.5

- Углеводы: 10

**Выходные данные:**

- Объект Food с корректно установленными свойствами (Name, Protein, Carbs)

**Проверяемые условия:**

- food.Name == "Яблоко"

- food.Protein == 0.5

- food.Carbs == 10

**2. Тест: Food\_CalculateQuality\_ReturnsCorrectValue**

**Назначение теста:** Проверить корректность расчёта качества для обычного продукта

**Входные данные:**

- Объект Food: "Банан", 1.2 (белки), 22 (углеводы)

**Выходные данные:**

- Рассчитанное качество: 92.8 (по формуле Carbs \* 4 + Protein \* 4)

**Проверяемые условия:**

- food.CalculateQuality() == 92.8

**3. Тест: KallorFood\_ShouldInitializePropertiesCorrectly**

**Назначение теста:** Проверить корректность инициализации объекта KallorFood

**Входные данные:**

- Название: "Шоколад"

- Белки: 3

- Углеводы: 50

- Калории: 500

**Выходные данные:**

- Объект KallorFood с корректно установленными свойствами (Name, Protein, Carbs, Calories)

**Проверяемые условия:**

- kallorFood.Name == "Шоколад"

- kallorFood.Protein == 3

- kallorFood.Carbs == 50

- kallorFood.Calories == 500

**4. Тест: KallorFood\_CalculateQuality\_ReturnsCorrectValue**

**Назначение теста:** Проверить переопределённый расчёт качества для KallorFood

**Входные данные:**

- Объект KallorFood: "Шоколад", 3 (белки), 50 (углеводы), 500 (калории)

**Выходные данные:**

- Рассчитанное качество: 3754.4 (по формуле (baseQuality \* 1.2) + (Calories \* 7))

**Проверяемые условия:**

- kallorFood.CalculateQuality() == 3754.4

**5. Тест: FoodManager\_InitialData\_ShouldHaveTwoItems**

**Назначение теста:** Проверить, что FoodManager инициализируется с двумя тестовыми продуктами

**Входные данные:**

- Конструктор FoodManager (предполагается, что он добавляет "Яблоко" и "Шоколад")

**Выходные данные:**

- Список FoodList с 2 элементами:

- Первый элемент: Food ("Яблоко")

- Второй элемент: KallorFood ("Шоколад")

**Проверяемые условия:**

- manager.FoodList.Count == 2

- manager.FoodList[0] is Food

- manager.FoodList[1] is KallorFood

**6. Тест: AddFood\_ShouldIncreaseListCount\_ByOne**

**Назначение теста:** Проверить добавление обычного продукта в список

**Входные данные:**

- Название: "Груша"

- Белки: 0.4

- Углеводы: 15

**Выходные данные:**

- Увеличение размера FoodList на 1

- Последний элемент списка - Food с именем "Груша"

**Проверяемые условия:**

- manager.FoodList.Count == initialCount + 1

- manager.FoodList.Last().Name == "Груша"

**7. Тест: AddKallorFood\_ShouldIncreaseListCount\_ByOne**

Назначение теста: Проверить добавление калорийного продукта в список

**Входные данные:**

- Название: "Мороженое"

- Белки: 2.0

- Углеводы: 20

- Калории: 250

**Выходные данные:**

- Увеличение размера FoodList на 1

- Последний элемент списка - KallorFood с именем "Мороженое"

**Проверяемые условия:**

- manager.FoodList.Count == initialCount + 1

- manager.FoodList.Last() is KallorFood

**8. Тест: RemoveFood\_ByIndex\_RemovesCorrectItem**

**Назначение теста:** Проверить удаление продукта по индексу

**Входные данные:**

- Индекс: 0 (первый элемент)

**Выходные данные:**

- Уменьшение размера FoodList на 1

- Первый элемент больше не "Яблоко"

**Проверяемые условия:**

- manager.FoodList.Count == initialCount - 1

- manager.FoodList[0].Name != "Яблоко"

**9. Тест: RemoveFood\_ByObject\_RemovesCorrectItem**

**Назначение теста:** Проверить удаление продукта по ссылке на объект

**Входные данные:**

- Объект Food с именем "Яблоко"

**Выходные данные:**

- Уменьшение размера FoodList на 1

- Отсутствие элемента с Name == "Яблоко"

**Проверяемые условия:**

- manager.FoodList.Count == initialCount - 1

- !manager.FoodList.Any(f => f.Name == "Яблоко")

**10. Тест: FilterByQuality\_ReturnsOnlyItemsWithHigherQuality**

**Назначение теста:** Проверить фильтрацию по качеству (CalculateQuality() > q)

**Входные данные:**

- Пороговое значение качества: 1000

**Выходные данные:**

- Список продуктов, у которых CalculateQuality() > 1000

**Проверяемые условия:**

- Все элементы в result удовлетворяют условию food.CalculateQuality() > 1000

**11. Тест: GetFullList\_ReturnsAllOriginalItems**

**Назначение теста:** Проверить, что GetFullList возвращает полный список (без фильтрации)

**Входные данные:**

- Предварительно отфильтрованный список (FilterByQuality(1000))

**Выходные данные:**

- Полный список продуктов (исходный размер)

**Проверяемые условия:**

- fullList.Count == manager.FoodList.Count

- fullList.Count != filtered.Count

## 1.7 Используемые инструменты

Язык программирования: C#

Среда разработки: Microsoft Visual Studio

Фреймворк: .NET Framework 5

## 1.8 Описание пользовательского интерфейса

**Кнопка "Добавить продукт":**

Пользователь вводит в текстовые поля следующие данные о продукте: название, количество белков, количество углеводов. После нажатия кнопки введенные данные отображаются в таблице DataGrid.

**Кнопка "Добавить продукт с калориями":**

Пользователь вводит в текстовые поля данные о продукте: название, количество белков, количество углеводов, а также калорийность. После нажатия кнопки вся введенная информация отображается в таблице DataGrid.

**Кнопка "Удалить продукт":**

Для удаления продукта пользователь выбирает соответствующую строку в таблице DataGrid и нажимает эту кнопку. Выбранный продукт удаляется из списка.

**Поле "Введите Q":**

Предназначено для ввода числового значения Q, которое используется для фильтрации списка продуктов по показателю качества.

**Кнопка "Фильтровать список":**

После нажатия этой кнопки в таблице DataGrid отображаются только те продукты, показатель качества которых превышает введенное значение Q.

**Кнопка "Убрать фильтрацию":**

Нажатие этой кнопки отменяет действующую фильтрацию и восстанавливает отображение полного списка продуктов в таблице DataGrid.1.9 Приложение (pr screen экранов)

**Поле "Введите название продукта":**

Предназначено для ввода названия, которое используется для вывода информации.

**Поле "Введите кол-во белков":**

Предназначено для ввода белков, которое используется для вывода информации и формулы.

**Поле "Введите кол-во углеводов":**

Предназначено для ввода углеводов, которое используется для вывода информации и формулы.

**Поле "Введите калорийность продукта":**

Предназначено для ввода калорийности, которое используется для вывода информации и формулы.

**DataGridView:**

Используется для вывода информации.

## 1.9 Приложение

