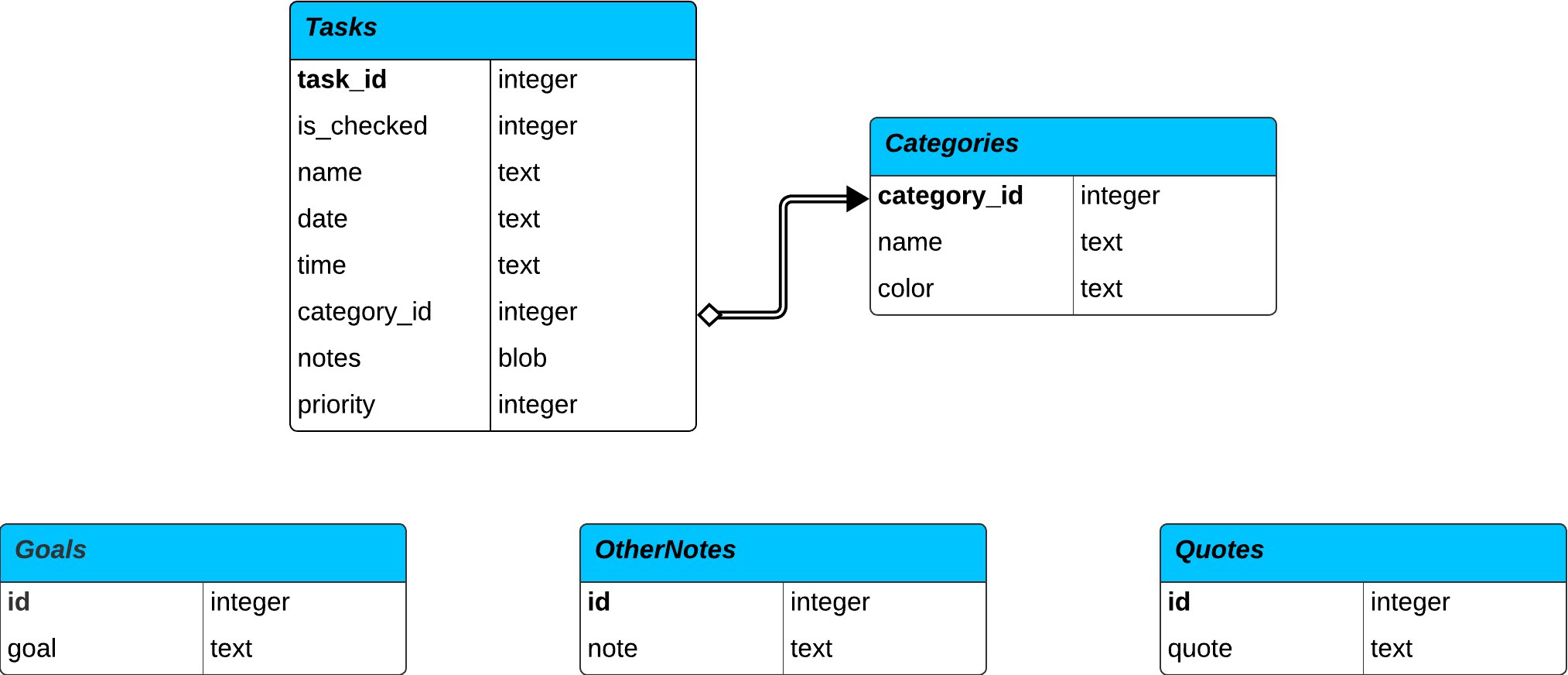
***Пояснительная записка к проекту “Планировщик задач”***

**Автор**: Тузова Дарья.

**Объем программного кода**: 625 строк.

**Идея**: создать программу, которая будет помогать организовывать повседневную жизнь.

**База данных**:



**Пояснения к приложению**:

Состоит из 3 вкладок: “Планы”, “Все задачи”, “Категории” (QTabWidget)

При открытии приложения показывается первая вкладка (“Планы”) с задачами на текущий день.

Заголовок “Планировщик задач” (QLabel) - heading.

Style Sheet:

background-color: rgb(0, 0, 0);

color: rgb(0, 196, 255);

Цитата (QLabel) - label\_quote.

Style Sheet:

background-color: rgb(0, 0, 0);

color: rgb(0, 196, 255);

На вкладке “Планы” находятся:

* Календарь (QCalendarWidget)
* Надпись “Задачи на …” (заместо … - день) (QLabel)
* Задачи на день, указанный в календаре в формате таблицы (QTableWidget)
* Кнопки для таблицы задач на день (строка выше) (QPushButton): “Добавить задачу”, “Сохранить изменения”, “Изменить задачу”, “Удалить задачу”
* Надпись “Просроченные задачи к …” (заместо … - день) (QLabel)
* Просроченные задачи к текущему дню в формате таблицы (QTableWidget)
* Кнопки для таблицы просроченных задач (строка выше) (QPushButton): “Сохранить изменения”, “Изменить задачу”, “Удалить задачу”
* Шкала прогресса для дня, указанного в календаре (QProgressBar)

Календарь - calendar.

Style Sheet:

alternate-background-color:rgb(0, 196, 255)

selection-background-color: rgb(0, 0, 0)

selection-color: rgb(255, 255, 255)

Надпись “Задачи на …” - label\_tasks\_date.

Style Sheet:

background-color: rgb(0, 0, 0)

color: rgb(255, 255, 255)

Задачи на день, указанный в календаре в формате таблицы - table\_tasks\_date.

Отображается: сделана задача или нет, сама задача, время задачи, категория задачи, заметки к задаче.

Кнопки для таблицы задач на день c Style Sheet: background-color: rgb(0, 196, 255):

* “Добавить задачу” - add\_tasks\_date
* “Сохранить изменения” - save\_tasks\_date
* “Изменить задачу” - change\_tasks\_date
* “Удалить задачу” - delete\_tasks\_date

При добавлении задачи открывается окно, в которое можно записать информацию по задаче. (описание позже)

Для того чтобы использовать кнопку save\_tasks\_date, нужно изменить какие-либо данные в таблице table\_tasks\_date и нажать на нее. Все преобразования сразу сохранятся.

Если надо поменять по задаче информацию, которая не показывается в save\_tasks\_date, то надо выделить ее в таблице table\_tasks\_date, нажать на кнопку change\_tasks\_datе, потом высветится аналогичное окно, что и при добавлении задачи, но уже с заполненными данными, которые можно изменить.

Чтобы удалить задачу, для начала необходимо ее выделить в таблице table\_tasks\_date, и нажать на кнопку delete\_tasks\_date.

Надпись “Просроченные задачи к …” - label\_tasks\_after\_srok.

Style Sheet:

background-color: rgb(0, 0, 0)

color: rgb(255, 255, 255)

Просроченные задачи к текущему дню в формате таблицы - table\_tasks\_after\_srok.

Отображается: сделана задача или нет, сама задача, категория задачи, заметки к задаче.

Под просроченными задачами подразумеваются невыполненные к текущему дню.

Кнопки для таблицы задач на день c Style Sheet: background-color: rgb(0, 196, 255):

* “Сохранить изменения” - save\_tasks\_after\_srok
* “Изменить задачу” - change\_tasks\_after\_srok
* “Удалить задачу” - delete\_tasks\_after\_srok

Эти кнопки работают аналогично с предыдущими, только используются таблица table\_tasks\_after\_srok и соответственно кнопки, описанные в списке выше.

К каждой таблице относятся свои кнопки, они находятся над или под ней. То есть нельзя выделить задачу в одной таблице, а использовать кнопку, относящуюся к другой.

Если что-то было введено в неверном формате или кнопка не сработала, то или выводится сообщение об ошибке или в левом нижнем углу появляется подсказка (QStatusBar - statusbar).

Шкала прогресса для дня, указанного в календаре - progress\_of\_tasks\_date.

Style Sheet:

QProgressBar

{

color: rgb(255, 255, 255);

background-color : rgba(0, 0, 0, 255);

border : 1px

}

QProgressBar::chunk

{

background-color: rgb(0, 196, 255);

}

Отображается процент выполненных задач на всех.

На вкладке “Все задачи” находятся:

* Все задачи в формате таблицы (QTableWidget)
* Кнопки для изменения таблицы всех задач (строка выше) (QPushButton): “Добавить задачу”, “Сохранить изменения”, “Удалить задачу”
* Параметр сортировки таблицы всех задач (QComboBox)
* Кнопка для сортировки таблицы всех задач (QPushButton)
* Надпись “Цели” (QLabel)
* Поле для целей (QTextEdit)
* Кнопка для сохранения целей (QPushButton)

Все задачи в формате таблицы - table\_tasks\_all.

Отображается: сделана задача или нет, сама задача, дата задачи, время задачи, категория задачи, заметки к задаче, приоритет.

Кнопки для изменения таблицы всех задач c Style Sheet: background-color: rgb(0, 196, 255):

* “Добавить задачу” - add\_tasks\_all
* “Сохранить изменения” - save\_tasks\_all
* “Удалить задачу” - delete\_tasks\_all

Эти кнопки работают аналогично с предыдущими, только используются таблица table\_tasks\_all и соответственно кнопки, описанные в списке выше.

Параметр сортировки таблицы всех задач - choice\_sort\_tasks\_all.

Критериями для сортировки являются: сделана задача или нет, сама задача, дата задачи, время задачи, категория задачи, заметки к задаче, приоритет.

Кнопка для c сортировки таблицы всех задач с Style Sheet: background-color: rgb(0, 196, 255) - sort\_tasks\_all.

Таблица table\_tasks\_all сортируется соответственно параметру choice\_sort\_tasks\_all.

Есть три вида сортировки:

* по алфавиту / по числам
* по трем значениям (“Все”, ”Да“, ”Нет”)
* по нескольким значениям

И каждый из них применяется к определенным критериям.

По алфавиту сортируется по столбцу “Задача” и “Заметки”, по числам - “Время”.

По трем значениям сортируется по столбцу “✓” (сделана задача или нет) и “Приоритет”. Когда выбрано “Все”, выводится вся таблица table\_tasks\_all с задачами, сгруппированными по одинаковому значению, то есть сначала невыполненные, потом сделанные или сначала с приоритетом, а потом без него. При ”Да“ показываются задачи или выполненные, или с приоритетом в зависимости от критерия, а при “Нет” - несделанные или без приоритета.

По нескольким значениям сортируется по столбцу “Дата” и “Категория”. Когда выбрано “Все”, выводится вся таблица table\_tasks\_all с задачами либо по возрастанию даты, или по порядку категорий в таблице table\_categories. Остальные значения можно выбрать из существующих, то есть дату, на которую запланированы дела, или категорию, к которой относятся задачи.

Надпись “Цели” - label\_goals.

Style Sheet:

background-color: rgb(0, 0, 0)

color: rgb(255, 255, 255)

Поле для целей - text\_goals.

В него можно записать любые цели, но и надо сохранить.

Кнопка для сохранения целей с Style Sheet: background-color: rgb(0, 196, 255) - save\_goals.

Если не нажать на нее, то цели не сохранятся, и при следующем использовании приложения они не отобразятся.

На вкладке “Категории” находятся:

* Все категории в формате таблицы (QTableWidget)
* Кнопки для изменения таблицы всех категорий (строка выше) (QPushButton): “Добавить категорию”, “Изменить категорию”, “Удалить категорию”
* Надпись “Заметки” (QLabel)
* Поле для целей (QTextEdit)
* Кнопка для сохранения целей (QPushButton)
* Надпись “Цитаты” (QLabel)
* Поле для цитат (QTextEdit)
* Кнопка для сохранения цитат (QPushButton)

Все категории в формате таблицы - table\_categories.

Отображается: сама категория, ее цвет.

Кнопки для изменения таблицы всех задач c Style Sheet: background-color: rgb(0, 196, 255):

* “Добавить категорию” - add\_category
* “Изменить категорию” - change\_category
* “Удалить категорию” - delete\_category

Эти кнопки работают аналогично с предыдущими, только используются таблица table\_categories и соответственно кнопки, описанные в списке выше, также при добавлении и удалении категории открывается другое окно. (описание позже)

Надпись “Заметки” - label\_notes.

Style Sheet:

background-color: rgb(0, 0, 0)

color: rgb(255, 255, 255)

Поле для целей - text\_notes.

В него можно записать любые заметки, но и надо сохранить.

Кнопка для сохранения заметок Style Sheet: background-color: rgb(0, 196, 255) - save\_notes.

Принцип работы точно такой же, как и у кнопки save\_goals.

Надпись “Цитаты” - label\_quotes.

Style Sheet:

background-color: rgb(0, 0, 0)

color: rgb(255, 255, 255)

Поле для цитат - text\_quotes.

В него можно записать любые цитаты, но и надо сохранить.

Кнопка для сохранения цитат Style Sheet: background-color: rgb(0, 196, 255) - save\_quotes.

Принцип работы точно такой же, как и у кнопки save\_goals и save\_notes.

Записанные цитаты показываются внизу приложения - label\_quote. При перелистывании вкладок они меняются в рандомном порядке.

Окно для добавления задачи состоит из:

* Надпись “Название” (QLabel)
* Поле для названия задачи (QLineEdit)
* Надпись “Категория” (QLabel)
* Поле выбора для категории задачи (QComboBox)
* Надпись “Дата” (QLabel)
* Поле выбора для даты задачи (QDateEdit)
* Надпись “Время” (QLabel)
* Поле для времени задачи (QLineEdit)
* Надпись “Заметки” (QLabel)
* Поле для заметок задачи (QPlainTextEdit)
* Надпись “Приоритет” (QLabel)
* Кнопки для выбора приоритета задачи (“Да”, “Нет”) (QRadioButton)
* Кнопка для сохранения задачи (QPushButton)
* Надпись “...” (“Сохранено”) (QLabel)

Изначально все данные пустые кроме категории, даты, приоритета.

Обязательно нужно ввести название, и если надо указать время, то сделать это в формате “часы:минуты”, иначе показывается сообщение об ошибке.

Надпись “Название” - label\_name\_task.

Style Sheet:

background-color: rgb(0, 0, 0)

color: rgb(255, 255, 255)

Поле для названия задачи - name\_task.

Надпись “Категория” - label\_category\_task.

Style Sheet:

background-color: rgb(0, 0, 0)

color: rgb(255, 255, 255)

Поле выбора для категории задачи - combo\_categories\_task.

Надпись “Дата” - label\_date\_task.

Style Sheet:

background-color: rgb(0, 0, 0)

color: rgb(255, 255, 255)

Поле выбора для даты задачи - date\_task.

Надпись “Время” - label\_date\_task.

Style Sheet:

background-color: rgb(0, 0, 0)

color: rgb(255, 255, 255)

Поле для времени задачи - time\_task.

Надпись “Заметки” - label\_notes\_task.

Style Sheet:

background-color: rgb(0, 0, 0)

color: rgb(255, 255, 255)

Поле для заметок задачи - text\_notes\_task.

Надпись “Приоритет” - label\_priorty

Style Sheet:

background-color: rgb(0, 0, 0)

color: rgb(255, 255, 255)

Кнопки для выбора приоритета задачи:

* “Да” - priorty\_yes\_task
* “Нет” - priorty\_not\_task

По умолчанию стоит кнопка priorty\_not\_task.

Кнопка для сохранения задачи с Style Sheet: background-color: rgb(0, 196, 255) - save\_task.

Надпись “...” (“Сохранено”) - label\_save.

Style Sheet:

background-color: rgb(0, 0, 0)

color: rgb(255, 255, 255)

Если в label\_save высветилось “Сохранено”, то кнопка save\_task сработала.

Окно для изменения задачи состоит из того, что и окно для добавления задачи (используется один виджет), но также есть:

* Кнопки для завершенности задачи (“Сделано”) (QCheckBox)
* Надпись “Номер” (QLabel)
* Надпись “...” - номер задачи (QLabel)

Все данные по задаче уже заполнены.

Кнопки для завершенности задачи (“Сделано”) - checkBox\_task.

Style Sheet:

background-color: rgb(0, 0, 0)

color: rgb(255, 255, 255)

Надпись “Номер” - label\_number\_task.

Style Sheet:

background-color: rgb(0, 0, 0)

color: rgb(255, 255, 255)

Надпись “...” - номер задачи - label\_id\_task.

Style Sheet:

background-color: rgb(0, 0, 0)

color: rgb(255, 255, 255)

Окно для добавления категории состоит из:

* Надпись “Название” (QLabel)
* Поле для названия задачи (QLineEdit)
* Надпись “Цвет” (QLabel)
* Кнопки для выбора цвета категории (QPushButton)
* Кнопка для сохранения категории “Сохранить” (QPushButton)
* Надпись “...” (“Сохранено”) (QLabel)

Надпись “Название” - label\_name.

Style Sheet:

background-color: rgb(0, 0, 0)

color: rgb(255, 255, 255)

Поле для названия задачи - text\_name\_category.

Нельзя, чтобы названия категорий повторялись.

Надпись “Цвет” - label\_color.

Style Sheet:

background-color: rgb(0, 0, 0)

color: rgb(255, 255, 255)

Кнопки для выбора цвета категории:

* “Рандомный цвет” - random\_color
* “Выбрать цвет” - choice\_color

При нажатии на random\_color цвет кнопки будет изменяться на случайный.

При нажатии на choice\_color высвечивается окно выбора цвета и можно использовать любой из них.

Кнопка для сохранения категории “Сохранить” с Style Sheet: background-color: rgb(0, 196, 255) - save\_category.

Надпись “...” (“Сохранено”) - label\_save.

Style Sheet:

background-color: rgb(0, 0, 0)

color: rgb(255, 255, 255)

Если в label\_save высветилось “Сохранено”, то кнопка save\_category сработала.

Окно для изменения категории состоит из того, что и окно для добавления задачи (используется один виджет), но данные уже заполнены.

**Пояснения к коду:**

Константы:

* NAMES\_COLUMNS - названия колонок таблиц с задачами
* SOOT - соответствие названий колонок с названиями в базе данных

Три класса:

* Planner(QMainWindow, , Ui\_MainWindow)
* AddOrChangeTask(QMainWindow, Ui\_MainWindow1)
* CategoryWindow(QMainWindow, Ui\_MainWindow2)

Методы Planner(QMainWindow, Ui\_MainWindow):

* self.\_\_init\_\_()
* self.initUI()
* self.save\_quotes\_to\_data() - для цитат
* self.save\_notes\_to\_data(): - для заметок
* self.save\_goals\_to\_data() - для целей
* self.save\_for\_quotes\_notes\_goals(pole, elem\_in\_data, data) - сохраняем заметки / цитаты / цели в базу данных
* self.set\_quotes() - для цитат
* self.set\_notes() - для заметок
* self.set\_goals() - для целей
* self.set\_for\_quotes\_notes\_goals(pole, elem\_in\_data, data)- Отображаем в pole заметки / цитаты / цели
* self.quotes()- отображение цитат в self.label\_quote
* self.sortirovka() - выбор сортировки
* self.choice\_dialog(text) - сортировка по нескольким значениям
* self.yes\_or\_not\_dialog(text)- сортировка по трем значениям
* self.calendar\_changed() - реакция на изменение даты в self.calendar
* self.date\_today()- сегодняшняя дата
* self.preporation\_usual\_for\_select\_data()- для таблицы table\_tasks\_date
* self.preporation\_prosrok\_for\_select\_data()- для таблицы table\_tasks\_after\_srok
* self.preporation\_all\_for\_select\_data(condition='', for\_sort=lambda x: (-x[0], -x[-1], x[1]))- для таблицы table\_tasks\_all
* self.select\_data\_all(table, condition, for\_sort, unvisible) - отображение таблиц задач
* self.color\_row(table, row, color) - красим row в color
* self.select\_data\_category()- отображение таблиц категорий
* self.selects() - обновление таблиц
* self.delete\_usual\_for\_delete\_all() - для таблицы table\_tasks\_date
* self.delete\_prosrok\_for\_delete\_all() - для таблицы table\_tasks\_after\_srok
* self.delete\_all\_for\_delete\_all()- для таблицы table\_tasks\_all
* self.for\_delete\_all(table) - проверка выделенной строки в table и подготовка для self.delete\_all(id\_task)
* self. delete\_all(id\_task) - удаление из таблиц с задачами
* self.delete\_category\_from\_data(self)- удаление из таблицы с категориями
* self.save\_usual\_for\_save\_all()– сохранение для таблицы table\_tasks\_date
* self.save\_prosrok\_for\_save\_all() – сохранение для таблицы table\_tasks\_after\_srok
* self.save\_all\_for\_save\_all() – сохранение для таблицы table\_tasks\_all
* self.save\_all(table) - сохранение изменений в таблице table
* self. - проверка на соответствие формата task, time, category, dates, priority
* self.status\_bar(self, row, what, act)- сообщение об ошибки в self.statusbar
* self.check\_two(task, time) - проверка на соответствие формата task, time
* self.check(task, time, category, dates, priority) - проверка на соответствие формата task, time, category, dates, priority
* self.message\_box\_delete - подтверждение удаления через QMessageBox
* self.message\_box(mess) - сообщение об ошибке через QMessageBox с текстом mess
* self.combo\_sort()- критерии для сортировки в self.choice\_sort\_tasks\_all
* self.progress() - изменение прогресса
* self.add\_or\_change\_task()- открытие окна, позволяющего добавить / изменить задачу
* self.add\_or\_change\_category() - открытие окна, позволяющего добавить / изменить категорию
* self.closeEvent(event)- разрывает связь с базой данных при закрытии приложения

Методы AddOrChangeTask(QMainWindow, Ui\_MainWindow1):

* self.\_\_init\_\_(action, parent=None)
* self.initUI\_for\_add() - initUI для добавления
* self.initUI\_for\_change() - initUI для изменения
* self.check() - проверка на соответствие формата
* self.save\_for\_add() - для добавления
* self.save\_for\_change() - для изменения
* self.save\_all(query)- cохранение задачи в базу данных
* self.combo() - установка категорий в self.combo\_categories\_task
* self.closeEvent(event)- разрывает связь с базой данных при закрытии приложения

Методы CategoryWindow(QMainWindow, Ui\_MainWindow2):

* self.\_\_init\_\_(action, parent=None)
* self.initUI\_for\_add() - initUI для добавления
* self.initUI\_for\_change() - initUI для изменения
* self.random\_colorr() - рандомный выбор цвета
* self.set\_color(сolor) - установка color
* self.run() - открытие окна с цветами
* self.check() - проверка на соответствие формата
* self.save\_for\_add() - для добавления
* self.save\_for\_change() - для изменения
* self.save\_all(query)- cохранение категории в базу данных
* self.closeEvent(event)- разрывает связь с базой данных при закрытии приложения