**Пояснительная записка на программу**

**по проекту на языке программирования**

**Python, с использованием Qt**

**Наименование разработанной программы:** Программа, представляющая собой календарь-планировщик. Название программы “Календарщик”.

**Разработчик программы:** Гусевский В.В, ученик “Академии Яндекс-лицея”, 10Т, Лицея “Дубна”.

1. **Описание работы программы**

Когда я узнал, что мне предстоит написать приложение для проекта, в процессе обучения в “Лицее академии Яндекса”, то я начал задумываться об идее для этого проекта. На ум пришло написать календарь, с помощью которого можно было бы планировать свои дела, и реализовать при этом некоторое количество упрощающих пользование подобным приложением функций. Мне бы самому было полезно пользоваться на компьютере таким приложением. Поэтому мне была очевидна полезность его применения.

Программа представляет собой планировщик на основе календаря. Пользователь может выбирать любую дату в календаре и назначать для этой даты события: чей-то день рождения, мероприятие, цель и т.п. Когда пользователь выбирает какую-либо дату в календаре, то в окне программы выводятся все события для этого дня. Также в окне программы используются поля ввода данных, для того чтобы назначить новое событие для выбранного дня. Также, пользователь может выбрать событие из списка событий на день и удалить, либо изменить его. Реализована возможность полной очистки базы данных, путём нажатия соответствующей кнопки в программе. Дополнительно в окне программы выводится информация о событиях ближайших 3 дней (за выбранный день и за два дня после него). В программе есть несколько типов событий (например, “Цель/задача” или “Важная дата”). В программе реализован механизм предупреждения пользователя звуковым сигналом о событии, если время в системе совпадает с назначенным в событии времени. Также, реализована возможность загрузки (импортирования) данных о событиях из таблицы (файл формата csv), и выгрузки (экспортирования) данных из базы данных в файл формата csv. При импорте в базу данных строк из csv таблицы происходит перевод изначального импортируемого файла из его родной кодировки в кодировку UTF-8, с помощью временного файла, после выполнения импорта временный файл удаляется. Реализована возможность установить собственное изображение, которое выводится в важные даты, а также установление этого изображения по умолчанию.

Для хранения информации о событиях, используется база данных SQLite (СУБД, написанная на языке C). В базе данных создана таблица с информацией по всем событиям календаря. У каждого события есть несколько параметров полей таблицы: дата события, описание события, время события, тип события. Объём данных в базе данных относительно небольшой, поэтому каких-то особенностей при работе с базой данных в Python (Qt) не было.

1. **Описание окон программы**

В главном окне программы расположены следующие основные элементы интерфейса:

1. Календарь.
2. Таблица вывода событий выбранного в календаре дня
3. Таблица с выводом событий за текущий и следующие два дня.
4. Поля для ввода информации о новом событии для выбранного дня.
5. Кнопки для добавления, изменения или удаления определённого события.
6. Кнопка для удаления всех событий из базы данных
7. Вывод текущей даты справа от поля заполнения информации о новом событии.

В окнах для экспорта/импорта файлов расположены следующие элементы интерфейса:

1. Кнопка для выбора директории, в которую будет сохранятся файл, или директории нужного файла, из которого импортируются события.
2. Кнопка для начала процесса экспорта/импорта файлов.

В окне для выбора изображения, которое выводится в важные даты расположены следующие элементы интерфейса:

1. Кнопка для выбора директории изображения, которое будет выводится в важные даты
2. Кнопка для установки этого изображения по умолчанию

В информационном окне расположены следующие элементы интерфейса:

1. Текст с общей информацией о программе, о разработчике программы.
2. **Ресурсы программы**

При разработке программы были использованы следующие библиотеки Python:

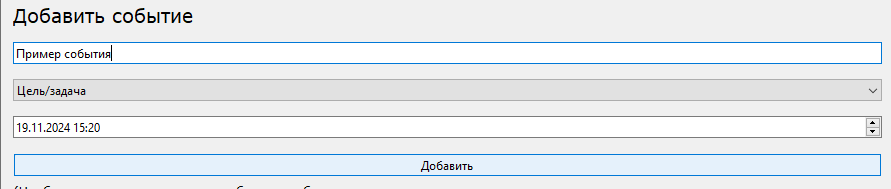
1. **PyQt6** — библиотека Python для создания приложений с графическим интерфейсом с помощью инструментария Qt. Созданная в Riverbank Computing, PyQt является свободным ПО (по лицензии GPL) и разрабатывается с 1999 года. Последняя версия PyQt6 — на основе Qt 6 — выпущена в 2021 году, и библиотека продолжает обновляться.

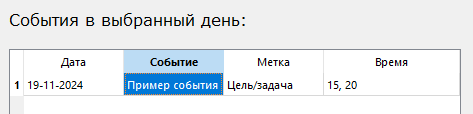
Основные использованные классы библиотеки PyQt6, которые представляют наибольший интерес:

1. Класс **QPushButton** (Работа с кнопками).
2. Класс **QTableWidget** (Вывод таблицы с данными из БД).
3. Класс **QCalendarWidget** (Виджет календаря).
4. Класс **QDateTimeEdit** (Ввод даты и времени).
5. Класс **QLineEdit** (Ввод события).
6. Класс **QComboBox** (Выбор метки).
7. Класс **QLabel** (Вывод информации)

Про остальные библиотеки:

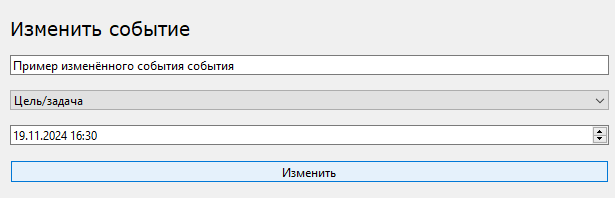
1. **io** ‒ Предоставляет основные средства Python для работы с различными типами ввода/вывода.
2. **datetime** ‒ Предоставляет классы для обработки времени и даты разными способами.
3. **sqlite3** ‒ C библиотека, реализующая легковесную дисковую базу данных (БД), не требующую отдельного серверного процесса и позволяющую получить доступ к БД с использованием языка запросов SQL.
4. **os** ‒ Функции для работы с операционной системой, не зависящие от используемой операционной системы.
5. **chardet** ‒ Модуль предназначенный для автоматического распознавания кодировок символов в тексте.
6. **csv** ‒ предоставляет функции для чтения и записи данных в формате CSV.
7. Примеры работы программы
8. Пример добавления события



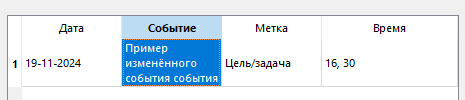


(Событие действительно добавилось)

1. Пример изменения события

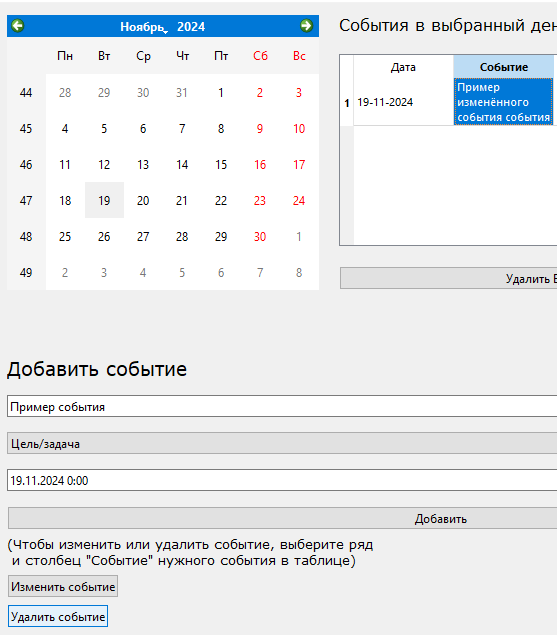


(Поменяли текст события и время)

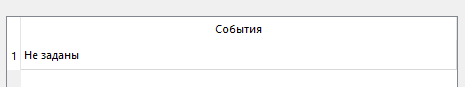


(Событие действительно изменилось)

1. Пример удаления события

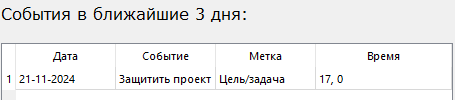


(Выбрали событие в таблице событий за день и нажимаем кнопку “Удалить”)



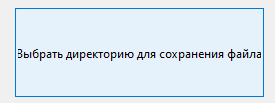
(Событие действительно удалилось)

1. Пример работы ленты с событиями

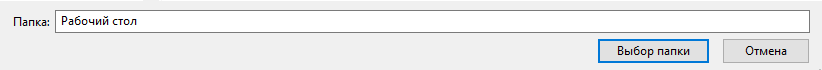


(Действительно, для 21 числа добавлено такое событие)

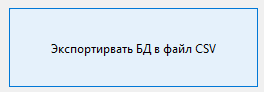
1. Пример экспорта БД



(Жмём кнопку для выбора директории)



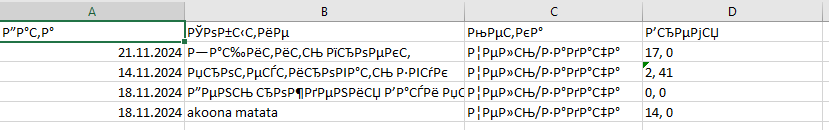
(Я выбрал рабочий стол)



(Жмём кнопку для экспорта)



(На рабочем столе появился файл с экспортированной БД)



(БД экспортировалась. Как мы видим, кириллица отображается некорректно, т. к. сохранение проводится в кодировке UTF-8, для более удобной работы с программой в дальнейшем. При желании таблицу можно перевести в кодировку Windows-1251, для корректного отображения в Excel. (прим. Notepad++))

1. Выводы

В результате работы над проектом была разработана программа “Календарщик”. Все пункты ТЗ были выполнены. Был реализован дополнительный функционал, расширяющий требования ТЗ. Программа была разработана с использованием современной библиотеки для Python PyQt6. Применение PyQt6 позволило работать с широкой номенклатурой классов. Поэтому разработка программы была более эффективной и быстрой. Современная разработка программ реализуется похожими способами. Мной был получен полезный опыт практической разработки современных программ.

При общении со своей классной руководительнице, о программе, которую я планирую написать, она отметила, что сама бы пользовалась ею, и предложила бы её использовать директрисе нашего лицея. Это доказывает актуальность программы.

Разработанная программа может использоваться по своему назначению пользователями ОС Windows 10/11 (В других операционных системах работа программы не проверялась). Считаю, что функционал разработанной программы “Каленадрщик” можно расширить в будущем.