Пояснительная записка

Помощник по составлению учебного расписания.

Авторы проекта: ученик лицея академии Яндекса – Федяев Александр.

Идея проекта: облегчить и автоматизировать человеку, отвечающему за учебное расписание, работу. Добавить многие функции, которые необходимы в работе данного человека.

Описание реализации: данное приложение включает в себя множество окон с необходимым функционалом. Здесь вы можете добавлять и изменять здания, предметы, учителей, классы и учебные планы. Всё это сохраняется в базу данных, поэтому данные никуда не пропадут. Доступна выгрузка нагрузки учителей в файл word. Также, помимо разнообразного интерфейса, разработан красивый и удобный интерфейс, а самое главное – интуитивно понятный.

Основные классы:

1. Класс QMy\_school\_plan. В данном классе реализована функция создания учебных планов. В ней вы добавляете учебные предметы и сохраняете в файл. Файл сохраняется с разрешением csv.
2. Класс QChange\_plan. Данное окно не менее важно чем предыдущее. В данном окне вы можете изменять, созданные вами ранее файлы. Свои изменения вы можете сохранить в этот же файл, или сохранить в новый.
3. Класс MyClasses. В данном классе реализована функция добавления классов. Здесь вы выбираете здание, номер, букву, а также учебный план. Все созданные классы сохраняются в базу данных.
4. Класс QNew\_teachers. В данном классе реализована функция добавления учителей. Здесь вы выбираете здание, где работает учитель, график работы и предметы, которые ведет данный учитель. Информация об учителе также сохраняется в базу данных.
5. Класс MyCreate\_workload. В данном окне реализована функция добавления нагрузки учителей, по заданным фильтрам вы выбираете учителя классу. Общая нагрузка учителей отображается в этом же окне. Также нагрузка учителей сохраняется в базу данных.
6. Класс MyWorkloadProblems. Здесь, после заполнения всех данных вам отобразятся предметы каждого класса, где не выбран учитель. Это окно очень хорошо помогает отслеживать все проблемные зоны.
7. Класс MyExportWorkload. В данном классе реализована функция экспорта нагрузки учителей в файл word. Каждая страница файла – это выгрузка одного учителя.

Взаимодействие классов и связь между ними: в данном проекте присутствует главный файл для связи всех классов программы. Все окна очень близко связанны между собой и зависят друг от друга. Для связи данных окон присутствует база данных со множеством таблиц, а также файлы с разрешением csv.

Классы и методы:

Класс QMyBuildings:

Функция del\_build:

1. Удаление здания из бд.

Функция add\_build:

1. Добавление здания и адреса в бд.

Функция update\_combo\_box:

1. Добавление наименований зданий в comboDelBuild.

Функция loadTable:

1. Запись значений из файла в tableWidget.

Функция clear\_builds:

1. Удаление всех значений из бд.

Функция closeEvent:

1. Закрытие окна.

Класс QMyLessons:

Функция closeEvent:

1. Закрытие окна.

Функция add\_subject:

1. Добавление в базу данных значения name.
2. Вызов функции select\_data.
3. Вызов функции update\_combo\_box

Функция select\_data:

1. Добавление значений из базы данных в tableLessons.

Функция update\_combo\_box:

1. Добавление значений из базы данных в combo\_del\_subject.

Функция clear\_all:

1. Удаление всех значений из базы данных.
2. Вызов функции update\_combo\_box.

Функция del\_dubject\_name:

1. Удаление из базы данных значения del\_subject\_name.
2. Вызов функции select\_data.
3. Вызов функции updata\_combo\_box:

Класс QChange\_plan:

Функция update\_combo\_files:

1. Обновление списка названий файлов.

Функция update\_combo\_choice:

1. Обновление списка предметов из учебного плана.

Функция add\_subject:

1. Добавление предмета и его количества в учебный план.

Функция new\_file:

1. Создание копии файла с учебным файлом.

Функция open\_file:

1. Запись значений файла в tableWidget.

Функция done\_1:

1. Получение массива предметов.
2. Изменение значений массива.

Функция write\_into\_file:

1. Перезапись файла с обновленными значениями.

Функция all\_done:

1. Перезапись учебного плана или сохранение в новый файл.

Класс MyClassTeachers:

Функция update\_combo\_box:

1. Добавление классов в combo\_classes.

Функция change:

1. Получение списка значений из бд.

Функция load\_to\_table:

1. Запись значений в tableWidget.

Функция del\_subject:

1. Удаление учебных часов с данным учителем по выбранному предмету.

Класс MyClasses:

Функция update\_build:

1. Добавление наименований зданий из бд.

Функция update\_school\_plans:

1. Получение школьных планов из бд.

Функция open\_plan:

1. Добавление значений в список из учебного плана.
2. Запись значений в table\_plan.

Функция add\_class:

1. Добавление класса в бд.

Функция select\_data:

1. Запись значений в tableClasses.
2. Удаление значений из table\_plan.

Класс MyCreate\_workload:

Функция update\_combo\_box:

1. Обновление значений QcomboBox().

Функция done\_1:

1. Запись значений в массив.

Функция select\_to\_table:

1. Запись значений в tableWidget.

Функция apply:

1. Получение списка классов по заданным параметрам.
2. Создание групп кнопок.

Функция save:

1. Запись значений в бд.

Класс MyDefiniteTeacher:

Функция choice\_teacher:

1. Поиск учителя в базе данных.

Функция load\_to\_table:

1. Запись значений в tableWidget.

Функция updata\_combo\_box:

1. Добавление списка учителей в combo\_teachers\_full\_name.

Функция update\_classes\_and\_subjects:

1. Добавление в combobox наименований классов и предметов.

Функция del\_load:

1. Получение значений класса и предметов.
2. Удаление у учителя урока у данного класса.

Класс My\_del\_classes:

Функция update\_combo\_box:

1. Добавление наименований классов в combo\_class.

Функция select\_data:

1. Запись значений в tableWidget.

Функция del\_class:

1. Удаление класса из бд.

Класс MyDelTeachers:

Функция update\_combo\_box:

1. Добавление ФИО учителя в combo\_SNP.

Функция loadTable:

1. Запись значений в tableWidget.

Функция del\_teacher:

1. Удаление учителя из бд.

Класс MyExport\_workload:

Функция export:

1. Получение данных об учителе.
2. Запись данных в word.

Класс QNew\_teachers:

Функция my\_builds:

1. Добавление чекбоксов со зданиями в scrollArea.

Функция get\_builds:

1. Извлечение наименований зданий из бд.

Функция update\_combo\_box:

1. Добавление наименований предметов в combo\_subjects.

Функция add\_subject:

1. Добавление предмета в список предметов.

Функция del\_subject:

1. Удаление предмета из списка предметов.

Функция all\_done:

1. Проверка на корректность введенных данных.
2. Сохранение данных в бд.

Функция set\_to\_file:

1. Запись данных в tableWidget.

Класс QMy\_school\_plan:

Функция add\_lesson:

1. Добавление предмета и количество часов в массив.

Функция del\_lesson:

1. Удаление предмета из массива.

Функция update\_combo\_del:

1. Добавление списка предметов в combo\_del\_lesson.

Функция write\_lessons:

1. Запись массива в tableWidget.

Функция save\_plan:

1. Сохранение файла в формате csv.

Класс MyWorkloadProblems:

Функция problem:

1. Добавление информации из класса.
2. Выявление проблемных предметов класса.
3. Запись проблем в массив

Функция load\_to\_table:

1. Запись значений массива в tableWidget.

Класс MyChange\_teachers:

Функция add\_subject:

1. Добавление предмета, который ведет учитель.

Функция del\_subject:

1. Удаление предмета, который ведет учитель.

Функция add\_build:

1. Добавления здания, в котором преподает учитель.

Функция del\_build:

1. Удаление здания, в котором преподает учитель.

Функция apply:

1. Открытие кнопок, для редактирования данных учителя.

Функция get\_id:

1. Получение id учителя.

Функция load\_subjects:

1. Выведение предметов, которые ведет учитель, в таблицу.

Функция load\_builds:

1. Выведение зданий, в которых ведет учитель, в таблицу.

Функция update\_combo\_box:

1. Добавление в combobox ФИО учителей.

Функция all\_done:

1. Очищение таблиц.
2. Применение всех изменений.

Применение технологий: в данном приложении используется база данных, а также фаллы формата csv и docx.

Необходимые библиотеки:

1. Pyqt5
2. Sqlite3
3. Csv
4. Docx



