

**ЧАСТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«ЛИЦЕЙ-ИНТЕРНАТ ЕСТЕСТВЕННЫХ НАУК»
Г. САРАТОВА**

**международная научно-практическая конференция
«ОТ ШКОЛЬНОГО ПРОЕКТА — К ПРОФЕССИОНАЛЬНОЙ КАРЬЕРЕ»
СЕКЦИЯ «В мире информатики и информационных технологий»**

**ПРИЛОЖЕНИЕ С ГРАФИЧЕСКИМ ИНТЕРФЕЙСОМ PyQt5
«Smart Food Calendar (SF Calendar)»**

Автор работы:

учащийся 9-1 класса

МАОУ “Физико-технический лицей №1”

Швецов Егор

Научный руководитель:

учитель информатики

МАОУ “Физико-технический лицей №1”

Рахманова Мария Николаевна

САРАТОВ, 2022

Оглавление

1. Введение	3
2. Основная часть	4
2.1 Выбор библиотеки для создания графического интерфейса	4
2.2 Разработка структуры приложения	4
2.3 Разработка БД	5
2.4 Разработка макетов окон	6
2.5 Выведение формул расчетов	7
2.6 Навигация по приложению.	8
3. Заключение.....	10
4. Список используемых ресурсов.....	11

1. Введение

Smart Food Calendar – приложение, помогающее следить за своим рационом питания. Производит автоматизированный поиск по базе данных питательных веществ различных продуктов, хранит данные дневника питания, приводит статистику рациона питания, дает необходимые советы по его корректировке.

Идея создания такого приложения появилась у меня из-за потребности следить за своим рационом питания. Как известно, секрет здоровья нашего организма заключается в продуктах, которые мы употребляем в пищу. Но как уследить за тем, что мы съели? Записывать в блокнот? – Неудобно, да и к тому же, как обработать на бумаге большое количество информации?! Тут и приходят на помощь использование информационных технологий.

Цель:

Написание приложения, способного выполнять следующие функции:

- поиск энергетической ценности продуктов,
- хранение дневника питания в удобном для пользователя формате,
- редактирование данных,
- анализ дневника питания, вывод советов по корректировке рациона питания.

Задачи, которые в течение проекта мне пришлось решить:

- Выбор языка программирования и библиотеки для создания графического интерфейса,
- Проектирование структуры приложения,
- Выведение формул расчетов норм содержания белков, жиров и углеводов,
- Поиск подходящих средств-виджетов для вывода или получения различной информации,
- Разработка макетов окон с интерфейсом,
- Выбор типа базы данных, содержащей информацию об энергетической ценности конкретных продуктов, и ее проектирование.

2. Основная часть

2.1 Выбор библиотеки для создания графического интерфейса

Для языка программирования Python есть много способов создания приложений с графическим интерфейсом, в частности, уже знакомая мне библиотека tkinter. Она используется в большом числе кроссплатформенных приложений, написанных на Python. В этом проекте я буду рассматривать библиотеку PyQt5, так как ее возможности значительно богаче.

Для реализаций функций приложения мне понадобилось изучить следующие технологии:

- Возможности библиотеки PyQt5 для создания графического интерфейса
- Работа с датами с использованием библиотеки datetime.
- Несколько форм, их взаимодействие между собой
- Стандартный диалог получения информации, вручную написанный класс диалогового окна.
- Работа с файловыми структурами в python
- Работа с реляционными базами данных, освоение языка запросов SQL
- Создание Exe-пакета, README.md, requirements.txt

2.2 Разработка структуры приложения

Для создания взаимодействия между объектами интерфейса использовалась технология ООП. Программа реализована через классы окон, отвечающих за различные функции.

Спроектированные классы:

1. MainWindow,
2. PersonalDialog,
3. Cabinet,
4. Statistic,
5. HelloScreen,
6. Info.

Для хранения дневника питания и информации о продуктах было принято решение создать две базы данных.

Персональная информация и рекомендуемые продукты сохраняются в текстовых файлах.

Код класса HelloScreen (приветственного окна) для примера:

```
class HelloScreen(QWidget, Ui_helloscreen):
    def __init__(self, other, SCREEN_SIZE):
        super().__init__()
        self.parent = other
        self.SCREEN_SIZE = SCREEN_SIZE
        self.setupUi(self)
        LOADING(self)
        self.setGeometry((SCREEN_SIZE[0] - 700) // 2,
                        (SCREEN_SIZE[1] - 500) // 2, 700, 500)
        self.label_4.resize(41, 41)
        self.pushButton.clicked.connect(self.dialog)

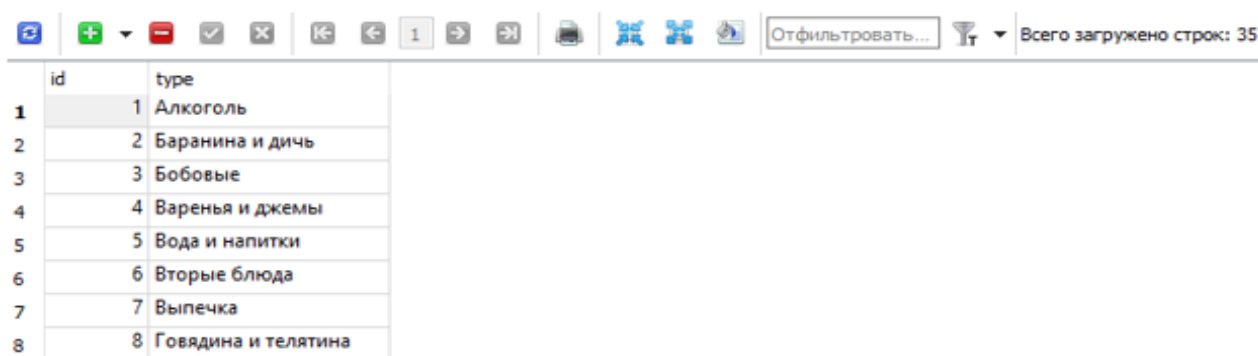
    def dialog(self):
        """
        Запрос личной информации, если пользователь подтвердил действие,
        то открытие рабочего стола
        """
        self.d = PersonalDialog(self, self.SCREEN_SIZE)
        self.d.exec_()
        if self.d.isHidden() and self.d.accepted:
            file = open('personal_data.txt', 'r', encoding='utf-8')
            info = [x.strip().split() for x in file.readlines()]
            file.close()
            self.hide()
            self.parent.workspace(info[0][0])
```

2.3 Разработка БД

В процессе разработки приложения мне пришлось проектировать и работать с реляционными базами данных sqlite3, освоить язык запросов SQL: чтение, добавление, обновление, удаление данных.

Сначала самой полной мне показалась база продуктов Министерства сельского хозяйства США. Но все названия там были на английском языке. Мне это не подходило. Тогда я принял решение составлять базу данных собственноручно, основываясь на данных с сайта «Мой здоровый рацион». Все продукты были разбиты на категории для удобства поиска. Связь разделов осуществляется по id между двумя таблицами в БД.

Рисунок 1 Структура таблицы категорий в БД



	id	type
1	1	Алкоголь
2	2	Баранина и дичь
3	3	Бобовые
4	4	Варенья и джемы
5	5	Вода и напитки
6	6	Вторые блюда
7	7	Выпечка
8	8	Говядина и телятина

Рисунок 2 Структура таблицы продуктов в БД

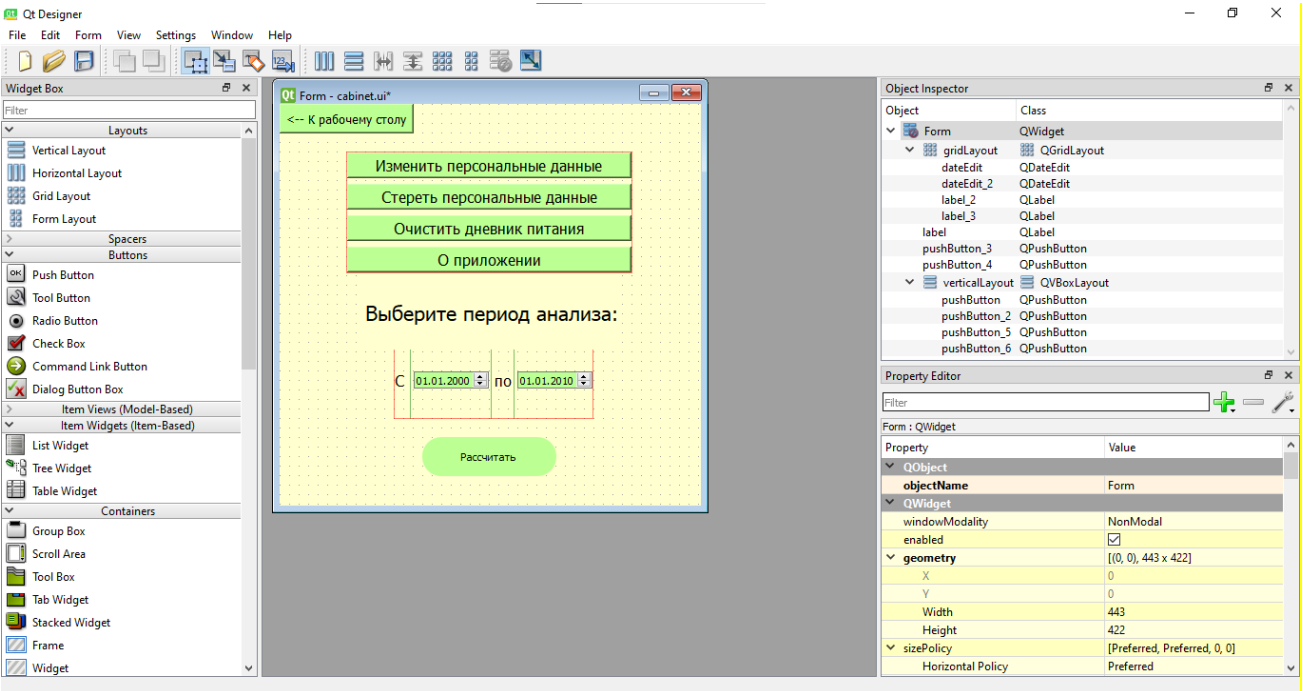
	index	product	kilocalories	protein	fats	carbohydrate	type	weight
1	0	LOMA LINDA Большие сосиски, низкожирные, консервированные, неприготовленные	154 кКал	23,1 г	4,7 г	4,9 г	3	100
2	1	MORI-NU, Тофу, мягкий, шелковый	55 кКал	4,8 г	2,7 г	2,9 г	3	100
3	2	MORI-NU, Тофу, твердый, шелковый	62 кКал	6,9 г	2,7 г	2,4 г	3	100
4	3	MORI-NU, Тофу, экстра твердый, шелковый	55 кКал	7,4 г	1,9 г	2 г	3	100
5	4	Арахис	552 кКал	26,3 г	45,2 г	9,9 г	3	100
6	5	Арахис	567 кКал	25,8 г	49,24 г	7,63 г	3	100
7	6	Арахис валенсия	570 кКал	25,09 г	47,58 г	12,21 г	3	100
8	7	Арахис валенсия жаренный на масле, без соли	589 кКал	27,04 г	51,24 г	7,4 г	3	100
9	8	Арахис валенсия жаренный на масле, с солью	589 кКал	27,04 г	51,24 г	7,4 г	3	100
10	9	Арахис варенный, с солью	318 кКал	13,5 г	22,01 г	12,46 г	3	100
11	10	Арахис вирджиния	563 кКал	25,19 г	48,75 г	8,04 г	3	100
12	11	Арахис вирджиния жаренный на масле, без соли	578 кКал	25,87 г	48,62 г	10,96 г	3	100
13	12	Арахис вирджиния, жаренный на масле, с солью	578 кКал	25,87 г	48,62 г	10,96 г	3	100
14	13	Арахис жаренный	626 кКал	26 г	52 г	13,4 г	3	100
15	14	Арахис жаренный без масла, без соли	587 кКал	24,35 г	49,66 г	12,86 г	3	100
16	15	Арахис жаренный без масла, с солью	587 кКал	24,35 г	49,66 г	12,86 г	3	100
17	16	Арахис жаренный на масле, без соли	599 кКал	28,03 г	52,5 г	5,86 г	3	100
18	17	Арахис жаренный на масле, с солью	599 кКал	28,03 г	52,5 г	5,86 г	3	100
19	18	Арахис испанский	570 кКал	26,15 г	49,6 г	6,33 г	3	100
20	19	Арахис испанский, жаренный на масле, без соли	579 кКал	28,01 г	49,04 г	8,55 г	3	100

2.4 Разработка макетов окон

Для каждого окна был спроектирован свой интерфейс с использованием цветов, сочетающихся между собой. Для создания объектов графического интерфейса использовались файлы коллекции ресурсов Qt Designer (Resource Collection Files, .qrc (формат файла основан на XML), в которой перечисляются файлы, используемые приложением.

На этапе разработки я подключаю интерфейс с помощью ui-файлов, для релиза сконвертировал весь интерфейс в классы Python с помощью утилиты ruic5.

Рисунок 3 Проектирование макета окна в Qt Designer



Виджеты, с помощью которых я реализовал функционал приложения:

Назначение виджета	Название виджета
Простая кнопка	QPushButton
Надпись	QLabel
Календарь	QCalendarWidget
Выбор даты из диапазона	QDateEdit
Круговая диаграмма	QChart
Вывод цифр, как на экране калькулятора	QLCDNumber
Строка редактирования	QLineEdit
Выбор варианта из списка	QComboBox
Таблица для отображения продуктов	QTableWidget
Текстовое поле	QTextEdit

2.5 Формулы расчетов

Для расчетов используются уравнения Харриса–Бенедикта, пересмотренные Розой и Шизгалом в 1984 году.

Ниже представлены сами формулы для мужчин и женщин.

Мужчины $BMR = 88.362 + (13.397 \times \text{вес в кг}) + (4.799 \times \text{рост в см}) - (5.677 \times \text{возраст в годах})$

Женщины $BMR = 447.593 + (9.247 \times \text{вес в кг}) + (3.098 \times \text{рост в см}) - (4.330 \times \text{возраст в годах})$

При расщеплении 1 грамма белков выделяется 17,6 кДж энергии, углеводов - 17,6 кДж энергии, жиров - 38,9 кДж энергии.

Энергия должна быть пропорциональная распределена между нутриентами.

Общая энергия (100%) = $(17,6 + 17,6 + 38,9) = 74,1$ КДж

Содержание нутриентов от общей массы:

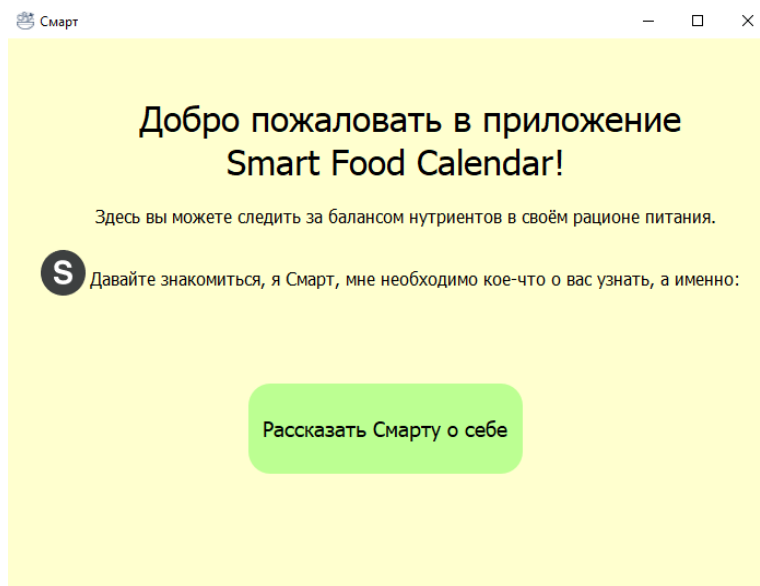
Белки	$17,6 / 74,1 = 0,2375 \sim 24 \%$
Жиры	$38,9 / 74,1 = 0,5249 \sim 52\%$
Углеводы	$17,6 / 74,1 = 0,2375 \sim 24 \%$

2.6 Навигация по приложению

Нас встречает приветственное окно, где с нами предлагает познакомиться виртуальный помощник Смарт.

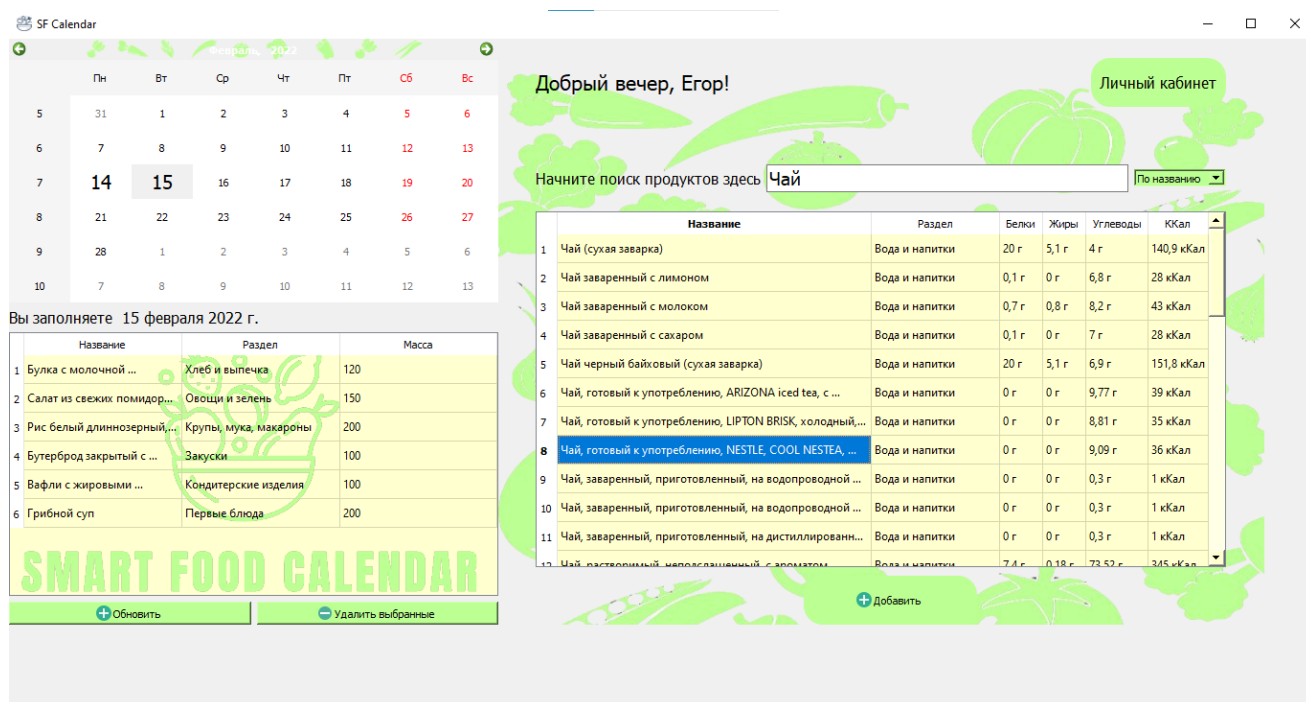
После нажатия на кнопку “Рассказать Смарту о себе”, возможно будет указать свою персональную информацию.

Рисунок 4 Приветственное окно



После этого мы попадем на рабочий стол приложения. Здесь нас встречают персонализированные надписи, виджет календаря и таблицы.

Рисунок 5 Рабочий стол



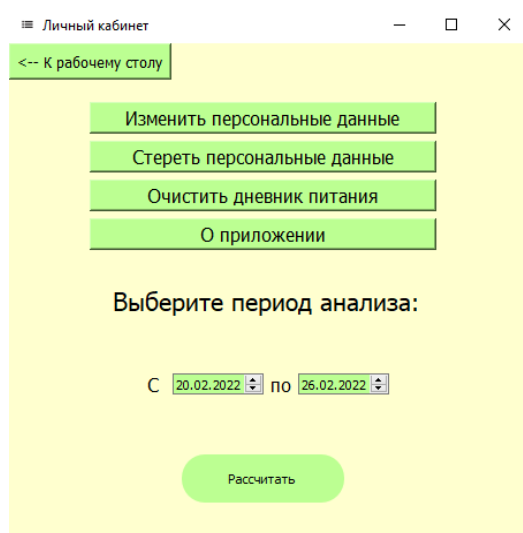
Таблицы нужны для того, чтобы работать с продуктами. В них указывается следующая информация о продуктах: раздел, информация о содержании БЖУ и Ккал.

Поиск продуктов можно начать в строке редактирования. Реализован поиск по названию, либо по категории.

Нажимая определенные кнопки, пользователь сможет добавить, удалить или обновить данные о питании.

В правом верхнем углу находится кнопка “Личный кабинет”.

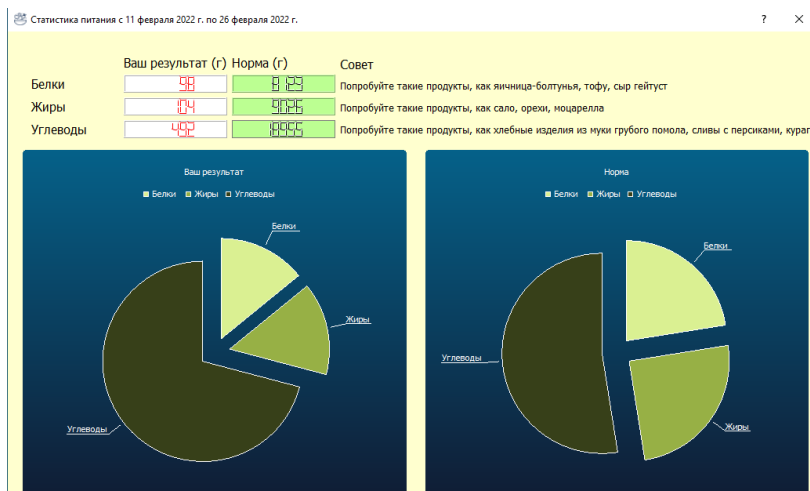
Рисунок 6 Личный кабинет



Здесь можно изменить или стереть персональные данные, а также очистить дневник питания или узнать информацию о приложении. Но самое главное, пользователь может выбрать период анализа и рассчитать норму нутриентов в своем рационе.

В окне статистики анализа выводятся показатели по БЖУ, а также строятся круговые диаграммы, отражающие их соотношение, но самое главное - выводятся советы по корректировке своего рациона питания.

Рисунок 7 Статистика питания



3. Заключение

В результате работы было создано приложение, выполняющее все функции, описанные в цели данного проекта. Приложение работоспособно и готово к использованию. Хочется, чтобы оно получило распространение и помогло людям следить за своим рационом питания.

Интересные идеи:

- Загрузочная анимация
- Непрерывный поиск в формате реального времени
- Приветственные, персонализированные надписи
- Работа с датами
- Построение круговой диаграммы с анимацией
- Добавление нескольких элементов в таблицу одновременно

Развитие:

В возможностях развития своей идеи я рассматриваю переход от десктопного приложения к мобильному, ведь заполнение дневника питания в телефоне удобнее, так как он всегда при себе. К тому же появится возможность воспользоваться камерой, чтобы считывать штрих-коды продуктов, тем самым получая информацию об их энергетической ценности.

4. Список используемых ресурсов

1. Официальная документация QT, URL: <https://doc.qt.io/qt.html#qtforpython>
2. Мой здоровый рацион, URL: <https://health-diet.ru/>
3. Википедия — свободной энциклопедии, Уравнение Харриса–Бенедикта, URL: https://en.wikipedia.org/wiki/Harris-Benedict_equation