

**СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ «КАТРАПС» (СУБД
«КАТРАПС») ВЕРСИИ 1.10.11**

МОДУЛЬ МАСКИРОВАНИЯ СУБД «КАТРАПС»

Листов 12

МОСКВА

2025

Оглавление

НАЗНАЧЕНИЕ МОДУЛЯ МАСКИРОВАНИЯ.....	2
РАЗВЁРТЫВАНИЕ МОДУЛЯ МАСКИРОВАНИЯ.....	3
СТРУКТУРА КАТАЛОГОВ ХРАНЕНИЯ.....	3
ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ МОДУЛЯ МАСКИРОВАНИЯ.....	4
ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ.....	9
ПРИЛОЖЕНИЕ 1. Версии Python (предустановлена в системе).....	9
ПРИЛОЖЕНИЕ 2. Заимствованные пакеты.....	9
ПРИЛОЖЕНИЕ 3. Структура модуля маскирования.....	10
ПРИЛОЖЕНИЕ 4. Хранимые процедуры.....	12

НАЗНАЧЕНИЕ МОДУЛЯ МАСКИРОВАНИЯ

СУБД «КАТРАПС», наряду со стандартными способами управления базами данных, обладает расширенным функционалом, в том числе обеспечивающим удаление остаточной информации в объектах баз данных. Его применение можно подразделить на два основных направления:

- маскирование остаточной информации, содержащейся в полях таблиц баз данных;
- маскирование остаточной информации, содержащейся в графических файлах изображений (*PNG, JPEG* и т.д.), текстовых файлах, таких как *JSON, XML* и т.п., хранящихся в специализированных каталогах, путь к размещению которых прописан в виде текстовых ссылок в полях таблиц базы данных.

Модуль выполнен в виде библиотеки функций, доступ к которой осуществляется в основном с помощью консольной утилиты. Он написан на языке программирования *python3* (далее фреймворк) с использованием встроенных в него плагинов. Список поддерживаемых для установки из доверенных репозиториях сертифицированных операционных систем версий языка python приведен в *Приложении 1. Версии Python*. Дополнительно используются заимствованные пакеты (устанавливаются в среде выполнения командой *pip install **, где *** означает название пакета, а *pip* является стандартным менеджером пакетов для *Python*).

Описание заимствованных *модулем маскирования* Python пакетов приведено в *Приложении 2. Заимствованные модули*.

Составные части модуля по их функционалу:

- Библиотека функций (оформлена в виде класса *k_lib.pyс*) (*Приложение 3. Структура модуля маскирования*);
- Хранимые процедуры, загружаемые при установке в схему базы данных *mysql* с назначением предустановленных прав на выполнение согласно ролевой модели СУБД «КАТРАПС» (*Приложении 4. Хранимые процедуры*).

Основной модуль представляет собой консольную утилиту *katraps_menu_class.pyс*, позволяющую в консольном режиме вызывать выполнение плагина в соответствии с настроенными действиями пользователя посредством ввода аргументов (наименование схемы в базе данных, действия с базой данных, символа маскирования), либо вызывать их на выполнение поочередно, выбирая из соответствующих подпунктов меню.

Библиотека функций *k_lib.pyс* также может быть использована в качестве внешнего API для доступа из пользовательских программ либо в версии GUI - десктопного приложения.

РАЗВЁРТЫВАНИЕ МОДУЛЯ МАСКИРОВАНИЯ

Для полноценной работы модуля маскирования необходимо:

- развернуть хранимые процедуры в схеме *mysql*:
 - ✓ *proc_kat_rec_all*;
 - ✓ *proc_kat_rec*;
 - ✓ *proc_kat_func_fields*;
 - ✓ *kat_find_schema*;
 - ✓ *kat_select*;
 - ✓ *kat_tables*;
 - ✓ *kat_mask*.
- создать следующие роли с предустановленными грантами для работы с СУБД:
 - ✓ *katrapsAdmin*;

- ✓ `katrapsDB`;
- ✓ `katrapsUser`.
- назначить пользователей для каждой роли доступа:
 - ✓ `userkatAdmin`;
 - ✓ `userkatDB`;
 - ✓ `userkatUser`.

Установка осуществляется следующим:

- выполнить скрипт создания ролевой модели, пользователей для этих ролей, хранимых процедур и функций в БД `mysql`:

```
> source /<путь установки>/sql/kat_mask.sql
```

- выполнить скрипт загрузки содержимого хранимых процедур и функций из отдельных файлов:

```
> source /<путь установки>/sql/kat_mask.sql
```

- выполнить скрипт развертывания и наполнения тестовой схемы `test`:

```
> source /<путь установки>/sql/kat_test.sql
```

СТРУКТУРА КАТАЛОГОВ ХРАНЕНИЯ

Структура каталогов установленного модуля выглядит следующим образом:

```
/<путь установки>/mask/  
--- config  
----- katraps.env  
----- katraps.txt  
--- graf_src  
----- dsp.png  
----- dsp.txt  
--- lib  
----- k_lib.pyс  
--- katraps_menu_class.pyс
```

**ПРИМЕРЫ
ИСПОЛЬЗОВАНИЯ
МАСКИРОВАНИЯ**

МОДУЛЯ

Пример вывода информации о использовании

```
> cd /<путь установки>/mask/  
> python3 katraps_menu_class.pyс
```

или для получения помощи по использованию:

```
> cd /<путь установки>/mask/  
> python3 katraps_menu_class.pyс --help
```

```
##### (C)2024 - ООО `Иновационные технологии` #####
### МОДУЛЬ УДАЛЕНИЯ остаточной информации в СУБД КАТРАПС ###
#####
РЕЛЯЦИОННАЯ СУБД - ... сертификат ФСТЭК РФ (2024 г.) .....
НАЗНАЧЕНИЕ МОДУЛЯ - .. маскирование остаточной информации ..
ООО Иновационные Технологии - https://techinnovations.ru...

katraps_arg_class.py
  [-m, ->Список баз данных]
  [-d, ->База данных для обработки]
  [-a, ->Действие по обработке БД]
  [-l, ->Символ маскирования остаточной информации]
  -> укажите в командной строке через пробел...

->Введено неверное значение аргумента (или не указано). Доступны:
  Схемы: catraps_test
  Действия: ['view', 'backup', 'restore', 'mask', '-> exit']
  Символы: ['*', '#', '-', '@', '$', '-> выйти']:
-h, --help          ->Введено неверное значение аргумента (или не указано). Доступны: Схемы: catraps_test
                    Действия: ['view', 'backup', 'restore', 'mask', '-> exit'] Символы: ['*', '#', '-', '@',
                    '$', '-> выйти']
-m                  ->Выбор базы данных из списка в меню
-d {*}              ->База данных для обработки
-a {просмотр содержимого table (view) БД,сделать dump,восстановить из dump,маскировать записи таблиц,-> выйти}
                    ->Действие по обработке БД
-l {*,#,-,@,$,-> выйти}
                    ->Символ маскирования остаточной информации
```

Пример вывода при неполном вводе

```
> python katraps_menu_class.py -m
```

```
***** Проверим наличие обязательных хранимых процедур. Всего:4
- mysql.kat_find_schema +
- mysql.kat_select +
- mysql.kat_tables +
- mysql.proc_kat_rec_all +
- mysql.proc_kat_rec +
- mysql.proc_kat_func_fields +

***** Найдены базы данных: *****
1 cargo
2 catraps_test
3 katraps_test
4 test
5 -> выйти

#####

### УДАЛЕНИЕ остаточной информации СУБД КАТРАПС ###

#####

Имя пользователя `root`, введите пароль (-или 0 для выхода): |
```

После прохождения идентификации:

```

***** Проверим наличие обязательных хранимых процедур. Всего:4
- mysql.kat_find_schema +
- mysql.kat_select +
- mysql.kat_tables +
- mysql.proc_kat_rec_all +
- mysql.proc_kat_rec +
- mysql.proc_kat_func_fields +

***** Найдены базы данных: *****
1 cargo
2 catraps_test
3 katraps_test
4 test
5 -> выйти

#####

### УДАЛЕНИЕ остаточной информации СУБД КАТРАПС ###

#####

Имя пользователя `root`, введите пароль (-или 0 для выхода-): *****
->Пароль корректен: *****

***** Найдены базы данных: *****
1 cargo
2 catraps_test
3 katraps_test
4 test
5 -> выйти
Выберите БД из меню по номеру: |

```

При неверной идентификации для удобства пользователю отображается введённый им неверный пароль:

```

#####

### УДАЛЕНИЕ остаточной информации СУБД КАТРАПС ###

#####

Имя пользователя `root`, введите пароль (-или 0 для выхода-): *****
->Введён некорректный пароль: 11111

#####

### УДАЛЕНИЕ остаточной информации СУБД КАТРАПС ###

#####

Имя пользователя `root`, введите пароль (-или 0 для выхода-): 

```

Пример вывода информации до маскирования:

```

***** Найдены базы данных: *****
 1 cargo
 2 catraps_test
 3 katraps_test
 4 test
 5 -> выйти
Выберите БД из меню по номеру: 2

-> выбрано: `catraps_test` и номер `2`

*** Действие с БД `catraps_test`: ***
 1 просмотр содержимого table (view) БД
 2 сделать dump
 3 восстановить из dump
 4 маскировать записи таблиц
 5 -> выйти
Выберите действие по номеру: 1
    
```

```

БД `catraps_test`, ТАБЛИЦА - `cards`
+-----+-----+-----+-----+-----+
| cards_id | cards_fio | cards_line | cards_station | cards_tabel |
+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 2 | 1 |
| 2 | 2 | 1 | 2 | 2 |
| 3 | 3 | 1 | 2 | 3 |
| 4 | 4 | 1 | 2 |  |
| 5 | 5 | 1 | 2 |  |
| 6 | 6 | 1 | 2 |  |
| 7 | 7 | 1 | 2 |  |
| 8 | 8 | 1 | 2 |  |
| 9 | 9 | 1 | 2 |  |
| 10 | 10 | 1 | 2 |  |
+-----+-----+-----+-----+-----+

БД `catraps_test`, ТАБЛИЦА - `employee`
+-----+-----+
| employee_id | employee_fio |
+-----+-----+
| 1 | ЧЖАМ Э.В. |
| 2 | ГЮЛ .. |
| 3 | ВФКИНОЕЭД Ж.Т. |
| 4 | НЦЕНК У.З. |
| 5 | КДОГОЧ Я.Н. |
| 6 | КЭАЖ .. |
| 7 | ПЛЛО .В. |
| 8 | НЖЯАУА Ф.. |
| 9 | САГП У.Ц. |
| 10 | ЮНА .. |
+-----+-----+
    
```

Пример вывода маскированной информации (выбираем символ `*`):

```

***** Найдены базы данных: *****
 1 cargo
 2 catraps_test
 3 katraps_test
 4 test
 5 -> выйти
Выберите БД из меню по номеру: 2

-> выбрано: `catraps_test` и номер `2`

*** Действие с БД `catraps_test`: ***
 1 просмотр содержимого table (view) БД
 2 сделать dump
 3 восстановить из dump
 4 маскировать записи таблиц
 5 -> выйти
Выберите действие по номеру: 4

-> выбрано: `маскировать записи таблиц` и номер `4`
*** Выберите символ маскирования для БД `catraps_test`: ***
 1 *
 2 #
 3 -
 4 @
 5 $
 6 -> выйти
Выберите маскирование по номеру: 1
    
```

```
-> выбрано: `*` и номер `1`
##### Маскирование #####
### Обработаны таблицы из `catraps_test`: cards, employee, line, station, worksheet
#####
***** Найдены базы данных: *****
 1 catraps_test
 2 katraps_test
 3 test
 4 -> выйти
Выберите БД из меню по номеру: |
```

Результат маскирования при последующем `просмотр содержимого table (view) БД` таблиц символом `*` и целочисленным кодом выбранного символа, в случае не символьного типа данных, выглядят следующим образом:

```
-> выбрана: catraps_test
БД `catraps_test`, ТАБЛИЦА - `cards`
+-----+-----+-----+-----+-----+
| cards_id | cards_fio | cards_line | cards_station | cards_tabel |
+-----+-----+-----+-----+-----+
| 1 | 42 | 42 | 42 | 42 |
| 2 | 42 | 42 | 42 | 42 |
| 3 | 42 | 42 | 42 | 42 |
| 4 | 42 | 42 | 42 | 42 |
| 5 | 42 | 42 | 42 | 42 |
| 6 | 42 | 42 | 42 | 42 |
| 7 | 42 | 42 | 42 | 42 |
| 8 | 42 | 42 | 42 | 42 |
| 9 | 42 | 42 | 42 | 42 |
| 10 | 42 | 42 | 42 | 42 |
+-----+-----+-----+-----+-----+
БД `catraps_test`, ТАБЛИЦА - `employee`
+-----+-----+
| employee_id | employee_fio |
+-----+-----+
| 1 | * |
| 2 | * |
| 3 | * |
| 4 | * |
| 5 | * |
| 6 | * |
| 7 | * |
| 8 | * |
| 9 | * |
| 10 | * |
+-----+-----+
```

При типе данных *Data* и *DataTime* значения конкретных дат и времени заменяются 0-ми в соответствии с выбранным форматом поля данных:

```
БД `catraps_test`, ТАБЛИЦА - `worksheet`
+-----+-----+-----+
| worksheet_id | worksheet_start | worksheet_end |
+-----+-----+-----+
| 1 | 8:00:00 | 17:00:00 |
| 2 | 9:00:00 | 18:00:00 |
| 3 | 10:00:00 | 19:00:00 |
+-----+-----+-----+
```

Маскируется следующим образом:

```

Бд `catraps_test`, ТАБЛИЦА - `worksheet`
+-----+-----+-----+
| worksheet_id | worksheet_start | worksheet_end |
+-----+-----+-----+
| 1 | 0:00:00 | 0:00:00 |
| 2 | 0:00:00 | 0:00:00 |
| 3 | 0:00:00 | 0:00:00 |
+-----+-----+-----+

```

ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ

ПРИЛОЖЕНИЕ 1. Версии Python (предустановлена в системе)

Сертифицированная операционная система	Менеджер пакетов для Python PIP, версия	Python
<i>Alt OS</i>	22.2.2	3.9

ПРИЛОЖЕНИЕ 2. Заимствованные пакеты

Наименование	Версия	Описание
<i>console-menu</i>	0.8.0	- user-interface меню для терминальных приложений
<i>ConsoleMenu</i>	1.0.1	- пакет реализации консольного меню на основе словарей (dict) без использования классов
<i>configparser</i>	7.1.0	- пакет предназначен для чтения значений конфигурации из текстового файла
<i>mysql-connector-python</i>	8.3.0	- пакет для доступа к СУБД
<i>pillow</i>	10.2.0	- пакет работы с графическими примитивами
<i>tabulate</i>	0.9.0	- форматированный вывод табличных данных в консоли
<i>getpass</i>	0.0.14.1	- ввод пароля для аутентификации без отображения на экране

ПРИЛОЖЕНИЕ 3. Структура модуля маскирования

1) Основной модуль консольного приложения `katraps_menu_class.py` состоит из следующего функционала:

```

### ***** * `ФУНКЦИИ`:
## __init__ - Инициализация класса `KATRAPS`
## fromTV - Чтение таблиц и вьюшек
## menu_dbase - Вывод меню с БД
## header_old - Вывод заголовков для набора меню
## func_viewer - Вывод содержимого таблиц и вьюшек
## func_sub - Вывод сообщения о выбранном пункте
## menuClick - Обработка выбранного пункта меню/подменю
## find_dump - Определим список dump`s
## func_dump - Создаём/загружаем дампы (таблиц и хранимых процедур)

## check_number - Проверка правильности введённого номера
## menu_sub - Работа с подменю
## main - Работа с меню (вызов многократно)

### ***** * `ЗАВИСИМОСТИ`:
## lib.k_lib - Библиотека вспомогательных функций `KLB`
    
```

2) Библиотечный модуль `k_lib.py` состоит из следующего функционала:

```

## - 14.10.2024 -
## - KLB - библиотека -

### ***** * `ФУНКЦИИ`:
## __init__ - инициализация класса `KLB`
## klb_cls - очистка консоли
## klb_read_file - чтение заголовка из файла
## klb_read - из основного файла конфигурации
## klb_conn_pars - чтение настроек основного файла конфигурации
## klb_write - сохраним выбранный символ маски
## klb_read_dsp - считаем из dsp конфигурации
## klb_is_number - проверка на числовой ввод
## klb_rus - руссификация сообщений
### ***** * `АРГУМЕНТЫ`:
## klb_arg_err_msg - сообщение об ошибке с аргументом
## klb_arg_msg - сообщение при отсутствии аргумента
## klb_arg_create - создание опциональных аргументов
## klb_arg_check - проверки аргументов
### ***** * `СОЕДИНЕНИЕ`:
## klb_connect - соединение с БД
## klb_path_mdb - определим путь к `KATRAPS` в ОС
### ***** * `МЕНЮ`:
## klb_menu_schema - получить все текущие схемы
## klb_menu_view - вывод списка схем
## klb_menu_create - коннект и формирование меню
## klb_menu_main - управление созданием меню
### ***** * `ГРАФИКА`:
## klb_klb_creg - очистка холста
## klb_coords - подсчёт блоков сетки
## klb_draw - сетка блоков на canvas
## klb_mask - выбор блока для canvas
### ***** * `ТЕКСТ`:
## klb_rpl_db - открыть таблицу ,считать ссылки, определить тип файла (graf / txt)
    
```

```
## klb_rpl_read           - чтение файла
## klb_rpl_preread       - вывод содержимого до/после
## klb_rpl_write         - запись маски в файл
## klb_rpl_chls          - проверка на space
## klb_txt               - вызов всей текстовой обработки
## klb_sh_no_arg         - шапка с вводом пароля без аргументов
## klb_proc_text         - вывод содержимого хранимых процедур / функций
## klb_minus_symb1      - удаляет двойные кавычки из элемента списка
```

3) Структура файла предустановленных настроек окружения `katraps.env`:

```
[INFO]
prog = СУБД КАТРАПС 1.11.0
port = 3306
host = localhost
user = root

[CONTENTS]
zgl_txt =
***** (C)2024 - ООО `Инновационные технологии` *****
РЕЛЯЦИОННАЯ СУБД - ... сертификат ФСТЭК РФ (2024 г.) .....
НАЗНАЧЕНИЕ МОДУЛЯ - .. маскирование остаточной информации ..
ООО Инновационные Технологии - https://techinnovations.ru...
*****
zgl = «КАТРАПС маскирование»
contents = 'ФЕДЕРАЛЬНАЯ СЛУЖБА','ПО ТЕХНИЧЕСКОМУ','И ЭКСПОРТНОМУ','КОНТРОЛЮ','(ФСТЭК
России)',',','Предназначено для служебного использования',';','Данный материал содежит информацию',';- не для
общественного использования',';- секретные сведения и т.д.'
```

```
[DB]
routines =
'mysql.proc_kat_schema','mysql.kat_select','mysql.kat_tables','mysql.proc_kat_rec_all','mysql.kat_mask','mysql.proc_
kat_rec'
functions = 'mysql.proc_kat_func_fields'
data = test

[MESSAGE]
f_footer = 'Выберите БД из меню по номеру: ','Выберите действие по номеру: ','Выберите маскирование по
номеру: ','-> выйти'
f_header = '*** Выберите действие для БД ','*** Действие с БД ','*** Выберите символ маскирования для БД
','*** Восстановить БД ''
f_sub = 'просмотр содержимого table (view) БД','сделать dump','восстановить из dump','маскировать записи
таблиц','маскировать внешние файлы по табличным ссылкам','-> выйти'
f_sub_eng = 'view', 'backup', 'restore', 'maskTable','maskFile','-> exit'
f_aster = '*','#','-', '@','$','-> выйти'
```

```
[ARG]
arg_help = '->Выбор базы данных из списка в меню','->База данных для обработки','->Действие по обработке
БД','->Символ маскирования остаточной информации','->Проверка прав пользователя'
active_mask = #
```

4) Описание заголовочного файла `katraps.txt`:

```
##### (C)2024 - ООО `Инновационные технологии` #####
## МОДУЛЬ МАСКИРОВАНИЯ остаточной информации СУБД КАТРАПС ##
#####
РЕЛЯЦИОННАЯ СУБД - ... сертификат ФСТЭК РФ (2024 г.) .....
НАЗНАЧЕНИЕ МОДУЛЯ - .. маскирование остаточной информации ..
ООО Инновационные Технологии - https://techinnovations.ru...
```

ПРИЛОЖЕНИЕ 4. Хранимые процедуры

В схеме ``mysql`` развёрнутой на сервере базы данных устанавливаются дополнительно следующие хранимые процедуры и функции:

Наименование	Описание
Процедуры:	
<code>proc_kat_schema</code>	- поиск существующих схем баз данных
<code>proc_kat_rec_all</code>	- выбор всех таблиц из выбранной схемы
<code>proc_kat_rec</code>	- выбор столбцов с определениями типов из таблицы
<code>kat_select</code>	- получение информации по выбранной таблице
<code>kat_tables</code>	- вспомогательная для <code>`proc_kat_rec`</code>
<code>kat_mask</code>	- процедура маскирования
Функции:	
<code>proc_kat_func_fields</code>	- функция присвоения выбранного символа маскирования

Установка хранимых процедур и функций в схему ``mysql`` осуществляется посредством запуска SQL-скрипта ``kat_main.sql`` из каталога ``<путь к папке>/sp_sql/``:

Наименование	Описание
Скрипты создания хранимых процедур, ролей, пользователей и назначения грантов (в схеме <code>`mysql`</code>):	
<code>kat_mask.sql</code>	- создание ролевой модели, пользователей, хранимых процедур и функций в БД <code>`mysql`</code>
<code>kat_mysql.sql</code>	- загрузка содержимого хранимых процедур и функций из отдельных файлов
Демо-приложение (в схеме <code>`test`</code>):	
<code>kat_test_load.sql</code>	- создание демо БД <code>`test`</code> с загрузкой данных <code>`tables`</code> и <code>`view`</code> для демонстрации работы модуля