

Министерство образования Республики Беларусь  
Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей  
Кафедра электронных вычислительных машин  
Дисциплина: Программирование на языках высокого уровня

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

МЕНЕДЖЕР ДИЕТЫ

БГУИР КП 1-40 02 01 407 ПЗ

Студент

Группы 150504

Д. А. Гринкевич

Руководитель

А. В. Марзалюк

Минск 2022

Министерство образования Республики Беларусь  
Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ  
Заведующий кафедрой ЭВМ

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
2022 г.

ЗАДАНИЕ  
по курсовому проектированию

Студенту Гринкевичу Денису Андреевичу

1. Тема проекта Менеджер диеты
2. Срок сдачи студентом законченного проекта 15 декабря 2022 г.
3. Содержание расчетно-пояснительной записки (список вопросов, которые подлежат развитию)  
Введение. 1. Обзор литературы. 2. Структурное проектирование.  
3. Функциональное проектирование. 4. Инструкция пользователя. Заключение.  
Приложение А. Приложение Б. Приложение В.  
Приложение Г. Приложение Д.
5. Список графического материала:  
1. Скриншоты работы программы  
2. UML-диаграмма классов.  
3. Схема основных алгоритмов программы.
6. Консультант по проекту А. В. Марзалюк
7. Дата выдачи задания 16 сентября 2022 г.
8. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и трудоемкости отдельных этапов):  
раздел 1 до 15 10 – 20 %;

*разделы 2, 3 до 15 11 – 40 %;*

*раздел 4 до 30 11 – 20 %;*

*оформление пояснительной записки и графического материала до 15 12 – 20 %*

*Защита курсового проекта с 19 по 30 декабря*

РУКОВОДИТЕЛЬ А. В. Марзалюк

(подпись)

Задание принял к исполнению Д. А. Гринкевич

(дата и подпись студента)

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	6
1 ОБЗОР ЛИТЕРАТУРЫ .....	8
1.1 Реализация с использованием формулы Харриса-Бенедикта.....	8
1.2 Реализация с использованием формулы Миффлина - Сан Жеора.....	9
1.3 Реализация с использованием формулы Кэтча - МакАрдла .....	9
1.4 Конечный расчёт TEE.....	9
1.5 Обоснование выбранных методов и алгоритмов.....	10
2 СТРУКТУРНОЕ ПРОЕКТИРОВАНИЕ .....	11
2.1 MyForm.cpp.....	11
2.2 MyForm.h.....	11
2.3 Authorization.h .....	11
2.4 Registration.h .....	11
2.5 UserInformation.h .....	12
2.6 UserInformationFunctions.h .....	12
2.7 GlobalFunctions.h .....	12
2.8 settings.h.....	12
2.9 container.h.....	12
2.10 ActionsClass.h.....	<b>Ошибка! Закладка не определена.</b>
2.11 MainWindow.h .....	<b>Ошибка! Закладка не определена.</b>
2.12 product.h.....	<b>Ошибка! Закладка не определена.</b>
2.13 productFunctions.h.....	14
2.14 usersfood.h .....	14
2.15 usersfoodFunctions.h .....	14
3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ .....	15
3.1 Диаграмма прецедентов .....	15
3.2 Метод проверки данных bool AddAction::infoChek().....	15
3.3 Метод добавления записи void AddAction::AddRecord().....	17
3.4 Метод вычисления калорий void BaseAction::sexSelectionHandler()	
	<b>Ошибка! Закладка не определена.</b>

4 ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЯ .....	21
ЗАКЛЮЧЕНИЕ .....	23
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ .....	24
ПРИЛОЖЕНИЕ А (обязательное) Листинг кода программного средства .....	25
ПРИЛОЖЕНИЕ Б (обязательное) Скриншоты работы программы .....	79
ПРИЛОЖЕНИЕ В (обязательное) UML-диаграмма классов программного средства .....	84
ПРИЛОЖЕНИЕ Г (обязательное) Блок-схемы алгоритмов функций MainWindow_Load(), fromStringToInt(), insertCalories() .....	86
ПРИЛОЖЕНИЕ Д (обязательное) Ведомость документов .....	89

## ВВЕДЕНИЕ

По всему миру люди пытаются сохранить своё здоровье, продлить свою жизнь. Правильная диета – одна из важнейших составляющих в вопросе о здоровье. Из-за избыточного веса или дефицита массы тела в организме запускаются процессы, нарушающие работу многих жизненно важных систем. При истощении происходит разрушение мышечных волокон, что приводит к дефициту мышечной массы, а ведь именно мышцы поддерживают «каркас» опорно-двигательного аппарата тела и влияют на силу и выносливость организма. Лишний вес – это не только эстетическая проблема. Избыток жира провоцирует возникновение проблем со здоровьем. В наибольшей степени страдают сосудистая и сердечная системы, поскольку из-за превышения веса на них тоже возрастает нагрузка. Как результат, у людей с ожирением развивается гипертония, что влечет за собой риск появления инсульта и инфаркта. Кроме того, гормоны, содержащиеся в жировой клетчатке, являются провокаторами отложения холестерина, запускают процессы старения сосудов.

Менеджер диеты требует удобный пользовательский интерфейс с возможностью записи данных пользователя в базу данных. Также необходимо использовать преимущества объектно-ориентированного программирования для оптимизации процессов обработки данных и взаимодействия с пользователем. Высокоуровневый объектно-ориентированный язык программирования C++ способен в полной мере удовлетворить эти потребности. С помощью различных фреймворков создается графический интерфейс оконного приложения, с элементами, управляемыми мышью и клавиатурой, что делает взаимодействие с программой удобным и интуитивно понятным.

Цель курсового проектирования заключается в использовании полученных знаний, умений и навыков в ходе изучения дисциплины «Программирование на языках высокого уровня» и языка программирования

с последующей разработкой программного средства для поддержания правильной диеты.

Для достижения цели были поставлены следующие задачи: изучить необходимую информацию по алгоритму и разработать программный продукт.

Курсовое проектирование представляет собой учебно-исследовательскую деятельность и необходимо определить объект и предмет исследования. Объектом исследования является расчёт по формулам здоровой диеты. Предметом исследования является программа «Менеджер диеты».

## 1 ОБЗОР ЛИТЕРАТУРЫ

Диета – строгое и сознательно ограничение количества энергии, потребляемой с пищей (калораж). Например, это может быть подсчет калорий и установление жестких рамок.

BMR (Basal metabolic rate) - базовый метаболизм, базовое количество калорий, которое позволяет человеку "существовать", ничего не делая.

NEAT (Non-exercise associated thermogenesis) – количество калорий, которое нужно прибавить за счет не спортивной дневной активности (стирка, разговоры, прогулки, работа, поход за покупками и т.д).

EAT (Exercise associated thermogenesis) – количество калорий, которое нужно прибавить за счет спорта.

TEF (Thermogenic effect of feeding) – термогенный эффект еды (сколько калорий тратится на усвоение и переваривание пищи).

TEE (Total Energy Expenditure) – общее количество необходимых человеку калорий ( $BMR + EAT + NEAT + TEF$ ).

Метод подсчёта зависит от множества параметров:

1. возраст и пол (в большинстве случаев мужчинам нужно потреблять больше, чем женщинам)
2. общий вес и мышечная масса
3. рацион питания
4. уровень спортивной активности
5. физиологический статус (здоровый, болеющий)
6. дневная активность

### 1.1 Реализация с использованием формулы Харриса-Бенедикта

Эта формула все еще довольно часто встречается. Проблема с ней в том, что она основана на исследовании, в котором участвовали молодые, ведущие активный образ жизни мужчины в холодных лабораториях много лет назад, и



она дает большее количество калорий, чем на самом деле нужно, особенно в случаях, если у человека лишний вес.

Для мужчин:

$$\text{BMR} = 66 + [13.7 \times \text{вес (кг)}] + [5 \times \text{рост (см)}] - [6.76 \times \text{возраст (в годах)}]$$

Для женщин:

$$\text{BMR} = 655 + [9.6 \times \text{вес (кг)}] + [1.8 \times \text{рост (см)}] - [4.7 \times \text{возраст (в годах)}]$$

## **1.2 Реализация с использованием формулы Миффлина – Сан Жеора**

Разработана в 90-х годах 20 века. Самая популярная формула для расчета необходимого уровня калорий. Более подходящая и более реалистичная для сегодняшнего образа жизни и питания. Но и она не берет в расчет разницу между различным процентом жира. Считается, что эта формула тоже несколько завышает потребности в калориях.

Для мужчин:

$$\text{BMR} = [9.99 \times \text{вес (кг)}] + [6.25 \times \text{рост (см)}] - [4.92 \times \text{возраст (в годах)}] + 5$$

Для женщин:

$$\text{BMR} = [9.99 \times \text{вес (кг)}] + [6.25 \times \text{рост (см)}] - [4.92 \times \text{возраст (в годах)}] - 161$$

## **1.3 Реализация с использованием формулы Кэтча - МакАрдла**

Считается наиболее точной из ей подобных формул, но для того, чтобы ей воспользоваться, нужно знать свой процент жира.

$$\text{BMR} = 370 + (21.6 \times \text{LBM})$$

$$\text{LBM} = [\text{вес (кг)} \times (100 - \% \text{жира})] / 100$$

## **1.4 Конечный расчёт ТЕЕ**

Чтобы получить финальное значение дневной калорийности, нужно умножить полученный BMR на коэффициент, который будет соответствовать физической активности (1 – 2). Далее расчёт в зависимости от целей: потерять

вес или набрать массу. Вместо использования стандартной схемы (плюс или минус 500 калорий в день от TEE) следует отниматься или прибавлять определенный процент от получившейся цифры.

### **1.5 Обоснование выбранных методов и алгоритмов**

В качестве фреймворка выбрана технология .NET в знакомой среде разработки Visual Studio, а метод хранения данных – база данных в СУБД MSAccess. Для выполнения поставленной задачи был выбран метод реализации с помощью формулы расчёта Миффлина – Сан Жеора. Формула включает в себя самые простые для измерения параметры о теле, которые знает практически каждый, в следствии именно эта формула является самой подходящей и универсальной.

## **2 СТРУКТУРНОЕ ПРОЕКТИРОВАНИЕ**

Код программы был разбит на модули для более удобного и наглядного процесса разработки программы. Подробное описание классов см. в UML-диаграмме в Приложении В.

### **2.1 MyForm.cpp**

Этот файл содержит точку входа в программу (функцию WinMain), которая запускает приложение и открывает форму авторизации (см. п.2.2).

### **2.2 MyForm.h**

Этот файл содержит код инициализации визуального графического интерфейса для авторизации (кнопки “Вход” и “Регистрация”), а также кнопку выхода. Код в этом файле создается автоматически при добавлении в конструкторе форм графических элементов и изменении их свойств.

### **2.3 Authorization.h**

В этом файле содержится код инициализации графического интерфейса. Отрисовываются поля для ввода логина и пароля. Содержит код обращения к БД для проверки и авторизации пользователя.

### **2.4 Registration.h**

В этом файле содержится код инициализации графического интерфейса. При создании этой формы отрисовываются поля для ввода логина и пароля. В этом же файле делается запрос обращения к базе данных, создаётся новая запись.

## **2.5 UserInformation.h**

Файл, где отрисовывается окно для ввода пользователем данных: возраст, вес, рост, идеальный вес, имя, фамилия. Также объявляются методы обработки нажатия. Создаётся поле проверки нажатия на кнопку выбора пола.

## **2.6 UserInformationFunctions.h**

В этом файле находятся определения для методов, необходимых для добавления записи в базу данных: `informationConfirmButton_Click`, `sexManButton_Click`, `sexWomanButton_Click`.

## **2.7 GlobalFunctions.h**

Заголовочный файл, содержащий объявления для некоторых функций, которые необходимы при разработке этой программы. Это функции проверки корректности ввода, функции преобразования типов данных, функция корректировки введенных и полученных из базы данных.

## **2.8 settings.h**

Содержит код инициализации графического интерфейса профиля. Подключение к базе данных с информацией о пользователе и выводит на экран.

## **2.9 container.h**

Содержит шаблонный класс `list<T>`, вложенный в него класс `iterator`, а также структуру `record`, объекты которой будут элементами контейнера `list`. Здесь же находятся и определения для методов работы с контейнером и

итератором (конструкторы, деструкторы, перегрузки операторов для итератора).

## **2.10 ActionsClass.h**

Этот заголовочный файл содержит абстрактный класс Food, от которого наследуются классы AddFood и FindFood. Эти классы нужны для обработки запроса пользователя на добавление нового продукта в дневном рационе и для поиска существующей записи о продукте соответственно.

Файл содержит абстрактный класс BaseActions, от которого наследуется класс AddAction, который наследуют методы ConvertToSystem для конвертации std::string в управляемую строку String^, autoEndWeightCalculate для расчёта идеального веса пользователя и поля list<record> Records, record\* Rec, int age, height, weight, end\_weight. Этот класс нужен для осуществления обработки действий пользователя и загрузке информации о нём в таблицу базы данных.

## **2.11 MainWindow.h**

Файл содержит код отрисовки основного окна приложения. Объявлены и определены методы кнопок: «Завтрак», «Обед», «Ужин», «Профиль», «Новый продукт», «Выход».

## **2.12 product.h**

Содержит код инициализации графического интерфейса добавления нового продукта и его калорийности. Объявлены поля управляемых строк: productName, kkal, product\_check. Объявлен метод класса product для обработки нажатия на кнопку «Добавить».

### **2.13 productFunctions.h**

В файле хранится определение метода класса `product` обработки нажатия. В методе происходит подключение к базе данных, поиск и проверка на наличие уже существующих записей, добавление нового продукта в базу данных. `usersfood.h`

### **2.14 usersfood.h**

Содержит код инициализации графического интерфейса добавления съеденной пищи в дневной рацион. Объявлены поля управляемых строк: `food`, `kcal`, `gram`. Объявлены методы класса `userdfood`: `FindButton_Click` (кнопка «Искать»), `AddFoodButton_Click` (кнопка «Добавить»).

### **2.15 usersfoodFunctions.h**

В файле хранится определение методов класса `product`: `FindButton_Click`, `AddFoodButton_Click`.

### 3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ

Для упрощения рассмотра кода и понимания работы программы ниже представлены UML-диаграмма прецедентов и алгоритмы по шагам для основных функций программы.

#### 3.1 Диаграмма прецедентов

Диаграмма прецедентов – UML-диаграмма, отражающая отношения между актором и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Диаграмма прецедентов представлена на рисунке 3.1.1

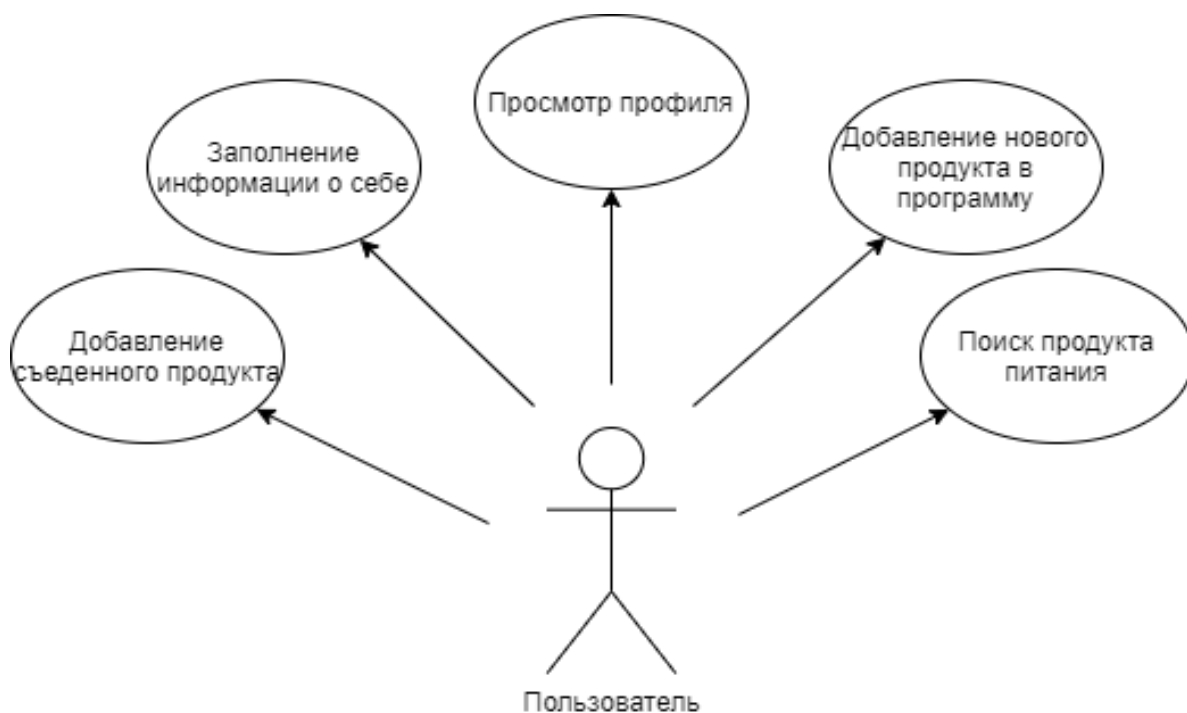


Рисунок 3.1.1 – UML-диаграмма прецедентов.

#### 3.2 Метод проверки данных `bool AddAction::infoChek()`

Шаг 1. Объявить переменную `est` типа `bool` и инициализировать значением «false».

Шаг 2. Объявить переменную `i` и инициализировать значением 0.

- Шаг 3. Проверить поля `ageRec[i]`, `heightRec[i]`, `weightRec[i]` наследованного объекта структуры `Rec`, если больше или равно «А» и меньше или равно «Z», или больше или равно «а» и меньше или равно «z», и равно пустой строке, проверить поле `end_weightRec[i]` структуры `Rec`, если больше или равно «А» и меньше или равно «Z», или больше или равно «а» и меньше или равно «z», то перейти к Шагу 4, иначе перейти к Шагу 5.
- Шаг 4. Присвоить переменной `est` значение «true».
- Шаг 5. Проверить переменную `est`, если «false» перейти к Шагу 6, иначе перейти к Шагу 13.
- Шаг 6. Присвоить полю `age` класса `BaseAction` значение функции `fromStringToInt(Rec->ageRec)`.
- Шаг 7. Присвоить полю `height` класса `BaseAction` значение функции `fromStringToInt(Rec-> heightRec)`.
- Шаг 8. Присвоить полю `weight` класса `BaseAction` значение функции `fromStringToInt(Rec-> weightRec)`.
- Шаг 9. Проверить поле `end_weightRec` наследованного объекта структуры `Rec`, если не равно пустой строке перейти к Шагу 12.
- Шаг 10. Присвоить полю `end_weight` класса `BaseAction` значение публичного метода `BaseAction::autoEndWeightCalculate(gender, height, age)`.
- Шаг 11. Присвоить полю `end_weightRec` наследованного объекта структуры `Rec` значение шаблонной функции `toString(end_weight)`. Перейти к Шагу 13.
- Шаг 12. Присвоить полю `end_weight` класса `BaseAction` значение функции `fromStringToInt(Rec-> end_weightRec)`.
- Шаг 13. Вернуть значение `est`.
- Шаг 14. Конец алгоритма.



### 3.3 Метод добавления записи `void AddAction::AddRecord()`

- Шаг 1. Объявить `bool` переменную `est` и инициализировать возвращённым значением метода проверки введённых данных `AddAction::infoChek()`.
- Шаг 2. Проверить поля класса `BaseAction` `age`, `height`, `weight`, `end_weight`. Если `age` больше 16, `height` больше или равен 130, `weight` больше 29, `end_weight` больше 29 перейти к Шагу 3, иначе перейти к Шагу 29.
- Шаг 3. Инициализировать строку подключения `String^ connectionString` и присвоить ей значение «provider=Microsoft.Jet,OleDB.4.0;Data Source= information.mdb».
- Шаг 4. Инициализировать подключение к базе данных `OleDbConnection^ dbConnection = gcnew OleDbConnection(connectionString)`.
- Шаг 5. Вызвать `dbConnection->Open()`.
- Шаг 6. Инициализировать строку запроса `String^ query` и присвоить ей значение «SELECT \* FROM [information]».
- Шаг 7. Инициализировать команду `OleDbCommand^ dbCommand = gcnew OleDbCommand(query, dbConnection)`.
- Шаг 8. Объявить объект `dbReader` класса `OleDbDataReader^` и присвоить значение «dbCommand->ExecuteReader()».
- Шаг 9. Если поле `Rec->nameRec` или `Rec->name2Rec` хранит пустую строку, или если переменная `est` равна «true» перейти к Шагу 10, иначе перейти к Шагу 13.
- Шаг 10. Вызвать окно сообщения об ошибке `MessageBox::Show` с текстом «Не все поля заполнены или заполнены некорректно, проверьте форму!», «Ошибка!».
- Шаг 11. Вызвать `dbReader->Close()`.
- Шаг 12. Вызвать `dbConnection->Close()`, перейти к Шагу 30.
- Шаг 13. Начать цикл `while` пока `dbReader->Read()` возвращает `true`.

- Шаг 14. Если экземпляр класса представляющий несуществующие значение не равен `dbReader["name"]` перейти к Шагу 15, иначе перейти к Шагу 13.
- Шаг 15. Объявление локальной `string` переменной `genderString`.
- Шаг 16. Если поле `BaseAction::gender` типа `bool` «false» присвоить локальной переменной `genderString` значение «М» и перейти к Шагу 18, иначе перейти к Шагу 17.
- Шаг 17. Присвоить локальной переменной `genderString` значение «Ж».
- Шаг 18. Инициализировать строку запроса `String^ query1` и присвоить ей значение «INSERT INTO [information] (name,name2, age,height,weight,end\_weight,gender, login)VALUES (' + `ConvertToSystem(Rec->nameRec)` + " , " + `ConvertToSystem(Rec->name2Rec)` + " , " + `ConvertToSystem(Rec->ageRec)` + " , " + `ConvertToSystem(Rec->heightRec)` + " , " + `ConvertToSystem(Rec->weightRec)` + " , " + `ConvertToSystem(Rec->end_weightRec)` + " , " + `ConvertToSystem(genderString)` + " , " + `ConvertToSystem(global_login)` + ")».
- Шаг 19. Инициализировать команду `OleDbCommand^ dbCommand1 = gcnew OleDbCommand(query1, dbConnection);`
- Шаг 20. Начало блока `try`, который содержит Шаг 21.
- Шаг 21. Выполнить инструкцию `dbCommand1->ExecuteNonQuery()`, которая может бросить исключение типа `InvalidOperationException^`.
- Шаг 22. Если поймано исключение типа `InvalidOperationException^`, вход в блок `catch`(Шаг 23 – Шаг 25);
- Шаг 23. Вызвать окно сообщения об ошибке `MessageBox::Show` с текстом “Ошибка выполнения запроса!”;
- Шаг 24. Закрывать соединение с базой данных методом `dbConnection->Close()`;
- Шаг 25. Выйти из функции командой `return`. Перейти к Шагу КОНЕЦ.
- Шаг 26. Прекратить чтение методом `dbReader->Close()`.
- Шаг 27. Закрывать соединение с базой данных методом `dbConnection->Close()`.

- Шаг 28. Вызвать метод `BaseAction::sexSelectionHandler` типа `void` для расчёта нормы потребления калорий в день. Перейти к Шагу 30.
- Шаг 29. Вызвать окно сообщения об ошибке `MessageBox::Show` с текстом «Ваши данные неправильны! Проверьте их!».
- Шаг 30. Конец алгоритма.

### **3.3 Метод вычисления калорий `void BaseAction::sexSelectionHandler()`**

- Шаг 1. Объявить переменную `physicalActivityCoefficient` типа `float` и инициализировать значением «12/10».
- Шаг 2. Объявить переменную `upDownWeight` типа `float` и инициализировать значением «1».
- Шаг 3. Объявить переменную `sexFactor` типа `int` и инициализировать значением «5».
- Шаг 4. Если поле `gender` типа `bool` класса `BaseAction` имеет значение «true» перейти к Шагу 5, иначе перейти к Шагу 6.
- Шаг 5. Присвоить переменной `sexFactor` значение «-181».
- Шаг 6. Если значение выражения поля класса `BaseAction` типа `int` (`end_weight - 2`) больше или равно значению `int` поля `weight` класса `BaseAction` перейти к Шагу 7, иначе перейти к Шагу 8.
- Шаг 7. Присвоить переменной `upDownWeight` значение «12/10».
- Шаг 8. Если значение поля класса `BaseAction` типа `int` (`end_weight + 2`) меньше или равно значению `int` поля `weight` класса `BaseAction` перейти к Шагу 9, иначе перейти к Шагу 10.
- Шаг 9. Присвоить переменной `upDownWeight` значение «9/10».
- Шаг 10. Объявить переменную `kal` типа `int` и инициализировать значением « $(9.99 * weight + 6.25 * height - 4.92 * age + sexFactor) * upDownWeight$ ». Где `weight`, `height`, `age` – приватные поля класса `BaseAction` типа `int`.
- Шаг 11. Присвоить переменной `kal` для расчёта дневной нормы калорий значение « $kal * physicalActivityCoefficient$ ».

- Шаг 12. Вызвать функцию типа `void insertCalories(kal)` для записи нормы калорий в день в базу данных.
- Шаг 13. Конец алгоритма.

## 4 ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЯ

Чтобы использовать программу Менеджер диеты необходимо пройти авторизацию введя свой логин и пароль или зарегистрироваться если учетная запись ещё не создана (приложение Б.1). При регистрации необходимо придумать и заполнить поля логин и пароль (приложение Б.2). Далее в случае успешной регистрации появиться окно для заполнения информации необходимой для расчётов (приложение Б.4). Поля помеченные «\*» являются полями обязательными для заполнения, в поле «Идеальный вес» по желанию заполняется конечный вес, если пропустить его заполнение идеальный вес будет рассчитан автоматически исходя из введенных данных (автоматический расчёт идеального веса не учитывает особенности строения тела, образа жизни, здоровья). Также важно выбрать пол нажав на соответствующую кнопку («М» – мужской, «Ж» – женский). Необходимо проверить правильность введенных данных и нажать кнопку «Подтвердить».

После успешной регистрации или авторизации появиться главное окно интерфейса Менеджера диеты (приложение Б.4). В правом верхнем углу отображается количество калорий необходимых для поддержания диеты (по-разному рассчитывается: при дефиците, избытке массы тела, при нормальной массе тела). В левом верхнем углу находится количество потребленных калорий, значение будет обновляться после нажатия на кнопки «Завтрак», «Обед» или «Ужин» и добавления съеденного в появившемся окне поиска и добавления продуктов (приложение Б.9). В соответствующих полях появившегося окна нужно записать название продукта и массу в граммах съеденного. Если продукта нет в базе данных программы появиться сообщение (приложение Б.10). Вернувшись в главное окно интерфейса программы, можно нажать кнопку «Новый продукт» и откроется окно добавления нового продукта в базу данных программы (приложение Б.11). Для добавления продукта в левом поле необходимо указать название продукта, а в поле справа количество калорий, содержащихся в 100 граммах этого продукта.

Найти калорийность какого-либо продукта можно в интернете или других источниках информации.

Просмотреть данные свои данные можно нажав кнопку «Профиль» в главном окне программы. После нажатия отобразится персональная страница с данными (приложение Б.8).

Чтобы завершить работу программы нужно нажать кнопку «Выход».

## ЗАКЛЮЧЕНИЕ

В этом курсовом проекте был реализован Менеджер диеты с использованием фреймворка .NET и языка C++/CLI на основе изученного материала и знаний из дисциплины "Программирование на языках высокого уровня". Основными плюсами этой программы являются: автоматический расчёт идеального веса, сохранение съеденной еды в базу данных для использования их в следующей сессии, возможность добавления пользователем еды помимо определенной по умолчанию, от пользователя не требуются знания о параметрах тела помимо веса и роста.

В дальнейшем данный проект может быть улучшен и дополнен в следующих аспектах:

1. Дневник питания, помогающий проследить процесс потери (или набора) массы тела посуточно.
2. Добавление настраиваемых уведомлений о необходимости попить воды, позаниматься или принять пищу.
3. Функция интерактивный помощник выбора диеты.
4. Использование серверной базы данных.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Под ред. О. Е. Степаненко Москва «Научная книга», Visual C++.NET Классика программирования. -Киев: Букинист, 2010. -768 с.: ил. — (В подлиннике) — с. 1 – 20.
2. Гордон Хогенсон, C++/CLI: язык Visual C++ для среды .NET. -Вильямс, 2007. – 464 с.: ил. — (В подлиннике) — с. 324 – 340.
3. Программирование .NET с использованием C++/CLI [Электронный ресурс]. — Режим доступа: <https://learn.microsoft.com/ru-ru/cpp/dotnet/dotnet-programming-with-cpp-cli-visual-cpp?view=msvc-160/>. —[Дата доступа] — 12.11.2022
4. Как посчитать калории, белки, углеводы и жиры. [Электронный ресурс].— Режим доступа: <http://goodlooker.ru/tablica-dlya-raschetov-bzhu.html>. —[Дата доступа] — 12.11.2022



**ПРИЛОЖЕНИЕ А**  
**(обязательное)**  
**Листинг кода программного средства**

```

\\ файл "Authorization.cpp"
#include "Authorization.h"

\\ файл " MainWindow.cpp"
#include "MainWindow.h"

\\ файл " MyForm.cpp"
#include "MyForm.h"
#include "UserInformationFunctions.h"

#include "MainWindowFunctions.h"
#include "productFunctions.h"
#include "usersfoodFunctions.h"

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;
using namespace System::Data::OleDb;
using namespace std;
[STAThreadAttribute]

int WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);
    MyForm^ myform = gcnew MyForm;
    Application::Run(myform);
    return 0;
}
//Здесь точка входа в программу, открытие первого окна

\\ файл "product.cpp"
#include "product.h"

\\ файл "Registration.cpp"
#include "Registration.h"

\\ файл "session.cpp"
#include "session.h"
//глобальная переменная для запоминания пользователя
std::string global_login;

\\ файл "settings.cpp"
#include "settings.h"

\\ файл "UserInformation.cpp"
#include "UserInformation.h"

\\ файл "usersfood.cpp"
#include "usersfood.h"

\\ файл "ActionsClass.h"
#pragma once
#include <Windows.h>

```

```

#include <string.h>
#include <msclr\marshal_cppstd.h>
#include <string>
#include <iostream>
#include "stdlib.h"
#include <cstring>

#include "GlobalFunctions.h"
#include "container.h"
#include "session.h"

using namespace DietManager;
using namespace System::Data::OleDb;
using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Runtime::InteropServices;
using namespace std;

class Food
{
protected:
    string food;
    string kkal;
    string food_check;
public:
    String^ ConvertToSystem(string str) { return gcnew String(str.data()); }
} //из stl в system string
};

class AddFood : public Food
{
protected:
    string gram;
    string login;
    int fl;
public:
    AddFood(String^ food, String^ gram, int fl);
    void AddFood::addFood();
    int kkalProduct(int lenght);
};

int AddFood::kkalProduct(int h) {
    string kkal1 = kkal;
    int kkal2 = 0, gram2 = 0;
    int k = 1;
    for (int i = 0; i < h - 1; i++) {
        k *= 10;
    }
    for (int i = 0; kkal1[i] != 0; i++) {
        kkal2 += (kkal1[i] - '0') * k;
        k /= 10;
    }
    string gram1 = gram;
    h = gram1.length();
    k = 1;
    for (int i = 0; i < h - 1; i++) {
        k *= 10;
    }
}

```

```

        for (int i = 0; gram1[i] != 0; i++) {
            gram2 += (gram1[i] - '0') * k;
            k /= 10;
        }
        return (gram2 * kkal2 / 100);
    }

AddFood::AddFood(String^ food, String^ gram, int fl)
{
    this->fl = fl;
    this->food = msclr::interop::marshal_as<std::string>(food);
    this->gram = msclr::interop::marshal_as<std::string>(gram);
    string login = global_login;
}

void AddFood::addFood()
{
    bool added = false;
    bool check = false;
    int fl1 = 0, gram2 = 0;
    String^ connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data
Source=product.mdb";
    OleDbConnection^ dbConnection = gcnew
OleDbConnection(connectionString);
    dbConnection->Open(); //открываем соединение
    String^ query = "SELECT * FROM [Таблица1]"; // запрос
    OleDbCommand^ dbCommand = gcnew OleDbCommand(query,
dbConnection); //команда
    OleDbDataReader^ dbReader1 = dbCommand->ExecuteReader(); // считываем
данные
    while (dbReader1->Read()) {
        food_check =
msclr::interop::marshal_as<std::string>((String^)dbReader1["Продукт"]);
        if (DBNull::Value != dbReader1["Продукт"]) {
            if (food_check == food) {
                check = true;
                MessageBox::Show(" " + ConvertToSystem(food) + " есть
в базе!", "", MessageBoxButtons::OK);
                kkal =
msclr::interop::marshal_as<std::string>((String^)dbReader1["калорийность"]);
                string kkal1 = kkal;
                int kkal3 = kkalProduct(kkal1.length());
                MessageBox::Show("Калорийность данного продукта = " +
ConvertToSystem(kkal) + "", "", MessageBoxButtons::OK);
                dbReader1->Close();
                dbConnection->Close();
                added = true;
                break;
            }
        }
    }
    if (added == true) {
        connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data
Source=user_food.mdb";
        dbConnection = gcnew OleDbConnection(connectionString);
        dbConnection->Open();
        query = "SELECT * FROM [Таблица1]";
        dbCommand = gcnew OleDbCommand(query, dbConnection);
        OleDbDataReader^ dbReader = dbCommand->ExecuteReader();
        string kkal1 = kkal;
        int kkal3 = kkalProduct(kkal1.length());
    }
}

```

```

        query = "INSERT INTO [Таблица1] (id,product,kkal,login )VALUES
('" + fl + "' , '" + ConvertToSystem(food) + "','" + kkal3 + "','" +
ConvertToSystem(global_login) + "')";
        dbCommand = gcnew OleDbCommand(query, dbConnection);
        try
        {
            dbCommand->ExecuteNonQuery();
        }
        catch (InvalidOperationException^)
        {
            MessageBox::Show("Ошибка выполнения запроса!");
            dbConnection->Close();
            return;
        }
        MessageBox::Show("продукт" + ConvertToSystem(food) + "",
"добавлен в сегодняшний рацион", MessageBoxButtons::OK);
        dbReader->Close();
        dbConnection->Close();
    }
    if (check == false) {
        MessageBox::Show("Извините, данного продукта пока нет в базе
данных, если вы точно знаете калорийность, то можете его добавить сами",
"Сообщение", MessageBoxButtons::OK, MessageBoxIcon::Warning);
        dbReader1->Close();// закрываем соединение
        dbConnection->Close();
    }
}

class FindFood : public Food
{
public:
    FindFood(String^ food);
    void FindFood::findFood();
};

FindFood::FindFood(String^ food)
{
    this->food = msclr::interop::marshal_as<std::string>(food);
}

void FindFood::findFood()
{
    bool check = false;
    String^ connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data
Source=product.mdb";
    OleDbConnection^ dbConnection = gcnew
OleDbConnection(connectionString);
    dbConnection->Open();//открываем соединение
    String^ query = "SELECT * FROM [Таблица1]";// запрос
    OleDbCommand^ dbCommand = gcnew OleDbCommand(query,
dbConnection);//команда
    OleDbDataReader^ dbReader1 = dbCommand->ExecuteReader();// считываем
данные
    while (dbReader1->Read()) {
        food_check =
msclr::interop::marshal_as<std::string>(((String^)dbReader1["Продукт"]));
        if (DBNull::Value != dbReader1["Продукт"]) {
            if (food_check == food) {
                MessageBox::Show(" " + ConvertToSystem(food) + " есть
в базе!", "", MessageBoxButtons::OK);
                kkal =
msclr::interop::marshal_as<std::string>((String^)dbReader1["калорийность"]);

```

```

        MessageBox::Show("Калорийность данного продукта = " +
ConvertToSystem(kkal) + "", "", MessageBoxButtons::OK);
        check = true;
        dbReader1->Close();// закрываем соединение
        dbConnection->Close();
        break;
    }
}

if (check == false) {
    MessageBox::Show("Извините, данного продукта пока нет в базе
данных, если вы точно знаете калорийность, то можете его добавить сами",
"Сообщение", MessageBoxButtons::OK, MessageBoxIcon::Warning);
    dbReader1->Close();// закрываем соединение
    dbConnection->Close();
}
}

```

```

//класс действий с информацией о пользователе
class BaseAction
{
protected:
    list<record> Records;
    record* Rec = new record;
    int age = 0, height = 0, weight = 0, end_weight = 0;
    bool gender;
public:
    String^ ConvertToSystem(string str) { return gcnew String(str.data());
} //из stl в system string
    void sexSelectionHandler(bool gender, int weight, int height, int age,
int end_weight);
    int autoEndWeightCalculate(bool gender, int height, int age);
    ~BaseAction() { delete Rec; }
};

```

```

//Расчет идеального веса по формуле Шульгиной
int BaseAction::autoEndWeightCalculate(bool gender, int height, int age) {
    if (!gender) { return (50 + (height - 150) * 0.75 + (age - 21) / 4); }
//for man
    else { return (50 + (height - 150) * 0.32 + (age - 21) / 5); } //for
woman
}

```

```

//Вычисляем кол-во калорий для поддержания текущего веса
void BaseAction::sexSelectionHandler(bool gender, int weight, int height, int
age, int end_weight) {
    float physicalActivityCoefficient = (12 / 10);
    //если начальный и конечный вес примерно равны и нужно поддерживать вес
    float upDownWeight = 1;
    int sexFactor = 5; //По умолчанию для мужчин
    if (gender) sexFactor = -181;
    if (end_weight - 2 >= weight) upDownWeight = (12 / 10);
    if (end_weight + 2 >= weight) upDownWeight = (9 / 10);
    // BMR РАССЧЁТ КАЛОРИЙ В ДЕНЬ
    int kal = (9.99 * weight + 6.25 * height - 4.92 * age +
sexFactor)*upDownWeight;
}

```

```

        //ТЕЕ кол-во калорий для поддержания/увеличения/уменьшения текущего
веса
        kal *= physicalActivityCoefficient;
        insertCalories(kal);
    }

//добавить запись в файл
class AddAction : public BaseAction
{
protected:
    //string login;
public:
    AddAction(String^ name, String^ name2, String^ age, String^ height,
String^ weight, String^ end_weight, bool gender);
    //bool CheckIfRepeating(); //проверим, не существует ли уже такой
записи
    bool AddAction::infoChek();
    void AddAction::AddRecord(); //добавим новую запись в файл
};

bool AddAction::infoChek() {
    bool est = false;
    int i = 0;
    if (((Rec->ageRec[i] >= 'A') && (Rec->ageRec[i] <= 'Z'))
        || ((Rec->ageRec[i] >= 'a') && (Rec->ageRec[i] <= 'z') || Rec-
>ageRec == "")) || ((Rec->heightRec[i] >= 'A') && (Rec->heightRec[i] <= 'Z'))
        || ((Rec->heightRec[i] >= 'a') && (Rec->heightRec[i] <= 'z') ||
Rec->heightRec == "") || ((Rec->weightRec[i] >= 'A') && (Rec->weightRec[i] <=
'Z'))
        || ((Rec->weightRec[i] >= 'a') && (Rec->weightRec[i] <= 'z') ||
Rec->weightRec == "") || ((Rec->end_weightRec[i] >= 'A') && (Rec-
>end_weightRec[i] <= 'Z'))
        || ((Rec->end_weightRec[i] >= 'a') && (Rec->end_weightRec[i] <=
'z'))))
    {
        est = true;
    }

    if (est == false) {
        age = fromStringToInt(Rec->ageRec);
        height = fromStringToInt(Rec->heightRec);
        weight = fromStringToInt(Rec->weightRec);
        if (Rec->end_weightRec == "") {
            end_weight = autoEndWeightCalculate(gender, height, age);
            Rec->end_weightRec = toString(end_weight);
        }
        else end_weight = fromStringToInt(Rec->end_weightRec);
    }
    return est;
}

//Добавляем записи в БД information
void AddAction::AddRecord()
{
    bool est = infoChek();
    if (age > 10 && height >= 130 && weight > 29 && end_weight > 29) {
        String^ connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data
Source=information.mdb";
        OleDbConnection^ dbConnection = gcnew
OleDbConnection(connectionString);
        dbConnection->Open();
    }
}

```

```

        String^ query = "SELECT * FROM [information]";
        OleDbCommand^ dbCommand = gcnew OleDbCommand(query,
dbConnection);
        OleDbDataReader^ dbReader = dbCommand->ExecuteReader();
        if (Rec->nameRec == "" || Rec->name2Rec == "" || est == true) {
            MessageBox::Show("Не все поля заполнены или заполнены
некорректно, проверьте форму!", "Ошибка!", MessageBoxButtons::OK);
            dbReader->Close();
            dbConnection->Close();
        }
        else {
            while (dbReader->Read()) {
                // Проверка: можно ли записать в поле таблицы
                if (DBNull::Value != dbReader["name"]) {
                    break;
                }
            }
            string genderString;
            if (!gender) { genderString = "М"; }
            else { genderString = "Ж"; }
            String^ query1 = "INSERT INTO [information] (name,name2,
age,height,weight,end_weight,gender, login)VALUES ('" + ConvertToSystem(Rec-
>nameRec) + "', '" + ConvertToSystem(Rec->name2Rec) + "', '" +
ConvertToSystem(Rec->ageRec) + "', '" + ConvertToSystem(Rec->heightRec) + "',
'" + ConvertToSystem(Rec->weightRec) + "', '" + ConvertToSystem(Rec-
>end_weightRec) + "', '" + ConvertToSystem(genderString) + "', '" +
ConvertToSystem(global_login) + "')";
            OleDbCommand^ dbCommand1 = gcnew OleDbCommand(query1,
dbConnection);
            try
            {
                dbCommand->ExecuteNonQuery();
            }
            catch (InvalidOperationException^)
            {
                MessageBox::Show("Ошибка выполнения запроса!");
                dbConnection->Close();
                return;
            }
            dbReader->Close();
            dbConnection->Close();
            sexSelectionHandler(gender, weight, height, age,
end_weight); //Расчёт нормы потребления калорий в день
        }
    }
    else
        MessageBox::Show("Ваши данные неправильны! Проверьте их!",
"Ошибка!", MessageBoxButtons::OK);
}

//Конструктор с параметрами
AddAction::AddAction(String^ name, String^ name2, String^ age, String^
height, String^ weight, String^ end_weight, bool gender)
{
    this->gender = gender;
    Rec->nameRec = msclr::interop::marshal_as<std::string>(name);
    Rec->name2Rec = msclr::interop::marshal_as<std::string>(name2);
    Rec->ageRec = msclr::interop::marshal_as<std::string>(age);
    Rec->heightRec = msclr::interop::marshal_as<std::string>(height);
    Rec->weightRec = msclr::interop::marshal_as<std::string>(weight);
    if (end_weight != "") Rec->end_weightRec =
msclr::interop::marshal_as<std::string>(end_weight);
}

```



```

        else Rec->end_weightRec = "";
    }

\\ файл "Authorization.h"
#pragma once
#include <msclr\marshal_cppstd.h>
#include "MainWindow.h"

namespace DietManager {
    using namespace System::Data::OleDb;
    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
    public ref class Authorization : public System::Windows::Forms::Form {
    public:
        Authorization(void) {
            InitializeComponent();
        }
    public:

    protected:
        ~Authorization() {
            if (components)
            {
                delete components;
            }
        }
    private: System::Windows::Forms::Label^ label2;
    protected:
    private: System::Windows::Forms::Label^ label1;
    private: System::Windows::Forms::TextBox^ textBox2;
    private: System::Windows::Forms::TextBox^ textBox1;
    private: System::Windows::Forms::Button^ button3;
    private: System::Windows::Forms::Button^ button2;

    private:
        settings^ nastr;
        MainWindow^ persona;
    private: System::ComponentModel::Container^ components;

#pragma region Windows Form Designer generated code
        void InitializeComponent(void) {
            System::ComponentModel::ComponentResourceManager^
resources = (gcnew
System::ComponentModel::ComponentResourceManager(Authorization::typeid));
            this->label2 = (gcnew System::Windows::Forms::Label());
            this->label1 = (gcnew System::Windows::Forms::Label());
            this->textBox2 = (gcnew
System::Windows::Forms::TextBox());
            this->textBox1 = (gcnew
System::Windows::Forms::TextBox());
            this->button3 = (gcnew
System::Windows::Forms::Button());
            this->button2 = (gcnew
System::Windows::Forms::Button());
            this->SuspendLayout();
            //
            // label2
            //

```

```

        this->label2->AutoSize = true;
        this->label2->BackColor =
System::Drawing::Color::Transparent;
        this->label2->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
        this->label2->Location = System::Drawing::Point(96, 95);
        this->label2->Name = L"label2";
        this->label2->Size = System::Drawing::Size(67, 22);
        this->label2->TabIndex = 25;
        this->label2->Text = L"логин :";
        //
        // label1
        //
        this->label1->AutoSize = true;
        this->label1->BackColor =
System::Drawing::Color::Transparent;
        this->label1->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
        this->label1->Location = System::Drawing::Point(84,
149);
        this->label1->Name = L"label1";
        this->label1->Size = System::Drawing::Size(79, 22);
        this->label1->TabIndex = 24;
        this->label1->Text = L"пароль :";
        //
        // textBox2
        //
        this->textBox2->Location = System::Drawing::Point(169,
146);
        this->textBox2->Margin =
System::Windows::Forms::Padding(3, 4, 3, 4);
        this->textBox2->Name = L"textBox2";
        this->textBox2->PasswordChar = '*';
        this->textBox2->Size = System::Drawing::Size(488, 28);
        this->textBox2->TabIndex = 23;
        //
        // textBox1
        //
        this->textBox1->Location = System::Drawing::Point(169,
92);
        this->textBox1->Margin =
System::Windows::Forms::Padding(3, 4, 3, 4);
        this->textBox1->Name = L"textBox1";
        this->textBox1->Size = System::Drawing::Size(488, 28);
        this->textBox1->TabIndex = 22;
        //
        // button3
        //
        this->button3->BackColor =
System::Drawing::Color::LightGray;
        this->button3->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
        this->button3->Location = System::Drawing::Point(383,
200);
        this->button3->Margin =
System::Windows::Forms::Padding(3, 4, 3, 4);
        this->button3->Name = L"button3";
        this->button3->Size = System::Drawing::Size(274, 37);
        this->button3->TabIndex = 21;

```

```

        this->button3->Text = L"Выйти";
        this->button3->UseVisualStyleBackColor = false;
        this->button3->Click += gcnew System::EventHandler(this,
&Authorization::button3_Click);
        //
        // button2
        //
        this->button2->BackColor =
System::Drawing::Color::LightGray;
        this->button2->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
        this->button2->Location = System::Drawing::Point(169,
200);
        this->button2->Margin =
System::Windows::Forms::Padding(3, 4, 3, 4);
        this->button2->Name = L"button2";
        this->button2->Size = System::Drawing::Size(200, 37);
        this->button2->TabIndex = 19;
        this->button2->Text = L"Авторизация";
        this->button2->UseVisualStyleBackColor = false;
        this->button2->Click += gcnew System::EventHandler(this,
&Authorization::button2_Click);
        //
        // Authorization
        //
        this->AutoScaleDimensions = System::Drawing::SizeF(10,
22);
        this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
        this->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::Stretch;
        this->ClientSize = System::Drawing::Size(774, 303);
        this->Controls->Add(this->button3);
        this->Controls->Add(this->button2);
        this->Controls->Add(this->label2);
        this->Controls->Add(this->label1);
        this->Controls->Add(this->textBox2);
        this->Controls->Add(this->textBox1);
        this->Font = (gcnew System::Drawing::Font(L"Microsoft
Sans Serif", 9, System::Drawing::FontStyle::Italic));
        this->Icon =
(cli::safe_cast<System::Drawing::Icon^>(resources-
>GetObject(L"$this.Icon")));
        this->Margin = System::Windows::Forms::Padding(3, 4, 3,
4);
        this->Name = L"Authorization";
        this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterParent;
        this->Text = L"Менеджер диеты";
        this->Load += gcnew System::EventHandler(this,
&Authorization::Authorization_Load);
        this->ResumeLayout(false);
        this->PerformLayout();

    }
#pragma endregion

    private: System::Void Authorization_Load(System::Object^ sender,
System::EventArgs^ e) {
    }

```

```

        private: System::Void button2_Click(System::Object^ sender,
System::EventArgs^ e) {
            bool check = false;
            String^ login;
            String^ password;
            String^ login_check, ^ password_check;
            login = textBox1->Text;
            password = textBox2->Text;
            String^ connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data
Source=logpass.mdb";
            OleDbConnection^ dbConnection = gcnew
OleDbConnection(connectionString);

            dbConnection->Open();//открываем соединение
            String^ query = "SELECT * FROM [Таблица1]";// запрос
            OleDbCommand^ dbCommand = gcnew OleDbCommand(query,
dbConnection);//открываем соединение с БД
            OleDbDataReader^ dbReader1 = dbCommand->ExecuteReader();//
считываем данные
            while (dbReader1->Read()) {

                login_check = (String^)dbReader1["log"];
                password_check = (String^)dbReader1["pass"];

                if (DBNull::Value != dbReader1["log"] && DBNull::Value !=
dbReader1["pass"]) {
                    if (login_check == login && password_check ==
password) {

                        String^ keylog = login; //получить логин
пользователя
                        //до завершения программы хранится логин
пользователя
                        global_login =
mscorlib::interop::marshal_as<std::string>(keylog);

                        MessageBox::Show("Пользователь " + login + "
успешно авторизован!", "Успешный вход!", MessageBoxButtons::OK);
                        check = true;
                        dbReader1->Close();// закрываем соединение
                        dbConnection->Close();

                        if (nastr) nastr->Close();
                        nastr = gcnew settings();
                        this->AddOwnedForm(nastr);
                        if (persona) persona->Close();
                        persona = gcnew MainWindow();
                        this->AddOwnedForm(persona);
                        MainWindow^ pers = gcnew MainWindow();//
создаем новый экземпляр
                        pers->Show();//открывается персональная
страница

                        Authorization::Hide();// закрываем
                        break;
                    }
                }
            }
            if (check == false) {
                MessageBox::Show(this, "Вы ввели неверный логин или
пароль", "Сообщение", MessageBoxButtons::OK, MessageBoxIcon::Warning);
                dbReader1->Close();// закрываем соединение
                dbConnection->Close();
            }
        }
    }
}

```

```

    }
    private: System::Void button3_Click(System::Object^ sender,
System::EventArgs^ e) {
        Application::Exit();
    }
};
}

\\ файл "container.h"
#pragma once
#include<iostream>

using namespace std;

struct record
{
    string nameRec;
    string name2Rec;
    string ageRec;
    string heightRec;
    string weightRec;
    string end_weightRec;
    record* next = nullptr;
    record* prev = nullptr;
};

template<class T>
class list
{
private:
    int size = 0;
    T* head = nullptr;
    T* tail = nullptr;
public:
    ~list();
    void push_front(T* value);
    void push_back(T* value);
    T* begin() { return head; }
    T* end() { return tail; }
    int GetSize() { return this->size; }
    class iterator
    {
private:
        T* node;
public:
        iterator() :node(0) {};
        iterator(record* other) :node(other) {};
        iterator(const iterator& it) :node(it.node) {}; //конструктор
копирования
        iterator& operator=(const iterator& other) { node = other.node;
return *this; }
        bool operator==(const iterator& it)const { return node ==
it.node; }
        bool operator!=(const iterator& it)const { return !(node ==
it.node); }
        iterator& operator++() { node = node->next; return *this; }
        T& operator*()const { return *node; }
    };
};
};

```

```

template<class T>
list<T>::~~list()
{
    T* tmp = tail;
    for (size_t i = size; i > 0; i--)
        if (i == 1)
            delete head;
        else
        {
            T* tmp = tail;
            tail = tail->prev;
            tail->next = nullptr;
            delete tmp;
        }
}

template<class T>
void list<T>::push_back(T* value)
{
    record* Rec = new record;
    Rec->nameRec = value->nameRec;
    Rec->name2Rec = value->name2Rec;
    Rec->ageRec = value->ageRec;
    Rec->heightRec = value->heightRec;
    Rec->weightRec = value->weightRec;
    Rec->end_weightRec = value->end_weightRec;
    if (!head && !tail) {
        head = Rec;
        tail = head;
    } else {
        Rec->prev = tail;
        tail->next = Rec;
        tail = Rec;
    } size++;
}

template <class T>
void list<T>::push_front(T* value)
{
    record* Rec = new record;
    Rec->nameRec = value->nameRec;
    Rec->name2Rec = value->name2Rec;
    Rec->ageRec = value->ageRec;
    Rec->heightRec = value->heightRec;
    Rec->weightRec = value->weightRec;
    Rec->end_weightRec = value->end_weightRec;
    if (!head && !tail) head = tail = Rec;
    else {
        Rec->prev = nullptr;
        Rec->next = head;
        head->prev = Rec;
        head = Rec;
    } size++;
}

\\ файл "GlobalFunctions.h"
#pragma once
#include <Windows.h>
#include <iostream>
#include "Registration.h"

```

```

#include <string>
#include <sstream>
using namespace std;
using namespace DietManager;
using namespace::System::Data::OleDb;

void insertCalories(int kal);
int fromStringToInt(std::string str);

String^ ConvertToSystem(string str) { return gcnew String(str.data()); } //из
stl в system string

template <typename T>
std::string toString(T val)
{
    std::ostringstream oss;
    oss << val;
    return oss.str();
}

//Функция для конвертации строки в целое число
int fromStringToInt(string str) {
    int number = 0;
    int lenght = str.length();
    int k = 1;
    for (int i = 0; i < lenght - 1; i++) {
        k *= 10;
    }
    //сначала записываем старшие разряды
    for (int i = 0; str[i] != 0; i++) {
        number += (str[i] - '0') * k;
        k /= 10;
    }
    return number;
}

//Записываем норму каллорий/день в таблицу
void insertCalories(int kal) {
    String^ connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data
Source=kcal.mdb";
    OleDbConnection^ dbConnection = gcnew
OleDbConnection(connectionString);
    dbConnection->Open();
    String^ query = "SELECT * FROM [Таблица1]";
    OleDbCommand^ dbCommand = gcnew OleDbCommand(query, dbConnection);
    OleDbDataReader^ dbReader = dbCommand->ExecuteReader();
    String^ query1 = "INSERT INTO [Таблица1] (калория,login)VALUES ('" +
kal + "', '" + ConvertToSystem(global_login) + "')";
    OleDbCommand^ dbCommand1 = gcnew OleDbCommand(query1, dbConnection);
    try
    {
        dbCommand1->ExecuteNonQuery();
    }
    catch (InvalidOperationException^)
    {
        MessageBox::Show("Ошибка выполнения запроса!");
        dbConnection->Close();
        return;
    }
    dbReader->Close();
}

```

```

        dbConnection->Close();
    }

\\ файл "MainWindow.h"
#pragma once
#include <msclr\marshal_cppstd.h>
#include "string.h"
#include <cstring>
#include "regex"
#include <string>
#include "iostream"
#include "stdlib.h"

// #include "GlobalFunctions.h"
#include "settings.h"
#include "product.h"
#include "usersfood.h"

namespace DietManager {
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
    using namespace System::Data::OleDb;
    using namespace msclr::interop;
    using namespace std;

    public ref class MainWindow : public System::Windows::Forms::Form {
    public:
        MainWindow(void) {
            InitializeComponent();
        }
        int Ufl;
        static String^ Ulogin;
        static int Ukkal;
    private: System::Windows::Forms::Button^ updateButton;
    public:
    private: System::Windows::Forms::Label^ kkalPerDay;
    public:
    public:

    protected:
        ~MainWindow() {
            if (components) {
                delete components;
                delete Ulogin;
            }
        }
        usersfood^ user;
    private: System::Windows::Forms::Label^ label1;
    protected:
    private: System::Windows::Forms::Label^ label2;

    private: System::Windows::Forms::Button^ button2;
    private: System::Windows::Forms::Button^ profileButton;

    private: System::Windows::Forms::Button^ breakfastButton;

```



```

private: System::Windows::Forms::Button^ lunchButton;
private: System::Windows::Forms::Button^ dinnerButton;

private: System::Windows::Forms::Button^ button8;
private: System::Windows::Forms::Label^ kkalEated;

private: System::ComponentModel::Container^ components;

#pragma region Windows Form Designer generated code
void InitializeComponent(void) {
    System::ComponentModel::ComponentResourceManager^
resources = (gcnew
System::ComponentModel::ComponentResourceManager(MainWindow::typeid));
    this->label1 = (gcnew System::Windows::Forms::Label());
    this->label2 = (gcnew System::Windows::Forms::Label());
    this->button2 = (gcnew
System::Windows::Forms::Button());
    this->profileButton = (gcnew
System::Windows::Forms::Button());
    this->breakfastButton = (gcnew
System::Windows::Forms::Button());
    this->lunchButton = (gcnew
System::Windows::Forms::Button());
    this->dinnerButton = (gcnew
System::Windows::Forms::Button());
    this->button8 = (gcnew
System::Windows::Forms::Button());
    this->kkalEated = (gcnew
System::Windows::Forms::Label());
    this->kkalPerDay = (gcnew
System::Windows::Forms::Label());
    this->updateButton = (gcnew
System::Windows::Forms::Button());
    this->SuspendLayout();
    //
    // label1
    //
    this->label1->AutoSize = true;
    this->label1->BackColor =
System::Drawing::Color::SpringGreen;
    this->label1->Font = (gcnew
System::Drawing::Font(L"Montserrat Medium", 12,
static_cast<System::Drawing::FontStyle>((System::Drawing::FontStyle::Bold |
System::Drawing::FontStyle::Italic))));
    this->label1->Location = System::Drawing::Point(37, 31);
    this->label1->Margin =
System::Windows::Forms::Padding(2, 0, 2, 0);
    this->label1->Name = L"label1";
    this->label1->Size = System::Drawing::Size(312, 33);
    this->label1->TabIndex = 0;
    this->label1->Text = L"потреблено калорий:";
    //
    // label2
    //
    this->label2->AutoSize = true;

```

```

        this->label2->BackColor =
System::Drawing::Color::SpringGreen;
        this->label2->Font = (gcnew
System::Drawing::Font(L"Montserrat Medium", 12,
static_cast<System::Drawing::FontStyle>((System::Drawing::FontStyle::Bold |
System::Drawing::FontStyle::Italic))));
        this->label2->Location = System::Drawing::Point(644,
31);
        this->label2->Margin =
System::Windows::Forms::Padding(2, 0, 2, 0);
        this->label2->Name = L"label2";
        this->label2->Size = System::Drawing::Size(233, 33);
        this->label2->TabIndex = 1;
        this->label2->Text = L"число кал/день:";
        //
        // button2
        //
        this->button2->BackColor =
System::Drawing::Color::Azure;
        this->button2->Font = (gcnew
System::Drawing::Font(L"Montserrat Medium", 12,
static_cast<System::Drawing::FontStyle>((System::Drawing::FontStyle::Bold |
System::Drawing::FontStyle::Italic))));
        this->button2->ForeColor =
System::Drawing::SystemColors::ControlText;
        this->button2->Location = System::Drawing::Point(786,
254);
        this->button2->Margin =
System::Windows::Forms::Padding(2);
        this->button2->Name = L"button2";
        this->button2->Size = System::Drawing::Size(265, 43);
        this->button2->TabIndex = 10;
        this->button2->Text = L"Новый продукт";
        this->button2->UseVisualStyleBackColor = false;
        this->button2->Click += gcnew System::EventHandler(this,
&MainWindow::to_product);
        //
        // profileButton
        //
        this->profileButton->BackColor =
System::Drawing::Color::Lavender;
        this->profileButton->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::None;
        this->profileButton->Font = (gcnew
System::Drawing::Font(L"Montserrat Medium", 12,
static_cast<System::Drawing::FontStyle>((System::Drawing::FontStyle::Bold |
System::Drawing::FontStyle::Italic))));
        this->profileButton->Location =
System::Drawing::Point(786, 181);
        this->profileButton->Margin =
System::Windows::Forms::Padding(2);
        this->profileButton->Name = L"profileButton";
        this->profileButton->Size = System::Drawing::Size(265,
48);
        this->profileButton->TabIndex = 9;
        this->profileButton->Text = L"Профиль пользователя";
        this->profileButton->UseVisualStyleBackColor = false;
        this->profileButton->Click += gcnew
System::EventHandler(this, &MainWindow::profileButton_Click);
        //
        // breakfastButton
        //

```

```

        this->breakfastButton->BackColor =
System::Drawing::Color::LemonChiffon;
        this->breakfastButton->Font = (gcnew
System::Drawing::Font(L"Montserrat Medium", 12,
static_cast<System::Drawing::FontStyle>((System::Drawing::FontStyle::Bold |
System::Drawing::FontStyle::Italic))));
        this->breakfastButton->Location =
System::Drawing::Point(363, 119);
        this->breakfastButton->Margin =
System::Windows::Forms::Padding(2);
        this->breakfastButton->Name = L"breakfastButton";
        this->breakfastButton->Size = System::Drawing::Size(363,
84);

        this->breakfastButton->TabIndex = 12;
        this->breakfastButton->Text = L"Завтрак";
        this->breakfastButton->UseVisualStyleBackColor = false;
        this->breakfastButton->Click += gcnew
System::EventHandler(this, &MainWindow::breakfastButton_Click);
        //
        // lunchButton
        //
        this->lunchButton->BackColor =
System::Drawing::Color::SeaShell;
        this->lunchButton->Font = (gcnew
System::Drawing::Font(L"Montserrat Medium", 12,
static_cast<System::Drawing::FontStyle>((System::Drawing::FontStyle::Bold |
System::Drawing::FontStyle::Italic))));
        this->lunchButton->Location =
System::Drawing::Point(363, 229);
        this->lunchButton->Margin =
System::Windows::Forms::Padding(2);
        this->lunchButton->Name = L"lunchButton";
        this->lunchButton->Size = System::Drawing::Size(363,
92);

        this->lunchButton->TabIndex = 13;
        this->lunchButton->Text = L"Обед";
        this->lunchButton->UseVisualStyleBackColor = false;
        this->lunchButton->Click += gcnew
System::EventHandler(this, &MainWindow::lunchButton_Click);
        //
        // dinnerButton
        //
        this->dinnerButton->BackColor =
System::Drawing::Color::Gray;
        this->dinnerButton->Font = (gcnew
System::Drawing::Font(L"Montserrat Medium", 12,
static_cast<System::Drawing::FontStyle>((System::Drawing::FontStyle::Bold |
System::Drawing::FontStyle::Italic))));
        this->dinnerButton->Location =
System::Drawing::Point(363, 349);
        this->dinnerButton->Margin =
System::Windows::Forms::Padding(2);
        this->dinnerButton->Name = L"dinnerButton";
        this->dinnerButton->Size = System::Drawing::Size(363,
85);

        this->dinnerButton->TabIndex = 14;
        this->dinnerButton->Text = L"Ужин";
        this->dinnerButton->UseVisualStyleBackColor = false;
        this->dinnerButton->Click += gcnew
System::EventHandler(this, &MainWindow::dinnerButton_Click);
        //
        // button8

```

```

        //
        this->button8->BackColor =
System::Drawing::Color::Transparent;
        this->button8->Font = (gcnew
System::Drawing::Font(L"Montserrat Medium", 12,
static_cast<System::Drawing::FontStyle>((System::Drawing::FontStyle::Bold |
System::Drawing::FontStyle::Italic))));
        this->button8->Location = System::Drawing::Point(814,
465);
        this->button8->Margin =
System::Windows::Forms::Padding(2);
        this->button8->Name = L"button8";
        this->button8->Size = System::Drawing::Size(212, 43);
        this->button8->TabIndex = 18;
        this->button8->Text = L"ВЫХОД";
        this->button8->UseVisualStyleBackColor = false;
        this->button8->Click += gcnew System::EventHandler(this,
&MainWindow::exit);
        //
        // kkalEated
        //
        this->kkalEated->AutoSize = true;
        this->kkalEated->BackColor =
System::Drawing::Color::Aquamarine;
        this->kkalEated->Font = (gcnew
System::Drawing::Font(L"Montserrat Medium", 12,
static_cast<System::Drawing::FontStyle>((System::Drawing::FontStyle::Bold |
System::Drawing::FontStyle::Italic))));
        this->kkalEated->Location = System::Drawing::Point(341,
31);
        this->kkalEated->Margin =
System::Windows::Forms::Padding(2, 0, 2, 0);
        this->kkalEated->Name = L"kkalEated";
        this->kkalEated->Size = System::Drawing::Size(32, 33);
        this->kkalEated->TabIndex = 20;
        this->kkalEated->Text = L"0";
        this->kkalEated->Click += gcnew
System::EventHandler(this, &MainWindow::kkalEated_Click);
        //
        // kkalPerDay
        //
        this->kkalPerDay->AutoSize = true;
        this->kkalPerDay->BackColor =
System::Drawing::Color::GreenYellow;
        this->kkalPerDay->Font = (gcnew
System::Drawing::Font(L"Montserrat Medium", 12,
static_cast<System::Drawing::FontStyle>((System::Drawing::FontStyle::Bold |
System::Drawing::FontStyle::Italic))));
        this->kkalPerDay->Location = System::Drawing::Point(877,
31);
        this->kkalPerDay->Margin =
System::Windows::Forms::Padding(2, 0, 2, 0);
        this->kkalPerDay->Name = L"kkalPerDay";
        this->kkalPerDay->Size = System::Drawing::Size(149, 33);
        this->kkalPerDay->TabIndex = 69;
        this->kkalPerDay->Text = L"//калории";
        //
        // updateButton
        //
        this->updateButton->BackColor =
System::Drawing::Color::LavenderBlush;

```

```

        this->updateButton->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"updateButton.BackgroundImage")));
        this->updateButton->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::Stretch;
        this->updateButton->Location =
System::Drawing::Point(520, 23);
        this->updateButton->Name = L"updateButton";
        this->updateButton->Size = System::Drawing::Size(63,
58);

        this->updateButton->TabIndex = 70;
        this->updateButton->UseVisualStyleBackColor = false;
        this->updateButton->Click += gcnew
System::EventHandler(this, &MainWindow::updateButton_Click);
        //
        // MainWindow
        //
        this->AccessibleRole =
System::Windows::Forms::AccessibleRole::Window;
        this->AutoScaleDimensions = System::Drawing::SizeF(9,
20);

        this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
        this->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"$this.BackgroundImage")));
        this->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::Stretch;
        this->ClientSize = System::Drawing::Size(1063, 519);
        this->Controls->Add(this->updateButton);
        this->Controls->Add(this->kkalPerDay);
        this->Controls->Add(this->kkalEated);
        this->Controls->Add(this->button8);
        this->Controls->Add(this->dinnerButton);
        this->Controls->Add(this->lunchButton);
        this->Controls->Add(this->breakfastButton);
        this->Controls->Add(this->button2);
        this->Controls->Add(this->profileButton);
        this->Controls->Add(this->label2);
        this->Controls->Add(this->label1);
        this->FormBorderStyle =
System::Windows::Forms::FormBorderStyle::FixedSingle;
        this->Icon =
(cli::safe_cast<System::Drawing::Icon^>(resources-
>GetObject(L"$this.Icon")));
        this->Margin = System::Windows::Forms::Padding(2);
        this->Name = L"MainWindow";
        this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScreen;
        this->Text = L"Менеджер диеты";
        this->Load += gcnew System::EventHandler(this,
&MainWindow::MainWindow_Load);
        this->ResumeLayout(false);
        this->PerformLayout();

    }
#pragma endregion
    int fromStringToInt(string str) {
        int number = 0;
        int lenght = str.length();
        int k = 1;
        for (int i = 0; i < lenght - 1; i++) {

```

```

        k *= 10;
    }
    //сначала записываем старшие разряды
    for (int i = 0; str[i] != 0; i++) {
        number += (str[i] - '0') * k;
        k /= 10;
    }
    return number;
}
int userEatAction(int fl) {
    usersfood^ us = gcnw usersfood();
    us->Show();
    Ufl = fl; //
    static int k;
    int k1 = 0;
    int kkal2 = 0;
    if (user) user->Close();
    user = gcnw usersfood();
    this->AddOwnedForm(user);
    this->user->Ufl = Ufl;
    String^ kalorisel, ^ login_check, ^ product;
    int kkal;
    String^ connectionString =
"provider=Microsoft.Jet.OLEDB.4.0;Data Source=kkal.mdb";
    OleDbConnection^ dbConnection = gcnw
OleDbConnection(connectionString);
    dbConnection->Open();
    String^ query = "SELECT * FROM [Таблица1]";
    OleDbCommand^ dbCommand = gcnw OleDbCommand(query,
dbConnection);

    OleDbDataReader^ dbReader = dbCommand->ExecuteReader();
    while (dbReader->Read()) {

        login_check = (String^)dbReader["login"];
        Ulogin = gcnw String(global_login.data());

        if (login_check == Ulogin) {
            kalorisel = (String^)dbReader["калория"];
            std::string kkal1 =
msclr::interop::marshal_as<std::string>(kalorisel); // в системную строку из ^
            //int al = kkal1.length();
            kkal2 = fromStringToInt(kkal1);
            break;
        }
    }
    dbReader->Close();
    dbConnection->Close();
    //Кол-во набранных калорий
    int amountKalisel = 0;
    int z = 1;
    if (kalorisel != "") {
        connectionString =
"provider=Microsoft.Jet.OLEDB.4.0;Data Source=user_food.mdb";
        dbConnection = gcnw
OleDbConnection(connectionString);
        dbConnection->Open();
        query = "SELECT * FROM [Таблица1]";
        dbCommand = gcnw OleDbCommand(query,
dbConnection);

        dbReader = dbCommand->ExecuteReader();
        while (dbReader->Read()) {
            login_check = (String^)dbReader["login"];

```

```

        if (login_check == Ulogin) {
            product =
                kkal = (int)dbReader["kkal"];
                int id = (int)dbReader["id"];
                k = kkal;
                z = 0;
                amountKalorise += k;
        }
    }
    dbReader->Close();
    dbConnection->Close();
    return amountKalorise;
    String^ temp = amountKalorise.ToString();
    kkalEated->Text = temp; // Вывести кол-во
набранных калорий
    }
    private: System::Void exit(System::Object^ sender, System::EventArgs^
e) {
        Application::Exit();
    }
    private: System::Void profileButton_Click(System::Object^ sender,
System::EventArgs^ e) {
        settings^ pers = gcnew settings();
        pers->Show();
    }
    private: System::Void MainWindow_Load(System::Object^ sender,
System::EventArgs^ e) {
        String^ kalorise; // цель калорий в день
        String^ login_check;
        String^ connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data
Source=kkal.mdb";
        OleDbConnection^ dbConnection = gcnew
OleDbConnection(connectionString);
        dbConnection->Open();
        String^ query = "SELECT * FROM [Таблица1]";
        OleDbCommand^ dbCommand = gcnew OleDbCommand(query,
dbConnection);
        OleDbDataReader^ dbReader = dbCommand->ExecuteReader();
        while (dbReader->Read()) {
            login_check = (String^)dbReader["login"];
            if (login_check == Ulogin) {
                kalorise = (String^)dbReader["калория"];
                kkalPerDay->Text = kalorise;
                break;
            }
        }
        dbReader->Close();
        dbConnection->Close();
    }
    private: System::Void to_product(System::Object^ sender,
System::EventArgs^ e) {
        product^ p = gcnew product();
        p->Show();
    }
    private: System::Void breakfastButton_Click(System::Object^ sender,
System::EventArgs^ e) {
        int fl = 1;
        kkalEated->Text = userEatAction(fl).ToString();
    }
}

```

```

        private: System::Void lunchButton_Click(System::Object^ sender,
System::EventArgs^ e) {
            int fl = 2;
            userEatAction(fl);
        }
        private: System::Void dinnerButton_Click(System::Object^ sender,
System::EventArgs^ e) {
            int fl = 3;
            userEatAction(fl);
        }
        //Показать съеденную пользователем еду
        private: System::Void button6_Click(System::Object^ sender,
System::EventArgs^ e) {
            usersfood^ us = gcnew usersfood();
            us->Show();
            int fl = 4;
            Ufl = fl;
            if (user) user->Close();
            user = gcnew usersfood();
            this->AddOwnedForm(user);
            this->user->Ufl = Ufl;
        }
        private: System::Void kkalEated_Click(System::Object^ sender,
System::EventArgs^ e) {
        }
        private: System::Void updateButton_Click(System::Object^ sender,
System::EventArgs^ e) {
            int fl = 4;
        }
    };
}

```

\\ файл "MyForm.h"

```

#pragma once
#include "Registration.h"
#include "Authorization.h"
namespace DietManager {
    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
    using namespace System::Data::OleDb;
    public ref class MyForm : public System::Windows::Forms::Form
    {
    public:
        MyForm(void) {
            InitializeComponent();
        }
    protected:
        ~MyForm()
        {
            if (components)
            {
                delete components;
            }
        }
    private: System::Windows::Forms::Label^ label1;
    private: System::Windows::Forms::Button^ button2;
    private: System::Windows::Forms::Button^ button3;
    private: System::Windows::Forms::Button^ button1;
}

```



```

        private: System::ComponentModel::IContainer^ components;
    private:
#pragma region Windows Form Designer generated code
        void InitializeComponent(void) {
            System::ComponentModel::ComponentResourceManager^ resources
= (gcnew System::ComponentModel::ComponentResourceManager(MyForm::typeid));
            this->label1 = (gcnew System::Windows::Forms::Label());
            this->button2 = (gcnew System::Windows::Forms::Button());
            this->button3 = (gcnew System::Windows::Forms::Button());
            this->button1 = (gcnew System::Windows::Forms::Button());
            this->SuspendLayout();
            //
            // label1
            //
            this->label1->AutoSize = true;
            this->label1->FlatStyle =
System::Windows::Forms::FlatStyle::Popup;
            this->label1->Location = System::Drawing::Point(674, 97);
            this->label1->Name = L"label1";
            this->label1->Size = System::Drawing::Size(0, 24);
            this->label1->TabIndex = 2;
            //
            // button2
            //
            this->button2->AutoSize = true;
            this->button2->BackColor =
System::Drawing::SystemColors::ControlLight;
            this->button2->FlatAppearance->BorderColor =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_cast<System::Byte>(255)),

            static_cast<System::Int32>(static_cast<System::Byte>(224)),
            static_cast<System::Int32>(static_cast<System::Byte>(192))));
            this->button2->FlatStyle =
System::Windows::Forms::FlatStyle::Popup;
            this->button2->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 20,
System::Drawing::FontStyle::Italic));
            this->button2->Location = System::Drawing::Point(130, 46);
            this->button2->Margin = System::Windows::Forms::Padding(3,
4, 3, 4);

            this->button2->Name = L"button2";
            this->button2->Size = System::Drawing::Size(336, 90);
            this->button2->TabIndex = 4;
            this->button2->Text = L"Вход";
            this->button2->UseVisualStyleBackColor = false;
            this->button2->Click += gcnew System::EventHandler(this,
&MyForm::button2_Click);
            //
            // button3
            //
            this->button3->AutoSize = true;
            this->button3->BackColor =
System::Drawing::SystemColors::ControlLight;
            this->button3->FlatStyle =
System::Windows::Forms::FlatStyle::Popup;
            this->button3->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
            this->button3->Location = System::Drawing::Point(317, 280);
            this->button3->Margin = System::Windows::Forms::Padding(3,
4, 3, 4);

```

```

        this->button3->Name = L"button3";
        this->button3->Size = System::Drawing::Size(238, 64);
        this->button3->TabIndex = 5;
        this->button3->Text = L"Выйти";
        this->button3->UseVisualStyleBackColor = false;
        this->button3->Click += gcnew System::EventHandler(this,
&MyForm::button3_Click);
        //
        // button1
        //
        this->button1->AutoSize = true;
        this->button1->BackColor =
System::Drawing::SystemColors::ControlLight;
        this->button1->FlatAppearance->BorderColor =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_cast<System::Byte>(255)),
        static_cast<System::Int32>(static_cast<System::Byte>(224)),
static_cast<System::Int32>(static_cast<System::Byte>(192))));
        this->button1->FlatStyle =
System::Windows::Forms::FlatStyle::Popup;
        this->button1->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 20,
System::Drawing::FontStyle::Italic));
        this->button1->Location = System::Drawing::Point(130, 144);
        this->button1->Margin = System::Windows::Forms::Padding(3,
4, 3, 4);
        this->button1->Name = L"button1";
        this->button1->Size = System::Drawing::Size(336, 96);
        this->button1->TabIndex = 6;
        this->button1->Text = L"Регистрация";
        this->button1->UseVisualStyleBackColor = false;
        this->button1->Click += gcnew System::EventHandler(this,
&MyForm::button1_Click_2);
        //
        // MyForm
        //
        this->AccessibleRole =
System::Windows::Forms::AccessibleRole::Window;
        this->AutoScaleDimensions = System::Drawing::SizeF(10, 23);
        this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
        this->BackColor = System::Drawing::Color::AliceBlue;
        this->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::Stretch;
        this->ClientSize = System::Drawing::Size(567, 357);
        this->Controls->Add(this->button1);
        this->Controls->Add(this->button3);
        this->Controls->Add(this->button2);
        this->Controls->Add(this->label1);
        this->Font = (gcnew System::Drawing::Font(L"Poor Richard",
10.125F));
        this->Icon =
(cli::safe_cast<System::Drawing::Icon^>(resources-
>GetObject(L"$this.Icon")));
        this->Margin = System::Windows::Forms::Padding(3, 4, 3, 4);
        this->Name = L"MyForm";
        this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScreen;
        this->Text = L"Менеджер диеты";
        this->Load += gcnew System::EventHandler(this,
&MyForm::MyForm_Load);

```

```

        this->ResumeLayout(false);
        this->PerformLayout();
    }
#pragma endregion
    public: System::Void button2_Click(System::Object^ sender,
System::EventArgs^ e) {
        Authorization^ autoriz = gcnew Authorization(); // создаем новый
экземпляр
        autoriz->Show(); //авторизация
        MyForm::Hide(); //закреть
    }
    public: System::Void button3_Click(System::Object^ sender,
System::EventArgs^ e) {
        Application::Exit();
    }
    private: System::Void button1_Click_2(System::Object^ sender,
System::EventArgs^ e) {
        Registration^ registr = gcnew Registration(); // создаем новый
экземпляр
        registr->Show(); //регистрация
        MyForm::Hide(); //закреть
    }
    private: System::Void MyForm_Load(System::Object^ sender,
System::EventArgs^ e) {
    }
};
}

\\ файл "product.h"
#pragma once
namespace DietManager {
    using namespace System::Data::OleDb;
    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
    public ref class product : public System::Windows::Forms::Form {
    public:
        product(void) {
            productName = "";
            kkal = "";
            product_check = "";
            InitializeComponent();
        }
    protected:
        String^ productName;
        String^ kkal, ^ product_check;
    protected:
        ~product() {
            if (components) {
                delete components;
                delete productName;
                delete kkal;
                delete product_check;
            }
        }
    private: System::Windows::Forms::Label^ label4;
    protected:
    private: System::Windows::Forms::TextBox^ textBox1;

```

```

        private: System::Windows::Forms::TextBox^ textBox2;
        private: System::Windows::Forms::Button^ button1;
        private: System::ComponentModel::Container^ components;
#pragma region Windows Form Designer generated code
        void InitializeComponent(void) {
            System::ComponentModel::ComponentResourceManager^
resources = (gcnew
System::ComponentModel::ComponentResourceManager(product::typeid));
            this->label4 = (gcnew System::Windows::Forms::Label());
            this->textBox1 = (gcnew
System::Windows::Forms::TextBox());
            this->textBox2 = (gcnew
System::Windows::Forms::TextBox());
            this->button1 = (gcnew
System::Windows::Forms::Button());
            this->SuspendLayout();
            //
            // label4
            //
            this->label4->AutoSize = true;
            this->label4->BackColor = System::Drawing::Color::Lime;
            this->label4->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 12,
System::Drawing::FontStyle::Italic));
            this->label4->ForeColor =
System::Drawing::SystemColors::ActiveCaptionText;
            this->label4->Location = System::Drawing::Point(30, 26);
            this->label4->Margin =
System::Windows::Forms::Padding(2, 0, 2, 0);
            this->label4->Name = L"label4";
            this->label4->Size = System::Drawing::Size(665, 29);
            this->label4->TabIndex = 3;
            this->label4->Text = L"Введите название продукта и его
калорийность на 100г";
            this->label4->Click += gcnew System::EventHandler(this,
&product::label4_Click);
            //
            // textBox1
            //
            this->textBox1->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
            this->textBox1->Location = System::Drawing::Point(35,
98);
            this->textBox1->Margin =
System::Windows::Forms::Padding(2);
            this->textBox1->Name = L"textBox1";
            this->textBox1->Size = System::Drawing::Size(403, 28);
            this->textBox1->TabIndex = 4;
            //
            // textBox2
            //
            this->textBox2->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
            this->textBox2->Location = System::Drawing::Point(500,
98);
            this->textBox2->Margin =
System::Windows::Forms::Padding(2);
            this->textBox2->Name = L"textBox2";
            this->textBox2->Size = System::Drawing::Size(195, 28);
            this->textBox2->TabIndex = 5;

```

```

        //
        // button1
        //
        this->button1->BackColor =
System::Drawing::Color::Transparent;
        this->button1->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
        this->button1->Location = System::Drawing::Point(500,
158);
        this->button1->Margin =
System::Windows::Forms::Padding(2);
        this->button1->Name = L"button1";
        this->button1->Size = System::Drawing::Size(190, 43);
        this->button1->TabIndex = 6;
        this->button1->Text = L"Добавить";
        this->button1->UseVisualStyleBackColor = false;
        this->button1->Click += gcnew System::EventHandler(this,
&product::button1_Click);
        //
        // product
        //
        this->AutoScaleDimensions = System::Drawing::SizeF(9,
20);
        this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
        this->BackColor = System::Drawing::Color::AquaMarine;
        this->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::Stretch;
        this->ClientSize = System::Drawing::Size(757, 237);
        this->Controls->Add(this->button1);
        this->Controls->Add(this->textBox2);
        this->Controls->Add(this->textBox1);
        this->Controls->Add(this->label4);
        this->Icon =
(cli::safe_cast<System::Drawing::Icon^>(resources-
>GetObject(L"$this.Icon")));
        this->Margin = System::Windows::Forms::Padding(2);
        this->Name = L"product";
        this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScreen;
        this->Text = L"DietManager";
        this->Load += gcnew System::EventHandler(this,
&product::product_Load);
        this->ResumeLayout(false);
        this->PerformLayout();

    }
#pragma endregion
    private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e);
    private: System::Void product_Load(System::Object^ sender,
System::EventArgs^ e) {
    }
    private: System::Void label4_Click(System::Object^ sender,
System::EventArgs^ e) {
    }
};
}

```

\\ файл "productFunctions.h"

```

#pragma once
#include <Windows.h>
#include <iostream>
#include "product.h"

using namespace std;
using namespace DietManager;
using namespace::System::Data::OleDb;

System::Void DietManager::product::button1_Click(System::Object^ sender,
System::EventArgs^ e) {
    productName = textBox1->Text;
    kkal = textBox2->Text;
    int fl = 0;
    String^ connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data
Source=product.mdb";
    OleDbConnection^ dbConnection = gcnew
OleDbConnection(connectionString);
    dbConnection->Open();
    String^ query = "SELECT * FROM [Таблица1]";
    OleDbCommand^ dbCommand = gcnew OleDbCommand(query, dbConnection);
    OleDbDataReader^ dbReader = dbCommand->ExecuteReader();
    if (productName == "" || kkal == "") {
        fl = 1;
        MessageBox::Show("Не все поля заполнены, проверьте форму!",
"Ошибка!", MessageBoxButtons::OK);
        dbReader->Close();
        dbConnection->Close();
    }
    else {
        while (dbReader->Read()) {
            if ((String^)dbReader["Продукт"] != "") product_check =
(String^)dbReader["Продукт"];
            if (DBNull::Value != dbReader["Продукт"])product_check =
(String^)dbReader["Продукт"];
            if (product_check == productName) {
                MessageBox::Show("Продукт " + productName + " уже
есть в базе!", "", MessageBoxButtons::OK);
                fl = 1;
                dbReader->Close();
                dbConnection->Close();
                break;
            }
        }
    }
    //Добавление нового продукта:
    if (fl == 0) {
        String^ query1 = "INSERT INTO [Таблица1] (Продукт,калорийность
)VALUES ('" + productName + "' , '" + kkal + "')";
        OleDbCommand^ dbCommand1 = gcnew OleDbCommand(query1,
dbConnection);
        dbCommand1->ExecuteNonQuery();
        MessageBox::Show("Новый продукт " + productName + " добавлен!",
"", MessageBoxButtons::OK);
        dbReader->Close();
        dbConnection->Close();
        this->Close();
    }
}

\\ файл "Registration.h"

```

```

#pragma once
#include "regex"
#include "string"
#include "UserInformation.h"
#include "string.h"
#include <msclr\marshal_cppstd.h>

// #include "GlobalFunctions.h"

// using namespace DietManager;

namespace DietManager {
    using namespace System::Data::OleDb;
    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
    using namespace System::Runtime::InteropServices;
    using namespace std;
    public ref class Registration : public System::Windows::Forms::Form {
    public:
        Registration(void) {
            login = "";
            password = "";
            Ulogin = "";
            login_check = "";
            InitializeComponent();
        }
    protected:
        ~Registration() {
            if (components) {
                delete components;
                delete login;
                delete password;
                delete Ulogin;
                delete login_check;
            }
        }
    protected:
        String^ login, ^ password;
        String^ login_check;
        static String^ Ulogin;
    private:
        settings^ nastr;
        MainWindow^ persona;
        usersfood^ user;
    private: System::Windows::Forms::Label^ label2;
    protected:
    private: System::Windows::Forms::Label^ label1;
    private: System::Windows::Forms::TextBox^ textBox2;
    private: System::Windows::Forms::TextBox^ textBox1;
    private: System::Windows::Forms::Button^ button3;
    private: System::Windows::Forms::Button^ button2;

    private: System::ComponentModel::Container^ components;
#pragma region Windows Form Designer generated code
        void InitializeComponent(void) {

```

```

        System::ComponentModel::ComponentResourceManager^
resources = (gcnew
System::ComponentModel::ComponentResourceManager(Registration::typeid));
        this->label2 = (gcnew System::Windows::Forms::Label());
        this->label1 = (gcnew System::Windows::Forms::Label());
        this->textBox2 = (gcnew
System::Windows::Forms::TextBox());
        this->textBox1 = (gcnew
System::Windows::Forms::TextBox());
        this->button3 = (gcnew
System::Windows::Forms::Button());
        this->button2 = (gcnew
System::Windows::Forms::Button());
        this->SuspendLayout();
        //
        // label2
        //
        this->label2->AutoSize = true;
        this->label2->BackColor =
System::Drawing::Color::Transparent;
        this->label2->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
        this->label2->Location = System::Drawing::Point(85, 87);
        this->label2->Name = L"label2";
        this->label2->Size = System::Drawing::Size(67, 22);
        this->label2->TabIndex = 18;
        this->label2->Text = L"логин :";
        //
        // label1
        //
        this->label1->AutoSize = true;
        this->label1->BackColor =
System::Drawing::Color::Transparent;
        this->label1->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
        this->label1->Location = System::Drawing::Point(65,
150);
        this->label1->Name = L"label1";
        this->label1->Size = System::Drawing::Size(79, 22);
        this->label1->TabIndex = 17;
        this->label1->Text = L"пароль :";
        //
        // textBox2
        //
        this->textBox2->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
        this->textBox2->Location = System::Drawing::Point(158,
150);
        this->textBox2->Margin =
System::Windows::Forms::Padding(3, 4, 3, 4);
        this->textBox2->Name = L"textBox2";
        this->textBox2->>PasswordChar = '*';
        this->textBox2->Size = System::Drawing::Size(440, 28);
        this->textBox2->TabIndex = 23;
        //
        // textBox1
        //

```



```

        this->textBox1->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
        this->textBox1->Location = System::Drawing::Point(158,
84);
        this->textBox1->Margin =
System::Windows::Forms::Padding(3, 4, 3, 4);
        this->textBox1->Name = L"textBox1";
        this->textBox1->Size = System::Drawing::Size(440, 28);
        this->textBox1->TabIndex = 22;
        //
        // button3
        //
        this->button3->BackColor =
System::Drawing::SystemColors::Control;
        this->button3->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
        this->button3->Location = System::Drawing::Point(364,
199);
        this->button3->Margin =
System::Windows::Forms::Padding(3, 4, 3, 4);
        this->button3->Name = L"button3";
        this->button3->Size = System::Drawing::Size(234, 37);
        this->button3->TabIndex = 21;
        this->button3->Text = L"Выйти";
        this->button3->UseVisualStyleBackColor = false;
        this->button3->Click += gcnew System::EventHandler(this,
&Registration::exit);
        //
        // button2
        //
        this->button2->BackColor =
System::Drawing::SystemColors::Control;
        this->button2->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
        this->button2->Location = System::Drawing::Point(158,
196);
        this->button2->Margin =
System::Windows::Forms::Padding(3, 4, 3, 4);
        this->button2->Name = L"button2";
        this->button2->Size = System::Drawing::Size(200, 40);
        this->button2->TabIndex = 19;
        this->button2->Text = L"Регистрация";
        this->button2->UseVisualStyleBackColor = false;
        this->button2->Click += gcnew System::EventHandler(this,
&Registration::Registr);
        //
        // Registration
        //
        this->AutoScaleDimensions = System::Drawing::SizeF(10,
23);
        this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
        this->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::Stretch;
        this->ClientSize = System::Drawing::Size(724, 300);
        this->Controls->Add(this->label2);
        this->Controls->Add(this->label1);
        this->Controls->Add(this->textBox2);
        this->Controls->Add(this->textBox1);

```

```

        this->Controls->Add(this->button3);
        this->Controls->Add(this->button2);
        this->Cursor = System::Windows::Forms::Cursors::Arrow;
        this->Font = (gcnew System::Drawing::Font(L"Poor
Richard", 10.125F));
        this->FormBorderStyle =
System::Windows::Forms::FormBorderStyle::Fixed3D;
        this->Icon =
(cli::safe_cast<System::Drawing::Icon^>(resources-
>GetObject(L"$this.Icon")));
        this->Margin = System::Windows::Forms::Padding(3, 4, 3,
4);
        this->Name = L"Registration";
        this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScreen;
        this->Text = L"Менеджер диеты";
        this->Load += gcnew System::EventHandler(this,
&Registration::Registration_Load);
        this->ResumeLayout(false);
        this->PerformLayout();

    }
#pragma endregion
    private: System::Void exit(System::Object^ sender, System::EventArgs^
e) {
        Application::Exit();
    }
    private: System::Void Registr(System::Object^ sender,
System::EventArgs^ e) {
        login = textBox1->Text;
        password = textBox2->Text;
        int fl = 0;
        String^ connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data
Source=logpass.mdb";
        OleDbConnection^ dbConnection = gcnew
OleDbConnection(connectionString);
        dbConnection->Open();
        String^ query = "SELECT * FROM [Таблица1]";
        OleDbCommand^ dbCommand = gcnew OleDbCommand(query,
dbConnection);
        OleDbDataReader^ dbReader = dbCommand->ExecuteReader();
        if (login == "" || password == "") {
            fl = 1;
            MessageBox::Show("Не все поля заполнены, проверьте форму!",
"Ошибка!", MessageBoxButtons::OK);
            dbReader->Close();
            dbConnection->Close();
        }
        else {
            while (dbReader->Read()) {
                if ((String^)dbReader["log"] != "") login_check =
(String^)dbReader["log"];
                if (DBNull::Value != dbReader["log"]) login_check =
(String^)dbReader["log"];
                if (login_check == login) {
                    MessageBox::Show("Пользователь " + login + "
уже зарегистрирован!", "", MessageBoxButtons::OK);
                    fl = 1;
                    dbReader->Close();
                    dbConnection->Close();
                    break;
                }
            }
        }
    }
}

```

```

    }
    if (fl == 0) {
        String^ keylog = login; //получить логин пользователя
        //до завершения программы хранится логин пользователя
        global_login =
msclr::interop::marshal_as<std::string>(keylog);

        String^ query1 = "INSERT INTO [Таблица1] (log,pass )VALUES
('" + login + "' , '" + password + "')";
        OleDbCommand^ dbCommand1 = gcnew OleDbCommand(query1,
dbConnection);
        dbCommand1->ExecuteNonQuery();
        MessageBox::Show("Новый пользователь " + login + "
зарегистрирован!", "", MessageBoxButtons::OK);
        dbReader->Close();
        dbConnection->Close();
        connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data
Source=logpass.mdb";
        dbConnection = gcnew OleDbConnection(connectionString);
        dbConnection->Open();
        query = "SELECT * FROM [Таблица1]";
        dbCommand = gcnew OleDbCommand(query, dbConnection);
        OleDbDataReader^ dbReader1 = dbCommand->ExecuteReader();
        while (dbReader1->Read()) {
            if ((String^)dbReader1["log"] != "") login_check =
(String^)dbReader1["log"];

            if (DBNull::Value != dbReader1["log"]) {
                if (login_check == login) {
                    Ulogin = login;
                    dbReader1->Close();
                    dbConnection->Close();
                    if (nastr) nastr->Close();
                    nastr = gcnew settings();
                    this->AddOwnedForm(nastr);
                    if (persona) persona->Close();
                    persona = gcnew MainWindow();
                    this->AddOwnedForm(persona);
                    this->persona->Ulogin = Ulogin;
                    if (user) user->Close();
                    user = gcnew usersfood();
                    this->AddOwnedForm(user);
                    this->user->Ulogin = Ulogin; //передать
логин из регистрации в usersfood
                    break;
                }
            }
        }
        UserInformation^ pers = gcnew UserInformation();
        pers->Show();
        Registration::Hide();
    }
}

private: System::Void Registration_Load(System::Object^ sender,
System::EventArgs^ e) {
}
private: System::Void label3_Click(System::Object^ sender,
System::EventArgs^ e) {
}

```

```

};
}

\\ файл "session.h"
#pragma once
#include <string>
#include <iostream>
#include "stdlib.h"
#include "windows.h"

extern std::string global_login;

\\ файл "settings.h"
#pragma once
#include "session.h"

namespace DietManager {
    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
    using namespace System::Data::OleDb;

    public ref class settings : public System::Windows::Forms::Form {
    public:
        settings(void) {
            InitializeComponent();
            Ulogin = "";
            age_check = height_check = weight_check = end_weight_check
= name_check = name2_check = "";
        }
        static String^ Ulogin;
    protected:
        String^ name_check, ^ name2_check;
        String^ age_check, ^height_check, ^weight_check,
^end_weight_check;

    protected:
        ~settings() {
            if (components) {
                delete components;
                delete name_check;
                delete name2_check;
            }
        }
    protected:

    private: System::Windows::Forms::Button^ button4;
    private: System::Windows::Forms::Label^ label6;
    private: System::Windows::Forms::Label^ label4;
    private: System::Windows::Forms::Label^ label5;
    private: System::Windows::Forms::Label^ label3;

    private: System::Windows::Forms::Label^ label9;

```

```

private: System::Windows::Forms::Label^ label10;
private: System::Windows::Forms::Label^ label11;
private: System::Windows::Forms::Label^ label12;
private: System::Windows::Forms::Label^ label13;
private: System::Windows::Forms::Label^ label14;
private: System::Windows::Forms::Label^ label7;
private: System::ComponentModel::IContainer^ components;
private:
#pragma region Windows Form Designer generated code
    void InitializeComponent(void) {
        System::ComponentModel::ComponentResourceManager^ resources
= (gcnew System::ComponentModel::ComponentResourceManager(settings::typeid));
        this->button4 = (gcnew System::Windows::Forms::Button());
        this->label6 = (gcnew System::Windows::Forms::Label());
        this->label4 = (gcnew System::Windows::Forms::Label());
        this->label5 = (gcnew System::Windows::Forms::Label());
        this->label3 = (gcnew System::Windows::Forms::Label());
        this->label9 = (gcnew System::Windows::Forms::Label());
        this->label10 = (gcnew System::Windows::Forms::Label());
        this->label11 = (gcnew System::Windows::Forms::Label());
        this->label12 = (gcnew System::Windows::Forms::Label());
        this->label13 = (gcnew System::Windows::Forms::Label());
        this->label14 = (gcnew System::Windows::Forms::Label());
        this->label7 = (gcnew System::Windows::Forms::Label());
        this->SuspendLayout();
        //
        // button4
        //
        this->button4->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button4->Location = System::Drawing::Point(339, 669);
        this->button4->Margin = System::Windows::Forms::Padding(2);
        this->button4->Name = L"button4";
        this->button4->Size = System::Drawing::Size(206, 48);
        this->button4->TabIndex = 4;
        this->button4->Text = L"ВЫХОД";
        this->button4->UseVisualStyleBackColor = false;
        this->button4->Click += gcnew System::EventHandler(this,
&settings::exit);
        //
        // label6
        //
        this->label6->AutoSize = true;
        this->label6->BackColor =
System::Drawing::Color::Transparent;
        this->label6->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->label6->Location = System::Drawing::Point(306, 628);
        this->label6->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);
        this->label6->Name = L"label6";
        this->label6->Size = System::Drawing::Size(145, 22);
        this->label6->TabIndex = 22;
        this->label6->Text = L"Идеальный вес:";
        this->label6->TextAlign =
System::Drawing::ContentAlignment::MiddleRight;

```

```

        this->label6->Click += gcnew System::EventHandler(this,
&settings::label6_Click);
        //
        // label4
        //
        this->label4->AutoSize = true;
        this->label4->BackColor =
System::Drawing::Color::Transparent;
        this->label4->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->label4->Location = System::Drawing::Point(410, 606);
        this->label4->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);

        this->label4->Name = L"label4";
        this->label4->Size = System::Drawing::Size(41, 22);
        this->label4->TabIndex = 20;
        this->label4->Text = L"Bec";
        this->label4->TextAlign =
System::Drawing::ContentAlignment::MiddleRight;
        //
        // label5
        //
        this->label5->AutoSize = true;
        this->label5->BackColor =
System::Drawing::Color::Transparent;
        this->label5->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->label5->Location = System::Drawing::Point(402, 584);
        this->label5->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);

        this->label5->Name = L"label5";
        this->label5->Size = System::Drawing::Size(49, 22);
        this->label5->TabIndex = 18;
        this->label5->Text = L"Рoct";
        //
        // label3
        //
        this->label3->AutoSize = true;
        this->label3->BackColor =
System::Drawing::Color::Transparent;
        this->label3->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->label3->Location = System::Drawing::Point(373, 562);
        this->label3->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);

        this->label3->Name = L"label3";
        this->label3->Size = System::Drawing::Size(78, 22);
        this->label3->TabIndex = 16;
        this->label3->Text = L"Boзпact";
        this->label3->TextAlign =
System::Drawing::ContentAlignment::MiddleRight;
        this->label3->Click += gcnew System::EventHandler(this,
&settings::label3_Click);
        //
        // label9
        //

```

```

        this->label9->AutoSize = true;
        this->label9->BackColor =
System::Drawing::Color::Transparent;
        this->label9->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->label9->Location = System::Drawing::Point(21, 9);
        this->label9->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);

        this->label9->Name = L"label9";
        this->label9->Size = System::Drawing::Size(58, 22);
        this->label9->TabIndex = 31;
        this->label9->Text = L"label9";
        this->label9->Click += gcnew System::EventHandler(this,
&settings::label9_Click);
        //
        // label10
        //
        this->label10->AutoSize = true;
        this->label10->BackColor =
System::Drawing::Color::Transparent;
        this->label10->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->label10->Location = System::Drawing::Point(21, 31);
        this->label10->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);

        this->label10->Name = L"label10";
        this->label10->Size = System::Drawing::Size(68, 22);
        this->label10->TabIndex = 32;
        this->label10->Text = L"label10";
        this->label10->Click += gcnew System::EventHandler(this,
&settings::label10_Click);
        //
        // label11
        //
        this->label11->AutoSize = true;
        this->label11->BackColor =
System::Drawing::Color::Transparent;
        this->label11->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->label11->Location = System::Drawing::Point(465, 563);
        this->label11->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);

        this->label11->Name = L"label11";
        this->label11->Size = System::Drawing::Size(68, 22);
        this->label11->TabIndex = 33;
        this->label11->Text = L"label11";
        this->label11->Click += gcnew System::EventHandler(this,
&settings::label11_Click);
        //
        // label12
        //
        this->label12->AutoSize = true;
        this->label12->BackColor =
System::Drawing::Color::Transparent;

```

```

        this->label12->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->label12->Location = System::Drawing::Point(179, 384);
        this->label12->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);

        this->label12->Name = L"label12";
        this->label12->Size = System::Drawing::Size(0, 22);
        this->label12->TabIndex = 36;
        //
        // label13
        //
        this->label13->AutoSize = true;
        this->label13->BackColor =
System::Drawing::Color::Transparent;
        this->label13->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->label13->Location = System::Drawing::Point(465, 606);
        this->label13->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);

        this->label13->Name = L"label13";
        this->label13->Size = System::Drawing::Size(68, 22);
        this->label13->TabIndex = 35;
        this->label13->Text = L"label13";
        //
        // label14
        //
        this->label14->AutoSize = true;
        this->label14->BackColor =
System::Drawing::Color::Transparent;
        this->label14->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->label14->Location = System::Drawing::Point(465, 584);
        this->label14->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);

        this->label14->Name = L"label14";
        this->label14->Size = System::Drawing::Size(68, 22);
        this->label14->TabIndex = 34;
        this->label14->Text = L"label14";
        //
        // label7
        //
        this->label7->AutoSize = true;
        this->label7->BackColor =
System::Drawing::Color::Transparent;
        this->label7->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->label7->Location = System::Drawing::Point(465, 628);
        this->label7->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);

        this->label7->Name = L"label7";
        this->label7->Size = System::Drawing::Size(58, 22);
        this->label7->TabIndex = 37;
        this->label7->Text = L"label7";
        //

```



```

        // settings
        //
        this->AutoScaleDimensions = System::Drawing::SizeF(9, 19);
        this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
        this->AutoSize = true;
        this->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"$this.BackgroundImage")));
        this->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::Stretch;
        this->ClientSize = System::Drawing::Size(550, 728);
        this->Controls->Add(this->label7);
        this->Controls->Add(this->label12);
        this->Controls->Add(this->label13);
        this->Controls->Add(this->label14);
        this->Controls->Add(this->label11);
        this->Controls->Add(this->label10);
        this->Controls->Add(this->label9);
        this->Controls->Add(this->label6);
        this->Controls->Add(this->label4);
        this->Controls->Add(this->label5);
        this->Controls->Add(this->label3);
        this->Controls->Add(this->button4);
        this->Font = (gcnew System::Drawing::Font(L"Cambria", 8,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->ForeColor =
System::Drawing::SystemColors::ControlText;
        this->FormBorderStyle =
System::Windows::Forms::FormBorderStyle::FixedSingle;
        this->Icon =
(cli::safe_cast<System::Drawing::Icon^>(resources-
>GetObject(L"$this.Icon")));
        this->Margin = System::Windows::Forms::Padding(2);
        this->Name = L"settings";
        this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScreen;
        this->Text = L"персональная страница";
        this->Load += gcnew System::EventHandler(this,
&settings::settings_Load);
        this->ResumeLayout(false);
        this->PerformLayout();
    }
#pragma endregion
    private: System::Void exit(System::Object^ sender, System::EventArgs^
e) {
        Application::Exit();
    }
    private: System::Void settings_Load(System::Object^ sender,
System::EventArgs^ e) {
        String^ connectionString = "provider=Microsoft.Jet.OLEDB.4.0;Data
Source=information.mdb";
        OleDbConnection^ dbConnection = gcnew
OleDbConnection(connectionString);
        dbConnection->Open();//открываем соединение
        String^ login_check;
        String^ query = "SELECT * FROM [information]";// запрос
        OleDbCommand^ dbCommand = gcnew OleDbCommand(query,
dbConnection);//команда
        OleDbDataReader^ dbReader = dbCommand->ExecuteReader();//
считываем данным

```

```

        Ulogin = gcnew String(global_login.data());

        while (dbReader->Read()) {
            login_check = (String^)dbReader["login"];
            if (login_check == Ulogin) {
                name_check = (String^)dbReader["name"];
                name2_check = (String^)dbReader["name2"];
                age_check = (String^)dbReader["age"];
                height_check = (String^)dbReader["height"];
                weight_check = (String^)dbReader["weight"];
                end_weight_check = (String^)dbReader["end_weight"];
                break;
            }
        }
        //Вывод данных пользователя на экран
        label9->Text = name_check;
        label10->Text = name2_check;
        label11->Text = age_check;
        label14->Text = height_check;
        label13->Text = weight_check;
        label7->Text = end_weight_check;
        dbReader->Close();
        dbConnection->Close();
    }
    private: System::Void button5_Click(System::Object^ sender,
System::EventArgs^ e) {
    }
    private: System::Void label6_Click(System::Object^ sender,
System::EventArgs^ e) {
    }
private: System::Void label3_Click(System::Object^ sender, System::EventArgs^
e) {
}
private: System::Void label11_Click(System::Object^ sender,
System::EventArgs^ e) {
}
private: System::Void label9_Click(System::Object^ sender, System::EventArgs^
e) {
}
private: System::Void label10_Click(System::Object^ sender,
System::EventArgs^ e) {
}
};
}

```

```

\\ файл "UserInformation.h"
#pragma once
#include "MainWindow.h"
#include <string.h>
#include <msclr\marshal_cppstd.h>
#include <string>
#include <iostream>
#include "stdlib.h"
#include <cstring>
namespace DietManager {
    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
}

```

```

using namespace System::Data::OleDb;
using namespace msclr::interop;
using namespace std;
public ref class UserInformation : public System::Windows::Forms::Form
{
public:
    UserInformation(void) {
        InitializeComponent();
    }
private:
    bool gender = false; // Man - 0 Woman - 1
    bool isClickedSexButton = false;
private: System::Windows::Forms::Label^ label9;
private: System::Windows::Forms::Label^ label8;
protected:
    ~UserInformation() {
        if (components) {
            delete components;
        }
    }
    MainWindow^ persona;
private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::TextBox^ textBox1;
private: System::Windows::Forms::TextBox^ textBox2;
private: System::Windows::Forms::TextBox^ textBox3;
private: System::Windows::Forms::Label^ label3;
private: System::Windows::Forms::TextBox^ textBox4;
private: System::Windows::Forms::Label^ label4;
private: System::Windows::Forms::TextBox^ textBox5;
private: System::Windows::Forms::Label^ label5;
private: System::Windows::Forms::TextBox^ textBox6;
private: System::Windows::Forms::Label^ label6;
private: System::Windows::Forms::Button^ sexManButton;
private: System::Windows::Forms::Button^ sexWomanButton;
private: System::Windows::Forms::Label^ label7;
private: System::Windows::Forms::Button^ informationConfirmButton;
private:
    System::ComponentModel::Container^ components;
#pragma region Windows Form Designer generated code
    void InitializeComponent(void) {
        System::ComponentModel::ComponentResourceManager^ resources
= (gcnew
System::ComponentModel::ComponentResourceManager(UserInformation::typeid));
        this->label1 = (gcnew System::Windows::Forms::Label());
        this->label2 = (gcnew System::Windows::Forms::Label());
        this->textBox1 = (gcnew System::Windows::Forms::TextBox());
        this->textBox2 = (gcnew System::Windows::Forms::TextBox());
        this->textBox3 = (gcnew System::Windows::Forms::TextBox());
        this->label3 = (gcnew System::Windows::Forms::Label());
        this->textBox4 = (gcnew System::Windows::Forms::TextBox());
        this->label4 = (gcnew System::Windows::Forms::Label());
        this->textBox5 = (gcnew System::Windows::Forms::TextBox());
        this->label5 = (gcnew System::Windows::Forms::Label());
        this->textBox6 = (gcnew System::Windows::Forms::TextBox());
        this->label6 = (gcnew System::Windows::Forms::Label());
        this->sexManButton = (gcnew
System::Windows::Forms::Button());
        this->sexWomanButton = (gcnew
System::Windows::Forms::Button());
        this->label7 = (gcnew System::Windows::Forms::Label());

```

```

        this->informationConfirmButton = (gcnew
System::Windows::Forms::Button());
        this->label8 = (gcnew System::Windows::Forms::Label());
        this->label9 = (gcnew System::Windows::Forms::Label());
        this->SuspendLayout();
        //
        // label1
        //
        this->label1->AutoSize = true;
        this->label1->BackColor =
System::Drawing::Color::Transparent;
        this->label1->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 16,
System::Drawing::FontStyle::Italic));
        this->label1->Location = System::Drawing::Point(96, 151);
        this->label1->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);

        this->label1->Name = L"label1";
        this->label1->Size = System::Drawing::Size(90, 37);
        this->label1->TabIndex = 0;
        this->label1->Text = L"Имя*";
        //
        // label2
        //
        this->label2->AutoSize = true;
        this->label2->BackColor =
System::Drawing::Color::Transparent;
        this->label2->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 16,
System::Drawing::FontStyle::Italic));
        this->label2->Location = System::Drawing::Point(30, 211);
        this->label2->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);

        this->label2->Name = L"label2";
        this->label2->Size = System::Drawing::Size(156, 37);
        this->label2->TabIndex = 1;
        this->label2->Text = L"Фамилия*";
        //
        // textBox1
        //
        this->textBox1->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
        this->textBox1->Location = System::Drawing::Point(191,
151);
        this->textBox1->Margin =
System::Windows::Forms::Padding(2);
        this->textBox1->Name = L"textBox1";
        this->textBox1->Size = System::Drawing::Size(292, 28);
        this->textBox1->TabIndex = 2;
        //
        // textBox2
        //
        this->textBox2->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
        this->textBox2->Location = System::Drawing::Point(191,
211);
        this->textBox2->Margin =
System::Windows::Forms::Padding(2);
        this->textBox2->Name = L"textBox2";
        this->textBox2->Size = System::Drawing::Size(292, 28);

```

```

        this->textBox2->TabIndex = 3;
        //
        // textBox3
        //
        this->textBox3->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
        this->textBox3->Location = System::Drawing::Point(191,
275);

        this->textBox3->Margin =
System::Windows::Forms::Padding(2);
        this->textBox3->Name = L"textBox3";
        this->textBox3->Size = System::Drawing::Size(292, 28);
        this->textBox3->TabIndex = 5;
        //
        // label3
        //
        this->label3->AutoSize = true;
        this->label3->BackColor =
System::Drawing::Color::Transparent;
        this->label3->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 16,
System::Drawing::FontStyle::Italic));
        this->label3->Location = System::Drawing::Point(37, 267);
        this->label3->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);

        this->label3->Name = L"label3";
        this->label3->Size = System::Drawing::Size(149, 37);
        this->label3->TabIndex = 4;
        this->label3->Text = L"Бозпакт*";
        //
        // textBox4
        //
        this->textBox4->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
        this->textBox4->Location = System::Drawing::Point(191,
409);

        this->textBox4->Margin =
System::Windows::Forms::Padding(2);
        this->textBox4->Name = L"textBox4";
        this->textBox4->PasswordChar = '*';
        this->textBox4->Size = System::Drawing::Size(292, 28);
        this->textBox4->TabIndex = 9;
        //
        // label4
        //
        this->label4->AutoSize = true;
        this->label4->BackColor =
System::Drawing::Color::Transparent;
        this->label4->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 16,
System::Drawing::FontStyle::Italic));
        this->label4->Location = System::Drawing::Point(101, 409);
        this->label4->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);

        this->label4->Name = L"label4";
        this->label4->Size = System::Drawing::Size(83, 37);
        this->label4->TabIndex = 8;
        this->label4->Text = L"Бек*";
        //
        // textBox5

```

```

        //
        this->textBox5->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
        this->textBox5->Location = System::Drawing::Point(191,
345);
        this->textBox5->Margin =
System::Windows::Forms::Padding(2);
        this->textBox5->Name = L"textBox5";
        this->textBox5->Size = System::Drawing::Size(292, 28);
        this->textBox5->TabIndex = 7;
        //
        // label5
        //
        this->label5->AutoSize = true;
        this->label5->BackColor =
System::Drawing::Color::Transparent;
        this->label5->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 16,
System::Drawing::FontStyle::Italic));
        this->label5->Location = System::Drawing::Point(88, 336);
        this->label5->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);
        this->label5->Name = L"label5";
        this->label5->Size = System::Drawing::Size(98, 37);
        this->label5->TabIndex = 6;
        this->label5->Text = L"Почт*";
        //
        // textBox6
        //
        this->textBox6->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
        this->textBox6->Location = System::Drawing::Point(336,
464);
        this->textBox6->Margin =
System::Windows::Forms::Padding(2);
        this->textBox6->Name = L"textBox6";
        this->textBox6->Size = System::Drawing::Size(147, 28);
        this->textBox6->TabIndex = 11;
        //
        // label6
        //
        this->label6->AutoSize = true;
        this->label6->BackColor =
System::Drawing::Color::Transparent;
        this->label6->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 16,
System::Drawing::FontStyle::Italic));
        this->label6->Location = System::Drawing::Point(88, 464);
        this->label6->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);
        this->label6->Name = L"label6";
        this->label6->Size = System::Drawing::Size(232, 37);
        this->label6->TabIndex = 10;
        this->label6->Text = L"Идеальный вес";
        //
        // sexManButton
        //
        this->sexManButton->BackColor =
System::Drawing::Color::DeepSkyBlue;

```

```

        this->sexManButton->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
        this->sexManButton->Location = System::Drawing::Point(191,
97);
        this->sexManButton->Margin =
System::Windows::Forms::Padding(2);
        this->sexManButton->Name = L"sexManButton";
        this->sexManButton->Size = System::Drawing::Size(141, 41);
        this->sexManButton->TabIndex = 12;
        this->sexManButton->Text = L"M";
        this->sexManButton->UseVisualStyleBackColor = false;
        this->sexManButton->Click += gcnew
System::EventHandler(this, &UserInformation::sexManButton_Click);
        //
        // sexWomanButton
        //
        this->sexWomanButton->BackColor =
System::Drawing::Color::Pink;
        this->sexWomanButton->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
        this->sexWomanButton->Location =
System::Drawing::Point(336, 97);
        this->sexWomanButton->Margin =
System::Windows::Forms::Padding(2);
        this->sexWomanButton->Name = L"sexWomanButton";
        this->sexWomanButton->Size = System::Drawing::Size(147,
41);
        this->sexWomanButton->TabIndex = 13;
        this->sexWomanButton->Text = L"Ж";
        this->sexWomanButton->UseVisualStyleBackColor = false;
        this->sexWomanButton->Click += gcnew
System::EventHandler(this, &UserInformation::sexWomanButton_Click);
        //
        // label7
        //
        this->label7->AutoSize = true;
        this->label7->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic));
        this->label7->Location = System::Drawing::Point(135, 534);
        this->label7->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);
        this->label7->Name = L"label7";
        this->label7->Size = System::Drawing::Size(0, 22);
        this->label7->TabIndex = 14;
        //
        // informationConfirmButton
        //
        this->informationConfirmButton->BackColor =
System::Drawing::Color::Honeydew;
        this->informationConfirmButton->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 16,
System::Drawing::FontStyle::Italic));
        this->informationConfirmButton->Location =
System::Drawing::Point(191, 543);
        this->informationConfirmButton->Margin =
System::Windows::Forms::Padding(2);
        this->informationConfirmButton->Name =
L"informationConfirmButton";

```

```

        this->informationConfirmButton->Size =
System::Drawing::Size(292, 69);
        this->informationConfirmButton->TabIndex = 15;
        this->informationConfirmButton->Text = L"Подтвердить";
        this->informationConfirmButton->UseVisualStyleBackColor =
false;
        this->informationConfirmButton->Click += gcnew
System::EventHandler(this, &UserInformation::informationConfirmButton_Click);
        //
        // label8
        //
        this->label8->AutoSize = true;
        this->label8->BackColor =
System::Drawing::Color::Transparent;
        this->label8->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 16,
System::Drawing::FontStyle::Italic));
        this->label8->Location = System::Drawing::Point(101, 97);
        this->label8->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);
        this->label8->Name = L"label8";
        this->label8->Size = System::Drawing::Size(85, 37);
        this->label8->TabIndex = 19;
        this->label8->Text = L"Пол*";
        //
        // label9
        //
        this->label9->AutoSize = true;
        this->label9->BackColor =
System::Drawing::Color::Transparent;
        this->label9->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 8,
System::Drawing::FontStyle::Italic));
        this->label9->ForeColor = System::Drawing::Color::Red;
        this->label9->Location = System::Drawing::Point(187, 57);
        this->label9->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);
        this->label9->Name = L"label9";
        this->label9->Size = System::Drawing::Size(296, 20);
        this->label9->TabIndex = 20;
        this->label9->Text = L"* - поле обязательно для
заполнения";
        //
        // UserInformation
        //
        this->AutoScaleDimensions = System::Drawing::SizeF(9, 20);
        this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
        this->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::Stretch;
        this->ClientSize = System::Drawing::Size(571, 647);
        this->Controls->Add(this->label9);
        this->Controls->Add(this->label8);
        this->Controls->Add(this->informationConfirmButton);
        this->Controls->Add(this->label7);
        this->Controls->Add(this->sexWomanButton);
        this->Controls->Add(this->sexManButton);
        this->Controls->Add(this->textBox6);
        this->Controls->Add(this->label6);
        this->Controls->Add(this->textBox4);
        this->Controls->Add(this->label4);
        this->Controls->Add(this->textBox5);

```



```

        this->Controls->Add(this->label5);
        this->Controls->Add(this->textBox3);
        this->Controls->Add(this->label3);
        this->Controls->Add(this->textBox2);
        this->Controls->Add(this->textBox1);
        this->Controls->Add(this->label2);
        this->Controls->Add(this->label1);
        this->Icon =
(cli::safe_cast<System::Drawing::Icon^>(resources-
>GetObject(L"$this.Icon")));
        this->Margin = System::Windows::Forms::Padding(2);
        this->Name = L"UserInformation";
        this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScreen;
        this->Text = L"Менеджер диеты";
        this->Load += gcnew System::EventHandler(this,
&UserInformation::UserInformation_Load);
        this->ResumeLayout(false);
        this->PerformLayout();

    }
#pragma endregion
        //Нажатие на кнопку Подтвердить
        private: System::Void informationConfirmButton_Click(System::Object^
sender, System::EventArgs^ e);
        //Нажатие на кнопку выбора пола Мужской
        private: System::Void sexManButton_Click(System::Object^ sender,
System::EventArgs^ e);
        //Нажатие на кнопку выбора пола Женский
        private: System::Void sexWomanButton_Click(System::Object^ sender,
System::EventArgs^ e);

        private: System::Void UserInformation_Load(System::Object^ sender,
System::EventArgs^ e) {
    }
};
}

\\ файл "UserInformationFunctions.h"
#pragma once
#include "UserInformation.h"
#include "ActionsClass.h"

#include <Windows.h>
#include <iostream>

using namespace std;
using namespace DietManager;
using namespace::System::Data::OleDb;

bool isClickedSexButton = false;

        //Нажатие на кнопку Подтвердить
        System::Void
DietManager::UserInformation::informationConfirmButton_Click(System::Object^
sender, System::EventArgs^ e) {
    if (isClickedSexButton) {
        AddAction AddRec1(textBox1->Text->ToString(), textBox2-
>Text->ToString(), textBox3->Text->ToString(),

```

```

        textBox5->Text->ToString(), textBox4->Text-
>ToString(), textBox6->Text->ToString(), gender);
        AddRecl.AddRecord();
    }
    MainWindow^ registr = gcnew MainWindow();// создаем новый
экземпляр
    registr->Show();// открыть регистрацию
    UserInformation::Hide();// скрыть элемент управления
}

//Нажатие на кнопку выбора пола Мужской
System::Void
DietManager::UserInformation::sexManButton_Click(System::Object^ sender,
System::EventArgs^ e) {
    gender = 0;
    isClickedSexButton = true; // Для проверки нажатия
}
//Нажатие на кнопку выбора пола Женский
System::Void
DietManager::UserInformation::sexWomanButton_Click(System::Object^ sender,
System::EventArgs^ e) {
    gender = 1; // Флаг определения пола = woman
    isClickedSexButton = true;
}

```

```

\\ файл "usersfood.h"
#pragma once
#include <msclr\marshal_cppstd.h>
#include "regex"
#include <string>
#include "iostream"
#include "stdlib.h"
#include "session.h"
namespace DietManager {
    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
    using namespace msclr::interop;
    using namespace System::Data::OleDb;
    public ref class usersfood : public System::Windows::Forms::Form {
    public:
        usersfood(void) {
            food = kkal = food_check = "";
            InitializeComponent();
        }
        static int Ufl;
        String^ Ulogin;
    protected:
        String^ food;
        String^ kkal;
        String^ gram;

    private: System::Windows::Forms::TextBox^ textBox2;
    protected:
    private: System::Windows::Forms::Label^ label2;
    private: System::Windows::Forms::Label^ label3;
        String^ food_check;
    protected:

```

```

        ~usersfood() {
            if (components)
            {
                delete components;
            }
        }
private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::Button^ FindButton;
protected:
private: System::Windows::Forms::Button^ AddFoodButton;

private: System::Windows::Forms::TextBox^ textBox1; private:
System::Windows::Forms::Button^ button3;
private:
    System::ComponentModel::Container^ components;

#pragma region Windows Form Designer generated code
    void InitializeComponent(void) {
        System::ComponentModel::ComponentResourceManager^ resources
= (gcnew
System::ComponentModel::ComponentResourceManager(usersfood::typeid));
        this->label1 = (gcnew System::Windows::Forms::Label());
        this->FindButton = (gcnew
System::Windows::Forms::Button());
        this->textBox1 = (gcnew System::Windows::Forms::TextBox());
        this->AddFoodButton = (gcnew
System::Windows::Forms::Button());
        this->textBox2 = (gcnew System::Windows::Forms::TextBox());
        this->label2 = (gcnew System::Windows::Forms::Label());
        this->label3 = (gcnew System::Windows::Forms::Label());
        this->SuspendLayout();
        //
        // label1
        //
        this->label1->AutoSize = true;
        this->label1->BackColor =
System::Drawing::Color::LightPink;
        this->label1->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 14,
System::Drawing::FontStyle::Italic));
        this->label1->Location = System::Drawing::Point(110, 57);
        this->label1->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);

        this->label1->Name = L"label1";
        this->label1->Size = System::Drawing::Size(353, 32);
        this->label1->TabIndex = 0;
        this->label1->Text = L"Поиск продуктов питания";
        this->label1->Click += gcnew System::EventHandler(this,
&usersfood::label1_Click);
        //
        // FindButton
        //
        this->FindButton->AutoSize = true;
        this->FindButton->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
        this->FindButton->Location = System::Drawing::Point(98,
222);

        this->FindButton->Margin =
System::Windows::Forms::Padding(2);
        this->FindButton->Name = L"FindButton";

```

```

        this->FindButton->Size = System::Drawing::Size(166, 42);
        this->FindButton->TabIndex = 1;
        this->FindButton->Text = L"Искать";
        this->FindButton->UseVisualStyleBackColor = true;
        this->FindButton->Click += gcnew System::EventHandler(this,
&usersfood::FindButton_Click);
        //
        // textBox1
        //
        this->textBox1->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->textBox1->Location = System::Drawing::Point(27, 159);
        this->textBox1->Margin =
System::Windows::Forms::Padding(2);
        this->textBox1->Name = L"textBox1";
        this->textBox1->Size = System::Drawing::Size(307, 28);
        this->textBox1->TabIndex = 2;
        //
        // AddFoodButton
        //
        this->AddFoodButton->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->AddFoodButton->Location = System::Drawing::Point(303,
222);
        this->AddFoodButton->Margin =
System::Windows::Forms::Padding(2);
        this->AddFoodButton->Name = L"AddFoodButton";
        this->AddFoodButton->Size = System::Drawing::Size(160, 42);
        this->AddFoodButton->TabIndex = 4;
        this->AddFoodButton->Text = L"Добавить";
        this->AddFoodButton->UseVisualStyleBackColor = true;
        this->AddFoodButton->Click += gcnew
System::EventHandler(this, &usersfood::AddFoodButton_Click);
        //
        // textBox2
        //
        this->textBox2->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->textBox2->Location = System::Drawing::Point(358,
159);
        this->textBox2->Margin =
System::Windows::Forms::Padding(2);
        this->textBox2->Name = L"textBox2";
        this->textBox2->Size = System::Drawing::Size(136, 28);
        this->textBox2->TabIndex = 5;
        //
        // label2
        //
        this->label2->AutoSize = true;
        this->label2->BackColor =
System::Drawing::Color::Transparent;
        this->label2->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->label2->Location = System::Drawing::Point(354, 135);

```

```

        this->label2->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);
        this->label2->Name = L"label2";
        this->label2->Size = System::Drawing::Size(132, 22);
        this->label2->TabIndex = 6;
        this->label2->Text = L"Macca (грамм)";
        this->label2->Click += gcnew System::EventHandler(this,
&usersfood::label2_Click);
        //
        // label3
        //
        this->label3->AutoSize = true;
        this->label3->BackColor =
System::Drawing::Color::Transparent;
        this->label3->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
        this->label3->Location = System::Drawing::Point(23, 135);
        this->label3->Margin = System::Windows::Forms::Padding(2,
0, 2, 0);
        this->label3->Name = L"label3";
        this->label3->Size = System::Drawing::Size(80, 22);
        this->label3->TabIndex = 7;
        this->label3->Text = L"Продукт";
        this->label3->Click += gcnew System::EventHandler(this,
&usersfood::label3_Click);
        //
        // usersfood
        //
        this->AutoScaleDimensions = System::Drawing::SizeF(9, 20);
        this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
        this->BackColor = System::Drawing::Color::LavenderBlush;
        this->ClientSize = System::Drawing::Size(554, 308);
        this->Controls->Add(this->label3);
        this->Controls->Add(this->label2);
        this->Controls->Add(this->textBox2);
        this->Controls->Add(this->AddFoodButton);
        this->Controls->Add(this->textBox1);
        this->Controls->Add(this->FindButton);
        this->Controls->Add(this->label1);
        this->Icon =
(cli::safe_cast<System::Drawing::Icon^>(resources-
>GetObject(L"$this.Icon")));
        this->Margin = System::Windows::Forms::Padding(2);
        this->Name = L"usersfood";
        this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScreen;
        this->Load += gcnew System::EventHandler(this,
&usersfood::usersfood_Load);
        this->ResumeLayout(false);
        this->PerformLayout();

    }
#pragma endregion
    private: System::Void FindButton_Click(System::Object^ sender,
System::EventArgs^ e);
    private: System::Void AddFoodButton_Click(System::Object^ sender,
System::EventArgs^ e);
    private: System::Void usersfood_Load(System::Object^ sender,
System::EventArgs^ e) {

```

```

    }
    private: System::Void label1_Click(System::Object^ sender,
System::EventArgs^ e) {
    }
private: System::Void label3_Click(System::Object^ sender, System::EventArgs^
e) {
}
private: System::Void label2_Click(System::Object^ sender, System::EventArgs^
e) {
}
};
}

\\ файл "usersfoodFunctions.h"
#pragma once
#include "usersfood.h"
#include "ActionsClass.h"

#include <Windows.h>
#include <iostream>

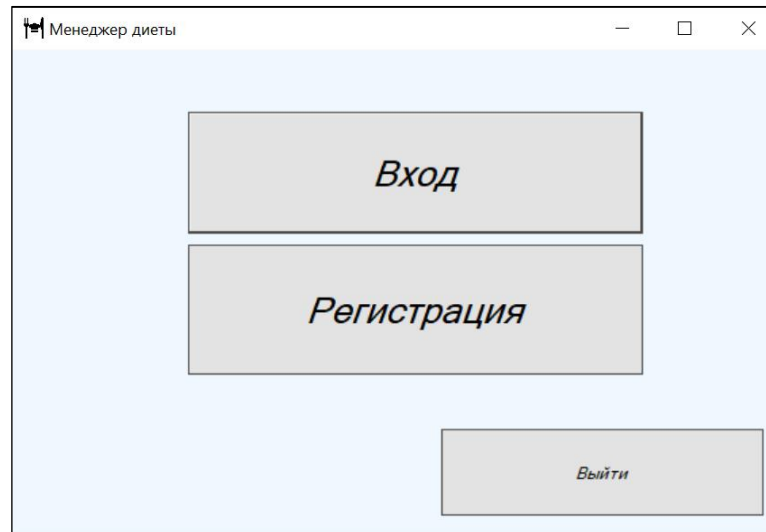
using namespace std;
using namespace DietManager;
using namespace::System::Data::OleDb;

System::Void DietManager::usersfood::FindButton_Click(System::Object^ sender,
System::EventArgs^ e) {
    FindFood search(textBox1->Text->ToString()); //создаём объект поиска
    search.findFood(); //ищем по таблице
}

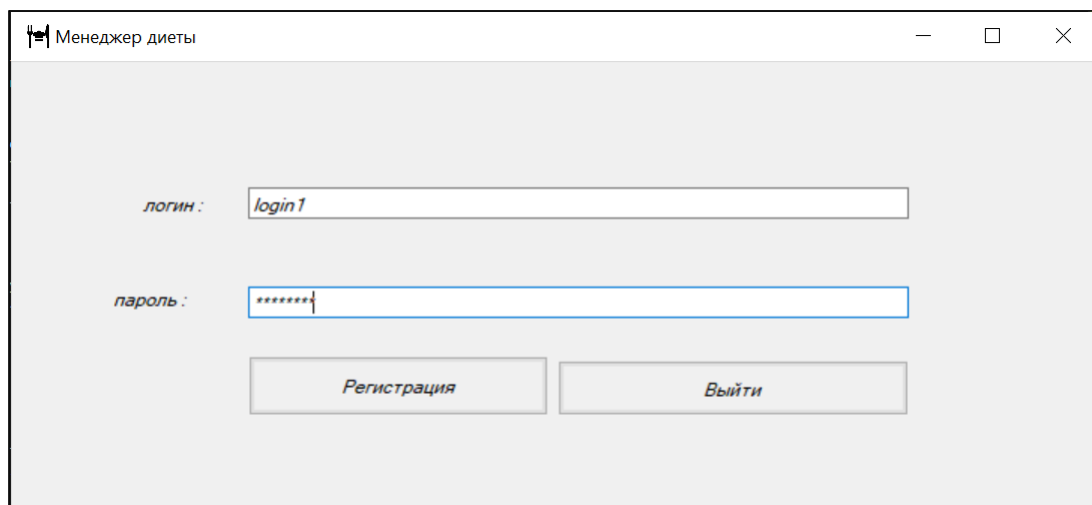
System::Void DietManager::usersfood::AddFoodButton_Click(System::Object^
sender, System::EventArgs^ e) {
    int fl = Ufl; // определяет приём пищи
    AddFood toAdd(textBox1->Text->ToString(), textBox2->Text->ToString(),
fl);
    toAdd.addFood();
}

```

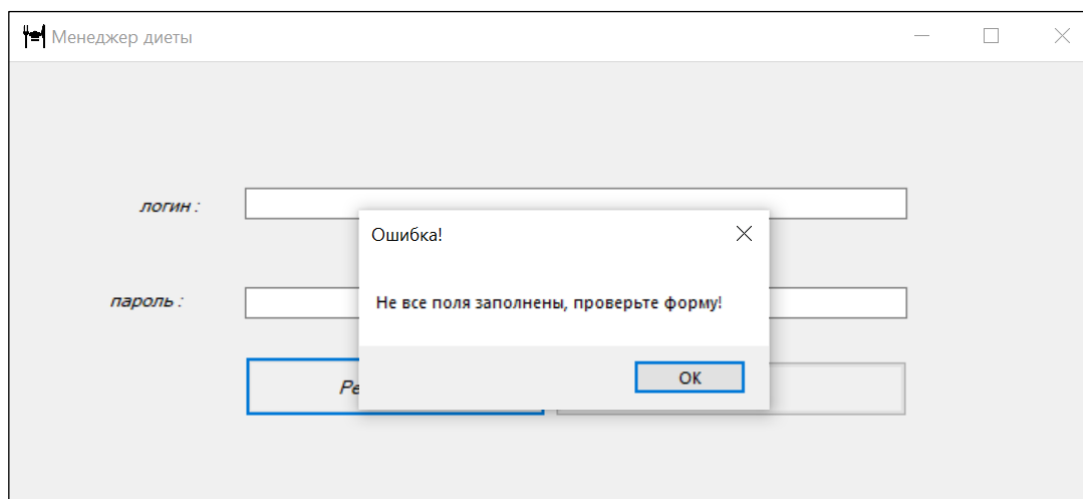
**ПРИЛОЖЕНИЕ Б**  
**(обязательное)**  
**Скриншоты работы программы**



Б.1 — Начальное окно регистрации и авторизации



Б.2 — Окно регистрации



Б.3 — Окно вывода ошибки регистрации

The image shows the "Менеджер диеты" (Diet Manager) application window. At the top, there is a red note: "\* - поле обязательно для заполнения" (\* - field is mandatory for filling). The form contains the following fields:

- Пол\*** (Gender\*): Two buttons, "М" (Male) in a blue box and "Ж" (Female) in a pink box. "М" is selected.
- Имя\*** (Name\*): A text input field containing "Name".
- Фамилия** (Surname): A text input field containing "Subname".
- Возраст\*** (Age\*): A text input field containing "33".
- Рост\*** (Height\*): A text input field containing "189".
- Вес\*** (Weight\*): A text input field containing "\*\*".
- Идеальный вес** (Ideal weight): An empty text input field.

At the bottom of the form is a large green button labeled "Подтвердить" (Confirm).

Б.4 — Окно заполнения пользовательских данных



Менеджер диеты

\* - поле обязательно для заполнения

Пол\* ☐ М ☐ Ж

Имя\*

Фамилия

Возраст\*

Ошибка!

Ваши данные неправильны! Проверьте их!

ОК

Подтвердить

Б.5 — Окно вывода ошибки заполнения данных

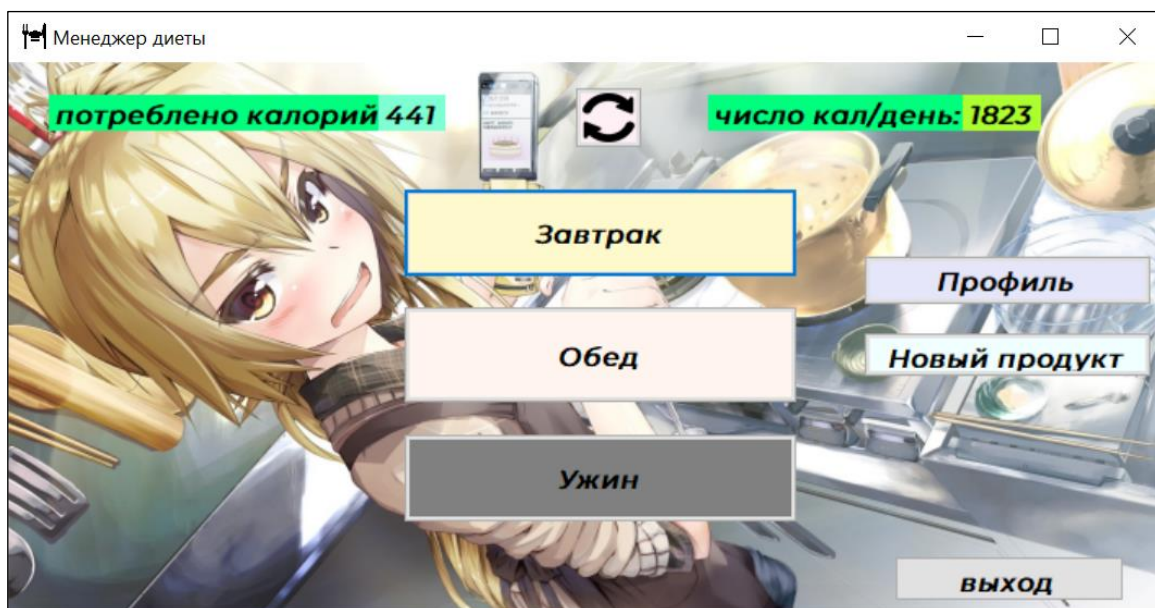
Менеджер диеты

логин :

пароль :

Авторизация Выйти

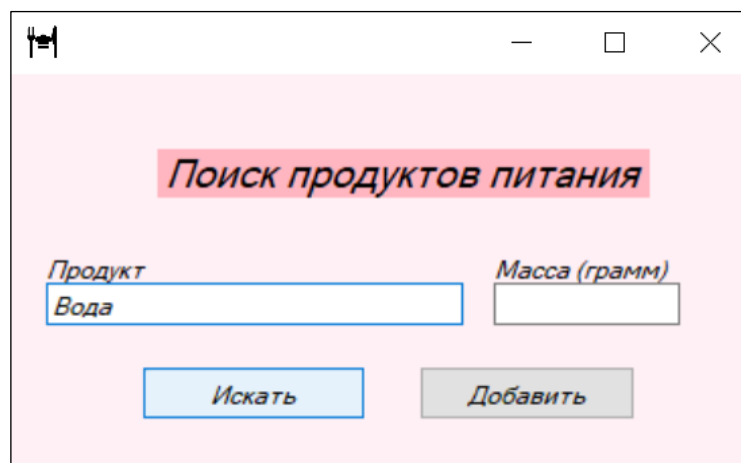
Б.6 — Окно авторизации



Б.7 — Главное окно интерфейса программы



Б.8 — Окно профиля пользователя



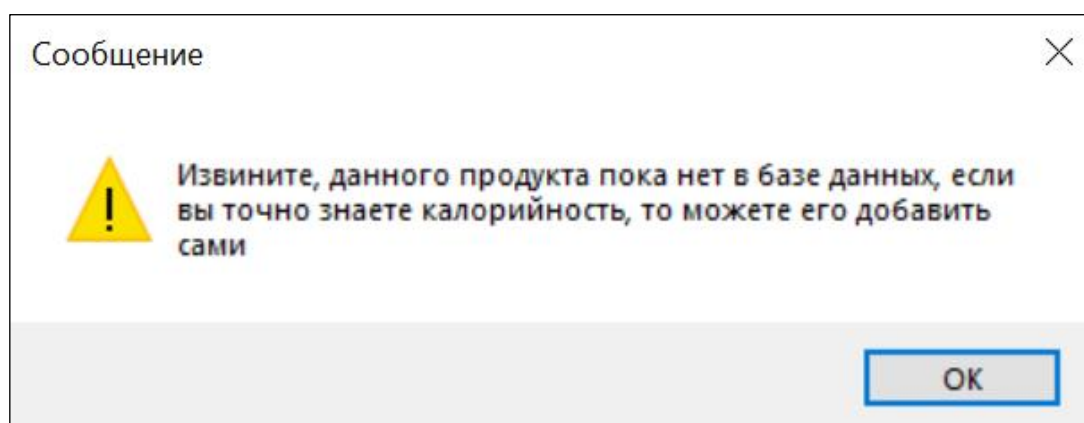
**Поиск продуктов питания**

Продукт: Вода

Масса (грамм):

Искать Добавить

Б.9 — Окно поиска и добавления продуктов

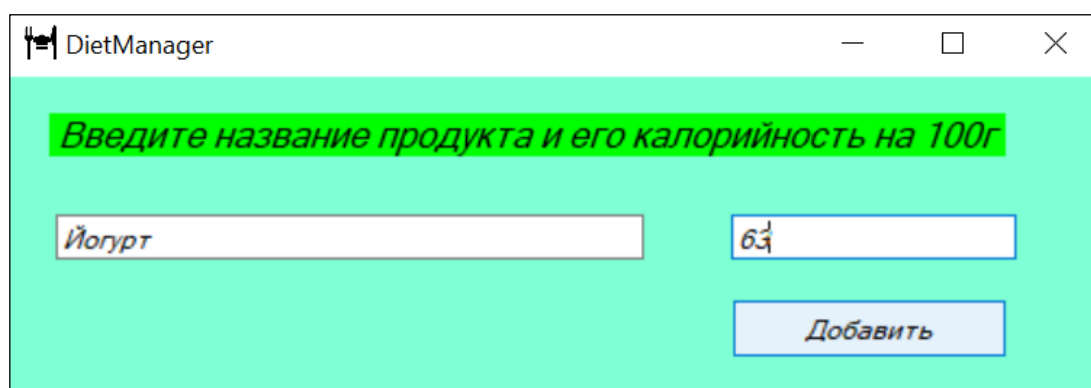


Сообщение

Извините, данного продукта пока нет в базе данных, если вы точно знаете калорийность, то можете его добавить сами

OK

Б.10 — Окно-сообщение ненайденного продукта



DietManager

**Введите название продукта и его калорийность на 100г**

Йогурт 63

Добавить

Б.11 — Окно добавления нового продукта в программу

**ПРИЛОЖЕНИЕ В**  
**(обязательное)**  
**UML-диаграмма классов программного средства**



## **ПРИЛОЖЕНИЕ Г**

**(обязательное)**

**Блок-схемы алгоритмов функций `MainWindow_Load()`,  
`fromStringToInt()`, `insertCalories()`**







**ПРИЛОЖЕНИЕ Д**  
**(обязательное)**  
**Ведомость документов**