МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики» (МТУСИ)**

Кафедра «Математическая кибернетика и информационные технологи»

**КУРСОВАЯ РАБОТА**

по дисциплине «Системы и алгоритмы обработки данных»

Выполнил: студент группы БВТ2205

Ковлавевский Стас

Проверила:

Преподаватель

Симонов С.Е.

Москва 2024

**СОДЕРЖАНИЕ**

1. Введение ............................................................................................. 3

2. Цель работы ....................................................................................... 4

3. Основные задачи программы ........................................................... 5

4. Среда разработки и библиотеки .......................................................6

4.1 Среда разработки ...................................................................6

4.2 Установка библиотек .............................................................7

5. Разработка интерфейса программы ..................................................8

5.1 Окно для регистрации водителей .........................................9

5.2 Поле для выбора дня недели .................................................9

5.3 Поле для ввода параметров маршрута ...............................10

5.4 Генерация расписания .........................................................11

6. Разработка функционала планирования маршрутов ....................12

6.1 Расчет времени начала и окончания маршрута ................12

6.2 Жадный алгоритм распределения маршрутов ..................13

6.3 Проверка пересечения маршрутов .....................................14

6.4 Проверка выходного дня ....................................................15

7. Тестирование приложения .............................................................16

7.1 Тестирование водителей типа А и типа Б ........................16

7.2 Тестирование расписания ..................................................17

8. Вывод по работе .............................................................................18

9. Список литературы ........................................................................19

**ЦЕЛЬ РАБОТЫ:** Необходимо разработать **оптимальную программу для генерации расписания водителей маршруток**, которая должна:

1. **Минимизировать количество водителей**, необходимых для выполнения всех маршрутов.
2. **Максимизировать количество выполненных маршрутов** за день в рамках установленных временных и трудовых ограничений.

**Основные задачи программы:**

1. **Распределение маршрутов среди водителей** с учётом:
   * Ограничений по рабочему времени:
     + Водители типа А могут работать максимум **8 часов** в будние дни и не работают по выходным.
     + Водители типа Б могут работать до **12 часов** и могут работать в выходные.
   * Общих временных рамок: рабочий день начинается в **6:00 утра** и заканчивается в **3:00 ночи** следующего дня.
2. **Проверка на пересечение маршрутов** для каждого водителя, чтобы каждый рейс назначался в уникальное время, без перекрытий.
3. **Поиск оптимальных временных промежутков** для рейсов, чтобы:
   * **Полностью использовать рабочее время водителей**, заполняя их смены максимально эффективно.
   * **Избежать переработки** и назначений рейсов на время после 3:00 ночи.
4. **Минимизация дублирования водителей** и обеспечение максимальной загруженности каждого водителя, прежде чем добавлять новых водителей.
5. **Гибкость планирования**: учёт как будних, так и выходных дней, с корректным распределением маршрутов между водителями разных типов в зависимости от дня недели.

**Ожидаемый результат:**

Программа должна выводить оптимальное расписание, в котором:

* Минимальное количество водителей используется для выполнения всех маршрутов.
* Каждый водитель работает максимально эффективно в рамках разрешённого рабочего времени.
* Все маршруты распределяются без временных пересечений, и рабочий день завершается до 3:00 ночи.

**Ход работы**

1. **Среда разработки**:
   * Для разработки программы будет использоваться **Visual Studio Code** — мощный редактор кода, который поддерживает Python и предоставляет удобные инструменты для работы с проектами.
   * Предполагается, что у вас уже установлен **Python** (рекомендуемая версия 3.7 или выше).
2. **Установка необходимых библиотек**: Программа будет использовать следующие библиотеки Python:
   * **pandas** — для работы с табличными данными и их удобного отображения в текстовом формате.
   * **tkinter** — для создания графического интерфейса (GUI), который позволит взаимодействовать с программой через окна и кнопки.
   * **datetime** — для работы с датами и временем, чтобы рассчитывать смены и маршруты водителей.
   * **random** — для случайного распределения маршрутов и выбора водителей при необходимости.

Установить необходимые библиотеки можно с помощью **pip**, выполнив следующие команды в терминале Visual Studio Code:

bash

Копировать код

pip install pandas

Библиотека **tkinter** обычно уже включена в стандартную библиотеку Python и не требует установки.

1. **Разработка интерфейса программы с использованием Tkinter**: В процессе работы создадим **графический интерфейс** программы с помощью Tkinter:
   * Окно для регистрации водителей (фамилия, тип А или Б).
   * Поле для выбора дня недели.
   * Поле для ввода количества маршруток и времени маршрута.
   * Кнопки для генерации расписания для водителей типа А, типа Б и комбинированного расписания.
   * Поле для отображения сгенерированного расписания и результатов.
2. **Разработка функционала планирования маршрутов**: Основная часть работы заключается в создании логики планирования маршрутов:
   * Написание функций для вычисления времени начала и окончания маршрутов.
   * Реализация алгоритма жадного распределения маршрутов среди водителей с учётом их рабочего времени.

ОСНОВНОЙ ФУНКЦИОНАЛ

Первым делом импортируем необходимые библиотеки:

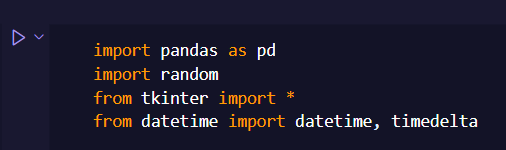


Рисунок 1 – импорт библиотек

Далее я инициализирую пустые списки водителей и их времени начала смен.

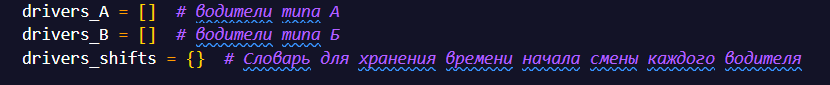


Рисунок 2 – инициализация списков

После инициализации списков пропишу следующие параметры: время работы смен,типы маргрутов,продолжительность смен для водителей типов А и Б, установлю начальное время маршрута (можно менять).

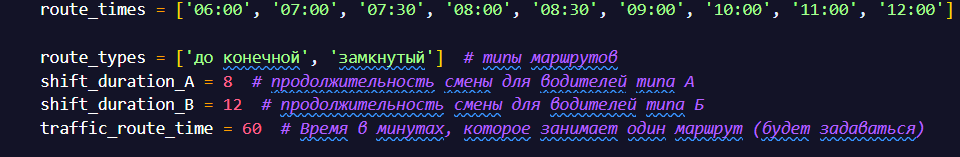


Рисунок 3 – вводим необходимые параметры

Также указываю начальное время работы и время окончания смены:



Рисунок 4 – начальное и конечное время смен

НАПИСАНИЕ ФУНКЦИЙ

Первая функция будет отвечать за расчёт окончания рейса с учётом его продолжительности:

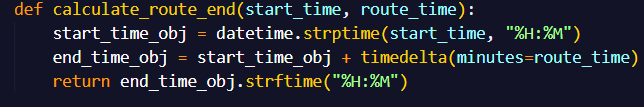


Рисунок 5 – функция расчёта конца рейса

Затем,важно учесть,что рейсы водителей не могут начинаться и заканчиваться в одно время. Допустима ситуация если у одного водителя смена начинается в 6 утра (пусть время рейса = 1 час), то его рейс окончится в 7 утра, и разрешено, что например в 6:30 следующий водитель начнет свою смену и тогда с учетом времени рейса – 1 час, закончит его в 7:30. Но ни в коем случае нельзя иметь одинаковое время начала рейса и окончания по одному маршруту. Поэтому нужно прописать соответствующую функцию:

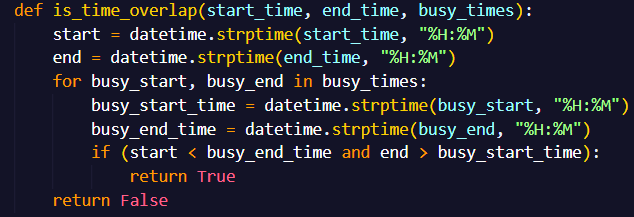


Рисунок 6 – функция для проверки пересекающихся рейсов

Далее пропишу функцию,на проверку дня недели – выходной ли?

Так как в дальнейшем нужно будет учитывать,что водители типа А не могут работать по выходным (и водители типа Б будут компенсировать этот момент)



Рисунок 7 – проверка дня недели на выходной

Итак,теперь займемся самим расписанием:

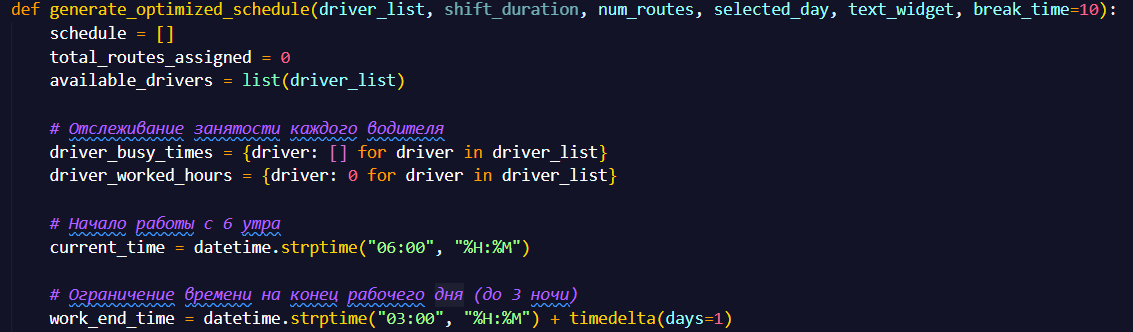


Рисунок 8 – функция оптимизации расписания

Здесь мы: инициализируем список для хранения расписания, задаем общее количество маршрутов,копируем список доступных водителей, создаем 2 словаря: словарь занятости водителей по времени и словарь уже отработанных часов водителем, устанавливаем текущее время на 6 утра,и вволим ограничение на окончание времени работы (3 часа ночи).

Теперь работа с циклом: цикл - по маршрутам:

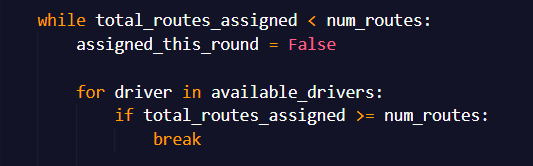


Рисунок 9 – работа с циклом маршрутов

Здесь мы назначаем каждому водителю свой маршрут и проверяем был ли он уже назначен.

Теперь проверка самих водителей. Проверка на то, является ли день для водителей типа А – выходным, также проверка каждого типа водителя на то,сколько он уже отработал:

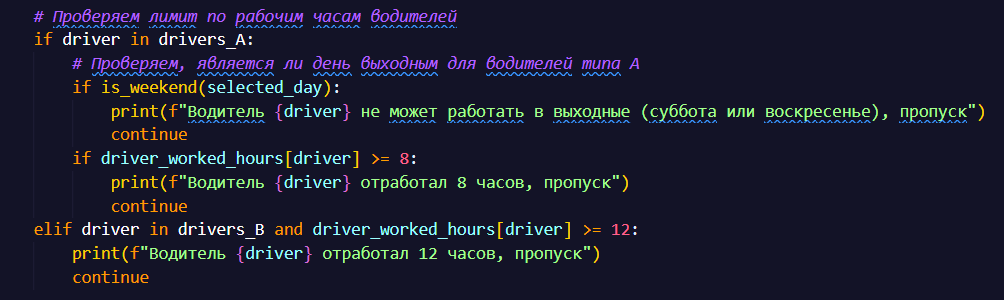


Рисунок 10 – проверка водителей

Затем еще немного поработаю с рейсами:

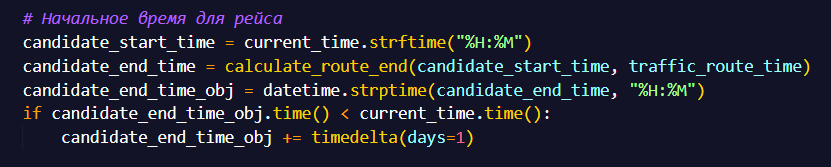


Рисунок 11 – расчёт времени для рейсов

Следующая проверка на то,чтобы рейс не выходил по времени за 3 час ночи: так как водители работают исключительно с 6 утра и максимум до 3 часов ночи.

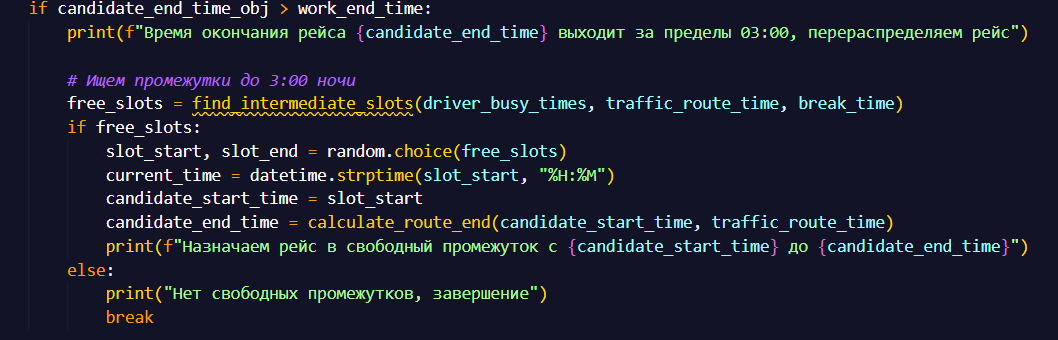


Рисунок 12 – проверка на то,чтобы рейс не выходил за 3 часа ночи

Далее проверка на пересечения рейсов:

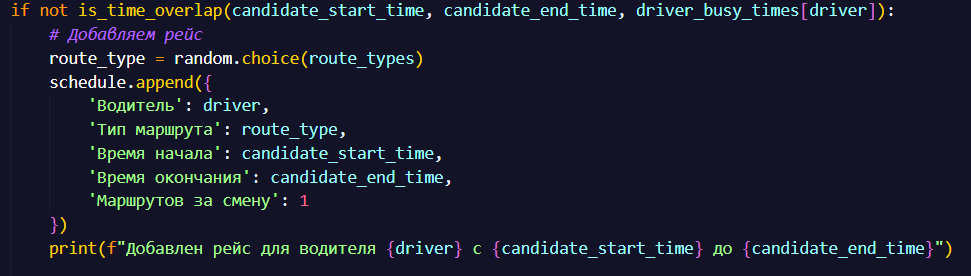


Рисунок 13 – работа с добавлением рейса в таблицу

Далее введу параметры,связанные с занятостью водителя

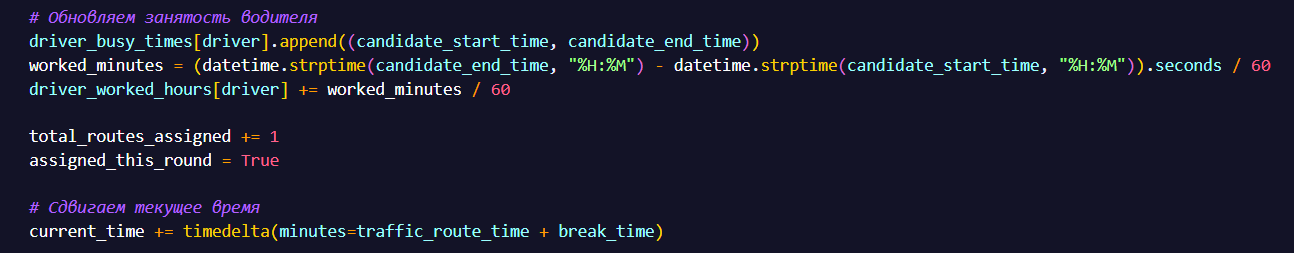


Рисунок 14 – работа с занятостью водителя

\*также мы увеличиваем количество назначенных маршрутов,устанавливаем флаг,что рейс был назначен и еще сдвигаем время на длительность маршрута + перерыв. Перерыв как правило у водителей составляет минут 10 (хотя бы банально на то,чтобы сменить друг друга).

Следующая проверка ситуации когда рейс не был назначен:

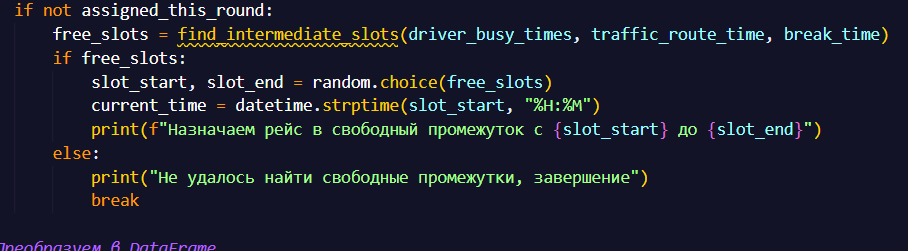


Рисунок 15 – проверка на то,если рейс не назначен

\*Если остались рейсы (например смена в 6 утра начинается на маршрут длиною в час,но владелец задал на день 25 маршрутов – то 21 будут разделены друг за другом почасово,а оставщиеся 4 маршрута будут добавлены в промежуточное время)

Затем поработаю с отображением расписания в оконном терминале.

Конкретнее,работа с Dataframe и его выводом в консоли.

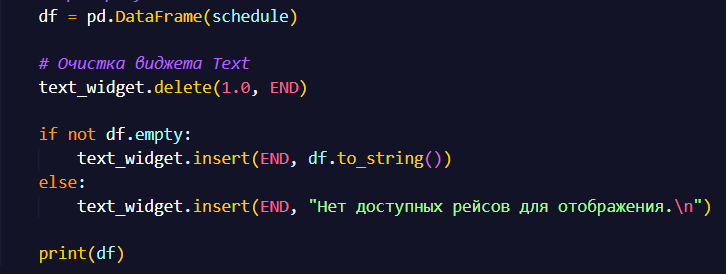


Рисунок 16 – работа с Dataframe

\*Логикак такова,что наш список расписания преобразуется в Dataframe,также очищается виджет «Text» перед выводом нового расписания,и также если Dtaframe НЕ пуст – то выводим расписание в «консоль»,а если же наоборот пуст, то выводим сообщение «Нет доступных рейсов для отображения».

Также я реализовал дополнительно (для удобства) функционал «управления водителями» для того, чтобы можно было во-первых отслеживать полный список своих сотрудников после их добавления,также удалять их или менять им тип «контракта» с типа А на тип Б и наоборот.

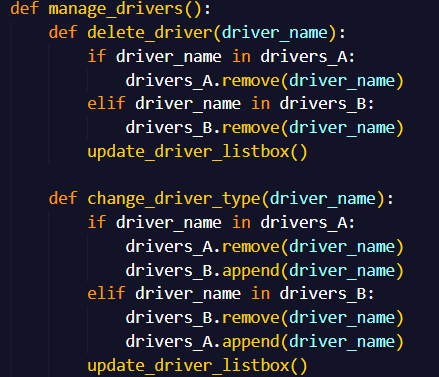


Рисунок 17 – управление водителями

И соответственно обновляю и перезаписываю всё это дело:

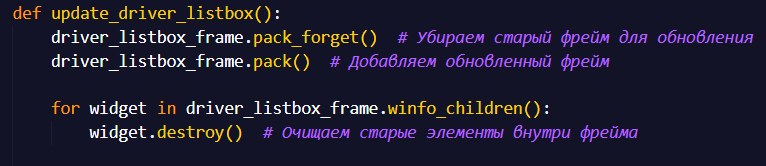


Рисунок 18 – перезапись фреймов

Еще один цикл – для того,чтобы создать (для удобства пользования) возле каждого водителя кнопки «Удалить» и «Сменить тип»

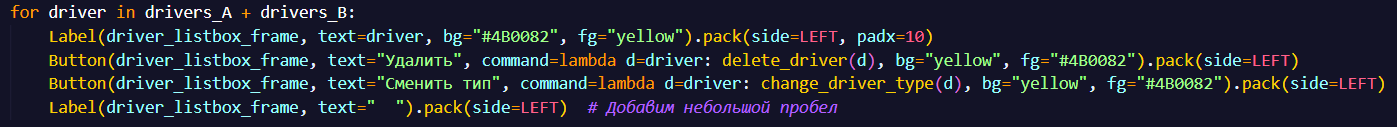


Рисунок 19 – работа с кнопками действий для водителей

И создаю уже новое окно для управления водителями:

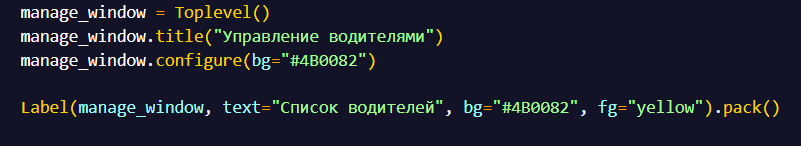


Рисунок 20 – новое окно управления

Также из своих дополнений я решил сделать не просто кнопку генерации расписания которая будет генерировать его из водителей типов А и Б, но и еще две кнопки с помощью которых можно будет отдельно генерировать расписание для водителей типа А и отдельно типа Б (по желанию).

Об этом далее.

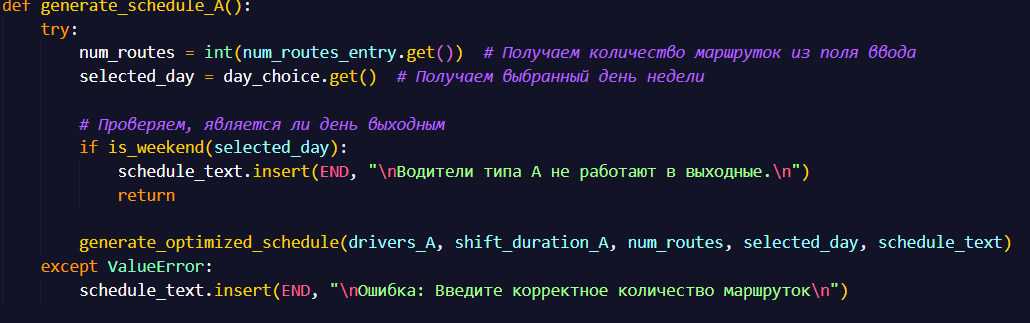


Рисунок 21 – функця генерации расписания для водителей типа А

\*в ней мы также делаем обязательную проверку на выходной день,так как водители типа А не могут работать по выходным,и если указан именно один из этих дней – то пользователю выводится ошикба-пояснение о том,что водители типа А не могут работать по выходным дням.

И также функция генерации расписания для водителей типа Б. (Здесь уже нет проверки на выходной день,поэтому просто оставляю проверку на корректность задания значения количество маршруток-рейсов).

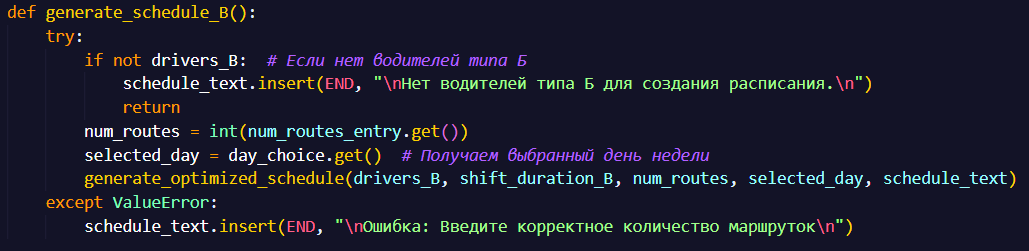


Рисунок 22 – генерация расписания для водителей типа Б

И также добавляю кнопку для генерации расписания совместного (водители типа А + водители типа Б):

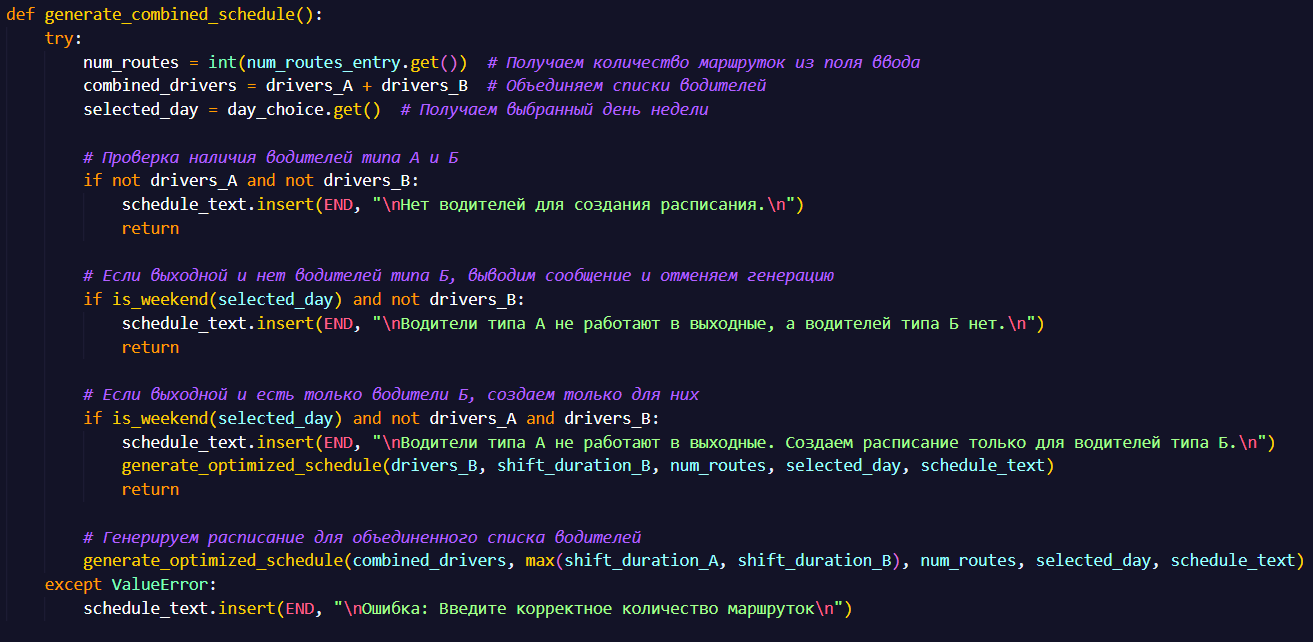


Рисунок 23 – генерация совместного расписания для водителей

Здесь я обьединяю списки водителей типа А и типа Б, делаю проверку на то,что водителей может не быть вообще (тогда вывожу в консоль сообщение о том,что нет водителей). Также проверка на то,что пользователем выбран выходной день,но нет водителей типа Б (поэтому также никто не сможет выйти на рейсы). И соответственно генерирую совместное расписание для водителей.

Также я решил добавить функцию сброса – которая очищает: терминал вывода расписания,количество маршруток (рейсов), время одного рейса.

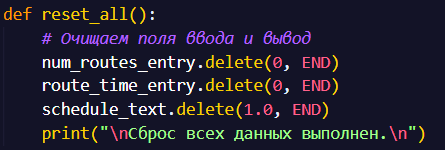


Рисунок 24 – функция очистки (сброса)

ЧАСТЬ КОДА – ОТРИСОВКА ИНТЕРФЕЙСА

Теперь я реализую отображение всех кнопок,полей ввода и самого термиала вывода расписания с помощью библиотеки Tkinter.

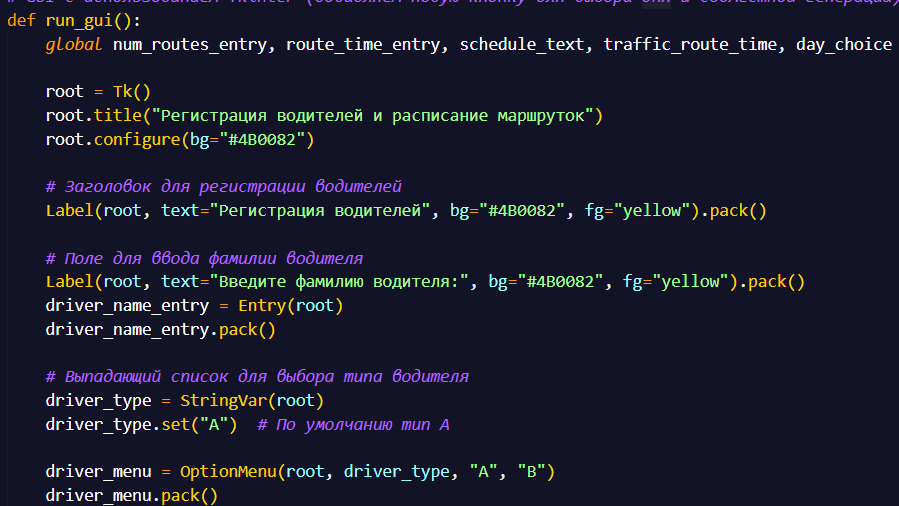


Рисунок 25 – реализация интерфейса (главное поле,главные кнопки)

Отдельно функция для кнопки добавления водителя:

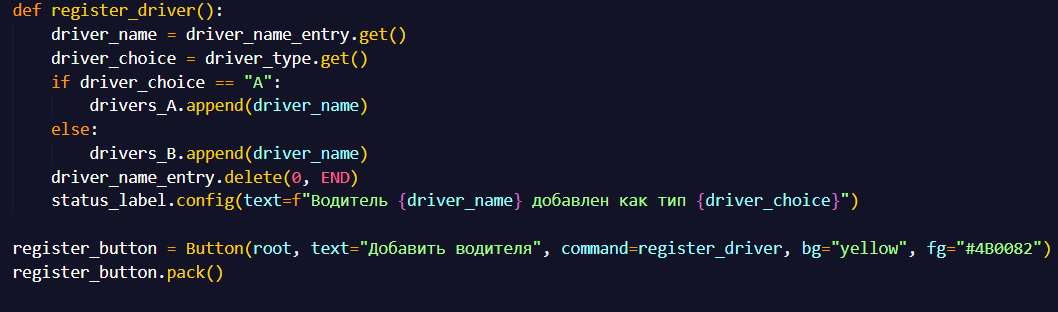


Рисунок 26 – добавление водителя (интерфейс)

Также добавляю статус добавления/удаления водителей:

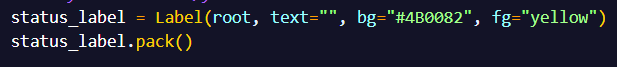


Рисунок 27 – добавление/удаление водителей

Добавляю поле для вывода дня недели:

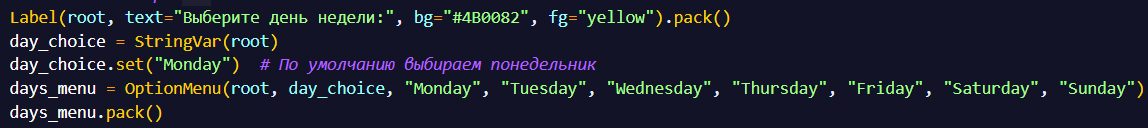


Рисунок 28 – добавление дня недели

Делаю поля для ввода количество маршруток (рейсов) и времени одного рейса:

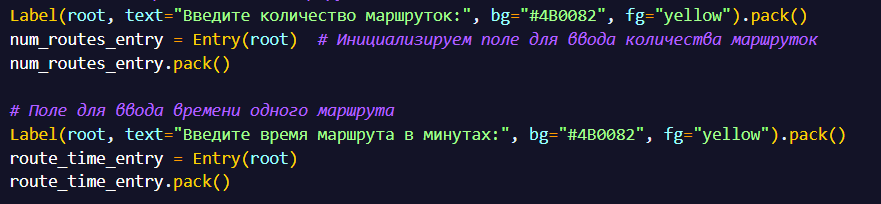


Рисунок 29 – отображение полей для ввода кол-ва маршурток и времени

Также проработаю момент допустимого ввода времени рейса.

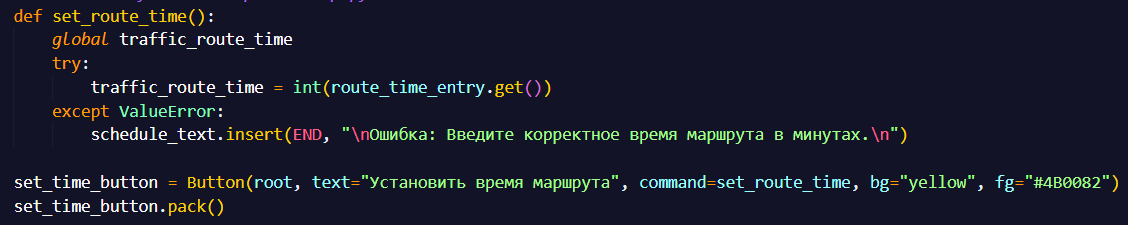


Рисунок 30 – проверка на корректность данных

И, наконец, отрисую все главные кнопки (генерация расписания – всех видов,кнопка сброса,управление водителями

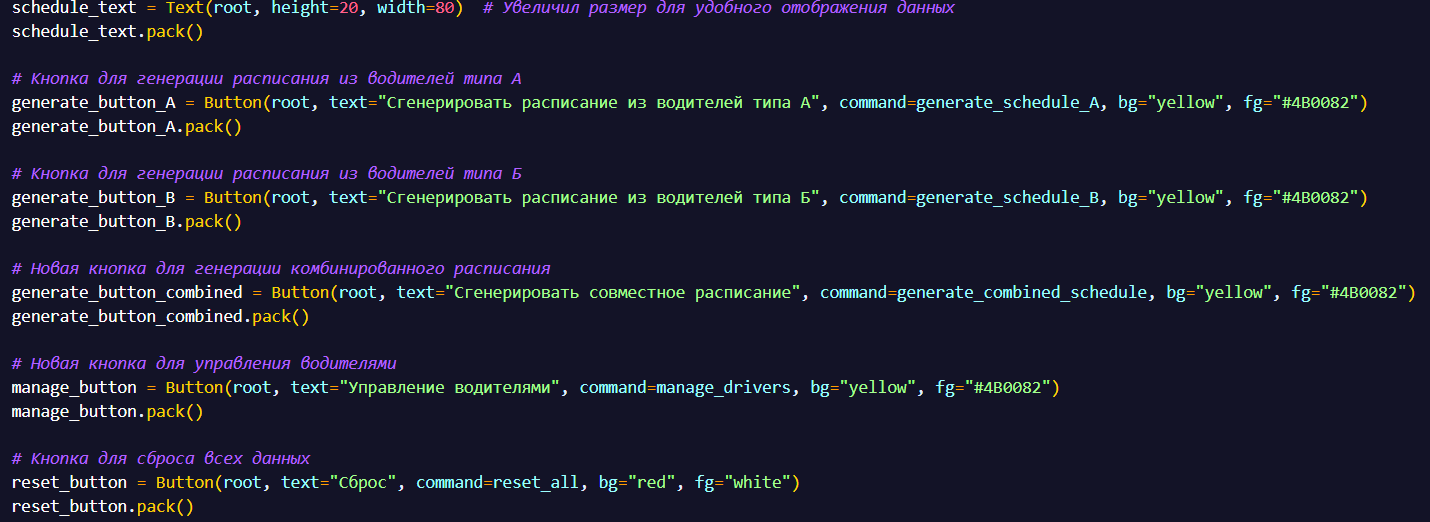


Рисунок 31 – отрисовка всех главных кнопок оконного приложения

И теперь прописываю запуск интерфейса:

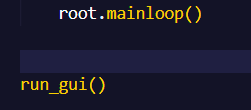


Рисунок 32 – запуск интерфейса

Далее наглядно продемонстрирую работу приложения.

ТЕСТИРОВКА ПРИЛОЖЕНИЯ

Так как код в файле формата Jupyter, то обойдусь без команд в терминале для запуска файла. Воспользуюсь готовой кнопкой запуска.

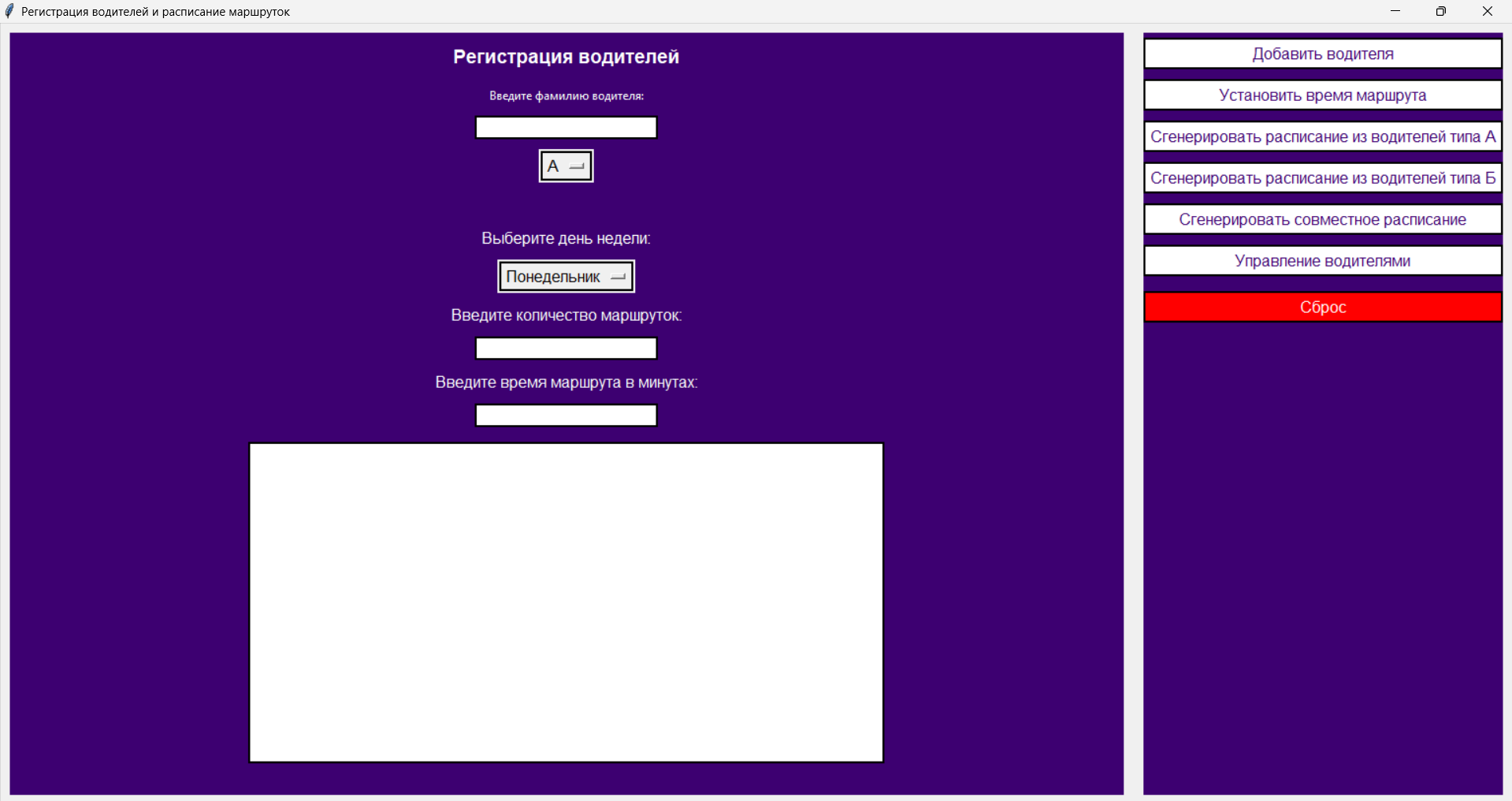


Рисунок 33 – запуск проекта

Как итог – успешный запуск проекта. Далее проведу более плодотворное тестирование: создам 4ёх водителей типа А, количество рейсов на день пусть будет 18 и также время одного рейса – 1 час (программа должна учитывать время на смену водителей)

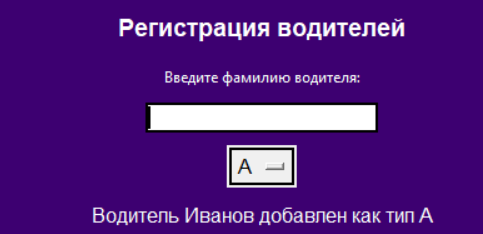


Рисунок 34 – добавление водителя

Можно проверить,что водитель действительно создан. Для этого я придумывал кнопку «управление водителями»

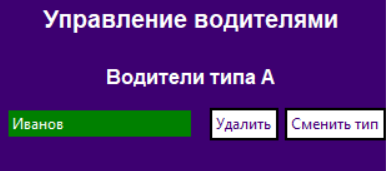


Рисунок 35 – управление созданными водителями

Появилось новое окно – спсиок моих водителей (пока создан только 1 и я могу его либо перенести на план работы типа Б или вовсе удалить из системы)

Создам еще 3 водителей,и пропишу количество рейсов на день и время рейса. Также изменю рабочий день (поставлю,например,вторник)

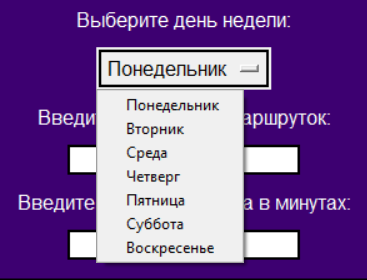


Рисунок 36 – смена рабочего дня

Задам количество рейсов на вторник – 18,а время одного рейса 1 час (60 минут). Также я ставил указание рейса именнов минутах – также для удобства.

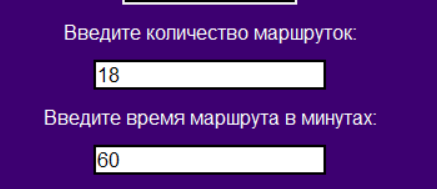


Рисунок 37 – выставление параметров маршрута

Жму генерировать (для водителей А – так как задавал только их. Мог сгенерировать совместное,так как я выбирал будний день – он подходит под условие проверки выходного дня)

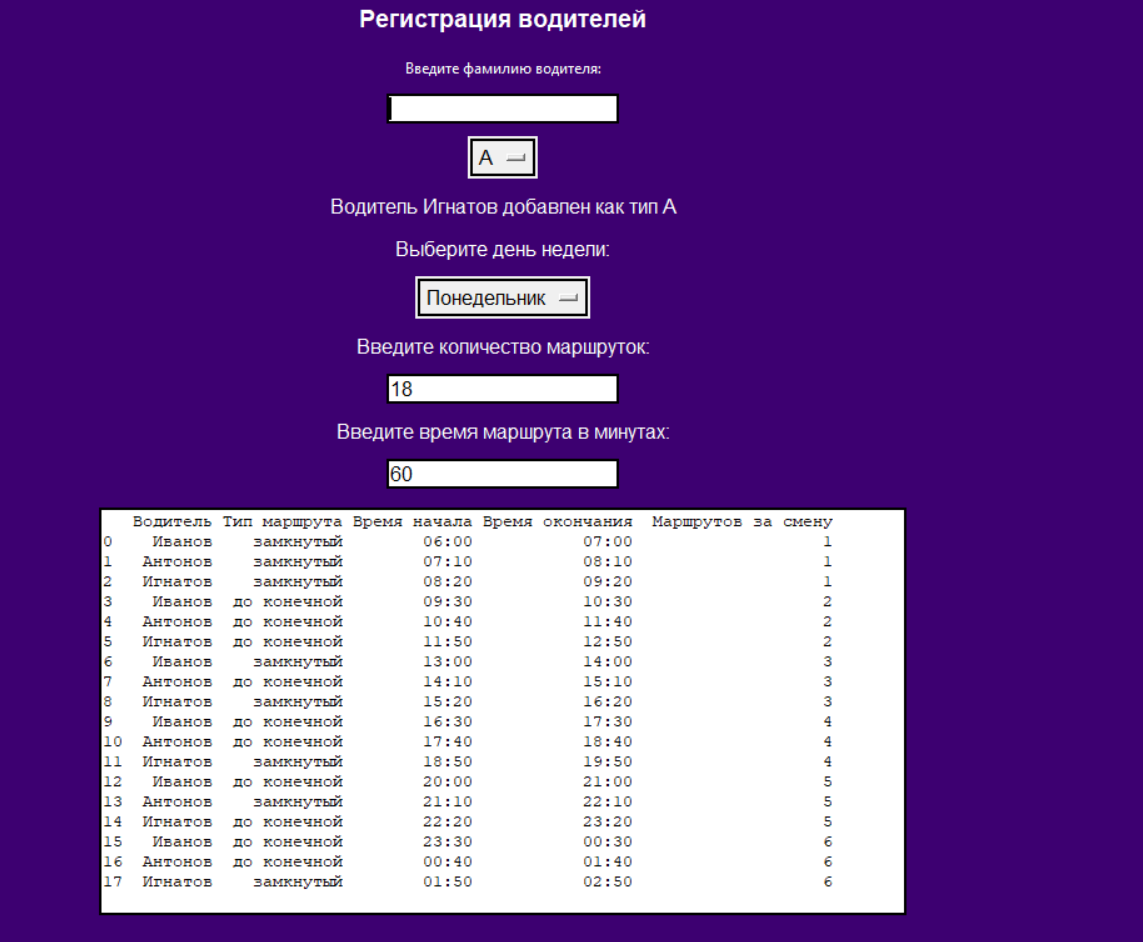


Рисунок 38 – расписание для водителей типа А

Также могу перегенирировать заново расписание. Тогда сменится расписания типов маршрута. Особо ничего больше не поменялось так как параметры я оставлял прежними.

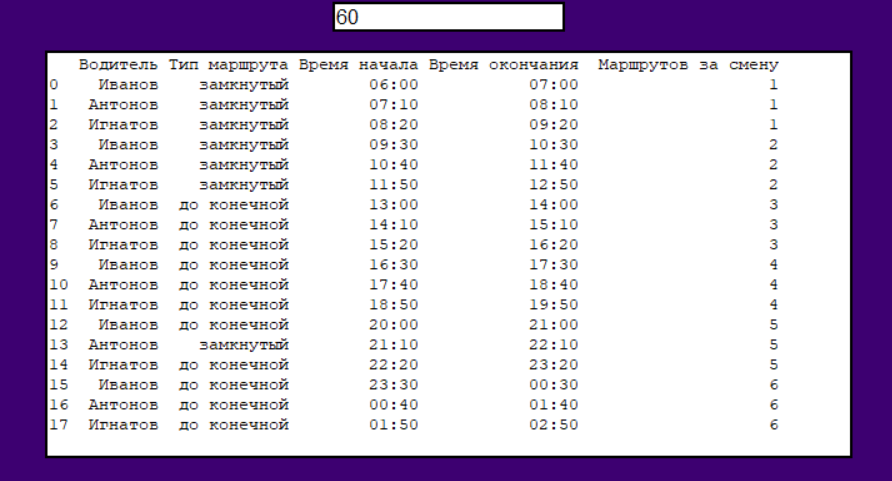


Рисунок 39 – перегенирация расписания

Проверки обработок ошибок:

* 1. Если я нажму создать расписания для водителей типа Б, то мне выдаст ошибку о том,что водители не созданы.



Рисунок 40 – проверка ошибки генерации

И кнопкой сброса у меня снова все очистится и можно заново все задавать.

Могу также с помощью кнопки сменить тип водителей перекинуть некоторых на другой план работы (например на тип Б). Задать новое количество рейсов и времени одного рейса и сгенерировать уже совместное расписание:

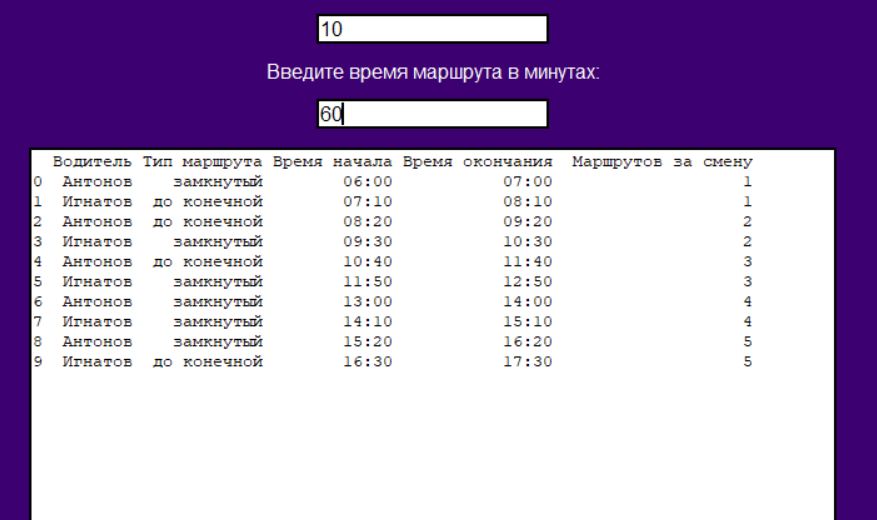


Рисунок 41 – генерация совместного расписания

Выполню теперь отработку вывода сообщения о том,что водители типа А не могут работать по выходным (перекину быстренько всех водителей в тип А и поставлю выходной день).

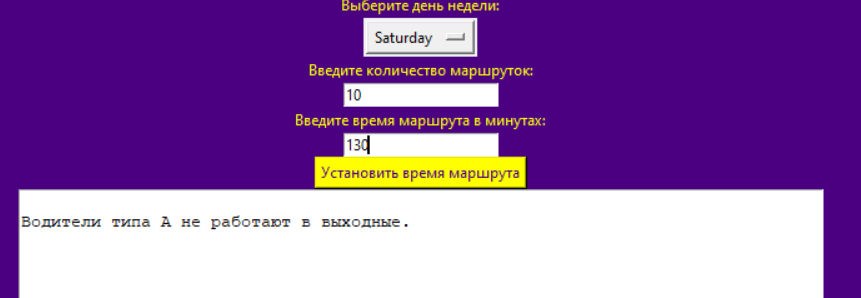


Рисунок 42 – отработка выдачи пояснения о выходном дне для водителей типа А при попытке генерации расписания для них

И сразу попробую сгенерировать расписания для водителей типа Б (должно выдать также ошибку так как я их не создавал)

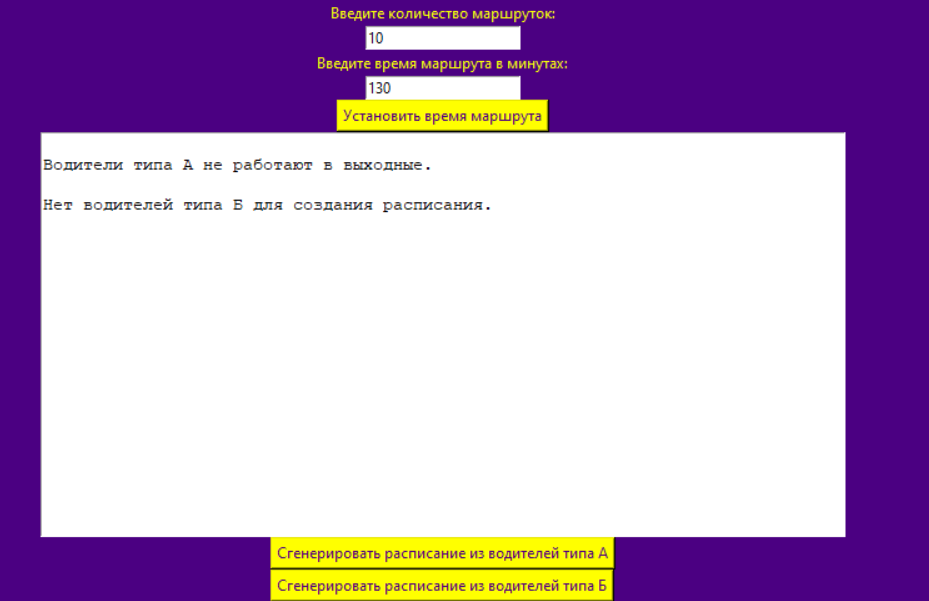


Рисунок 43 – отработка ошибки для расписания водителей типа Б

\*Решил еще чуть улчшить окно управления водителями:

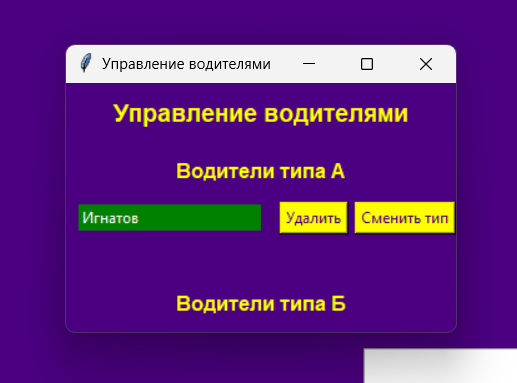


Рисунок 44 – обновленный внешний вид окна управления водителями

### Вывод по работе:

Я разработал программу для **оптимальной генерации расписания водителей маршруток**, которая учитывает множество факторов и ограничений. Программа оснащена удобным графическим интерфейсом (GUI), что делает взаимодействие с ней более простым и интуитивно понятным.

### Основные достижения разработки:

1. **Распределение водителей и маршрутов**:
   * Программа позволяет регистрировать водителей типа А и типа Б, а также автоматически распределять маршруты между ними.
   * Реализованы все необходимые ограничения по рабочему времени: водители типа А могут работать до 8 часов в будни и не работают в выходные, а водители типа Б могут работать до 12 часов и работают в выходные.
   * Программа гарантирует, что маршруты водителей не выходят за пределы рабочего дня, который заканчивается в 3:00 ночи.
2. **Оптимизация графика работы**:
   * Программа эффективно минимизирует пересечения рейсов и проверяет, чтобы они укладывались в установленные временные рамки.
   * Реализован жадный алгоритм, который максимально эффективно использует рабочее время водителей, прежде чем привлекать дополнительных сотрудников.
3. **Гибкость и учет условий**:
   * Я добавил возможность генерировать расписания как для одного типа водителей, так и для комбинированного расписания (типы А и Б).
   * Программа корректно учитывает день недели, адаптируя расписание в зависимости от того, является ли день выходным или будним.
4. **Улучшенный интерфейс управления водителями**:
   * В программе можно легко управлять списком водителей: добавлять новых, удалять существующих и менять их тип (А или Б).
   * Для улучшения восприятия информации реализовано цветовое кодирование: водители типа А выделены зелёным цветом, а водители типа Б — синим, что помогает легко различать категории сотрудников.
5. **Интуитивно понятный интерфейс (GUI)**:
   * Я создал графический интерфейс на базе Tkinter, который позволяет пользователю:
     + Регистрировать водителей.

**Список литературы**

1. Обучение Python GUI на Python <https://pythonru.com/uroki/obuchenie-python-gui-uroki-po-tkinter>
2. Tkinter Документация <https://docs.python.org/3/library/tkinter.html>