**Рабочий slack-бот**

**Описание программы**

Оглавление

[1. Описание 4](#_Toc510453786)

[1.1. Файловая структура, описание основных функций 4](#_Toc510453787)

[2. Файл constants.py 5](#_Toc510453788)

[2.1. Описание 5](#_Toc510453789)

[3. Файл calendarSearcher.py 5](#_Toc510453790)

[3.1. Класс calendarSearcher ( ) 5](#_Toc510453791)

[3.1.1. Описание, структура класса 5](#_Toc510453792)

[3.1.2. Метод \_\_init\_\_ ( ) 5](#_Toc510453793)

[3.1.3. Метод cs\_search\_days\_week ( ) 5](#_Toc510453794)

[3.1.3.1. Метод parse\_month ( ) 6](#_Toc510453795)

[3.1.4. Метод cs\_search\_days\_month ( ) 6](#_Toc510453796)

[3.1.5. Метод cs\_search\_same\_weekdays ( ) 7](#_Toc510453797)

[4. Файл commandChecker.py 8](#_Toc510453798)

[4.1. Класс commandChecker ( ) 8](#_Toc510453799)

[4.1.1. Описание, структура класса 8](#_Toc510453800)

[4.1.2. Метод \_\_init\_\_ ( ) 8](#_Toc510453801)

[4.1.3. Метод make\_employees\_list ( ) 8](#_Toc510453802)

[4.1.4. Метод make\_locations\_list ( ) 9](#_Toc510453803)

[4.1.5. Метод cc\_check\_name ( ) 9](#_Toc510453804)

[4.1.6. Метод cc\_check\_date ( ) 9](#_Toc510453805)

[4.1.7. Метод check\_location ( ) 10](#_Toc510453806)

[4.1.8. Метод cc\_set\_day\_input\_check ( ) 10](#_Toc510453807)

[4.1.8.1. Метод check\_time ( ) 11](#_Toc510453808)

[4.1.8.2. Метод check\_time\_period ( ) 11](#_Toc510453809)

[4.1.9. Метод cc\_get\_day\_input\_check ( ) 12](#_Toc510453810)

[4.1.10. Метод cc\_get\_week\_input\_check ( ) 12](#_Toc510453811)

[5. Файл mysqlScheduleParser.py 13](#_Toc510453812)

[5.1. Класс mysqlScheduleParser ( ) 13](#_Toc510453813)

[5.1.1. Описание, структура класса 13](#_Toc510453814)

[5.1.2. Метод \_\_init\_\_ ( ) 13](#_Toc510453815)

[5.1.3. Метод msp\_add\_employee ( ) 14](#_Toc510453816)

[5.1.3.1. Метод make\_query ( ) 14](#_Toc510453817)

[5.1.4. Метод msp\_add\_location ( ) 14](#_Toc510453818)

[5.1.4.1. Метод make\_query ( ) 15](#_Toc510453819)

[5.1.5. Метод search\_user\_id ( ) 15](#_Toc510453820)

[5.1.6. Метод search\_location\_id ( ) 16](#_Toc510453821)

[5.1.7. Метод msp\_set\_day\_schedule ( ) 16](#_Toc510453822)

[5.1.7.1. Метод make\_query ( ) 16](#_Toc510453823)

[5.1.8. Метод msp\_set\_day\_schedule\_repeat ( ) 17](#_Toc510453824)

[5.1.9. Метод msp\_get\_day\_schedule ( ) 17](#_Toc510453825)

[5.1.9.1. Метод make\_query ( ) 18](#_Toc510453826)

[5.1.10. Метод msp\_get\_day\_schedule\_everybody ( ) 18](#_Toc510453827)

[5.1.10.1. Метод make\_query ( ) 18](#_Toc510453828)

[5.1.11. Метод msp\_get\_week\_schedule ( ) 19](#_Toc510453829)

[5.1.12. Метод msp\_get\_week\_schedule\_everybody ( ) 19](#_Toc510453830)

[6. Файл Slackbot.py 20](#_Toc510453831)

[6.1. Функция show\_commands ( ) 20](#_Toc510453832)

[6.2. Функция show\_commands\_examples ( ) 20](#_Toc510453833)

[6.3. Функция main ( ) 20](#_Toc510453834)

# Описание

## Файловая структура, описание основных функций

Программа написана на языке Python 3.

В программе используются нестандартные модули:

* slackclient
* MySQLdb

Данный бот работает с БД MySql, которая хранит расписание работы сотрудников, а также место работы (офис/удалённо). По команде бот может добавлять расписание сотрудника в конкретный день, показывать время работы сотрудника в указанный день или во все дни текущей, либо следующей недели.

Также бот может показывать время работы всех сотрудников в указанный день или во все дни текущей, либо следующей недели.

Программа состоит из 5 файлов:

* constants.py – файл, содержащий постоянные.
* calendarSearcher.py – файл, содержащий класс calendarSearcher ( ). Этот класс работает календарём – находит все даты в указанном месяце или в указанной неделе.
* commandChecker.py – файл, содержащий класс commandChecker ( ). Этот класс содержит набор методов, которые проверяют правильность введённых команд.
* mysqlScheduleParser.py – файл, содержащий класс mysqlScheduleParser ( ). Этот класс содержит набор методов, которые добавляют или получают данные из БД Mysql.
* slackbot.py – файл, содержащий функции, которые управляют поведением бота.

# Файл constants.py

## Описание

Файл содержит следующие константы:

* MYSQL\_CONNECT – список с данными для подключения к БД (ip сервера, логин, пароль, имя БД)
* WEEKDAYS – список из названий дней недели (понедельник, вторник и т.д.)
* HOLIDAYS – список из дат праздничных дней в 2018 году.

# Файл calendarSearcher.py

## Класс calendarSearcher ( )

### Описание, структура класса

Класс, который находит все даты в указанном месяце или в указанной неделе.

Класс содержит следующие методы:

* \_\_init\_\_ (self)
* cs\_search\_days\_week (self, week)
* cs\_search\_days\_month (self, month)
* cs\_search\_same\_weekdays (self, date)

### Метод \_\_init\_\_ ( )

Описание

Инициализация класса.

Параметры

Отсутствуют.

Возвращаемые значения

Отсутствуют.

### Метод cs\_search\_days\_week ( )

Описание

Метод, который находит все дни в указанной неделе текущего года.

Для первой недели предусмотрен особый алгоритм. Это связано с тем, что в питоне 53-ая неделя года (например, 31.12.2018) считается за 1-ую неделю следующего года.

Параметры

(int) week – номер недели.

Возвращаемые значения

(list) week\_days\_list – список дат в этой неделе.

#### Метод parse\_month ( )

Описание

Метод перебирает все месяцы года, ищет неделю, а затем заполняет список с датами.

Параметры

* (int) year – номер года.
* (int) month – номер месяца.
* (int) week1 – номер недели.
* (list) list1 – список дат.

Возвращаемые значения

(list) list1 – список дат.

### Метод cs\_search\_days\_month ( )

Описание

Метод, который составляет список всех дат в указанном месяце текущего года.

Параметры

(int) month – номер месяца.

Возвращаемые значения

(list) month\_days\_list – список дат в этом месяце.

### Метод cs\_search\_same\_weekdays ( )

Описание

Для переданной даты метод определяет день недели, далее метод находит даты ВСЕХ таких же дней недели в ближайшие 3 месяца (например, метод находит даты всех вторников в этом и ближайших трёх месяцах).

Параметры

(string) date – дата в формате гггг.мм.дд.

Возвращаемые значения

(list) same\_weekdays\_list – список дат таких же дней недели в этом и ближайших трёх месяцах.

# Файл commandChecker.py

## Класс commandChecker ( )

### Описание, структура класса

Класс, который выполняет проверку входящей команды.

Класс содержит следующие методы:

* \_\_init\_\_ (self)
* make\_employees\_list (self)
* make\_locations\_list (self)
* cc\_check\_name (self, flag, answer, text)
* cc\_check\_date (self, flag, answer, text)
* check\_location (self, flag, answer, text)
* cc\_set\_day\_input\_check (self, text)
* cc\_get\_day\_input\_check (self, text)
* cc\_get\_week\_input\_check (self, text)

### Метод \_\_init\_\_ ( )

Описание

Инициализация класса, создание переменных (bool) flag, (string) answer, соединения с БД MySql.

Параметры

Отсутствуют.

Возвращаемые значения

Отсутствуют.

### Метод make\_employees\_list ( )

Описание

Метод составляет список сотрудников (из таблицы Users БД).

Параметры

Отсутствуют.

Возвращаемые значения

(list) employees\_list – список имён всех сотрудников.

### Метод make\_locations\_list ( )

Описание

Метод составляет список рабочих мест (из таблицы Locations БД).

Параметры

Отсутствуют.

Возвращаемые значения

(list) locations\_list – список названий всех мест работы.

### Метод cc\_check\_name ( )

Описание

Метод проверяет правильность введенного имени. Сотрудник с таким именем должен быть в таблице Users БД.

Параметры

* (bool) flag – флаг.
* (string) answer – ответ.
* (string) text – проверяемое имя.

Возвращаемые значения

* (string) answer – ответ. Если имя входит в список, то ответ пустой, если не входит, то ответ 'Ошибка в имени. '.
* (bool) flag – флаг. True, если имя входит в список, False – если имя не входит в список.

### Метод cc\_check\_date ( )

Описание

Метод проверяет правильность введённой даты и преобразует её в вид гггг.мм.дд.

Параметры

* (bool) flag – флаг.
* (string) answer – ответ.
* (string) text – проверяемая дата.

Возвращаемые значения

* (string) answer – ответ. Если дата введена правильно, то ответ пустой, если нет - ответ 'Ошибка в дате. '.
* (bool) flag – флаг. True, если имя входит в список, False – если имя не входит в список.
* (string) date – дата в формате гггг.мм.дд.

### Метод check\_location ( )

Описание

Метод проверяет правильность введенного места работы. Место работы с таким названием должно быть в таблице Locations БД.

Параметры

* (bool) flag – флаг.
* (string) answer – ответ.
* (string) text – проверяемое место работы.

Возвращаемые значения

* (string) answer – ответ. Если место работы входит в таблицу, , то ответ пустой, если нет - ответ 'Ошибка в месте работы. '.
* (bool) flag – флаг. True, если место работы содержится в таблице, False – если нет.

### Метод cc\_set\_day\_input\_check ( )

Описание

Метод проверяет правильность введённой команды. Метод проверяет имя сотрудника, место работы, дату, время начала и конца рабочего дня, а также проверяет, чтобы время начала не было больше, чем время конца.

Команда должна иметь один из следующих вариантов вида:

* 'name workplace date time\_start time finish' /
* 'name workplace date time\_start +' /
* 'name workplace date +'

Параметры

(string) text – полный текст команды.

Возвращаемые значения

* (bool) flag – флаг. True, если вся команда введена правильно, False – если нет.
* (string) answer – ответ. Если вся команда введена правильно, то ответ пустой, если нет – один из вариантов ответов, в зависимости от вида ошибки (ошибка в дате / ошибка во времени и т.д.).
* (list) list – список из слов, входящих в команду.

#### Метод check\_time ( )

Описание

Метод проверяет правильность ввода времени. Время должно иметь вид HH.MM или HH:MM.

Параметры

* (bool) flag – флаг.
* (string) answer – ответ.
* (string) text – проверяемое время.

Возвращаемые значения

* (string) answer – ответ. Если время введено правильно, то ответ пустой, если нет - ответ 'Ошибка во времени. '.
* (bool) flag – флаг. True, если время введено правильно, False – если нет.
* (string) time – время в формате HH:MM.

#### Метод check\_time\_period ( )

Описание

Метод проверяет, чтобы время начала было меньше, чем время конца.

Параметры

* (bool) flag – флаг.
* (string) answer – ответ.
* (string) text1 – проверяемое время начала рабочего дня.
* (string) text2 – проверяемое время окончания рабочего дня.

Возвращаемые значения

* (string) answer – ответ. Если время начала меньше времени окончания, то ответ пустой, если нет - ответ 'Ошибка во времени – начало позже конца. '.
* (bool) flag – флаг. True, если время начала меньше времени конца, False – если наоборот.

### Метод cc\_get\_day\_input\_check ( )

Описание

Метод проверяет правильность введённой команды для получения расписания человека в указанный день.

Команда должна иметь вид 'Name date'.

Параметры

(string) text – полный текст проверяемой команды.

Возвращаемые значения

* (bool) flag – флаг. True, если вся команда введена правильно, False – если нет.
* (string) answer – ответ. Если вся команда введена правильно, то ответ пустой, если нет – один из вариантов ответов, в зависимости от вида ошибки (ошибка в имени / ошибка в дате / неполная или избыточная команда).
* (list) list – список из слов, входящих в команду.

### Метод cc\_get\_week\_input\_check ( )

Описание

Метод проверяет правильность введённой команды для получения расписания человека за указанную неделю.

Команда должна иметь вид 'Name'.

Параметры

(string) text – полный текст проверяемой команды.

Возвращаемые значения

* (bool) flag – флаг. True, если вся команда введена правильно, False – если нет.
* (string) answer – ответ. Если вся команда введена правильно, то ответ пустой, если нет – один из вариантов ответов, в зависимости от вида ошибки (ошибка в имени / неполная или избыточная команда).
* (string) text – текст команды (в этом случае – имя).

# Файл mysqlScheduleParser.py

## Класс mysqlScheduleParser ( )

### Описание, структура класса

Класс, содержащий набор методов для добавления либо получения данных из БД Mysql.

Класс содержит следующие методы:

* \_\_init\_\_ (self)
* msp\_add\_employee (self, name)
* msp\_add\_location (self, location)
* search\_user\_id (self, name)
* search\_location\_id (self, location)
* msp\_set\_day\_schedule (self, text)
* msp\_set\_day\_schedule\_repeat (self, text)
* msp\_get\_day\_schedule (self, text)
* msp\_get\_day\_schedule\_everybody (self, text)
* msp\_get\_week\_schedule (self, week, text)
* msp\_get\_week\_schedule\_everybody (self, week)

### Метод \_\_init\_\_ ( )

Описание

Инициализация класса, создание переменных – экземпляра класса commandChecker ( ), экземпляра класса calendarSearcher ( ), соединения с БД MySql.

Параметры

Отсутствуют.

Возвращаемые значения

Отсутствуют.

### Метод msp\_add\_employee ( )

Описание

Метод добавляет сотрудника в таблицу Users БД. Если сотрудник с таким именем уже есть в БД, то метод не делает ничего.

Параметры

(string) name – имя сотрудника.

Возвращаемые значения

Один из вариантов ответа:

* (string) 'Новый сотрудник добавлен.'
* (string) 'Ошибка в добавлении сотрудника.'
* (string) 'Такой сотрудник уже есть в БД.'

#### Метод make\_query ( )

Описание

Метод проверяет есть ли сотрудник с таким именем в БД. Если нет, то метод формирует запрос для БД Mysql. Этот запрос добавит сотрудника в БД.

Параметры

(string) text – имя сотрудника.

Возвращаемые значения

(string) mysql\_query – сформированный запрос на языке Sql.

### Метод msp\_add\_location ( )

Описание

Метод добавляет место работы в таблицу Locations БД. Если место работы с таким названием уже есть в БД, то метод не делает ничего.

Параметры

(string) location – место работы.

Возвращаемые значения

Один из вариантов ответа:

* (string) 'Новое место работы добавлено.'
* (string) 'Ошибка в добавлении места работы.'
* (string) 'Такое место работы уже есть в БД.'

#### Метод make\_query ( )

Описание

Метод проверяет есть ли рабочее место с таким названием в БД. Если нет, то метод формирует запрос для БД Mysql. Этот запрос добавит место работы в БД.

Параметры

(string) text – место работы.

Возвращаемые значения

(string) mysql\_query – сформированный запрос на языке Sql.

### Метод search\_user\_id ( )

Описание

Метод находит id сотрудника по его имени.

Параметры

(string) name – имя сотрудника.

Возвращаемые значения

(int) id сотрудника.

### Метод search\_location\_id ( )

Описание

Метод находит id места работы по его названию.

Параметры

(string) location – название места работы.

Возвращаемые значения

(int) id места работы.

### Метод msp\_set\_day\_schedule ( )

Описание

Метод, который добавляет расписание в БД MySql. Сначала, метод проверяет правильность введённой команды с помощью методов класса commandChecker ( ).

Если команда введена верно, то формируется и выполняется запрос на языке Sql, который добавляет расписание в БД. Если команда введена неверно, то возвращается сообщение об ошибке.

Параметры

(string) text – полный текст команды.

Возвращаемые значения

Один из вариантов ответа:

* (string) 'Расписание успешно добавлено.'
* (string) 'Ошибка в добавлении расписания.'
* (string) answer – сообщение об ошибке.

#### Метод make\_query ( )

Описание

Формирует запрос для БД Mysql. Этот запрос добавляет данные в таблицу Schedule БД.

Параметры

* (list) command\_list – список слов из введённой команды.

Возвращаемые значения

(string) mysql\_query – сформированный запрос на языке Sql.

### Метод msp\_set\_day\_schedule\_repeat ( )

Описание

Метод, который добавляет расписание в БД MySql. Сначала, метод проверяет правильность введённой команды с помощью методов класса commandChecker ( ).

Если команда введена верно, то метод определяет текущий день недели, затем составляет список дат таких же дней недели в текущем и ближайших 3 месяцах. и во все эти дни добавляет одинаковое расписание.

Параметры

(string) text – полный текст команды.

Возвращаемые значения

Один из вариантов ответа:

* (string) 'Расписание успешно добавлено на 3 месяца вперёд.'
* (string) answer – сообщение об ошибке.

### Метод msp\_get\_day\_schedule ( )

Описание

Метод, который получает данные из БД MySql. Сначала, метод проверяет правильность введённой команды с помощью методов класса commandChecker ( ).

Если команда введена верно, то формируется и выполняется запрос на языке Sql, который получает из БД расписание одного сотрудника в указанный день. Если команда введена неверно, то возвращается сообщение об ошибке.

Параметры

(string) text – полный текст команды.

Возвращаемые значения

(string) – расписание человека в указанный день, либо сообщение, что в этот день человек не работает.

#### Метод make\_query ( )

Описание

Формирует запрос для БД Mysql. Этот запрос получает данные из БД Mysql.

Параметры

(list) command\_list – список слов из введённой команды.

Возвращаемые значения

(string) mysql\_query – сформированный запрос на языке Sql.

### Метод msp\_get\_day\_schedule\_everybody ( )

Описание

Метод, который получает данные из БД MySql. Сначала, метод проверяет правильность введённой команды с помощью методов класса commandChecker ( ).

Если команда введена верно, то формируется и выполняется запрос на языке Sql, который получает из БД расписание всех сотрудников в указанный день. Если команда введена неверно, то возвращается сообщение об ошибке.

Параметры

(string) text – полный текст команды.

Возвращаемые значения

(string) – расписание человека в указанный день, либо сообщение, что в этот день человек не работает.

#### Метод make\_query ( )

Описание

Формирует запрос для БД Mysql. Этот запрос получает данные из БД Mysql.

Параметры

(list) command\_list – список слов из введённой команды.

Возвращаемые значения

(string) mysql\_query – сформированный запрос на языке Sql.

### Метод msp\_get\_week\_schedule ( )

Описание

Метод, который получает расписание работы сотрудника во все дни указанной недели.

Параметры

* (int) week – номер недели.
* (string) text – имя сотрудника.

Возвращаемые значения

Один из вариантов ответа:

* (list) result\_list – список из расписаний сотрудника на каждый день недели.
* (list) – сообщение об ошибке.

### Метод msp\_get\_week\_schedule\_everybody ( )

Описание

Метод, который получает расписание работы ВСЕХ сотрудников во все дни указанной недели.

Параметры

* (int) week – номер недели.

Возвращаемые значения

Один из трех вариантов ответа:

* (list) result\_list – список из расписаний всех сотрудников на каждый день недели

# Файл Slackbot.py

## Функция show\_commands ( )

Описание

Функция, которая выводит список команд.

Параметры

(string) channel – канал.

Возвращаемые значения

Отсутствуют.

## Функция show\_commands\_examples ( )

Описание

Функция, которая выводит образцы всех команд.

Параметры

(string) channel – канал.

Возвращаемые значения

Отсутствуют.

## Функция main ( )

Описание

В зависимости от введённого текста выполняет действия.

Параметры

Отсутствуют.

Возвращаемые значения

Отсутствуют.