

Федеральное государственное автономное образовательное
учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра «Информационные системы»

УТВЕРЖДАЮ
Зав. кафедрой ИС
 А. И. Легалов
« » 2016 г.

БАКАЛАВРСКАЯ РАБОТА

090301.02 Информатика и вычислительная техника
«Мультиплатформенный ассистент для работы с компьютером»
Тема

Руководитель _____
подпись, дата

О.В. Шмелев
инициалы, фамилия

Выпускник _____
подпись, дата

В.П. Ларионов
инициалы, фамилия

Нормоконтролер _____
подпись, дата

В.И. Иванов
инициалы, фамилия

Красноярск 2016

СОДЕРЖАНИЕ

Введение	4
1. Подготовка к разработке программного обеспечения	5
1.1. Анализ предметной области	5
1.2. Описание инструментария	5
1.3. Техническое задание	6
1.3.1. Основное окно.....	6
1.3.2. Выполнитель	7
1.3.3. Менеджер сеансов.....	8
1.3.4. Окно напоминания.....	8
1.3.5. Окно настроек	8
2. Разработка приложения.....	12
2.1. Класс MainForm	12
2.2. Класс Executer	14
2.3. Класс MyButton.....	16
2.4. Класс Commands	16
2.5. Класс Statement	17
2.6. Класс Skins	18
2.7. Класс Settings	18
2.8. Класс SkinLable.....	22
2.9. Класс RemainForm	23
2.10. Класс CommandForm	25
2.11. Класс MultyCommand	26
2.12. Класс commandList	27

3. Руководство пользователя	28
3.1. Использование главного окна программы	28
3.2. Использование исполнителя	29
3.3. Использование напоминаний	33
3.4. Использование менеджера сеансов	34
Заключение	36
Список использованных источников	37

ВВЕДЕНИЕ

Автоматизация процессов часто ведет к убыстрению и увеличению качества работы человека.

Автоматизация так же не прошла стороной мимо процесса использования компьютера. В настоящее время уже создано несколько программных продуктов, которые помогают пользователю с помощью электронных средств. Некоторые из них с помощью оповещений напоминают людям о важных делах, другие помогают пользователю через более короткий путь запускать нужные ему программы и файлы.

Так же, некоторым любителям игр и фильмов приятно видеть своего любимого персонажа у на экране. Хоть это и уменьшает рабочую область экрана, люди идут на подобные жертвы ради этого персонажа.

И в данной выпускной квалификационной работе (далее ВКР) было создано специальное программное обеспечение под названием С.А.Project. Функционал С.А.Project был сформулирован благодаря опросу, в котором мои знакомые могли предоставить требования к будущей программе. После анализа их требований было создано техническое задание, которое описано ниже.

1 Подготовка к разработке программного обеспечения

1.1 Анализ предметной области

Вообще, после анализа интернета, был найден один аналог, схожий по функционалу с C.A.Project. Это - The SAO Utilities Project [1]. Эта программа позволяет пользователю создавать свои команды, такие как:

- перейти на сайт;
- открыть файл;
- открыть картинку;

Так же эта программа позволяет пользователю подписаться на рассылку сайта. Так же она имеет довольно удобный интерфейс. Однако в этой программе у пользователя нет возможности выполнять команды по времени, так же нет системы напоминаний. Собственно, этот функционал и будет добавлен в C.A.Project.

Так же, The SAO Utilities Project может работать только на операционных системах семейства Windows. C.A.Project лишен этого недостатка, так как он будет написан с помощью Qt – фреймворка для создания кроссплатформенных приложений.

1.2 Описание инструментария

Программа была написана с помощью языка программирования C++. Этот язык обладает большой скоростью и большой гибкостью [2]. Так же это объектно-ориентированный язык, который идеально подходит для создания этого приложения.

C.A.Project реализован с помощью фреймворка Qt [3]. Одна из самых важных особенностей этого фреймворка – кроссплатформенность [4]. Так же с помощью Qt Designer можно легко создать нужный графический интерфейс [5]. Для определения стандартного цвета программы использовался сайт

ColorScheme [6]. Этот сайт часто дизайнеры используют для подбора цветовых схем.

Однако, далеко не весь функционал программы можно сделать кросс-платформенным. Для такого рода вещей есть специальный синтаксис [7], чтоб писать разный код для разной операционной системы. Для операционной системы Windows использовались функции из Windows API [8].

И конечно же, код был написан в стиле, как советовалось в книге «Совершенный код» [9].

Для удобного отображения структуры классов была использована диаграмма классов UML [10].

1.3 Техническое задание

1.3.1 Основное окно

Основное окно отображает картинку, которую выберет или загрузит пользователь. Размер картинки изменяется в окне настроек.

При двойном клике по окну должно показываться окно исполнителя.

При зажатии левой кнопки мыши по окну и перемещении перемещается и основное окно программы.

При нажатии на основное окно правой кнопкой мыши показывается контекстное меню, которое содержит следующие пункты:

- Поверх всех окон – чек-бокс. Если галочка установлена, то программа всегда отображается поверх других окон;
- Использовать исполнитель – чек-бокс. Если галочка установлена, то при наведении мышью на правую часть экрана, будет показываться кнопка для запуска исполнителя;
- Показывать напоминания – чек-бокс. Если галочка установлена, то приложение будет показывать напоминания;
- Создать напоминание – кнопка для вызова окна создания напоминания;

- Настройки – кнопка для вызова окна настроек;
- О программе – кнопка для открытия *.html файла с информацией о программе и мануалом;
- Дополнительно – подменю, которое содержит следующие элементы:
 - Подписаться на новости – кнопка для перехода на страничку группы приложения на сайте vk.com;
 - Оставить отзыв или предложение - кнопка для перехода на страничку поста группы приложения на сайте vk.com;
 - ARProject@yandex.ru – электронный почтовый ящик для писем проекта.
- Свернуть – кнопка, по нажатию которой приложение сворачивается;
- Выход – кнопка, по нажатию которой приложение закрывается;

1.3.2 Выполнитель

Выполнитель является средством для запуска некоторых приложений и команд. Выполнитель запускается через основное окно двойным кликом левой кнопки мыши, а также через специальную кнопку, которая показывается, если пользователь навел курсором на правый край экрана.

Команды для исполнителя создаются в окне настроек. Пользователь сам выбирает, что команда должна делать, дает ей название и настраивает другие параметры команды.

Окно исполнителя содержит следующие элементы:

- Список режимов;
- Кнопка для выхода из исполнителя;
- Кнопка для открытия окна настроек;
- Строка ввода команды с автозаполнением;
- Строка состояния выполнения команды;

- Список команд для запуска мышью.

1.3.3 Менеджер сеансов

Менеджер сеансов контролирует активные приложения компьютера. А именно: пользователь добавляет запрещенные приложения в определенный сеанс, и он не может пользоваться этими приложениями во время сеанса. Приложение активно, если на него направлен фокус. Пользователь может менять сеанс с помощью исполнителя, а так же добавлять их с помощью окна настроек. Так же пользователь может добавлять и удалять запрещенные приложения для сеанса.

Существует три режима:

- Легкий – показывает напоминание о том, что вы используете запрещенное приложение;
- Умеренный – сворачивает запрещенное активное приложение;
- Хардкор – закрывает запрещенное активное сообщение;

1.3.4 Окно напоминания

Окно с текстом напоминания. Содержит текст напоминания, кнопка для закрытия напоминания. При нажатии на окно останавливается мелодия напоминания. Окно напоминания по умолчанию появляется над основным окном. Если место не хватает, то окно напоминания передвигается около основного окна, чтоб его было полностью видно. От окна напоминания до основного тянется «стрелка».

1.3.5 Окно настроек

Окно настроек содержит четыре вкладки:

- 1) Исполнитель;
- 2) Напоминания;
- 3) Менеджер сеансов;
- 4) Скинны.

На вкладке «Выполнитель» расположены следующие кнопки:

- Добавить – кнопка для добавления команды. При нажатии этой кнопки появляется окно настройки команды. В этом окне доступно поле ввода имени команды, а также выпадающий список, который определяет что команда будет делать. Список содержит следующие строки:
 - Запустить программу – выбирается файл программы;
 - Открыть файл – выбирается файл, который надо открыть;
 - Открыть папку – выбирается папка;
 - Открыть сайт – в поле ввода вводится URL нужного сайта;
 - Выключить компьютер;
 - Перезагрузить компьютер;
 - Переключить в спящий режим;
- Добавить многозадачную команду – кнопка для добавления многозадачной команды. При нажатии открывается окно создания команды. В этом окне есть поле ввода для название команды, кнопка «Добавить команду», «Удалить команду», «Сохранить» и «Отмена». Так же есть поле числового ввода для определения интервала между запуском команд;
- Редактировать – при нажатии на кнопку появляется окно для редактирования команды. Окно такое же, как для создания команды;
- Удалить – при нажатии на кнопку удаляется выбранная команда.

На вкладке «Напоминания» доступен список с существующими напоминаниями, а так же текстовое с описанием текущего напоминания. Так же есть следующие кнопки:

- Добавить – вызывает открытие окна для создания напоминания. Это окно содержит в себе:
 - Название напоминания;

- Выпадающий список с режимом напоминания (один раз, каждый день, в некоторые дни);
- Поле ввода времени напоминания;
- Поле ввода даты напоминания;
- Поле ввода текста напоминания;
- Выпадающий список с мелодией напоминания;
- Кнопка для загрузки мелодии напоминания (при загрузке звуковой файл копируется в место установки программы);
- Кнопки для прослушивания напоминания;
- Регулятор громкости мелодии напоминания;
- Кнопка для выбора команды, которая будет выполняться с напоминанием;
- Чек-бокс с выбором режима напоминания (показывать окно напоминания или нет);
- Текстовое поле с кратким обзором напоминания;
- Кнопка для сохранения напоминания;
- Кнопка для отмены создания напоминания;
- Редактировать – вызывает показ окна настройки напоминания (такое же, как при создании напоминания);
- Удалить – кнопка для удаления выделенного напоминания;
- Изменить цвет – изменяет цвет окна напоминания;
- Изменить прозрачность – изменяет прозрачность окна напоминания;
- Изменить цвет текста – изменяет текст в окне напоминания;

На вкладке «Менеджер сеансов» пользователь видит кнопку для создания и удаления сеанса, а также выпадающий список для выбора сеанса. Так же пользователю доступен выпадающий список с режимом работы сеанса. Ниже доступны кнопки для добавления запрещенной программы и удаления ее из

сеанса. Так же пользователь может видеть список запрещенных программ для текущего сеанса.

На вкладке «Скины» расположены следующие элементы:

- Миниатюра текущего скина с шестиугольником, с помощью которого он может отобразить положения рта скина (это положение нужно для напоминания);
- Числовое поле ввода данных для изменения масштаба изображения относительно оригинала;
- Справа расположен графический список всех скинов;
- Кнопка удаления скина;
- Кнопка добавления скина.

1.4

2 Разработка приложения

Сначала была разработана архитектура приложения. Эта архитектура представлена на рисунке 2.1.

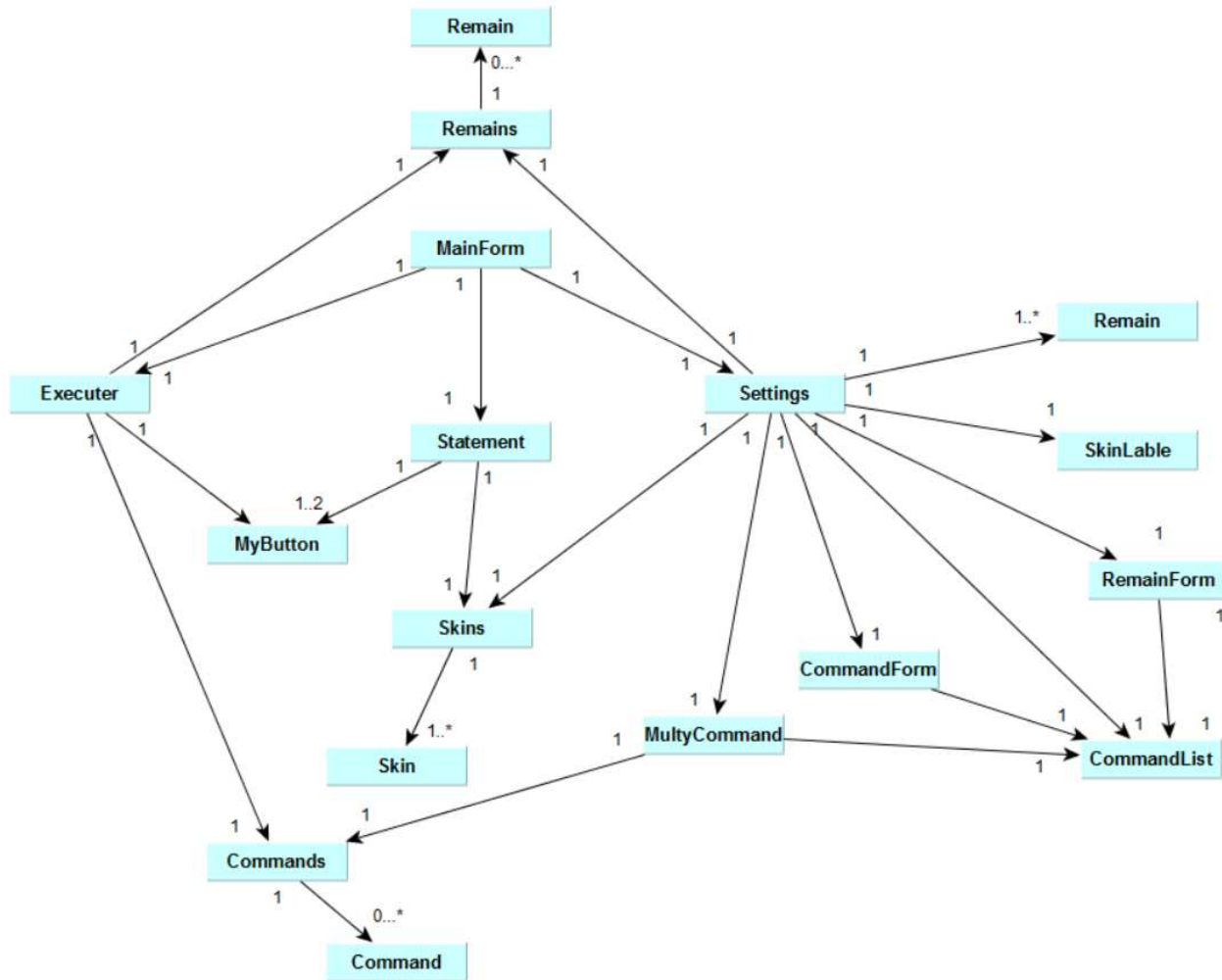


Рисунок 2.1 – Диаграмма классов UML.

Далее подробно рассмотрим каждый класс.

2.1 Класс MainForm

Этот класс является основным в работе программы. Через него открываются настройки, запускается исполнитель и выполняются другие процессы. Так же одна из основных задач этого класса – отображение программы. Именно его мы видим при запуске программы.

Также этот класс содержит контекстное меню для работы с программой.

Основное окно отвечает за работу с запрещенными программами. А именно, оно берет указатель на текущее окно, берет *.exe файл и смотрит есть ли этот файл в списке запрещенных программ. Если в запрещенных программах будет найден этот файл, то в зависимости от режима будут происходить дальнейшие действия программы.

Так же этот класс отвечает за показ напоминаний.

Опишем алгоритм определения является ли текущая секунда временем показа какого-то напоминания.

Так же введем переменные:

- secondsCount – количество секунд от времени напоминания до текущего времени;
- period – период между появления одного напоминания;
- count – количество появлений одного напоминания;

Каждую секунду для всех напоминаний в списке напоминаний проверяем режим напоминания. Если напоминание показывается:

- один раз. Тогда определяем логическое условие: $(secondsCount \% period = 0)$ И $(secondsCount / period \leq count)$ И $(secondsCount \geq 0)$.
(1)

Если условие выполняется, то показываем напоминание;

- некоторые дни. Тогда для каждого дня в списке дней считаем сколько секунд пройдет с того дня до текущего момента, и смотрим выполняется ли условие (1). Если выполняется, то показываем напоминание;
- каждый день. Тогда берется ближайшее время напоминания до текущего момента. От этого времени и считается количество секунд и проверяется выполнение условия (1). Если условие выполнено, то показываем напоминание. Если $(secondsCount / period > count)$, то

делаем тоже самое для предыдущего дня, пока `secondsCount / period` станет меньше либо равно `period`.

2.2 Класс `Executer`

Класс окна исполнителя. Содержит в себе две кнопки типа `MyButton`. Также содержит класс `Commands`, который содержит в себе команды, поле ввода команды, список команд, выпадающий список сеансов и строку состояния. Скрин этого класса изображен на рисунке 2.2.1.

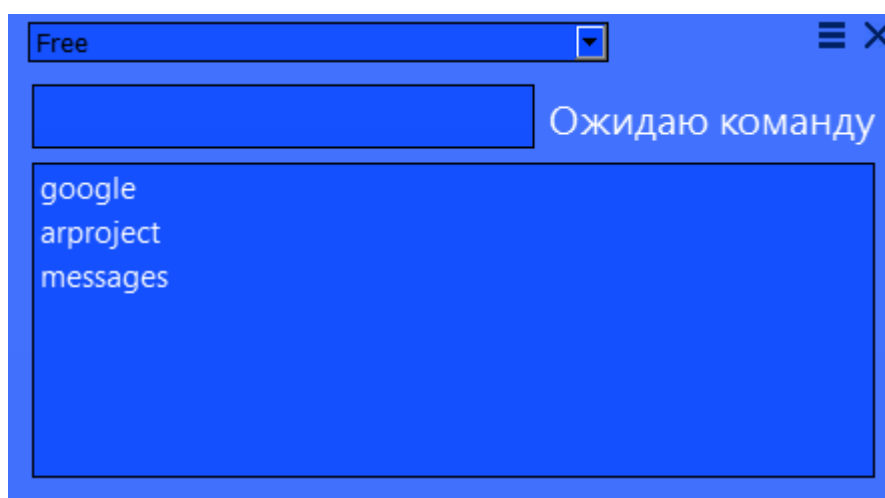


Рисунок 2.2.1 – Скрин окна исполнителя.

Основные методы этого класса:

- `processCommand()` – выполняет команду, название которой ввел пользователь в поле ввода. Выполняется после нажатия клавиши «Enter»;
- `processComandFromList()` – выполняет команду, на которую пользователь нажал в списке команд;
- `processCommandHiden(QString cmd)` – выполняет команду. Используется в предыдущих функциях. А так же при показе напоминания, если там есть команда. Блок схема этого метода представлена на рисунке 2.2.2.

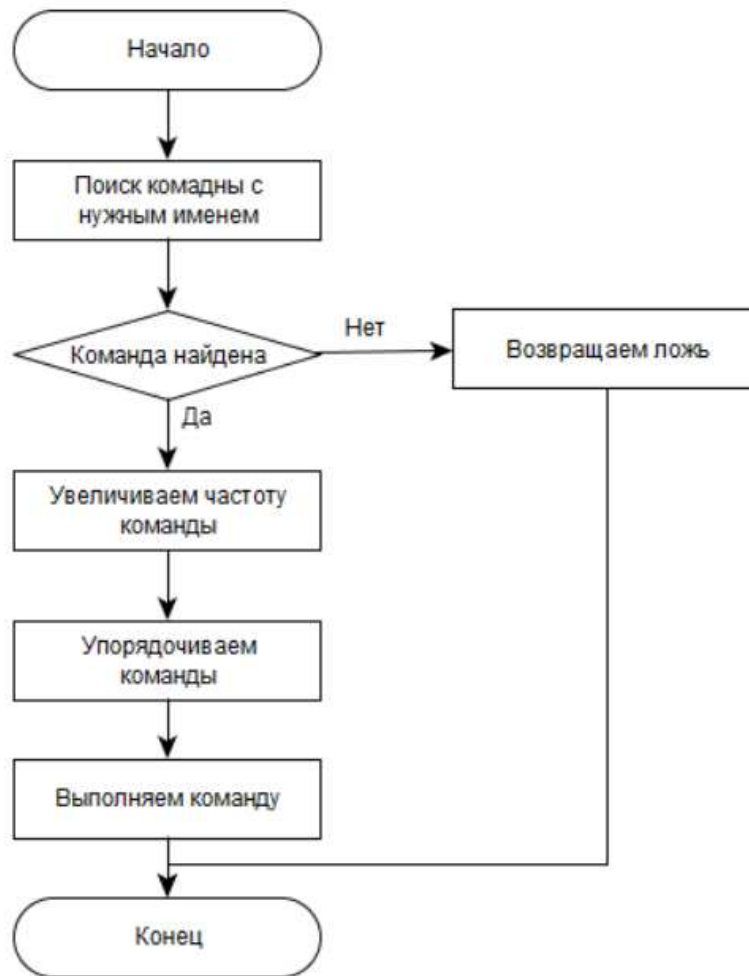


Рисунок 2.2.2 – Блок-схема алгоритма запуска команды

Открытие файла происходит с помощью класса `QDesktopServices`. Он предоставляет метод `openUrl`, который открывает нужный файл или сайт. Выключение компьютера выполняется через исполнитель, переход в спящий режим – через запуск `*.bat` файла.

- `updateCommands()` – обновляет список команд. Так же увеличивает частоту выполненной только что команды;
- `showStandart()` – показывает исполнитель, очищая строку ввода и устанавливая стандартное состояние строки состояния;
- `hideTerminal()` – скрывает окно исполнителя;

2.3 Класс MyButton

Класс, который делает кнопку в форме картинки. При отпускании левой кнопки мыши этого класса генерируется сигнал `clicked`, который сообщает о том, что кнопка была отпущена.

Методы этого класса:

- `setFileName(QString str)` – устанавливает картинку кнопки. `Str` – строка с путем до картинки;
- `mouseReleaseEvent(QMouseEvent *e)` – метод, который вызывается, если пользователь отпускает кнопку мыши. Как раз тут и генерируется сигнал `clicked`.

2.4 Класс Commands

Класс, который создан с использованием шаблона Одиночка. Класс содержит в себе команды программы. Он отвечает за доступ, сохранение и загрузку в текстовой файл команд.

Основные методы этого класса:

- `load` – загружает команды из текстового файла `commands.txt`;
- `save` – сохраняет команды в текстовой файл `commands.txt`;
- `getCommand(int i)` – возвращает команду с номером `i`.

Так же класс содержит описание структуры команды. Код структуры показан на рисунке 2.4.1.

```
struct comand{
    int index;
    int rating;
    int interval;
    QString path;
    QString site;
    QString cmd;
    QStringList cmds;
};
```

Рисунок 2.4.1 – Описание структуры `comand`

2.5 Класс Statement

Класс напоминания. По своей сути является окошком, которое отображает текст, проигрывает музыку и запускает на выполнение определенную команду.

Основные методы класса:

- `setText(QString text)` – отображает текст. Проблема этого метода в том, что иногда текст бывает слишком большой для того, чтоб его отобразить в маленьком окне. Поэтому размер шрифта подстраивается под размер окна. Алгоритм показан на рисунке 2.5.1.

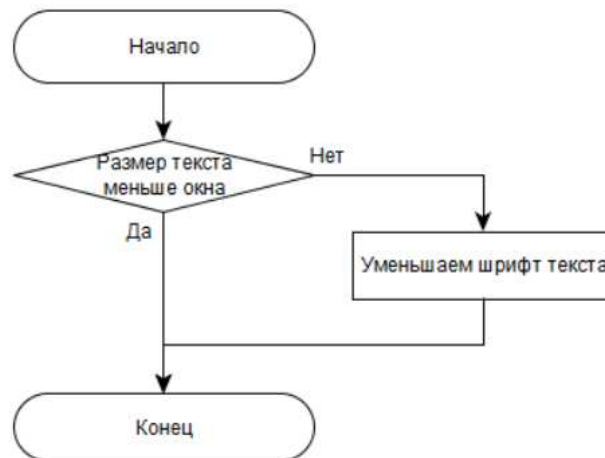


Рисунок 2.5.1 – Алгоритм работы подстройки размера текста.

- `showStatement(bool soundOnly)` – показывает окно напоминания. Также запускает звуковой файл;
- `showInformation(QString text)` – показывает текст;
- `eventFilter(QObject *obj, QEvent *event)` – отлавливает события класса. При наведении курсора на окно показывает кнопки отключения звука и кнопку закрытия окна. Когда курсор уходит из области окна, скрывает эти кнопки;

- `paintEvent(QPaintEvent *)` – рисует окно программы. Рисует скругленный прямоугольник, а также треугольник, который направлен в сторону «рта» картинки.

Положение этого окна контролируется классом, который содержит в себе это окно. По умолчанию оно показывается сверху над основным окном. Если места там не хватает, то оно отображается снизу основного окна. Так же это окно сдвигается для того, чтоб оно полностью отображалось на рабочем столе.

2.6 Класс Skins

Класс, который создан по принципу шаблона одиночки. По своей сути является изоморфным классу `Commands`. Хранит в себе структуру скина, код которой изображен на рисунке 2.6.1.

```
struct skin{
    QStringList files;
    QVector<QPoint> headPositions;
    QPoint position;
    double scale;
    bool standart;
};
```

Рисунок 2.6.1 – Структура skin

2.7 Класс Settings

Класс окна настроек. Это окно изображено на рисунке 2.7.1.

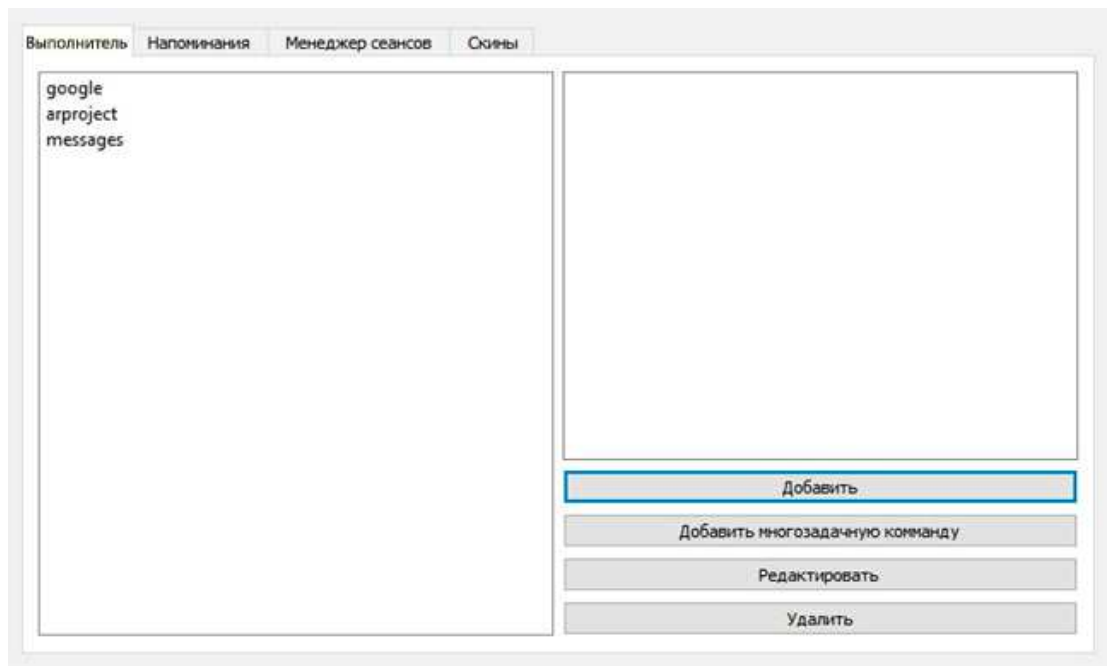


Рисунок 2.7.1 – Окно настроек

Содержит в себе 4 вкладки. На первой вкладке находятся настройки для **выполнителя**.

Основные методы для этой вкладки:

- `updateCommandList()` – обновляет список команд в форме `CommandList`;
- `showCommandSettings()` – обновляет информацию текстового поля описанием команды;
- `addNewCommand()` – вызывает окно добавления команды;
- `editCommand()` – вызывает окно редактирования команды;
- `removeCommand()` – удаляет выбранную команду;
- `addNewMultyCommand()` – вызывает окно `MultyCommand`;

Вкладка напоминаний изображена на рисунке 2.7.2.



Рисунок 2.7.2 – Вкладка «Напоминания»

Содержит следующие кнопки:

- «Добавить» - при нажатии на эту кнопку появляется окно RemainForm;
- «Редактировать» - при нажатии на эту кнопку появляется окно редактирования выбранного напоминания;
- «Удалить» - при нажатии на эту кнопку удаляется выбранное напоминание;
- «Изменить цвет» - при нажатии на эту кнопку появляется стандартное окно выбора цвета в Qt . После нажатия кнопки «Ок» новый цвет сохраняется в настройках. Теперь класс Statement будет иметь такой цвет;
- «Изменить прозрачность» - при нажатии на эту кнопку появляется окно с полем ввода целого числа от 0 до 255. После сохранения Statement будет иметь выбранную прозрачность;

- «Изменить цвет текста» - при нажатии на эту кнопку появится окно с выбором цвета. При сохранении цвет текста напоминаний будет новым.

На вкладке «Менеджер сеансов» пользователь может редактировать настройки сеанса. Окно изображено на рисунке 2.7.3.

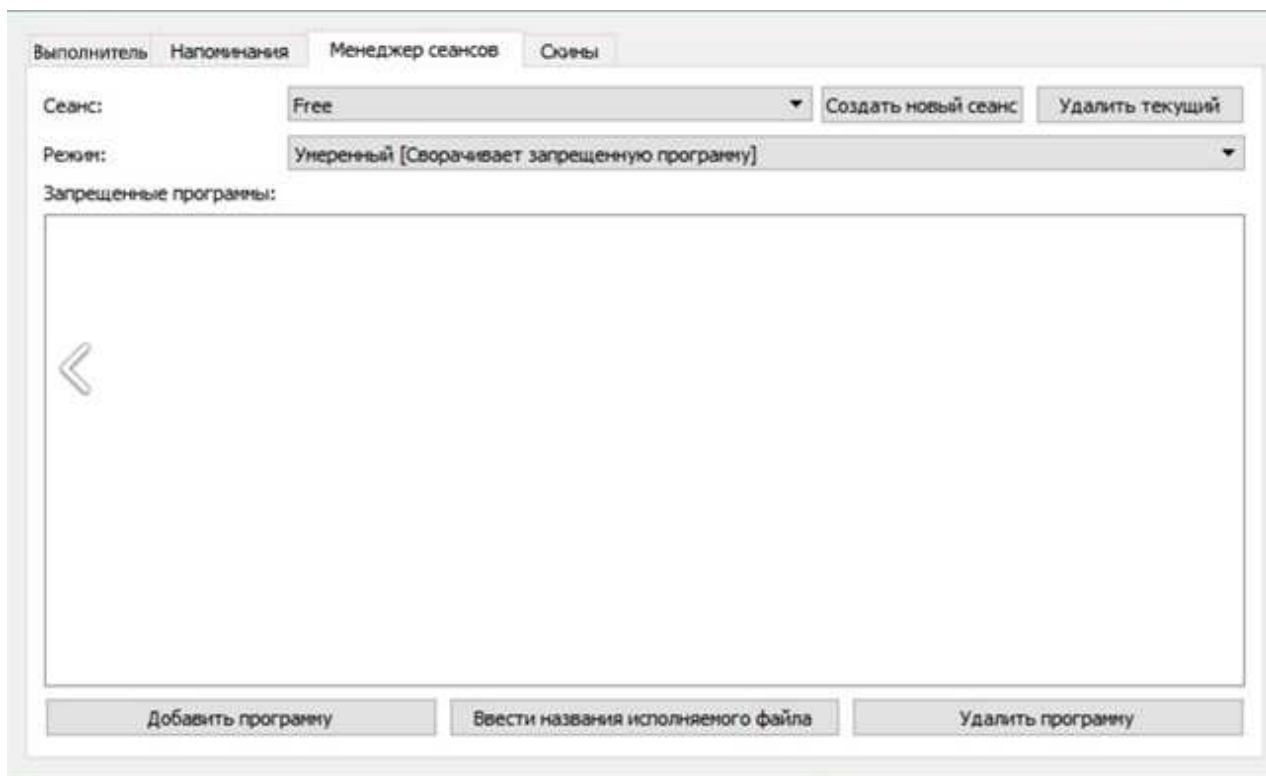


Рисунок 2.7.3 – Вкладка «Менеджер сеансов»

На вкладке «Скины» пользователь может редактировать внешний вид программы. Вкладка изображена на рисунке 2.7.4.

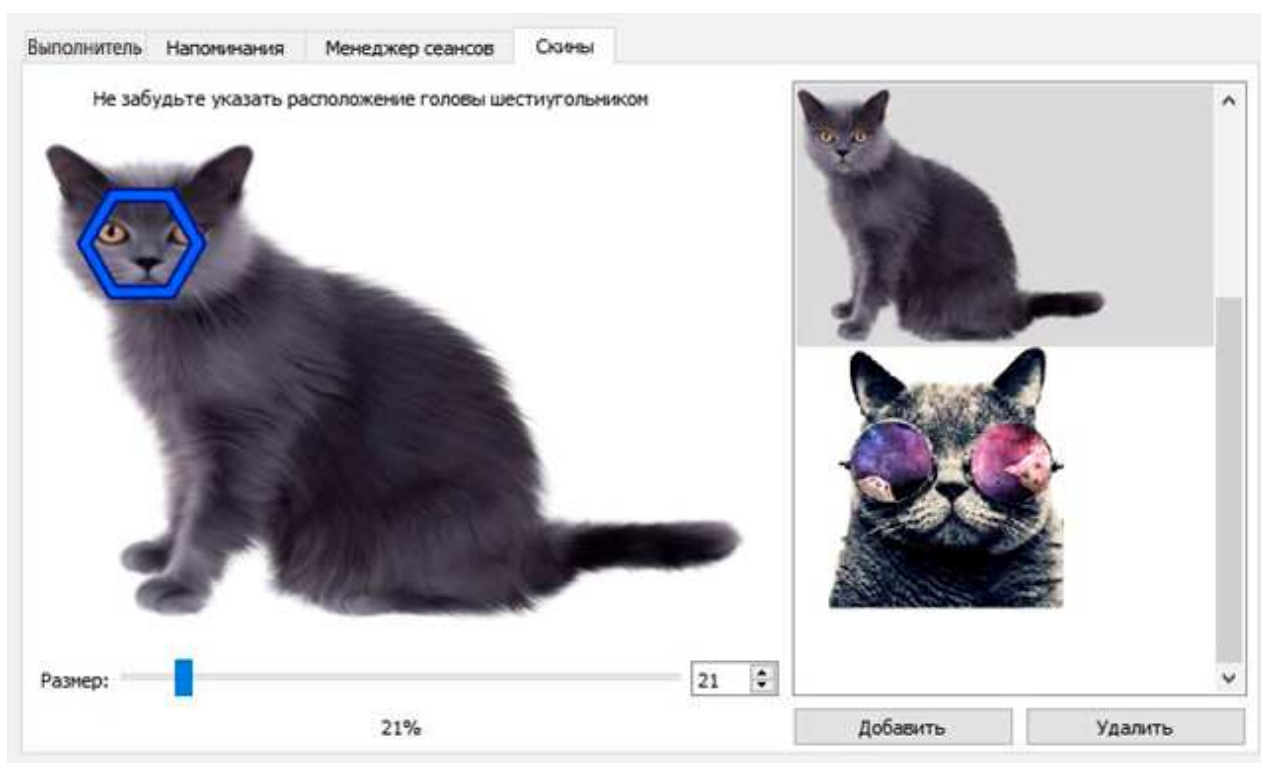


Рисунок 2.7.4 – Вкладка «Скины»

Шестиугольником (SkinLable) пользователь отмечает положение «рта» картинки. Ползунком меняет размер окна, с помощью списка скинов редактирует текущий внешний вид. Так же пользователь может удалить и добавить скин с помощью соответствующих кнопок.

Добавление скина так же осуществляется с помощью технологии drag-and-drop [11]. Нужный файл картинки нужно просто перетащить на основное окно программы.

2.8 Класс SkinLable

Класс, который отображает картинку. Особенность его в том, что положение этой картинки можно изменять с помощью мыши.

Основные методы:

- `setFileName(QString str)` – отображает картинку с путем `str`;

- `mouseMoveEvent(QMouseEvent* e)` – событие, которое вызывается, когда пользователь двигает мышкой над классом. Собственно, в этом методе и изменяется положение картинки;

2.9 Класс RemainForm

Окно для настройки напоминания. Напоминание – некоторая сущность, которая может выполняться в определенное время. Напоминание может показывать класс `Statement`, выполнять команду, включать звуки и выполнять все это вперемешку. Окно этого класса представлено на рисунке 2.9.1.

Основное

Название: Напоминать: Время:

← Сентябрь, 2016 →						
Пн	Вт	Ср	Чт	Пт	Сб	Сб
29	30	31	1	2	3	3
5	6	7	8	9	10	10
12	13	14	15	16	17	17
19	20	21	22	23	24	24
26	27	28	29	30	1	1

Выбранный день: 20:09.

Текст напоминания:

Выберите мелодию: Загрузить свою Громкость

Команда:

Оповещение только с помощью мелодии (без появления окна напоминания)

Итог

Рисунок 2.9.1 – Окно настроек напоминания

Для напоминаний есть несколько режимов:

- напоминать один раз – пользователь выбирает дату с помощью календаря;

- напоминать каждый день – календарь блокируется;
- напоминать в некоторые дни – пользователь может выбрать несколько дней в календаре, нажимая на нужный день. Если на выбранный день нажать еще раз, он перестанет быть выбранным.

Так же для напоминания можно выбрать мелодию. Она начнет проигрываться во время показа напоминания. С помощью кнопки «Загрузить свою» можно выбрать музыкальный файл на компьютере пользователя. Так же можно прослушать файл и установить громкость проигрывания файла.

Помимо этого, для напоминания можно установить команду, которую она будет выполнять во время показа.

Так же напоминание может быть активировано без окна, то есть напоминание просто проигрывает музыкальный файл или запускает команду в нужное время.

Так же внизу окна содержится текстовое поле с описанием настроенного напоминания.

Класс содержит в себе описание структуры напоминания. Код структуры изображен на рисунке 2.9.2.

```

struct remain{
    QString name;
    QString mode;
    QString ed;
    QString com;
    QVector<QDate*> dates;
    QTime time;
    QString text;
    int countOfRep;
    int repeatCount;
    int period;
    int loud;
    QString fileName;
    int number;
    bool soundOnly;
};

```

Рисунок 2.9.2 – Структура remain

Основные методы класса RemainForm:

- makeRemain() – создает и возвращает структуру remain по выбранным настройкам;
- standart() – устанавливает значение всех настроек по-умолчанию;
- setRemain(remain r) – устанавливает настройки напоминания r;
- loadMusicList() – загружает список музыкальных файлов;
- saveMusicList() – сохраняет список музыкальных файлов;
- setCommandList(QStringList list) – устанавливает список доступных команд;
- makeDescription() – составляет описание команды и выводит его в текстовое поле;
- acceptedRem() – сигнал, который отправляется, когда пользователь нажмет на кнопку «Сохранить».

2.10 Класс CommandForm

Класс формы для создания и редактирования команды. Окно формы представлено на рисунке 2.10.1.

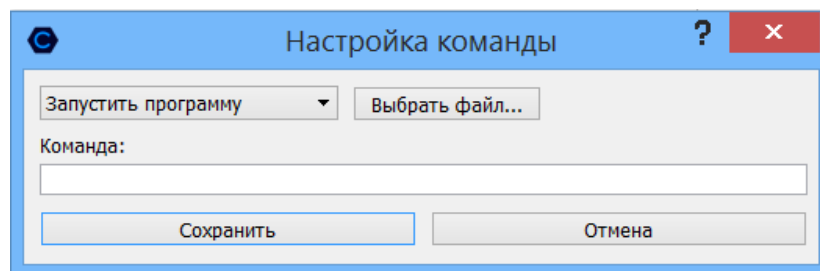


Рисунок 2.10.1 – Окно настройки команды.

Окно содержит поле ввода названия команды. Так же кнопки «Сохранить» и «Отмена». Выпадающий список содержит доступные типы команды. Элементы справа от списка меняются в зависимости от типа команды.

Кнопка выбора файла или папки показывается для следующих типов команд:

- запустить программу;
- открыть файл;
- открыть папку.

Для команды типа «открыть сайт» (рисунок 2.10.2) показывается поле ввода URL сайта.

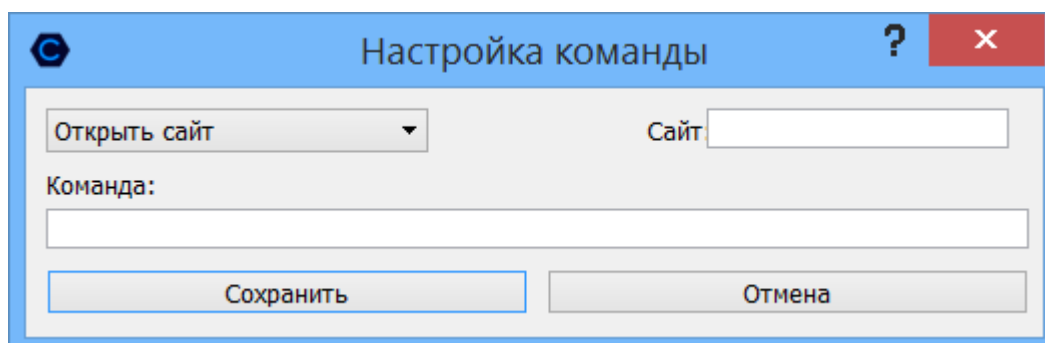


Рисунок 2.10.2 – Настройка напоминания.

Для команд «Выключить компьютер», «Перезагрузить компьютер» и «Перейти в спящий режим» отсутствуют элементы справа.

Основные методы этого класса:

- `setStandart()` – устанавливает всем полям стандартное значение;
- `setCommand(команд _с)` – устанавливает настройки команды `_с`;
- `checkExisting(QString str)` – проверяет есть ли в списке команд команда `str`;
- `createCommand()` – создает новую команду;

2.11 Класс MultyCommand

Окно для создания многозадачной команды. Это окно изображено на рисунке 2.11.1.

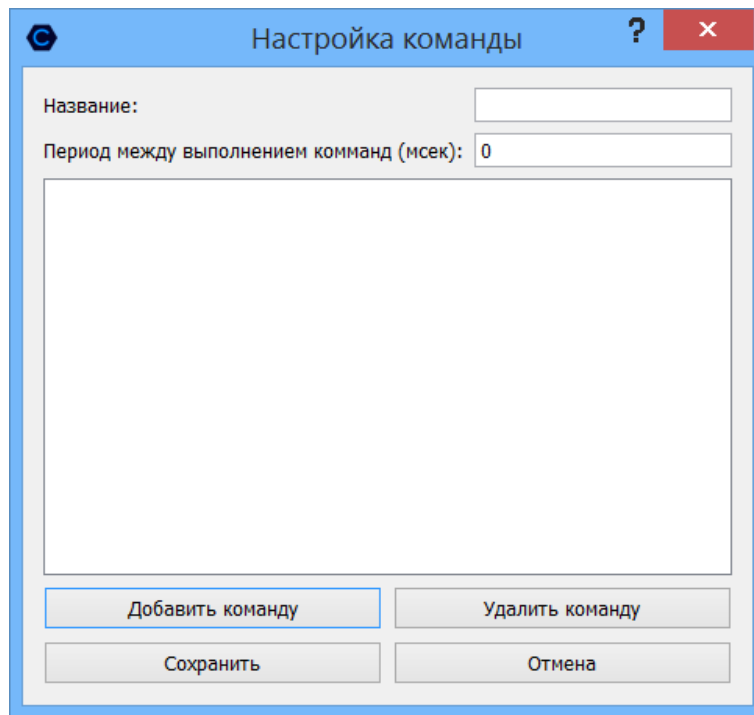


Рисунок 2.11.1 – Окно настройки мультикомандной задачи.

При нажатии на окно «Добавить команду» появляется окно класса `commandList`. Пользователь выбирает команду. Эта команда с помощью сигнала передается в класс `MultyCommand`, и она добавляется в список команд.

Основные методы этого класса:

- `addNewCommand(QString s)` – добавляет команду `s` в список команд;
- `removeCommand()` – удаляет выбранную команду;
- `saveCommands()` – сохраняет многозадачную команду, вызывает сигнал `accepted(QStringList s)`;
- `accepted(QStringList s)` – сигнал, который отправляет в окно настроек список команд многозадачной команды;
- `clearAll()` – очищает список команд.

2.12 Класс `commandList`

Класс окна со списком команд. Содержит в себе список команд, кнопку «Выбрать» и кнопку «Отмена». Окно изображено на рисунке 2.12.1.

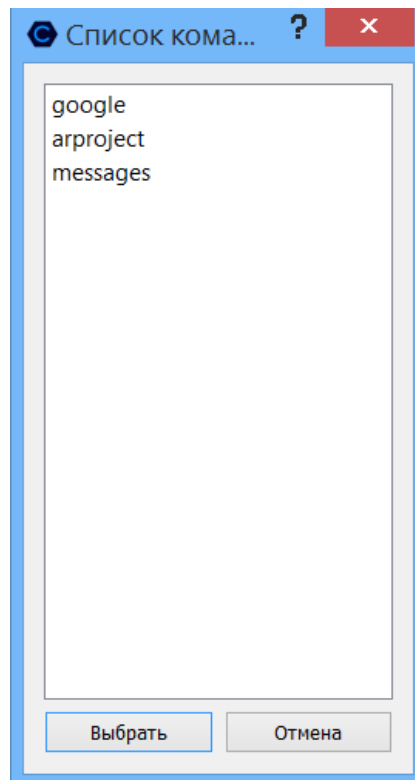


Рисунок 2.12.1 – Окно выбора из списка команд.

Основные методы этого класса:

- `setCommandList(QStringList list)` – устанавливает команды в список команд;
- `accepted(QString)` – сигнал, который генерируется, когда пользователь выбирает команду. Отправляет выбранную команду.

3 Руководство пользователя

В этом разделе будет приведена инструкция по использованию C.A.Project.

3.1 Использование главного окна программы

Для изменения текущего скина программы нажмите ПКМ на программу или иконку программы в трее, и в контекстном меню выберете раздел «Настройки...». В появившемся окне перейдите на вкладку «Скины». В правой части окна выберете понравившийся вам скин и нажмите на нем ЛКМ. Там же

вы можете изменить размер кожи. Измените значение в элементах, которые находятся справа от надписи «Размер». Так же необходимо переместить шестиугольник на место «рта» картинке. Вкладка «Скины» изображена на рисунке 3.1.

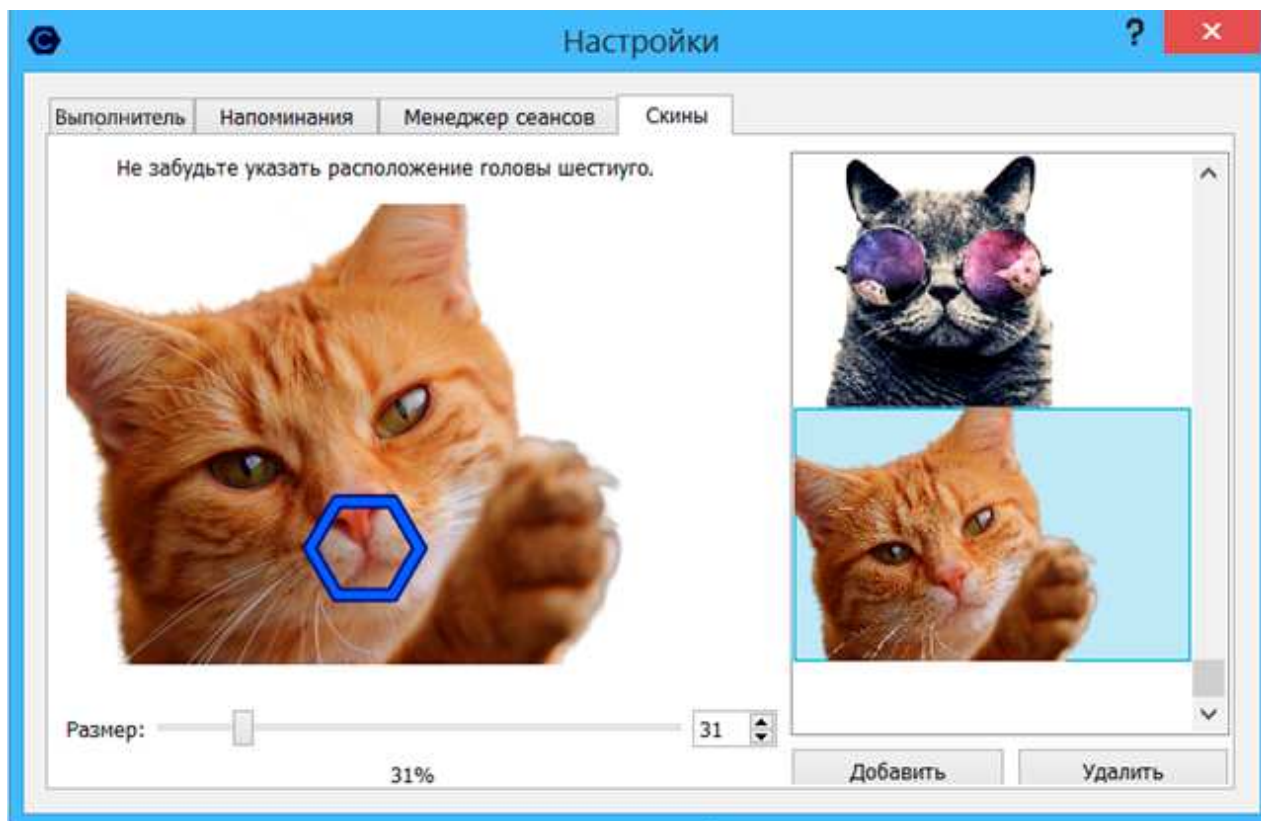


Рисунок 3.1 – Выбор кожи

Для изменения положения главного окна необходимо зажать левой кнопкой мыши и передвинуть окно.

Добавление кожи происходит следующими способами:

- перетаскивание файла картинки на главное окно;
- перетаскивание файла картинки в окно настроек;
- нажатие на кнопку «Добавить» во вкладке «Скины». После необходимо выбрать нужный файл картинки.

3.2 Использование исполнителя

Добавление и изменение команд происходит на вкладке «выполнитель» в окне настроек. С помощью кнопки «Добавить» можно

создать новую команду. В окне настройки команды нужно выбрать в выпадающем списке что команда будет делать, и позже добавить дополнительные настройки (выбрать файл, папку или сайт). Окно настроек изображено на рисунках 3.2.1-2.

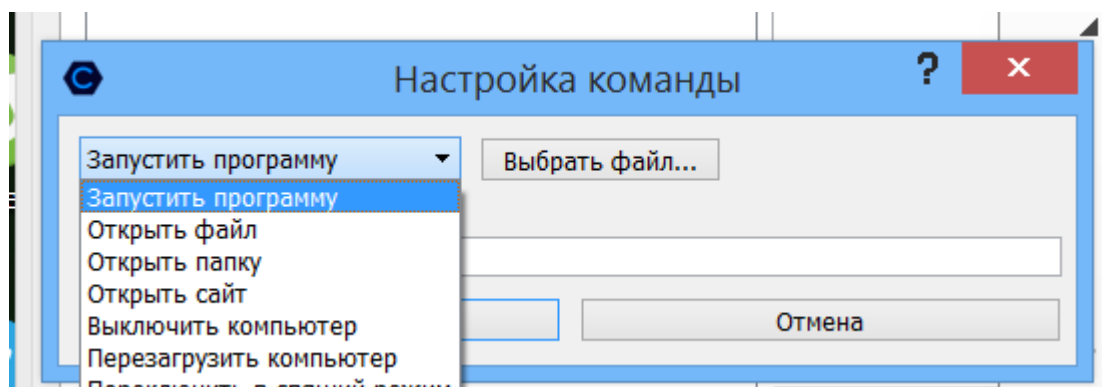


Рисунок 3.2.1 – Настройка команды

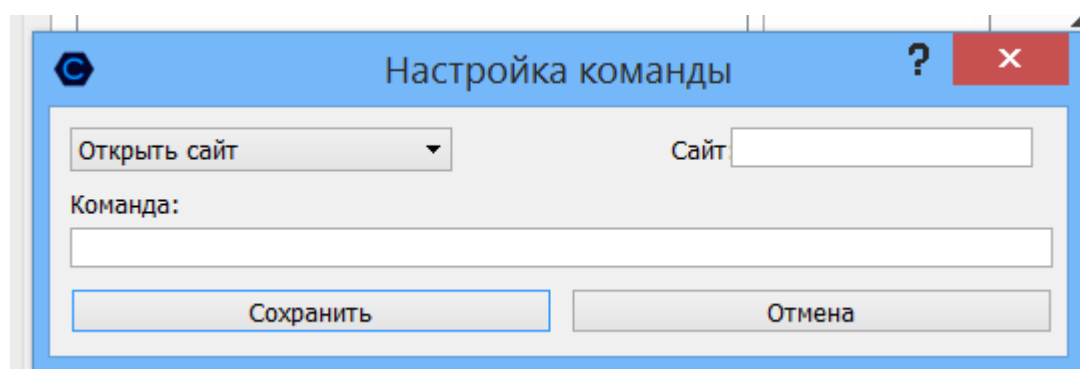


Рисунок 3.2.2 – Запуска сайта

С помощью кнопки «Добавить многозадачную программу» можно создать команду, которая будет последовательно выполнять другие команды. Окно редактирование многозадачной команды изображено на рисунке 3.2.3.

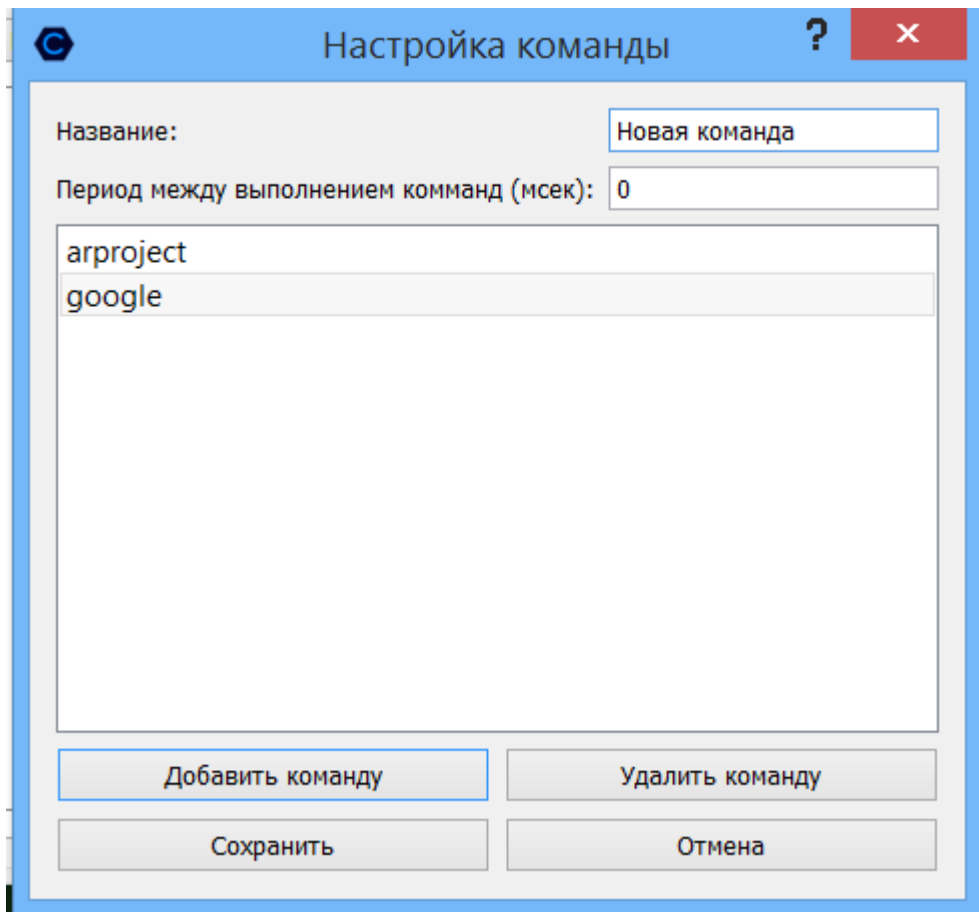


Рисунок 3.2.3 – Настройка мультикомандной команды

Для запуска команды необходимо вызвать окно исполнителя. Для этого необходимо два раза нажать на главное окно программы, или же навести на правую часть экрана, после чего появится зеленая кнопка. После нажатия на эту кнопку появится окно исполнителя. Зеленая кнопка изображена на рисунке 3.2.4. Исполнитель изображен на рисунке . Исполнитель содержит выпадающий список с сеансами (сверху), поле ввода названия команды, строка состояния команды и список всех команд. Так же справа находятся кнопка для запуска окна настроек и кнопка для закрытия окна исполнителя.

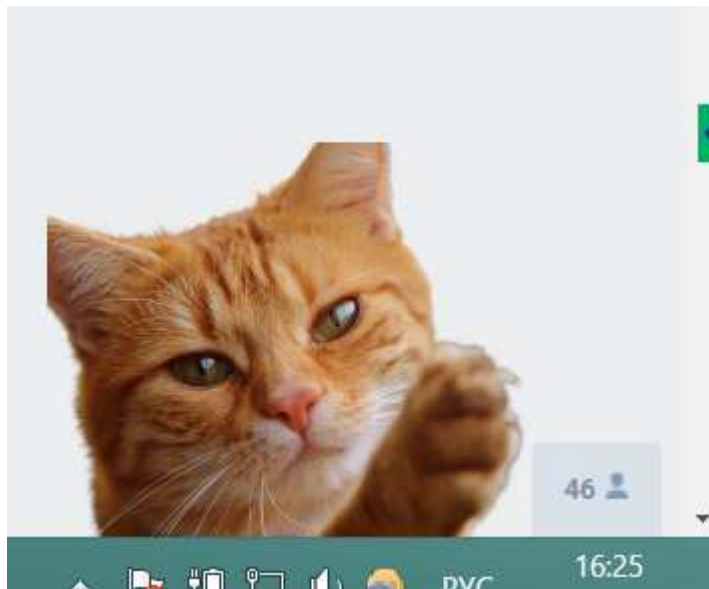


Рисунок 3.2.4 – Зеленая кнопка

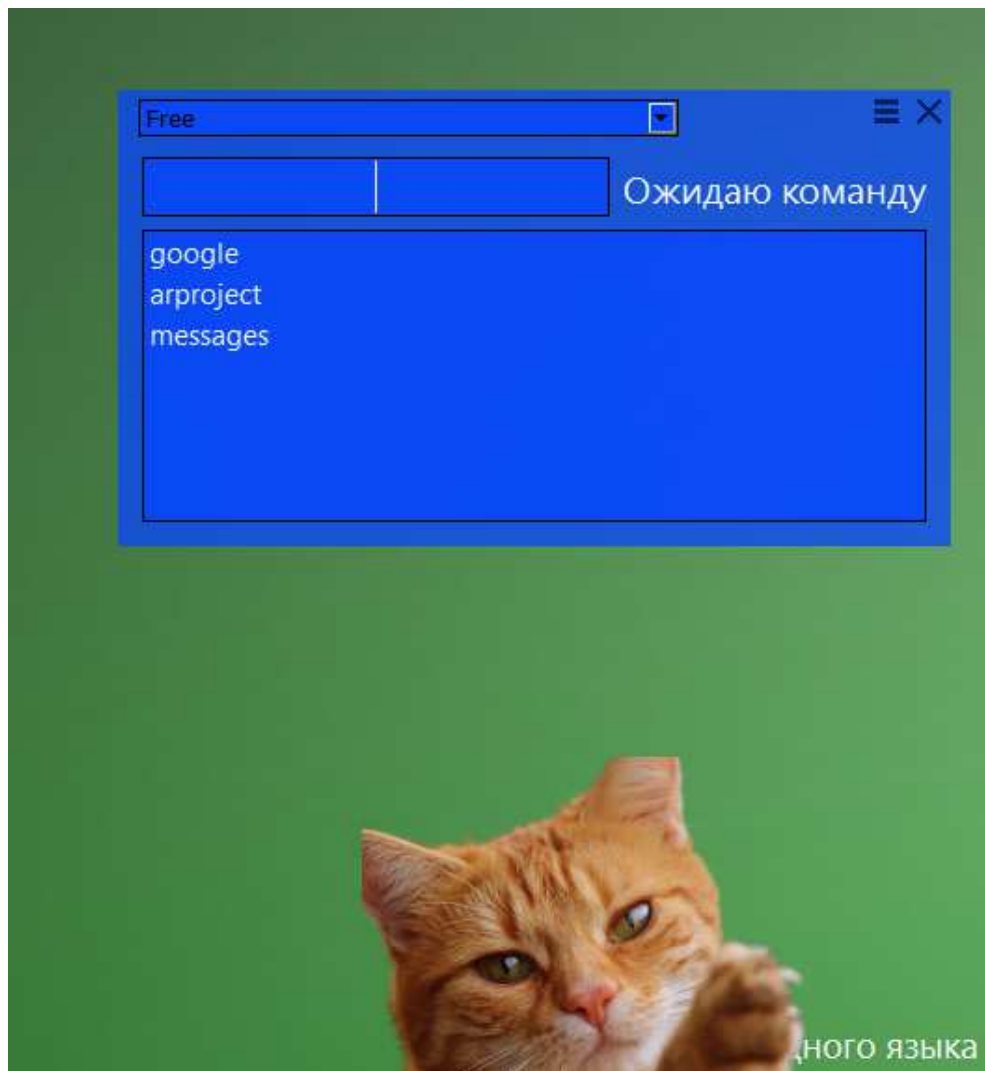
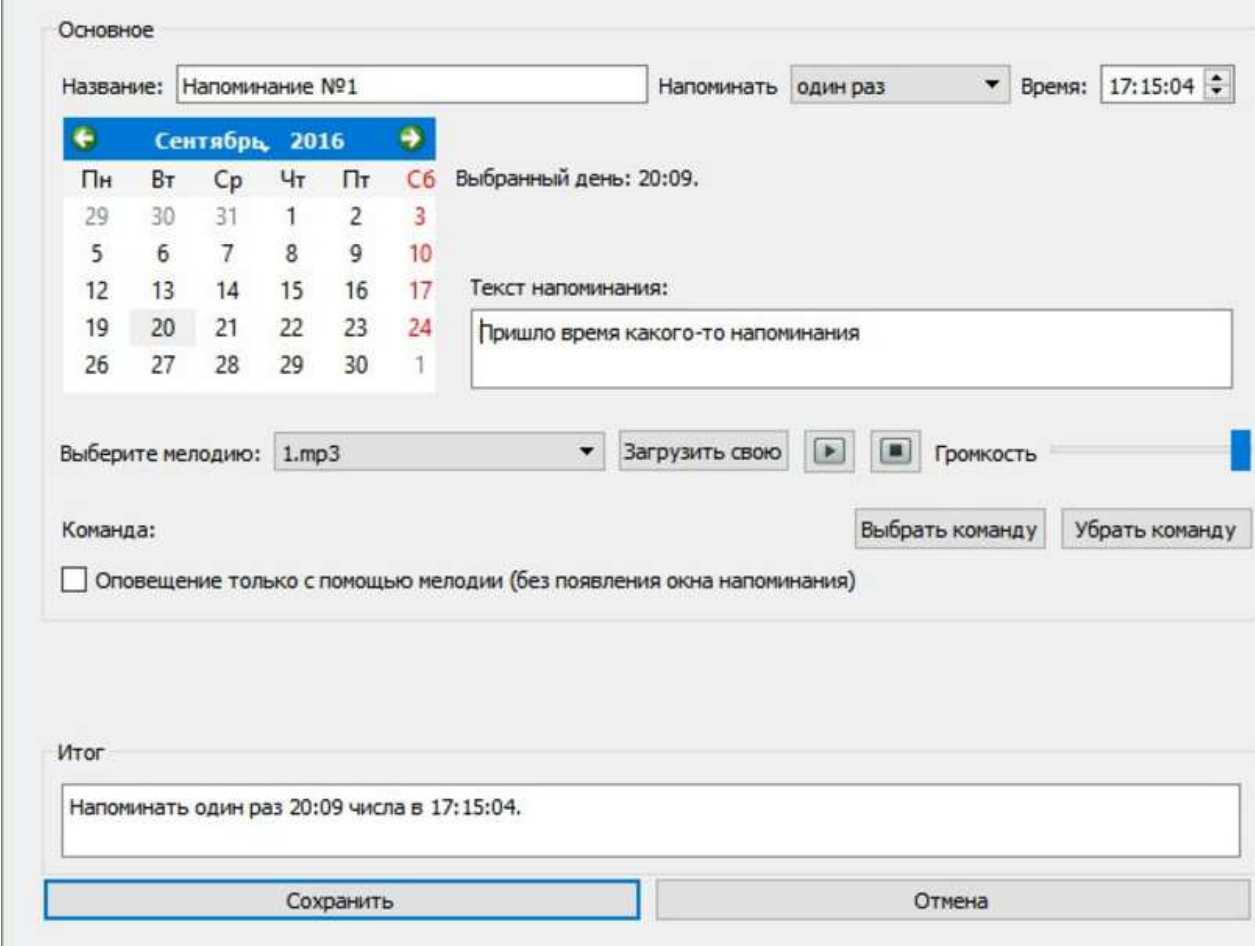


Рисунок 3.2.5 – Окно исполнителя

Для запуска команды можно нажать на нее в списке команд или ввести название команды и нажать на «Enter». В строке ввода команды используется автодополнение, которое ускоряет процесс ввода названия команды.

3.3 Использование напоминаний

Редактировать настройки напоминаний можно на вкладке «Напоминания». На этой же вкладке можно добавлять, редактировать и удалять напоминания с помощью соответствующих кнопок. При добавлении нового напоминания появится окно настроек напоминания, которое изображено на рисунке .



Основное

Название: Напоминать: Время:

← Сентябрь, 2016 →

Пн	Вт	Ср	Чт	Пт	Сб	Выбранный день: 20:09.
29	30	31	1	2	3	
5	6	7	8	9	10	
12	13	14	15	16	17	
19	20	21	22	23	24	
26	27	28	29	30	1	

Текст напоминания:

Выберите мелодию: Загрузить свою Громкость

Команда:

Оповещение только с помощью мелодии (без появления окна напоминания)

Итог

Рисунок 3.3.1 – Окно настройки напоминания.

Так же с помощью кнопки «Изменить цвет» можно выбрать цвет напоминания. Окно редактирования цвета напоминания показано на рисунке.

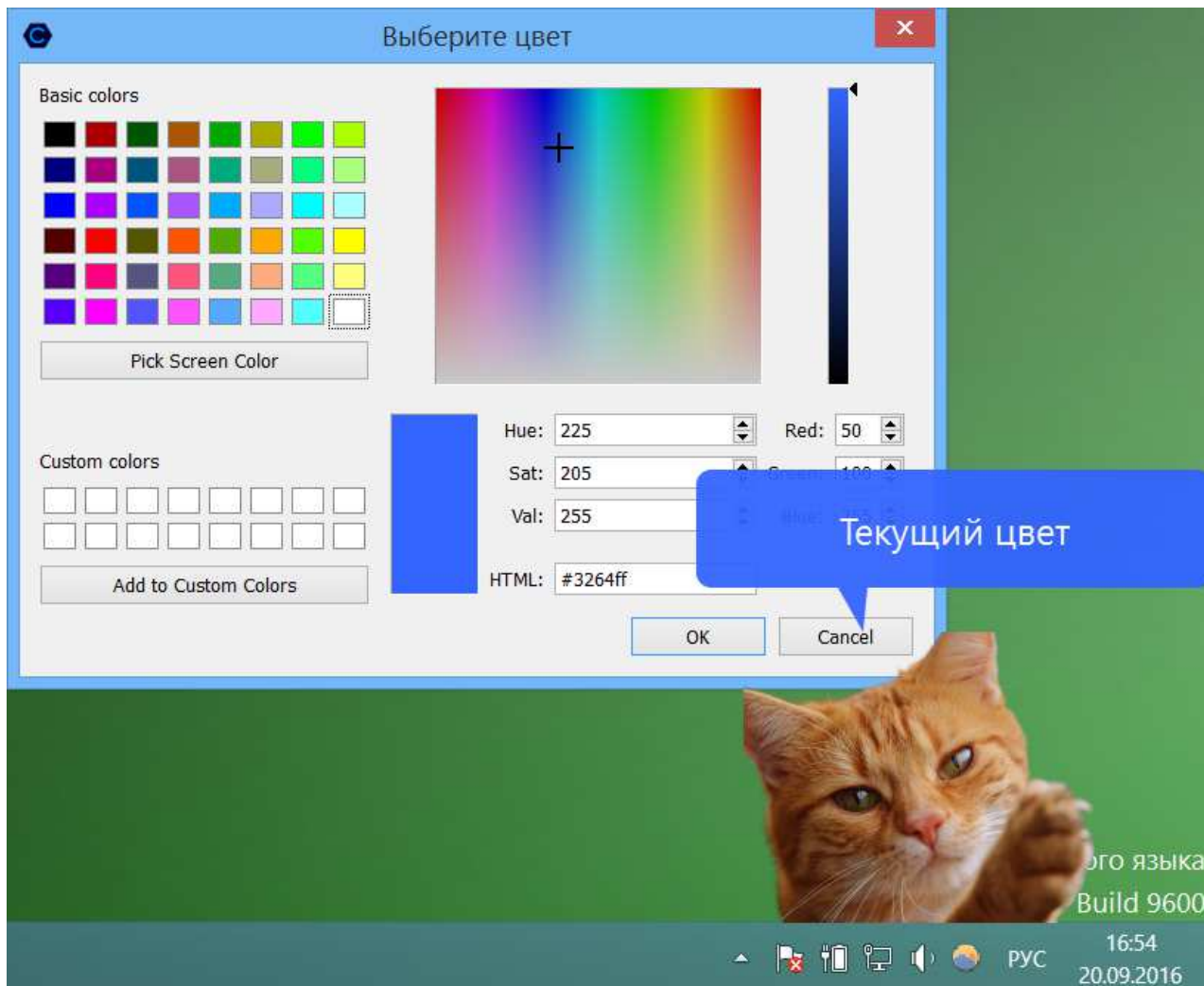


Рисунок 3.3.2 – Окно выбора цвета

Также можно изменять цвет текста напоминаний с помощью того же окна.

3.4 Использование менеджера сеансов

Для редактирования настроек менеджера необходимо перейти на вкладку «Менеджер сеансов». Эта вкладка изображена на рисунке 3.4.1.

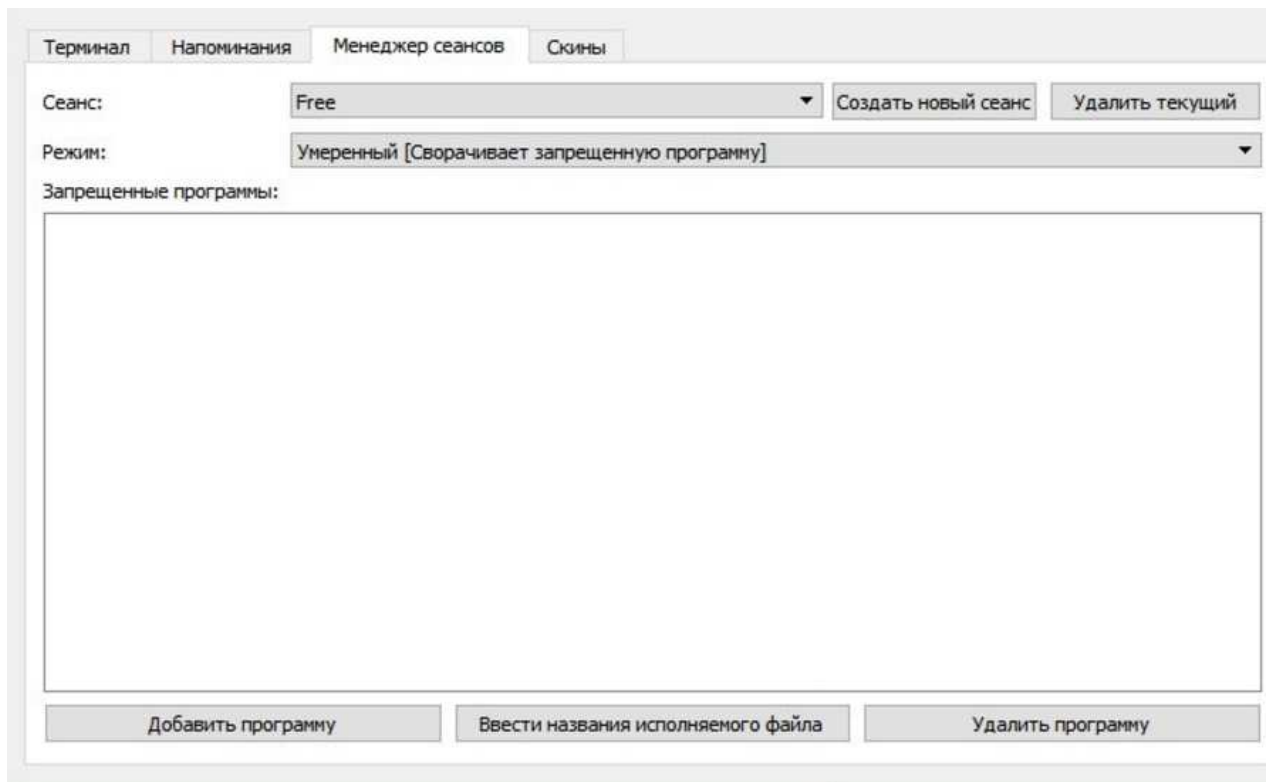


Рисунок 3.4.1 – Менеджер сеансов.

В текущий сеанс можно добавить запрещенную программу с помощью кнопки «Добавить программу» или с помощью кнопки «Ввести названия исполняемого файла».

Также можно создать новый сеанс и удалить текущий. Так же можно изменять режим текущего сеанса.

Текущий сеанс переключается в исполнителе.

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы был реализован программный продукт для повышения удобства использования компьютера. В данной пояснительной записке был описан процесс разработки этого программного продукта.

В ходе выполнения работы были выполнены следующие задачи:

- анализ предметной области;
- проектирование и реализация удобной для пользователей системы для выполнения некоторых задач компьютера;

Основные результаты:

- написана программная система, которая повышает эффективность использования пользователем компьютера.

В результате была получен программный продукт, который позволяет пользователю создавать и использовать свои команды. Также пользователь может менять внешний вид этого программного продукта. И конечно, эта программа показывает пользователю напоминания, которые он создал.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. The SAO Utilities Project – development log [Электронный ресурс] : официальный сайт проекта. Режим доступа : <http://www.gpbeta.com/en/post/develop/sao-utils-alpha2>;
2. Страуструп, Б. Язык программирования C++ : учебник / Б. Страуструп. – Москва : Бином, 2011. – 1136 с;
3. Qt [Электронный ресурс] : официальный сайт проекта. Режим доступа : <https://www.qt.io/ru>;
4. Кроссплатформенность – это круто [Электронный ресурс] : Хабрахабр. Режим доступа : <https://habrahabr.ru/post/133897>;
5. Qt Designer Manual [Электронный ресурс] : официальная документация Qt. Режим доступа : <http://doc.qt.io/qt-5/qtdesigner-manual.html>;
6. ColorScheme [Электронный ресурс] : официальный сайт проекта. Режим доступа : <http://colorscheme.ru/>;
7. How to detect reliably Mac OS X, iOS, Linux, Windows in C preprocessor? [Электронный ресурс] : StackOverflow . Режим доступа: <http://stackoverflow.com/questions/5919996/how-to-detect-reliably-mac-os-x-ios-linux-windows-in-c-preprocessor>;
8. Win32 API по шагам [Электронный ресурс] : Первые шаги . Режим доступа: <http://www.firststeps.ru/mfc/winapi/winapi1.html>;
9. Макконнелл, С. Совершенный код. Мастер-класс : Учебник / С. Макконнелл. – Москва: Русская Редакция, 2016. – 896 с;
10. Моделирование на UML [Электронный ресурс] : Моделирование на UML. Режим доступа : http://book.uml3.ru/sec_1_5
11. Шлее, М. Qt 5.3. Профессиональное программирование на C++ : учебник / М.Шлее. – Санкт Петербург: БХВ-Петербург, 2015. – 923 с.

