

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра інженерії програмного забезпечення

КУРСОВА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Об'єктно-орієнтоване програмування»

на тему:

«Віконний додаток "Калькулятор калорій"»

студента I курсу групи ІПЗ-24-1
спеціальності 121 «Інженерія програмного
забезпечення»
СЕМЕНОВА Олександра Петровича
(прізвище, ім'я та по-батькові)

Керівник Старший викладач кафедри ІПЗ
Олексій ЧИЖМОТРЯ

Дата захисту: " ____ " _____ 20__ р.
Національна шкала _____
Кількість балів: _____
Оцінка: ECTS _____

Члени комісії

(підпис)

Тетяна ВАКАЛЮК

(прізвище та ініціали)

(підпис)

Олексій ЧИЖМОТРЯ

(прізвище та ініціали)

(підпис)

Олег ВЛАСЕНКО

(прізвище та ініціали)

(підпис)

Дмитро ПРОХОРЧУК

(прізвище та ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Факультет інформаційно-комп'ютерних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень: бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри ІПЗ

_____ Тетяна ВАКАЛЮК

«____»_____ 2025 р.

ЗАВДАННЯ
НА КУРСОВУ РОБОТУ СТУДЕНТУ
Семенову Олександру Петровичу

1. Тема роботи: Розробка віконного додатку «Калькулятор калорій»
керівник роботи: ст. викладач кафедри ІПЗ ЧИЖМОТРЯ Олексій
2. Срок подання студентом: « 19 » травня 2025 р.
3. Вихідні дані до роботи: Розробити програму "Калькулятор калорій" – розрахунок калорій продуктів які вводить користувач
4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)
 1. Постановка завдання
 2. Аналіз аналогічних розробок
 3. Алгоритми роботи програми
 4. Опис роботи програми
 5. Програмне дослідження
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
1.https://docs.google.com/presentation/d/1lcwBxDtFFslyFoyTH-82Lad6uD1nLTZ4xPsQOudMLg/edit?usp=sharing
2. Посилання на репозиторій: https://git.ztu.edu.ua/ipz/2024-2028/ipz-24-1/Semenov-Oleksandr/calculatorcalories/-/tree/master?ref_type=heads
6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посади консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	ЧИЖМОТРЯ Олексій	04.03.2025	04.03.2025

7. Дата видачі завдання “ 04 ” березня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів курсового проєктування	Строк виконання етапів роботи	Примітки
1	Постановка задачі	15.03.2025	виконано
2	Пошук, огляд та аналіз аналогічних розробок	20.03.2025	виконано
3	Формулювання технічного завдання	24.03.2025	виконано
4	Опрацювання літературних джерел	26.03.2025	виконано
5	Проектування структури	04.04.2025	виконано
6	Написання програмного коду	15.04.2025	виконано
7	Відлагодження	18.05.2025	виконано
8	Написання пояснівальної записки	20.05.2025	виконано
9	Захист		

Студент _____
(підпис)

Олександр СЕМЕНОВ
(прізвище та ініціали)

Керівник роботи _____
(підпис)

Олексій ЧИЖМОТРЯ
(прізвище та ініціали)

РЕФЕРАТ

Завданням на курсову роботу було створення програми для розрахунку калорій.

Пояснювальна записка до курсової роботи на тему «Віконний додаток “Калькулятор калорій”», складається з вступу, трьох розділів, висновків, списку використаної літератури та додатків.

Текстова частина викладена на 19 сторінках друкованого тексту.

Пояснювальна записка має 7 сторінок додатків. Список використаних джерел містить 10 найменувань і займає 1 сторінку.

В роботі наведено 16 рисунків. Загальний обсяг роботи 38 сторінок.

Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка». 25.121.10.000 - ПЗ		
Розроб.	Семенов О.П.				Розробка віконного додатку «Калькулятор калорій».		
Перевір.							
Керівник	Чижмотря О.В.						
Н. контр.							
Зав. каф.					Літ.	Арк.	Аркушів
						4	38
					ФІКТ, Гр. ІПЗ-24-1[2]		

Зміст

ВСТУП.....	6
РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ.....	7
1.1.Аналіз задачі, засобів та методів її вирішення.....	7
1.2.Аналіз існуючого програмного забезпечення за тематикою курсового проекту (роботи).....	7
Висновки до першого розділу.....	11
РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	13
2.1 Проектування загального алгоритму роботи програми	13
2.2 Розробка функціональних алгоритмів роботи програми	14
2.3 Розробка програмного забезпечення.....	15
Висновки до другого розділу	19
РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ	20
3.1 Опис роботи з програмним додатком (Опис інтерфейсу)	20
3.2 Тестування роботи програмного забезпечення.....	23
Висновки до третього розділу	26
ВИСНОВКИ.....	27
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	28
ДОДАТКИ.....	29

		Семенов.О.П.		
		Чижмотря О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

ВСТУП

Актуальність теми. У сучасному світі, де здоровий спосіб життя набуває все більшої популярності, контроль за споживанням калорій є важливим аспектом підтримки фізичної форми та запобігання різним захворюванням. Автоматизація цього процесу через цифрові засоби, зокрема калькулятор калорій, дозволяє користувачам ефективно планувати харчування, регулювати раціон та досягати індивідуальних цілей щодо ваги та здоров'я. Впровадження такого інструменту сприяє підвищенню обізнаності про вплив харчування на організм і дозволяє зробити управління калорійністю доступним та зручним для широкого кола користувачів.

Об'єктом дослідження є програмне забезпечення для розрахунку калорійності продуктів та визначення норми калорій для людини.

Предметом дослідження є алгоритми та методи розрахунку калорійності продуктів, їх інтеграція у програмне забезпечення та можливості оптимізації процесу обчислення.

Метою роботи є розробка та оптимізація калькулятора калорій, що дозволяє користувачам швидко та зручно обчислювати калорійність харчових продуктів. Це передбачає створення інтуїтивно зрозумілого інтерфейсу, інтеграцію JSON-файлу з базою харчових продуктів, а також реалізацію методів точного розрахунку калорійності. Дослідження спрямоване на покращення користувацького досвіду та підвищення ефективності контролю раціону.

		Семенов.О.П.			ДУ «Житомирська політехніка».25.121.10.000 – ПЗ	Арк.
		Чижмотря О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

1.1.Аналіз задачі, засобів та методів її вирішення

Коли розробляєш калькулятор калорій, перш за все треба розібратися, що взагалі потрібно людям від такого інструменту. Суть задачі – допомогти користувачам швидко і легко рахувати калорійність продуктів, щоб краще контролювати свій рацион. Для цього треба не просто зробити формули для підрахунку, а й продумати, як люди взаємодіятимуть із програмою.

З одного боку, є готові бази даних, які містять інформацію про калорійність різних продуктів. Але якщо просто створити калькулятор, що бере дані з таблиць, це буде занадто примітивно. Треба, щоб користувач міг сам вводити вагу продукту, комбінувати інгредієнти, змінювати порції та отримувати точний розрахунок.

Тому реалізація включає декілька важливих моментів: потрібно зібрати коректні дані для розрахунку, розробити механізм обчислень, підключити базу з продуктами і їхньою калорійністю і створити зручний інтерфейс. Найскладніше – зробити все логічно і просто для користувача.

Мова програмування C# добре підходить для цього, бо дає змогу чітко структурувати код і реалізувати всі необхідні алгоритми. Якщо розробляти десктопний додаток, можна використати WinForms, головне – зробити так, щоб калькулятор був швидким і зручним у використанні.

Крім того, важливо протестувати програму та оптимізувати її, щоб все працювало без збоїв. В ідеалі вона має бути достатньо гнучкою, щоб користувач міг додавати власні продукти, зберігати результати і легко переглядати історію обчислень.

1.2.Аналіз існуючого програмного забезпечення за тематикою курсового проекту (роботи).

Є купа програм для підрахунку калорій, і всі вони різні за можливостями. Найбільш популярні – це мобільні додатки, бо їх зручно використовувати на

		Семенов.О.П.			ДУ «Житомирська політехніка».25.121.10.000 – ПЗ	Арк.
		Чижмоторя О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

ходу. Веб-сервіси теж є, але вони менш інтерактивні, а десктопні програми зустрічаються рідше, бо люди хочуть щось доступне зі смартфона.

Один із найвідоміших – MyFitnessPal. Його розробила компанія Under Armour, і це свого роду стандарт у цій сфері. Він має величезну базу даних продуктів, дозволяє рахувати калорії, інтегрується з фітнес-трекерами. Але мінус у тому, що платна версія відкриває багато корисних функцій, а дані іноді бувають неточними через внески користувачів.

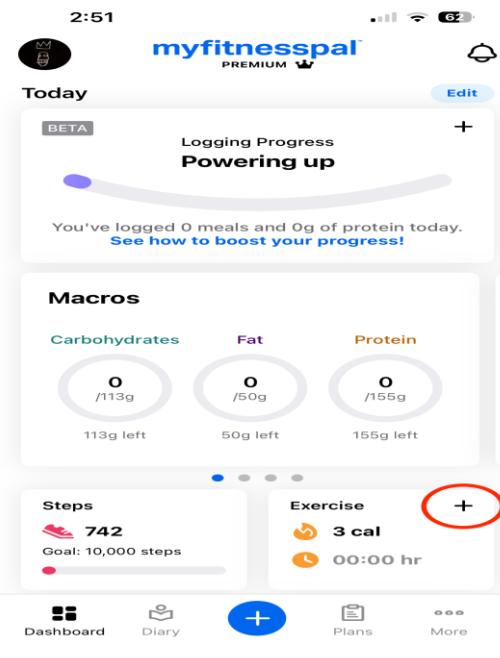


Рис.1.1.Інтерфейс MyFitnessPal

Lose It! теж популярний варіант, від FitNow, Inc. У нього простий інтерфейс, є функція сканування штрих-кодів, що зручно для швидкого введення продуктів. Але його база даних трохи менша, ніж у MyFitnessPal.

		Семенов.О.П.			ДУ «Житомирська політехніка».25.121.10.000 – ПЗ	Арк.
		Чижмоторя О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		

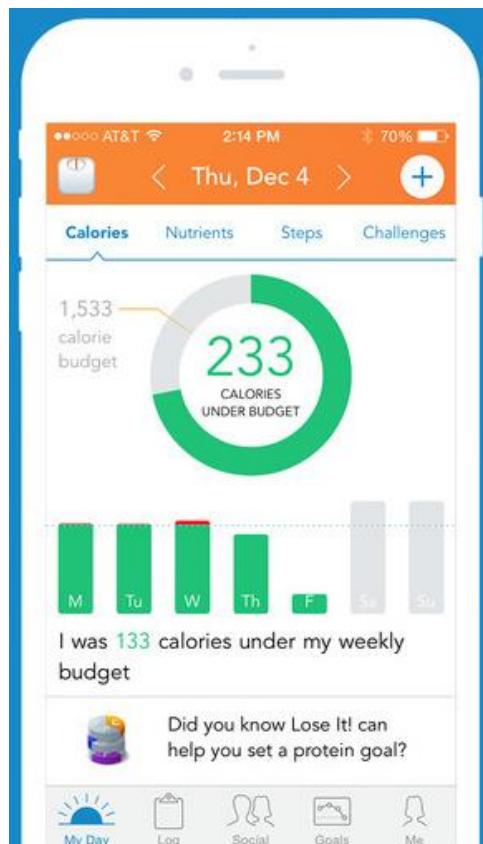


Рис.1.2.Інтерфейс LoseIt!

FatSecret це безкоштовна альтернатива, в якій можна вести щоденник харчування та обговорювати раціон зі спільнотою. Це класно для тих, хто шукає підтримку. Проте база даних продуктів тут не така велика, а аналіз поживних речовин менш детальний.

		Семенов.О.П.		
		Чижмоторя О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

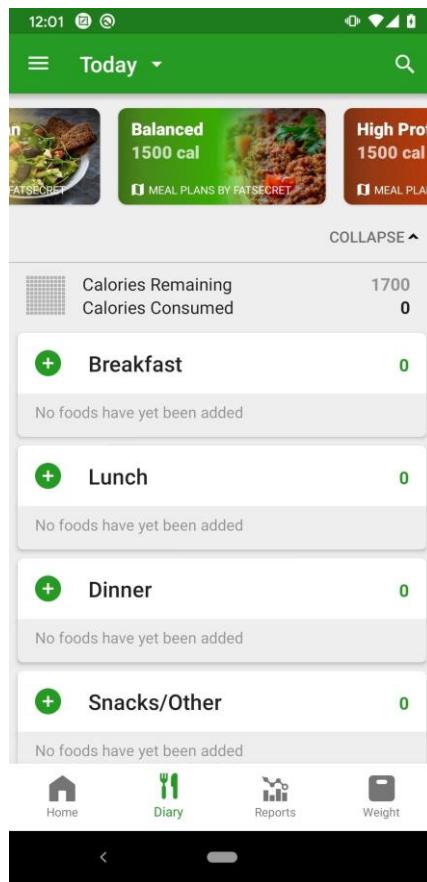


Рис.1.3.Інтерфейс FatSecret

Cronometer більш точний і глибокий у плані аналізу харчування. Його створила компанія Cronometer Inc., і він більше підходить для тих, кому важлива не просто калорійність, а й поживні речовини. Його дані дуже точні, але сам інтерфейс трохи складніший, ніж у конкурентів.

		<i>Семенов.О.П.</i>			ДУ «Житомирська політехніка».25.121.10.000 – ПЗ	<i>Арк.</i>
		<i>Чижмоторя О.В.</i>				
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>10</i>

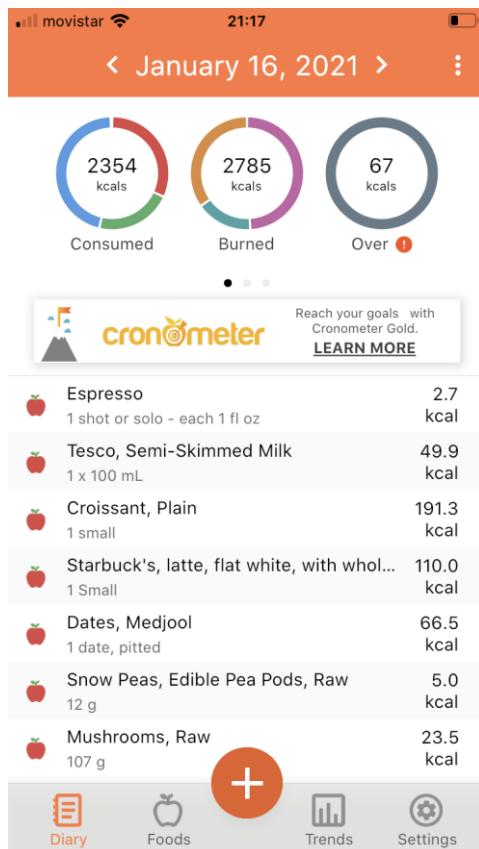


Рис.1.4.Інтерфейс Cronometr

Якщо підсумувати, то найкращим варіантом для нового додатка буде щось просте й зручне, але без залежності від сторонніх баз даних. Було б круто мати калькулятор, де користувачі можуть додавати свої продукти, коригувати порції, отримувати точні розрахунки та не відчувати зайвих обмежень.

Висновки до першого розділу

Якщо підсумувати все це, то виходить, що калькулятор калорій потрібен, щоб людям було простіше слідкувати за своїм харчуванням. Він має бути зручним, швидким і давати точні результати, а не просто витягувати дані з таблиці.

У розробці важливо правильно вибрати технології — C# добре підходить для програмування логіки, а WinForms допоможе зробити інтерфейс зрозумілим. Також треба подбати про базу даних, в якій зберігатимуться продукти та їх калорійність.

Серед вже існуючих додатків є MyFitnessPal, Lose It!, FatSecret, Cronometer. Всі вони мають плюси й мінуси: одні залежать від онлайн-баз, інші

		Семенов.О.П.		
		Чижмоторя О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

обмежують користувачів безкоштовною версією. Тому краще зробити таку програму, яка працюватиме автономно, дозволяючи додавати свої продукти та налаштовувати порції без зайвих складнощів.

		<i>Семенов.О.П.</i>		
		<i>Чижмотря О.В.</i>		
Змн.	Арк.	№ докум.	Підпис	Дата

ДУ «Житомирська політехніка».25.121.10.000 – ПЗ

Арк.

12

РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Проектування загального алгоритму роботи програми

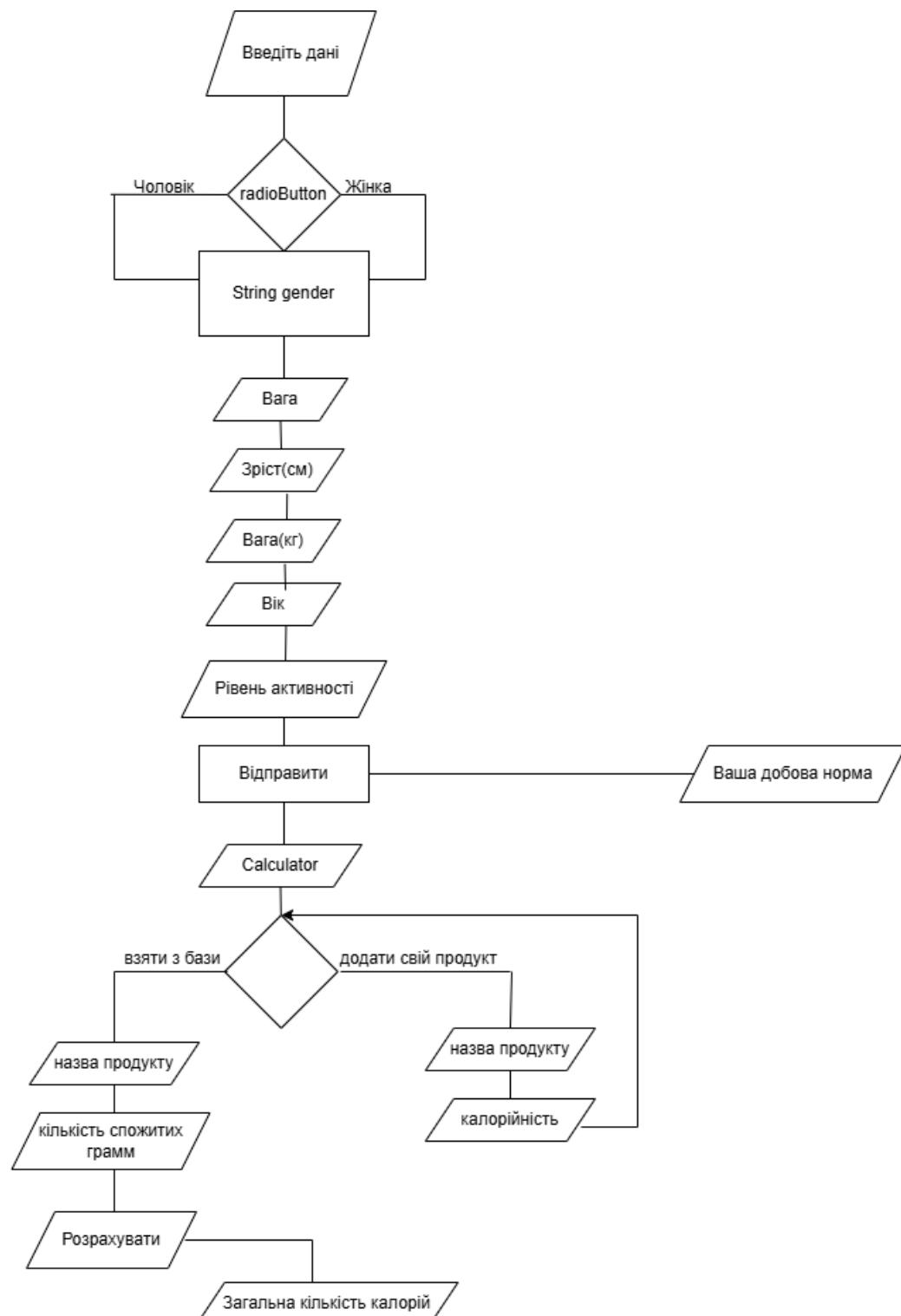


Рис.2.1. Блок-схема

		Семенов.О.П.		
		Чижмоторя О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

Початок програми, відкривається вікно UserDataForm, користувач вибирає стать через radioButton, (результат записується в string). Далі користувач вводить свої параметри (вага, зріст, вік, та рівень активності), далі користувач натискає на кнопку «Відправити» ці дані відправляються на іншу форму Calculator та виводиться денна норма калорій. У вікні Calculator з'являється можливість вибирати продукти з готової бази, яка знаходиться у JSON файлі foodDatabase.json, або додати нові продукти по кліку на текст «Додати свій продукт», після цього коли користувач зібрав своє меню, натиснувши на кнопку «Розрахувати», підраховуються калорії усіх продуктів які були додані у меню.

2.2 Розробка функціональних алгоритмів роботи програми

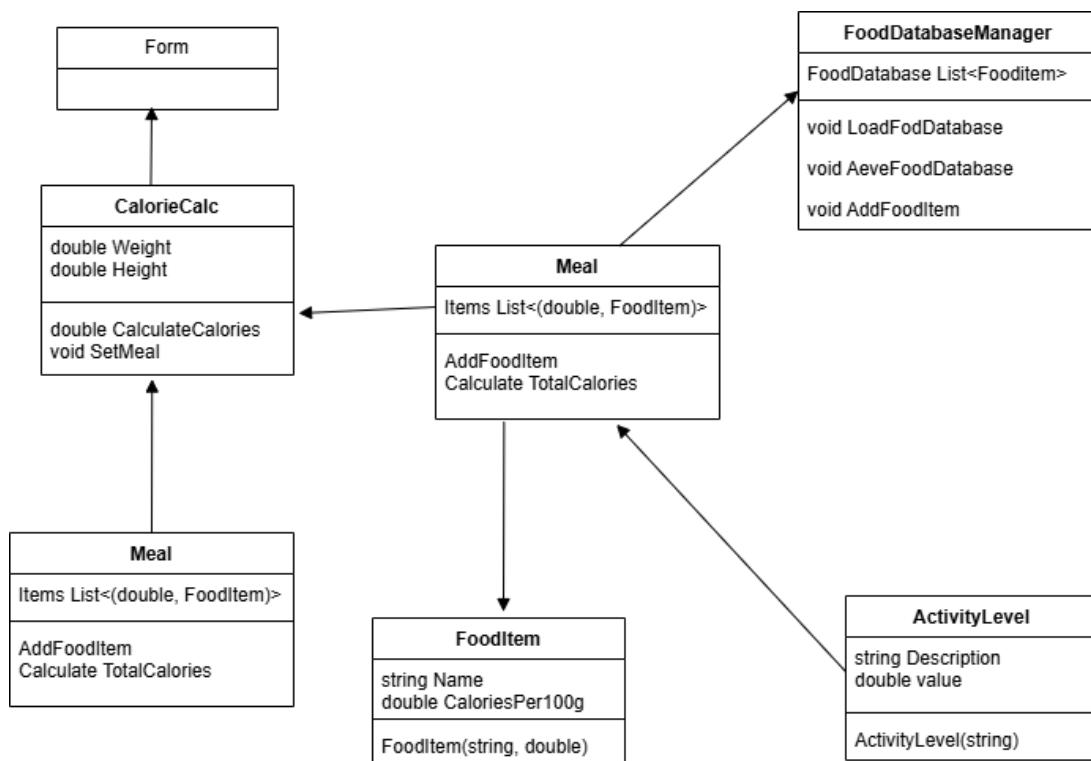


Рис.2.2. Діаграма класів

Form — головне вікно, де користувач вводить свої параметри й натискає кнопку. Вираховується норма калорій, а потім створюється **CalorieCalc**, в якому вже можна додавати продукти.

CalorieCalc отримує норму калорій із **Form**, дозволяє вибирати їжу та додає її в **Meal**.

		Семенов.О.П.		
		Чижмоторя О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

Meal — це, по суті, список продуктів, які користувач додає під час прийому їжі. Воно рахує загальну калорійність.

FoodDatabaseManager — завантажує список продуктів із JSON-файлу. Якщо база порожня, додає стандартні продукти.

FoodItem — описує окремий продукт (його назву та калорійність).

ActivityLevel — просто містить рівні фізичної активності та їх коефіцієнти. Використовується у **Form**, щоб правильно розрахувати потребу в калоріях.

Це все взаємодіє за рахунок того, що користувач вводить дані у **Form**, розраховує свою норму калорій. **CalorieCalc** отримує ці дані й дає можливість вибирати їжу.

Користувач додає продукти, вони йдуть у **Meal**, і калорії підраховуються.

FoodDatabaseManager керує списком їжі, а **FoodItem** представляє кожен продукт.

2.3 Розробка програмного забезпечення

Структура проекту(рис. 2.4.) являє собою 3 шари:

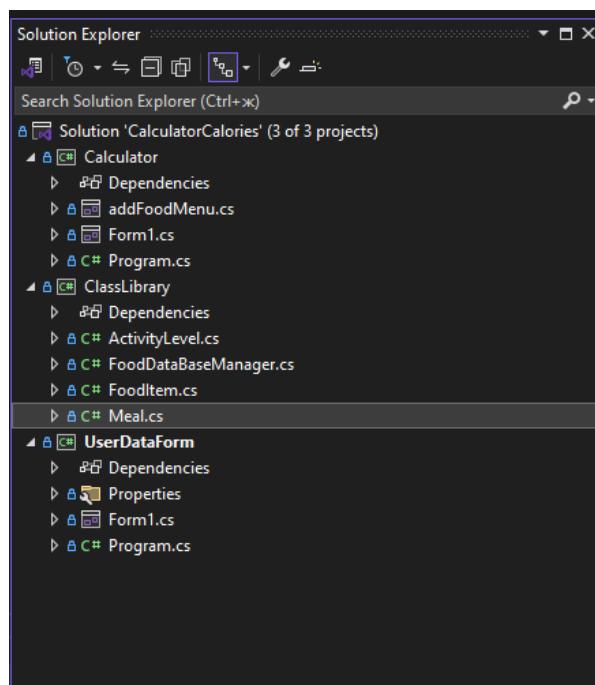


Рис.2.4.Структура проекту

Calculator – де знаходиться вся логіка розрахунку калорій.

Лістинг форми Calculator:

		Семенов.О.П.		
		Чижмоторя О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

public CalorieCalc()
{
    InitializeComponent();
    foodDatabaseManager = new FoodDatabaseManager();
    currentMeal = new Meal();
    InitializeFoodComboBox();
}

public void InitializeFoodComboBox()
{
    FoodComboBox.Items.Clear();
    foreach (var food in foodDatabaseManager.FoodDatabase)
    {
        FoodComboBox.Items.Add(food);
    }

    if (FoodComboBox.Items.Count > 0)
        FoodComboBox.SelectedIndex = 0;
}

private void buttonAdd_Click(object sender, EventArgs e)
{
    if (FoodComboBox.SelectedItem is FoodItem selectedFood &&
        double.TryParse(textBoxWeight.Text, out double weight) && weight > 0)
    {
        currentMeal.AddFoodItem(selectedFood, weight);
        listBoxItems.Items.Add($"{selectedFood.Name} - {weight} г");
        labelTotalCalories.Text = $"Загальна кількість калорій:
{currentMeal.CalculateTotalCalories():F2} ккал";
        textBoxWeight.Clear();
    }
    else
    {
        MessageBox.Show("Будь ласка, введіть коректну вагу продукту!",
        "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
public CalorieCalc(double weight, double height, string gender, int age,
double activityLevel, double dailyCalories, FoodDatabaseManager foodManager)
{
    InitializeComponent();
    foodDatabaseManager = foodManager;
    currentMeal = new Meal();
    this.dailyCalorieNorm = dailyCalories;
    calorieNorm.Text = $"Денна норма: {dailyCalorieNorm:F0} ккал";
    InitializeFoodComboBox();
}

```

Спочатку створюється база даних FoodDatabaseManager, щоб підтягнути список продуктів. Також є клас Meal, який зберігає все, що користувач додає до своєї "тарілки".

Коли людина відкриває форму, InitializeFoodComboBox() заповнює випадаючий список їжею. Якщо користувач вибирає продукт і вводить його вагу, натисканням кнопки buttonAdd_Click() додає його до Meal та оновлює відображення загальних калорій.

		Семенов.О.П.		
		Чижмоторя О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

ClassLibrary у цьому проекті — це набір ключових класів, що відповідають за логіку роботи з продуктами, активністю та підрахунком калорій. Тут є ActivityLevel, який містить рівні фізичної активності й допомагає правильно розрахувати добову норму калорій. Є FoodItem — клас, що просто зберігає назву продукту та його калорійність на 100 грамів, він використовується у багатьох місцях, зокрема в Meal та FoodDatabaseManager. Останній керує списком продуктів: він або завантажує дані з JSON-файлу, або додає нові продукти, якщо користувач хоче розширити базу. Сам Meal потрібен, щоб зберігати прийоми їжі та вираховувати загальну калорійність, тобто він фактично є контейнером для FoodItem зі збереженням ваги кожного доданого продукту. Вся ця система побудована так, щоб відокремити логіку від UI: Form1 працює з ActivityLevel для отримання коефіцієнта активності, CalorieCalc отримує список продуктів від FoodDatabaseManager і додає їх у Meal, а вже там підраховуються калорії. Усе це робить код структурованим і розширюваним: якщо потрібно змінити або доповнити якусь функцію, достатньо працювати з конкретним класом, а не переписувати весь проект.(Кожен клас є в Додатках (ст.25)).

UserDataTable – є головним вікном для введення даних користувача, яке потім використовується для розрахунку добової норми калорій. Вона містить змінні для ваги, зросту, віку, статі та рівня фізичної активності, а також поле для обчисленої кількості калорій. При створенні форми ініціалізуються рівні активності, що дозволяє користувачеві вибрати відповідний коефіцієнт. Коли користувач вводить свої параметри та натискає кнопку, відбувається перевірка введених даних, розрахунок BMR на основі формули Харриса-Бенедикта і врахування рівня активності. Якщо введення коректне, відкривається нове вікно CalorieCalc, передаючи туди всі параметри, які користувач вніс. У випадку помилки, наприклад, некоректного введення ваги або зросту, показується повідомлення з поясненням. Форма також має обробники подій для кнопки закриття, змінюючи її вигляд при наведенні. Код добре організований, ідея розділення логіки розрахунку та графічного інтерфейсу через окремі класи

		Семенов.О.П.			ДУ «Житомирська політехніка».25.121.10.000 – ПЗ	Арк.
		Чижмотря О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		17

робить його зручним для підтримки та розширення. Якщо буде потрібно додати нові параметри або змінити принцип розрахунку, можна легко внести зміни.

Лістинг форми UserDataForm:

```
using ClassLibrary;
using Calculator;
namespace UserDataForm
{
    public partial class Form1 : System.Windows.Forms.Form
    {
        public double Weight { get; private set; }
        public double Height { get; private set; }
        public string Gender { get; private set; }
        public int Age { get; private set; }
        public double ActivityLevel { get; private set; }
        public double dailyCalories { get; private set; }
        public FoodDatabaseManager foodManager { get; private set; }
        public Form1()
        {
            InitializeComponent();
            InitializeActivityLevels();
        }

        private void InitializeActivityLevels()
        {
            cmbActivityLevel.Items.Add(new ActivityLevel("Сидячий спосіб життя",
1.2));
            cmbActivityLevel.Items.Add(new ActivityLevel("Легка активність (1-3
тренування/тиждень)", 1.375));
            cmbActivityLevel.Items.Add(new ActivityLevel("Помірна активність (3-5
тренувань/тиждень)", 1.55));
            cmbActivityLevel.Items.Add(new ActivityLevel("Висока активність (6-7
тренувань/тиждень)", 1.725));
            cmbActivityLevel.Items.Add(new ActivityLevel("Дуже висока активність
(важка фізична робота)", 1.9));
            cmbActivityLevel.SelectedIndex = 0;
        }

        private void buttonSend_Click_1(object sender, EventArgs e)
        {
            try
            {
                string gender = radioButtonMale.Checked ? "Чоловік" : "Жінка";
                double weight = double.Parse(textBoxWeight.Text);
                double height = double.Parse(textBoxHeight.Text);
                int age = int.Parse(txtAge.Text);
                double activityLevel =
((ActivityLevel)cmbActivityLevel.SelectedItem).Value;

                double dailyCalories = CalculateDailyCalories(gender, weight, height,
age, activityLevel);
                this.Hide();
                FoodDatabaseManager foodManager = new FoodDatabaseManager();
                CalorieCalc calorieCalculator = new CalorieCalc(Weight, Height,
Gender, Age, ActivityLevel, dailyCalories, foodManager);
                calorieCalculator.Show();
            }
            catch (Exception ex)
            {

```

Змн.	Арк.	Семенов.О.П.		
		Чижмоторя О.В.		

```

        MessageBox.Show($"Помилка: {ex.Message}", "Помилка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }

}

public double CalculateDailyCalories(string gender, double weight, double
height, int age, double activityLevel)
{
    double bmr;

    if (gender == "Чоловік")
    {
        bmr = 10 * weight + 6.25 * height - 5 * age + 5;
    }
    else if (gender == "Жінка")
    {
        bmr = 10 * weight + 6.25 * height - 5 * age - 161;
    }
    else
    {
        throw new ArgumentException("Невірно вказана стать");
    }

    return bmr * activityLevel;
}

```

Висновки до другого розділу

Програма калькулятора калорій має працювати просто і зручно: користувач вводить свої параметри, отримує розрахунок добової норми калорій, додає продукти і бачить загальну калорійність. Важливі алгоритми – правильний підрахунок BMR, оновлення загальної суми калорій та збереження списку продуктів.

У коді логіка розподілена між FoodDatabaseManager, Meal, Form1 і CalorieCalc, що забезпечує чітку структуру та можливість розширення. Головний акцент – зробити калькулятор автономним, щоб користувач міг гнучко керувати даними без обмежень.

Змн.	Арк.	Семенов.О.П.		
		Чижмотря О.В.		

РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ

3.1 Опис роботи з програмним додатком (Опис інтерфейсу)

Інтерфейс програмного додатку для розрахунку калорій розроблений таким чином, щоб користувачеві було максимально зручно вводити необхідні дані та отримувати результати. Основне вікно містить кілька основних полів для введення: вага, зріст, вік, стать і рівень фізичної активності. Для зручності рівень активності вибирається зі списку, що вже містить описані категорії – від сидячого способу життя до високої фізичної активності. Коли користувач заповнює всі ці параметри, він натискає кнопку для розрахунку добової норми калорій, і програма обчислює необхідну кількість калорій, після чого відкривається нове вікно.

У другому вікні калькулятора користувач працює з продуктами харчування. У верхній частині розташований випадаючий список, у якому містяться всі продукти з бази даних. Користувач вибирає один із них, вводить вагу порції і додає її до списку спожитих продуктів. Програма миттєво оновлює загальну кількість калорій, що відображається у відповідному полі. Okрім вибору готових продуктів, у користувача є можливість додавати нові позиції в базу. Натискання спеціального посилання відкриває форму для введення нового продукту з вказанням його калорійності на 100 грамів.

Зручність використання також забезпечується можливістю редагування списку продуктів: якщо користувач вводить некоректне значення, йому відається повідомлення про помилку. Інтерактивні елементи, такі як кнопки закриття, змінюють свій вигляд при наведенні, що робить взаємодію з програмою більш зрозумілою. У підсумку весь інтерфейс створений так, щоб користувач міг максимально швидко ввести свої дані, отримати розрахунки та керувати власним раціоном без зайвих складнощів. Якщо потрібно внести якісь зміни або розширити функціонал, завжди можна вдосконалити систему, додавши нові можливості збереження або аналізу харчування.

		Семенов.О.П.			ДУ «Житомирська політехніка».25.121.10.000 – ПЗ	Арк.
		Чижмотря О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		20

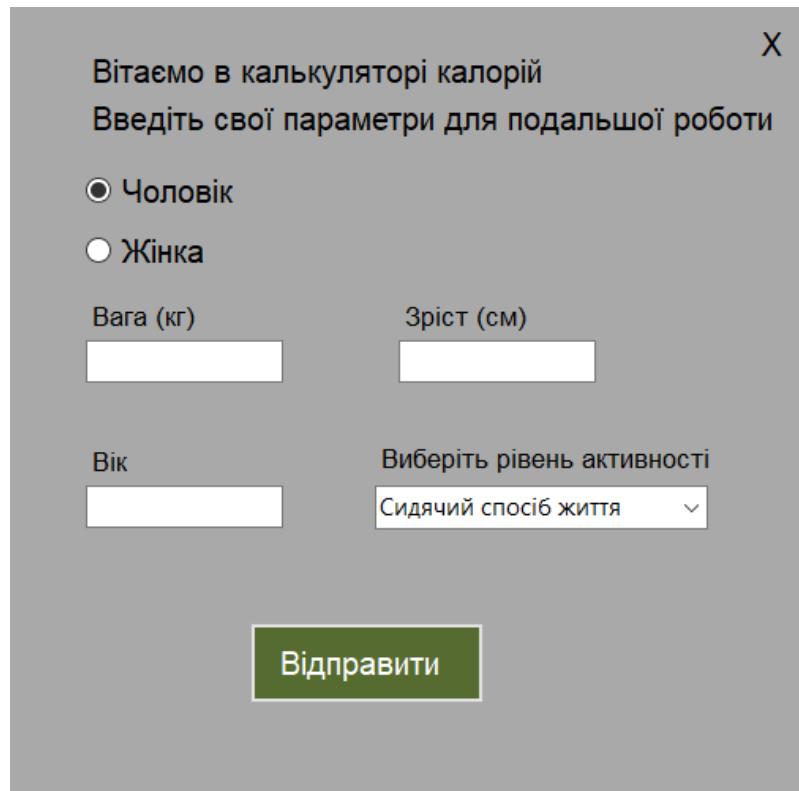


Рис.3.1.Головне меню програми, після натискання на кнопку «Відправити», відкривається вікно «Calculator» (рис.3.2.).

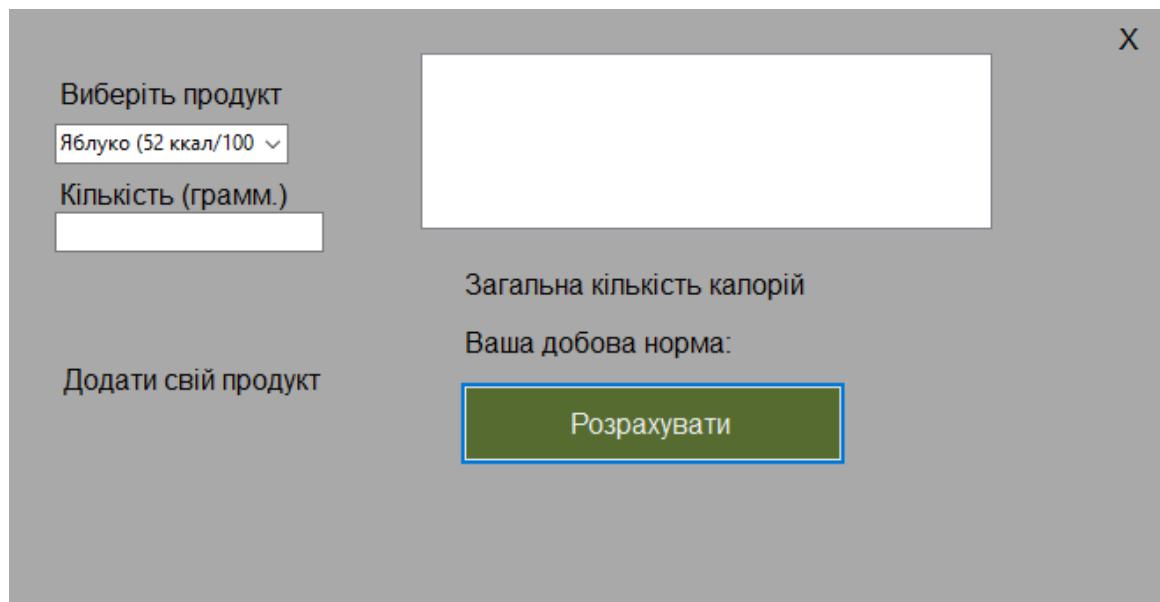


Рис.3.2. Меню «Calculator». Тут можна вибрати продукт з готового випадаючого списку, та ввести кількість продукту в граммах(рис.3.3.), або ввести продукт самостійно, та додати його до бази продуктів(рис.3.4.), також тут показується загальна кількість підсумованих калорій, та денна норма, з

		Семенов.О.П.		
		Чижмотря О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

даними які були розраховані з введених в UserDataForm(головному меню) даних.

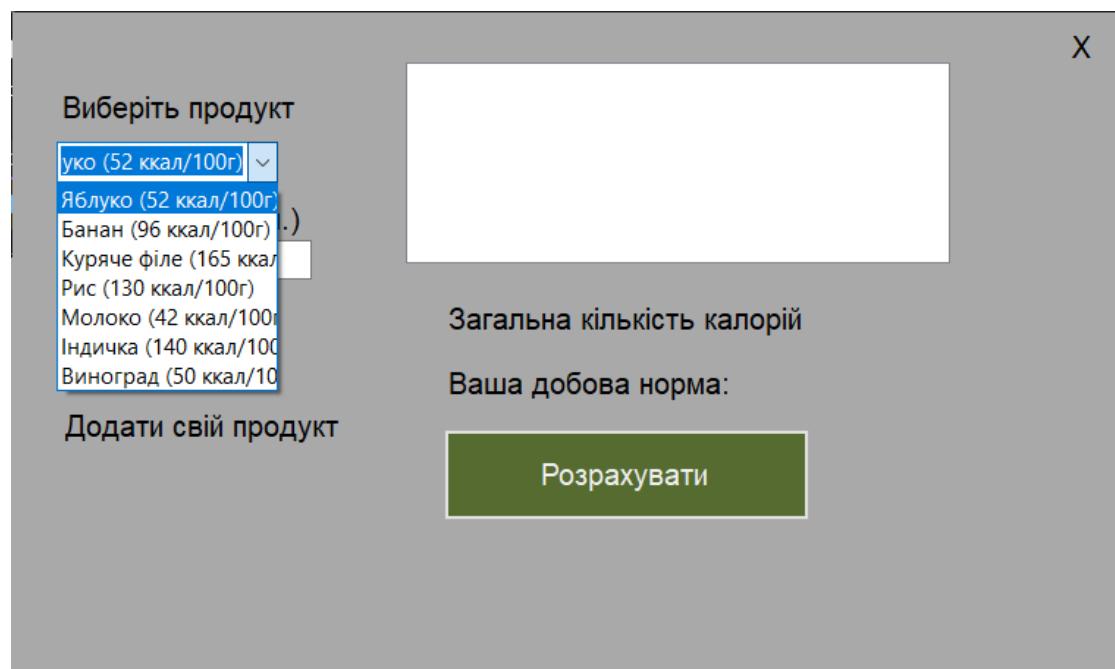


Рис.3.3. Продукти з випадаючого списку. Це стандартний набір продуктів, які зберігаються у JSON-файлі.

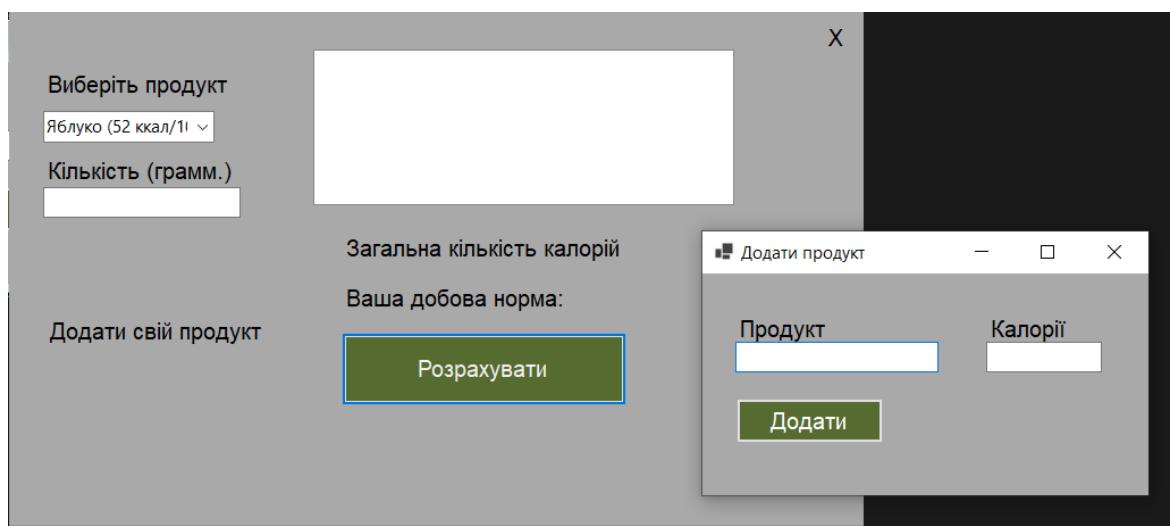


Рис.3.4. Додавання свого продукту. Через клік на текст «Додати свій продукт», відкривається нова форма з додаванням продукту(вводиться назва та калорійність)

		Семенов.О.П.		
		Чижомотря О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

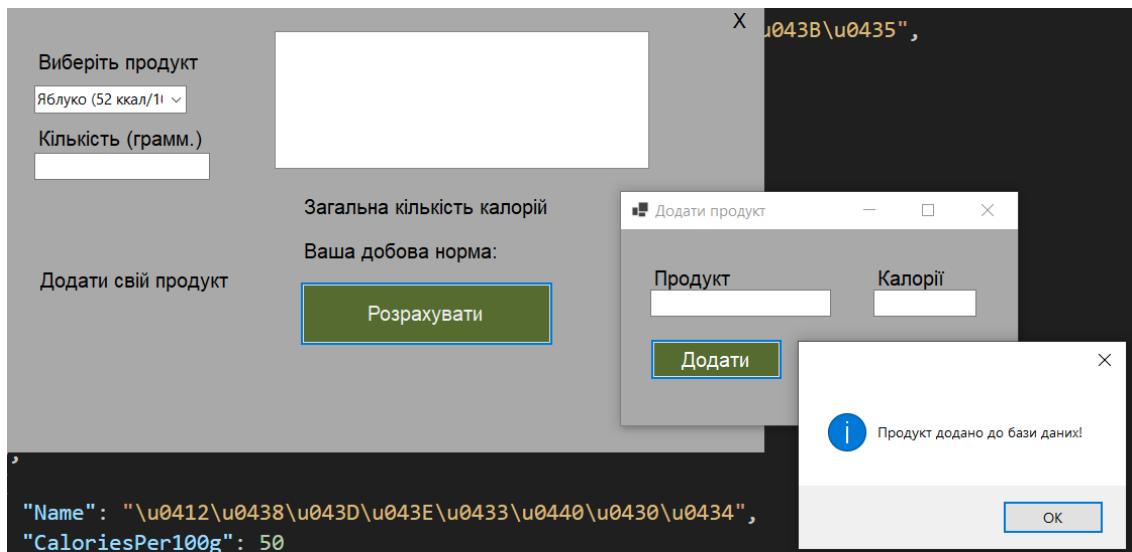


Рис.3.5. Демонстрація того, що продукт додано успішно у JSON-файл. Та відображається у comboBox.

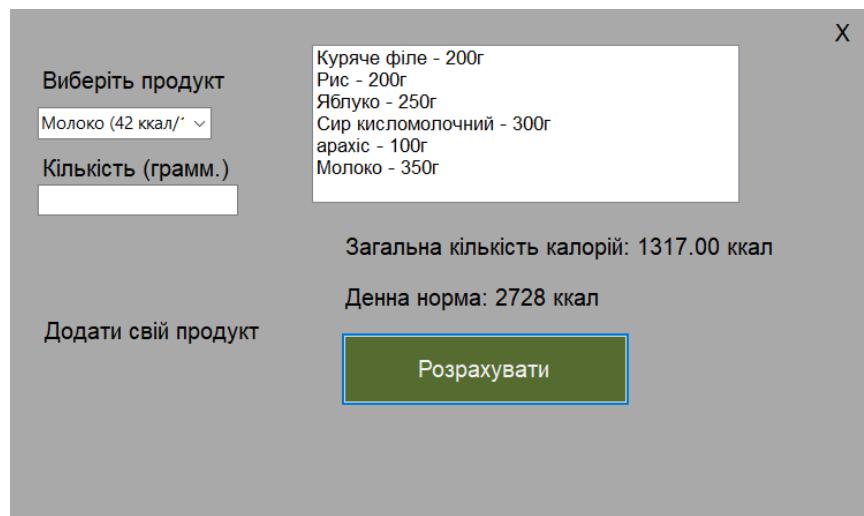


Рис.3.6. Демонстрація того, що калорії рахуються коректно. Та продукти додаються у listBox корректно.

3.2 Тестування роботи програмного забезпечення

Коли я почав працювати над калькулятором калорій, спочатку визначив основні параметри, які користувач має вводити: вагу, зріст, вік, стать і рівень активності. Це було ключовим для розрахунку добової потреби в калоріях, тому я створив форму, де користувач міг зручно вказати всі ці значення. Було важливо зробити так, щоб рівні активності заповнювалися автоматично, тому я ініціалізував список ActivityLevel, аби можна було вибрати відповідний коефіцієнт.

		Семенов.О.П.		
		Чижмоторя О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

Коли користувач натискає кнопку, програма перевіряє введені значення, щоб вони були коректними. Потім запускається розрахунок BMR, використовуючи формулу Харріса-Бенедикта, і множиться отримане значення на коефіцієнт активності. Якщо дані введені правильно, форма приховується, а я створюю CalorieCalc, у яку передаю всі параметри. Там уже користувач може додати продукти і контролювати скільки калорій він споживає.

Окремий момент, над яким я працював, — це база даних продуктів. Я реалізував FoodDatabaseManager, щоб програма могла завантажувати список їжі з JSON-файлу. Це дозволяє додавати нові продукти, які будуть збережені та доступні при наступному запуску. В CalorieCalc я зробив так, щоб користувач міг вибрати продукт, вказати його вагу і додати в прийом їжі, який зберігається в Meal. Тут уже підраховується загальна калорійність, і користувач може контролювати свій раціон.

На етапі тестування я перевіряв, як програма реагує на некоректні дані. Виправляв помилки введення, додавав перевірки на коректність чисел, тестував, як зберігаються продукти в базі. Було важливо зробити так, щоб все працювало плавно: введення даних розрахунок калорій вибір продуктів підрахунок загальної калорійності. У процесі я вдосконалив механізми обробки помилок, перевірив, як змінюється інтерфейс залежно від дій користувача, і оптимізував взаємодію між класами.

Зрештою, калькулятор працює так, як я задумав: людина вводить дані, отримує норму калорій, додає продукти та бачить, скільки калорій вона споживає. Якщо потрібно буде допрацювати систему, можна легко розширити її, наприклад, додавши збереження історії прийомів їжі або створивши більш детальні звіти.

		Семенов.О.П.		
		Чижмотря О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

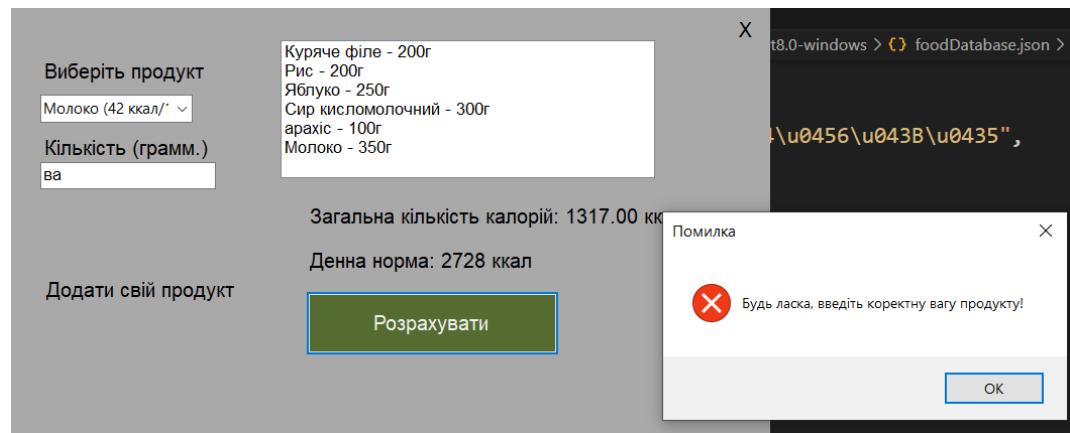


Рис.3.7. В цей textBox можна вводити лише числові значення.

Виводиться помилка, «Введіть вагу коректно».

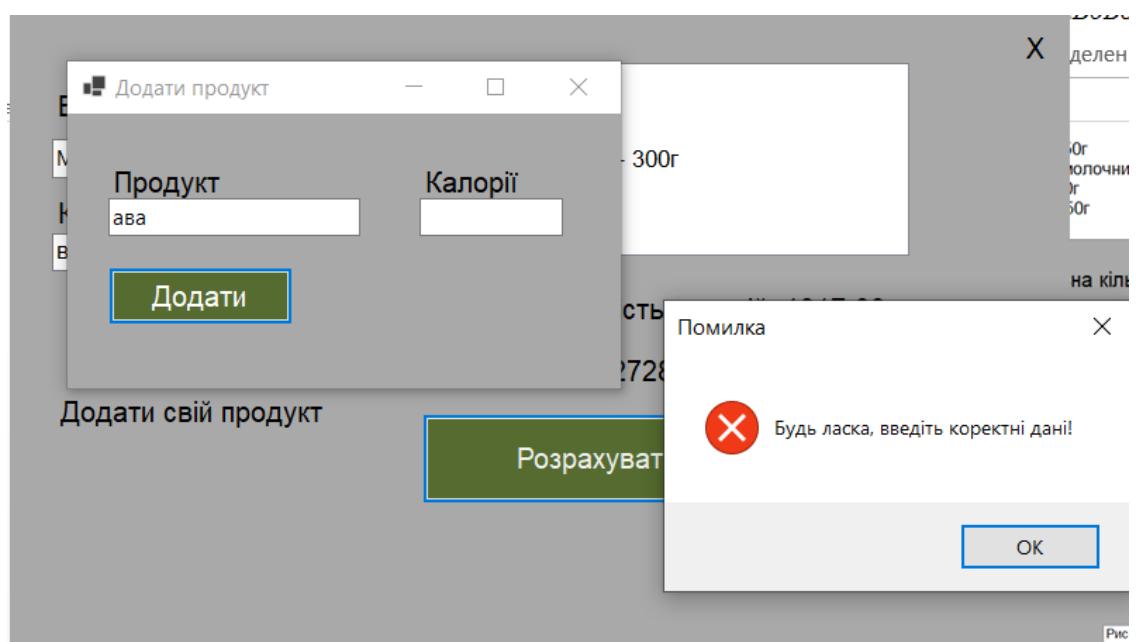


Рис.3.8. В цей textBox можна вводити лише літери. Виводиться помилка

«Введіть дані коректно».

		Семенов.О.П.			ДУ «Житомирська політехніка».25.121.10.000 – ПЗ	Арк.
		Чижмоторя О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		

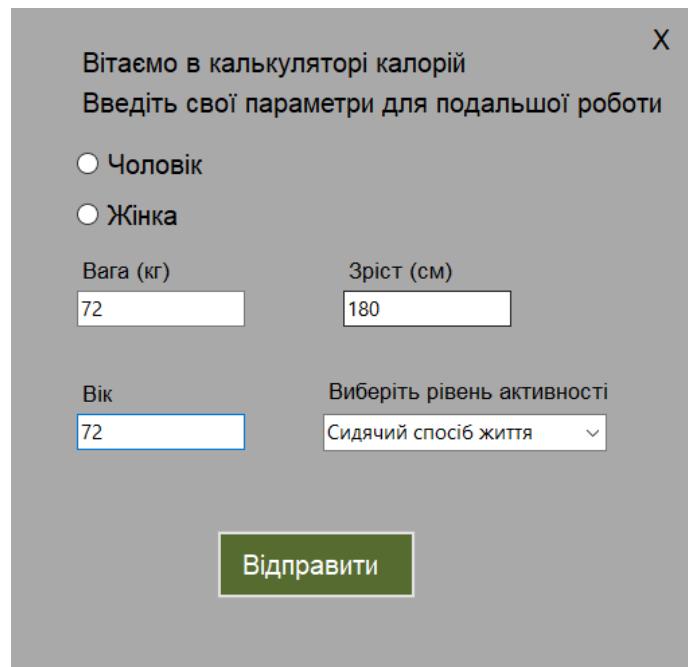


Рис.3.9. В головному меню, якщо не вибрати стать, не розрахуються калорії і приходиться перезапускати програму.

Висновки до третього розділу

Програма має простий інтерфейс для введення параметрів і додавання продуктів, а тестування підтвердило її стабільну роботу. Все працює швидко та інтуїтивно, але є можливості для покращення зручності.

		Семенов.О.П.		
		Чижмоторя О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

ВИСНОВКИ

Розробка калькулятора калорій на WinForms у C# була цікавою і водночас вимагала уваги до деталей. Важливо було правильно організувати структуру проекту, щоб все працювало злагоджено. Наприклад, система збереження продуктів і калорій була окремим класом, що дозволило гнучко керувати базою продуктів і не перевантажувати основний код.

Оформлення інтерфейсу виявилося важливим моментом. Зручне розміщення елементів забезпечило гарний вигляд на будь-якому розмірі екрана.

Автоматичні розрахунки калорій та можливість збереження даних додали функціональності додатку. Використання JSON для роботи з даними допомогло зробити збереження швидким та надійним.

Ну і, звичайно, не обійшлося без рефакторингу — усунення дублювань у коді, використання повторюваних методів значно спростили роботу над додатком. Загалом, це вийшов зручний, стабільний і перспективний калькулятор калорій, який можна доповнювати новими функціями.

		Семенов.О.П.			ДЧ «Житомирська політехніка».25.121.10.000 – ПЗ	Арк.
		Чижмоторя О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Об'єктно-орієнтоване програмування [електронний ресурс] / Освітній портал ДУ “Житомирська політехніка” – 2020. – Режим доступу до ресурсу: <https://learn.ztu.edu.ua/course/view.php?id=1628>
2. Data Structures and Algorithms Using C# / Michael McMillan. - 2007, 60p.
3. Офіційний сайт Microsoft. C# Guide [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.microsoft.com/en-us/dotnet/csharp/>
4. Troelsen A. Pro C# 10 with .NET 6 [Electronic resource] / Andrew Troelsen, Phil Japikse. – Berkeley, CA : Apress, 2022. – Mode of access: <https://doi.org/10.1007/978-1-4842-7869-7>.
5. Richter J. CLR via C#, second edition (pro developer) / Jeffrey Richter. – 2nd ed. – [S. 1.] : Microsoft Press, 2006. – 736 p.
6. The Pragmatic Programmer / David Thomas / Andrew Hunt [Longman] – 2019, 343p.
7. Design Patterns: Elements of Reusable Object-Oriented Software / Erich Gamma / Richard Helm / Ralph Johnson / John Vlissides [Addison-Wesley Professional] – 1995, 396p.
8. Патерни проєктування / Head First / Ерік Фрімен / Елізабет Робсон, [Фабула] – 2004, 634p.
9. The Object-Oriented Thought Process / Matt Weisfeld [Longman] – 2018. 368p.
10. Head First Object-Oriented Analysis and Design: A Brain Friendly Guide to OOA&D / Brett McLaughlin / Gary Pollice/ David West [O'Reilly Media] – 2007, 634p.

Змн.	Арк.	Семенов.О.П.		
		Чижмотря О.В.		

ДУ «Житомирська політехніка».25.121.10.000 – 73

Арк.

28

ДОДАТКИ

		<i>Семенов.О.П.</i>			ДЧ «Житомирська політехніка».25.121.10.000 – ПЗ	Арк.
		<i>Чижмоторя О.В.</i>				
Змн.	Арк.	№ докум.	Підпис	Дата		

Вихідний код програми:

UserDataTable:

```

using ClassLibrary;
using Calculator;
namespace UserDataForm
{
    public partial class Form1 : System.Windows.Forms.Form
    {
        public double Weight { get; private set; }
        public double Height { get; private set; }
        public string Gender { get; private set; }
        public int Age { get; private set; }
        public double ActivityLevel { get; private set; }
        public double dailyCalories { get; private set; }
        public FoodDatabaseManager foodManager { get; private set; }
        public Form1()
        {
            InitializeComponent();
            InitializeActivityLevels();
        }

        private void InitializeActivityLevels()
        {
            cmbActivityLevel.Items.Add(new ActivityLevel("Сидячий спосіб життя", 1.2));
            cmbActivityLevel.Items.Add(new ActivityLevel("Легка активність (1-3
тренування/тиждень)", 1.375));
            cmbActivityLevel.Items.Add(new ActivityLevel("Помірна активність (3-5
тренування/тиждень)", 1.55));
            cmbActivityLevel.Items.Add(new ActivityLevel("Висока активність (6-7
тренування/тиждень)", 1.725));
            cmbActivityLevel.Items.Add(new ActivityLevel("Дуже висока активність (важка фізична
робота)", 1.9));
            cmbActivityLevel.SelectedIndex = 0;
        }

        private void buttonSend_Click_1(object sender, EventArgs e)
        {
    
```

		Семенов.О.П.		
		Чижмоторя О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

try
{
    string gender = radioButtonMale.Checked ? "Чоловік" : "Жінка";
    double weight = double.Parse(textBoxWeight.Text);
    double height = double.Parse(textBoxHeight.Text);
    int age = int.Parse(txtAge.Text);
    double activityLevel = ((ActivityLevel)cmbActivityLevel.SelectedItem).Value;

    double dailyCalories = CalculateDailyCalories(gender, weight, height, age,
activityLevel);
    this.Hide();
    FoodDatabaseManager foodManager = new FoodDatabaseManager();
    CalorieCalc calorieCalculator = new CalorieCalc(Weight, Height, Gender, Age,
ActivityLevel, dailyCalories, foodManager);
    calorieCalculator.Show();
}

catch (Exception ex)
{
    MessageBox.Show($"Помилка: {ex.Message}", "Помилка", MessageBoxButtons.OK,
MessageBoxIcon.Error);
}

}

public double CalculateDailyCalories(string gender, double weight, double height, int
age, double activityLevel)
{
    double bmr;

    if (gender == "Чоловік")
    {
        bmr = 10 * weight + 6.25 * height - 5 * age + 5;
    }
    else if (gender == "Жінка")
    {
        bmr = 10 * weight + 6.25 * height - 5 * age - 161;
    }
    else
    {
        throw new ArgumentException("Невірно вказана стать");
    }
}

```

		Семенов.О.П.		
		Чижмоторя О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

    return bmr * activityLevel;
}

private void CloseButton_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void CloseButton_MouseEnter(object sender, EventArgs e)
{
    CloseButton.ForeColor = Color.White;
}

private void CloseButton_MouseLeave(object sender, EventArgs e)
{
    CloseButton.ForeColor = Color.Black;
}
}
}
}

```

Calculator:

```

using System;
using System.Text.Json;
using System.IO;
using ClassLibrary;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;
namespace Calculator
{
    public partial class CalorieCalc : Form
    {
        private FoodDatabaseManager foodDatabaseManager;
        private Meal currentMeal;
        private double dailyCalorieNorm;

        public CalorieCalc()
        {
            InitializeComponent();
            foodDatabaseManager = new FoodDatabaseManager();
            currentMeal = new Meal();
            InitializeFoodComboBox();
        }
    }
}

```

		Семенов.О.П.			ДУ «Житомирська політехніка».25.121.10.000 – 73	Арк.
		Чижмоторя О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

```

private void InitializeFoodComboBox()
{
    FoodComboBox.Items.Clear();
    foreach (var food in foodDatabaseManager.FoodDatabase)
    {
        FoodComboBox.Items.Add(food);
    }

    if (FoodComboBox.Items.Count > 0)
        FoodComboBox.SelectedIndex = 0;
}

private void buttonAdd_Click(object sender, EventArgs e)
{
    if (FoodComboBox.SelectedItem is FoodItem selectedFood &&
        double.TryParse(textBoxWeight.Text, out double weight) && weight > 0)
    {
        currentMeal.AddFoodItem(selectedFood, weight);
        listBoxItems.Items.Add($"{selectedFood.Name} - {weight} г");
        labelTotalCalories.Text = $"Загальна кількість калорій:
{currentMeal.CalculateTotalCalories():F2} ккал";
        textBoxWeight.Clear();
    }
    else
    {
        MessageBox.Show("Будь ласка, введіть коректну вагу продукту!", "Помилка",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void buttonAddFood_Click(object sender, EventArgs e)
{
    try
    {
        string name = textBoxNewFoodName.Text.Trim();
        if (double.TryParse(textBoxNewFoodCalories.Text, out double calories))
        {
            foodDatabaseManager.AddFoodItem(new FoodItem(name, calories));
            InitializeFoodComboBox();
        }
    }
}

```

		Семенов.О.П.		
		Чижмоторя О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

        textBoxNewFoodName.Clear();
        textBoxNewFoodCalories.Clear();
        MessageBox.Show("Продукт додано до бази даних!", "", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
}
else
{
    MessageBox.Show("Будь ласка, введіть коректні дані!", "Помилка", MessageBoxButtons.OK,
    MessageBoxIcon.Error);
}
}

catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

public CalorieCalc(double weight, double height, string gender, int age, double
activityLevel, double dailyCalories, FoodDatabaseManager foodManager)
{
    InitializeComponent();
    foodDatabaseManager = foodManager;
    currentMeal = new Meal();
    this.dailyCalorieNorm = dailyCalories;
    calorieNorm.Text = $"Денна норма: {dailyCalorieNorm:F0} ккал";
}

private void CloseButton_MouseEnter(object sender, EventArgs e)
{
    CloseButton.ForeColor = Color.White;
}

private void CloseButton_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void CloseButton_MouseLeave(object sender, EventArgs e)
{
    CloseButton.ForeColor = Color.Black;
}

```

		Семенов.О.П.		
		Чижмоторя О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```
}
```

```
}
```

```
}
```

ClassLibrary:

ActivityLevel:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ClassLibrary
{

public class ActivityLevel
{
    public string Description { get; }
    public double Value { get; }

    public ActivityLevel(string description, double value)
    {
        Description = description;
        Value = value;
    }

    public override string ToString()
    {
        return Description;
    }
}

}
```

FoodDatabaseManager:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.Json;
using System.Threading.Tasks;

namespace ClassLibrary
```

		Семенов.О.П.		
		Чижмоторя О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

{
public class FoodDatabaseManager
{
private readonly string jsonFilePath = "FoodDataBase.json";
public List<FoodItem> FoodDatabase { get; private set; }

public FoodDatabaseManager()
{
LoadFoodDatabase();
}

private void LoadFoodDatabase()
{
try
{
if (File.Exists(jsonFilePath))
{
string jsonString = File.ReadAllText(jsonFilePath);
FoodDatabase = JsonSerializer.Deserialize<List<FoodItem>>(jsonString) ?? new
List<FoodItem>();
}
else
{
FoodDatabase = new List<FoodItem>
{
new FoodItem("Яблуко", 52),
new FoodItem("Банан", 96),
new FoodItem("Куряче філе", 165),
new FoodItem("Рис", 130),
new FoodItem("Молоко", 42)
};
}

SaveFoodDatabase();
}
}

catch (Exception ex)
{
throw new Exception($"Помилка завантаження бази продуктів: {ex.Message}");
}
}
}

```

		Семенов.О.П.		
		Чижмоторя О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

public void SaveFoodDatabase()
{
try
{
var options = new JsonSerializerOptions { WriteIndented = true };
string jsonString = JsonSerializer.Serialize(FoodDatabase, options);
File.WriteAllText(jsonPath, jsonString);
}
catch (Exception ex)
{
throw new Exception($"Помилка при збереженні бази даних: {ex.Message}");
}
}

```

```

public void AddFoodItem(FoodItem food)
{
if (FoodDatabase.Any(f => f.Name.Equals(food.Name,
 StringComparison.OrdinalIgnoreCase)))
throw new Exception("Цей продукт вже є в базі даних!");

FoodDatabase.Add(food);
SaveFoodDatabase();
}
}
}

```

FoodItem:

```

namespace ClassLibrary
{
public class FoodItem
{
public string Name { get; set; }
public double CaloriesPer100g { get; set; }
public FoodItem(string name, double caloriesPer100g)
{
Name = name;
CaloriesPer100g = caloriesPer100g;
}

public override string ToString()

```

		Семенов.О.П.		
		Чижмоторя О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

{
    return $"{Name} ({CaloriesPer100g} ккал/100г)";
}
}
}
}

```

Meal:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ClassLibrary
{
    public class Meal
    {
        public List<(FoodItem, double)> Items { get; private set; }

        public Meal()
        {
            Items = new List<(FoodItem, double)>();
        }

        public void AddFoodItem(FoodItem item, double weightInGrams)
        {
            Items.Add((item, weightInGrams));
        }

        public double CalculateTotalCalories()
        {
            double total = 0; foreach (var (food, weight) in Items){
                total += (food.CaloriesPer100g * weight) / 100;
            }
            return total;
        }
    }
}

```

		Семенов.О.П.		
		Чижмоторя О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата