

## Аннотация к дипломной работе Гусева Михаила

В ходе комплексного тестирования программных средств возникает необходимость интерпретации результатов множества тестов, написанных по различным правилам и для различных целей. Для решения задачи автоматизации запуска, сбора информации и интерпретации результатов тестирования необходимо привести интерфейсную часть всех тестирующих программ к единообразному виду позволяющему с наименьшими затратами решать поставленную задачу. Для наиболее простого решения, исключающего возможность административного взаимонепонимания при реализации интерфейсов для обработки входных параметров и систематизации выходных данных, предлагается использовать единую библиотеку с небольшим прикладным программным интерфейсом — API. При этом, в силу того что приложения тестирования обеспечивают взаимодействие с оборудованием, необходимо обеспечить полную поддержку многопоточности в такой библиотеке.

Функция инициализации библиотеки

```
int tioInit(const char* version, const char* help,  
            const tio_param _param[], int argc, char *argv[]);
```

Функция `tioInit` осуществляет первоначальную инициализацию библиотеки и списка параметров доступных приложению. Функция принимает 5 параметров:

1. версия теста
2. короткое описание назначения теста
3. список параметров принимаемых приложением и тех ключей для параметров что используются в данном приложении. Признаком конца списка параметров является структура `tio_param` у которой все поля имеют нулевое значений.
4. количество аргументов командной строки
5. список аргументов командной строки

`tio_param` представляет собой структуру вида:

```
typedef struct _tio_param  
{  
    char *key;  
    char *name;  
    char* description;  
} tio_param;
```

Где key ключ используемый при вызове из командной строки, name — имя параметра используемое при взаимодействии приложения с библиотекой, а description - короткое описание предназначение параметра.

Функции получения параметров.

Для получения параметров используется семейство функций вида tioGet\* и tioGetDef\*. Эти функции предназначены для получения параметров по их имени. функции оканчиваются на суффикс, означающий возвращаемое значение: L - long, D - double, C - char, S - char\* ( string ).

Коды ошибок при неверном завершении функции:

TESUC	успешное завершение ( 0 )
TENOPAR	параметр не зарегистрирован при инициализации библиотеки
TEINCTYPE	Параметр не может быть приведен к запрошенному типу
TENOTSET	параметр не передан при вызове приложения.
TENES	размер буфера недостаточно велик для помещения параметра
TEFAILL	Отказ по непонятным причинам

Функции вывода

```
int tioPrint(const char* messge);  
int tioPrintF(const char* template, ... );
```

Функции размещают в потоке вывода сообщения message или форматирования сообщение по шаблону template.

Функции табличного вывода

Предназначены для вывода информации структура полей которой может быть четко описана по числу полей. Для работы с табличной формой представления определено следующее семейство функций:

```
int tioTableBegin(const char* format, ... )  
int tioTableEnd();  
int tioTableRecord(int num; ... );
```

Первая из функций отвечает за инициализацию таблицы и назначение ей заголовков и типов выводимых значений. tioTableRecord — предназначена для добавления новой строки данных в таблицу. tioTableRecord — предназначена для вывода таблицы на экран и освобождения ресурсов, выделенных для создания таблицы.