

Рабочий slack-бот

Описание программы

Оглавление

1. Описание	4
1.1. Файловая структура, описание основных функций	4
2. Файл constants.py	5
2.1. Описание	5
3. Файл calendarSearcher.py	5
3.1. Класс calendarSearcher ()	5
3.1.1. Описание, структура класса	5
3.1.2. Метод __init__ ()	5
3.1.3. Метод cs_search_days_week ()	5
3.1.3.1. Метод parse_month ()	6
3.1.4. Метод cs_search_days_month ()	6
3.1.5. Метод cs_search_same_weekdays ()	7
4. Файл commandChecker.py	8
4.1. Класс commandChecker ()	8
4.1.1. Описание, структура класса	8
4.1.2. Метод __init__ ()	8
4.1.3. Метод make_employees_list ()	8
4.1.4. Метод make_locations_list ()	9
4.1.5. Метод cc_check_name ()	9
4.1.6. Метод cc_check_date ()	9
4.1.7. Метод check_location ()	10
4.1.8. Метод cc_set_day_input_check ()	10
4.1.8.1. Метод check_time ()	11
4.1.8.2. Метод check_time_period ()	11
4.1.9. Метод cc_get_day_input_check ()	12
4.1.10. Метод cc_get_week_input_check ()	12
5. Файл mysqlScheduleParser.py	13
5.1. Класс mysqlScheduleParser ()	13
5.1.1. Описание, структура класса	13

5.1.2. Метод <code>__init__ ()</code>	13
5.1.3. Метод <code>mzp_add_employee ()</code>	14
5.1.3.1. Метод <code>make_query ()</code>	14
5.1.4. Метод <code>mzp_add_location ()</code>	14
5.1.4.1. Метод <code>make_query ()</code>	15
5.1.5. Метод <code>search_user_id ()</code>	15
5.1.6. Метод <code>search_location_id ()</code>	16
5.1.7. Метод <code>mzp_set_day_schedule ()</code>	16
5.1.7.1. Метод <code>make_query ()</code>	16
5.1.8. Метод <code>mzp_set_day_schedule_repeat ()</code>	17
5.1.9. Метод <code>mzp_get_day_schedule ()</code>	17
5.1.9.1. Метод <code>make_query ()</code>	18
5.1.10. Метод <code>mzp_get_day_schedule_everybody ()</code>	18
5.1.10.1. Метод <code>make_query ()</code>	18
5.1.11. Метод <code>mzp_get_week_schedule ()</code>	19
5.1.12. Метод <code>mzp_get_week_schedule_everybody ()</code>	19
6. Файл <code>Slackbot.py</code>	20
6.1. Функция <code>show_commands ()</code>	20
6.2. Функция <code>show_commands_examples ()</code>	20
6.3. Функция <code>main ()</code>	20

1. Описание

1.1. 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 введенных 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.

2. Файл constants.py

2.1. 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.

3. Файл calendarSearcher.py

3.1. Класс calendarSearcher ()

3.1.1. 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)

3.1.2. Метод __init__ ()

Описание

Инициализация класса.

Параметры

Отсутствуют.

Возвращаемые значения

Отсутствуют.

3.1.3. 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.

3.1.3.1. 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.

3.1.4. 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.

3.1.5. 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.

4. Файл commandChecker.py

4.1. Класс commandChecker ()

4.1.1. 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)

4.1.2. Метод __init__ ()

Описание

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

Параметры

Отсутствуют.

Возвращаемые значения

Отсутствуют.

4.1.3. Метод make_employees_list ()

Описание

Метод составляет список сотрудников (из таблицы Users БД).

Параметры

Отсутствуют.

Возвращаемые значения

(list) employees_list – список имён всех сотрудников.

4.1.4. 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.

4.1.5. Метод ss_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.

4.1.6. Метод ss_check_date ()

Описание

Метод проверяет правильность введенной 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.

4.1.7. 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.

4.1.8. Метод `cc_set_day_input_check ()`

Описание

Метод проверяет правильность введенной 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.

4.1.8.1. 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.

4.1.8.2. 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.

4.1.9. 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.

4.1.10. 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).

5. Файл mysqlScheduleParser.py

5.1. Класс mysqlScheduleParser ()

5.1.1. 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)

5.1.2. Метод __init__ ()

Описание

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

Параметры

Отсутствуют.

Возвращаемые значения

Отсутствуют.

5.1.3. 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.'

5.1.3.1. 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.

5.1.4. 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.'

5.1.4.1. 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.

5.1.5. Метод search_user_id ()

Описание

Метод находит id сотрудника по его имени.

Параметры

(string) name – имя сотрудника.

Возвращаемые значения

(int) id сотрудника.

5.1.6. 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.

5.1.7. Метод `msp_set_day_schedule ()`

Описание

Метод, который добавляет расписание в БД MySQL. Сначала, метод проверяет правильность введенной 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.

5.1.7.1. Метод `make_query ()`

Описание

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

Параметры

✓ (list) `command_list` – список слов из введенной команды.

Возвращаемые значения

(string) `mysql_query` – сформированный запрос на языке Sql.

5.1.8. Метод `mzp_set_day_schedule_repeat ()`

Описание

Метод, который добавляет расписание в БД MySQL. Сначала, метод проверяет правильность введенной 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 вперед.'
- ✓ (string) `answer` – сообщение об ошибке.

5.1.9. Метод `mzp_get_day_schedule ()`

Описание

Метод, который получает данные из БД MySQL. Сначала, метод проверяет правильность введенной 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.

5.1.9.1. Метод `make_query ()`

Описание

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

Параметры

(list) `command_list` – список слов из введенной команды.

Возвращаемые значения

(string) `mysql_query` – сформированный запрос на языке Sql.

5.1.10. Метод `mvp_get_day_schedule_everybody ()`

Описание

Метод, который получает данные из БД MySql. Сначала, метод проверяет правильность введенной 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.

5.1.10.1. Метод `make_query ()`

Описание

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

Параметры

(list) `command_list` – список слов из введенной команды.

Возвращаемые значения

(string) `mysql_query` – сформированный запрос на языке Sql.

5.1.11. Метод `mzp_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.

5.1.12. Метод `mzp_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

6. Файл Slackbot.py

6.1. Функция show_commands ()

Описание

Функция, которая выводит список команд.

Параметры

(string) channel – канал.

Возвращаемые значения

Отсутствуют.

6.2. Функция show_commands_examples ()

Описание

Функция, которая выводит образцы всех команд.

Параметры

(string) channel – канал.

Возвращаемые значения

Отсутствуют.

6.3. Функция main ()

Описание

В зависимости от введённого текста выполняет действия.

Параметры

Отсутствуют.

Возвращаемые значения

Отсутствуют.