

СОГЛАСОВАНО

Начальник
межгалактической комиссии

А.Б. Заказчиков
«___» 2022 г.

УТВЕРЖДАЮ

Главный конструктор
изделия АБВГД.12345

А.Б. Главный
«___» 2022 г.

ИЗДЕЛИЕ АБВГД.12345
ПРОГРАММНЫЙ КОМПЛЕКС
ГАЛАКТИЧЕСКИЙ ТРАНКЛЮКАТОР

Пример оформления документации

ЛИСТ УТВЕРЖДЕНИЯ

01234.56789-01 ПРИМЕР 03-ЛУ

Инф. № подл.	Подл.и дата	Взам. инф. №	Инф. № дубл.	Подл.и дата

От межгалактической комиссии
А.Б. Галактионов
«___» 2022 г.

Начальник Центра
А.Б. Чатланин
«___» 2022 г.

Зам. гл. конструктора
А.Б. Заместителев
«___» 2022 г.

Разработчик
А.Б. Разработчиков
«___» 2022 г.

Исполнитель
А.Б. Пацак
«___» 2022 г.

Нормоконтроллер
«___» 2022 г.

2022

Перф. признак.
АБВГД.12345

УТВЕРЖДЕН
01234.56789-01 ПРИМЕР 03-ЛУ

ИЗДЕЛИЕ АБВГД.12345
ПРОГРАММНЫЙ КОМПЛЕКС
ГАЛАКТИЧЕСКИЙ ТРАНКЛЮКАТОР

Пример оформления документации

01234.56789-01 ПРИМЕР 03

Листов 28

Инф. № подл.	Подл.и.дата	Ведм. инф. №	Инф. № дубл.	Подл.и.дата

2022

АННОТАЦИЯ

Данный документ является примером оформления текста с использованием системы верстки (La)TeX. Ссылка: <https://en.wikipedia.org/wiki/LaTeX>. Отличительной чертой проекта, намного повышающей удобство использования, является использование файла UseLatex.cmake, который позволяет легко и просто собирать исходные тексты из *.tex файлов путем написания соответствующего CMakeLists.txt (пример имеется в директории проекта) и вызова процесса сборки стандартным способом: `mkdir build && cd build && cmake .. && make`.

Доработанный класс espd.cls позволяет легко и просто оформлять титульную страницу и лист утверждения по ГОСТ-19, а также включает все необходимое оформление. Таким образом, использование данного класса и языка разметки (La)TeX позволяет техническому писателю сконцентрироваться на главном — написании текста. Оформление формул, таблиц, вставка рисунков также значительно упрощаются, исключается их «съезжание», как часто случается при использовании текстового редактора Word, особенно разных версий.

Далее изложены наиболее часто встречающиеся конструкции, необходимые для написания текста технического задания и остальной документации по ГОСТ-19.

СОДЕРЖАНИЕ

1. Оформление иерархии вложенности разделов	5
1.1. Подраздел 1	5
1.2. Подраздел 2	5
1.2.1. Пункт 1	5
1.2.2. Пункт 2	5
1.2.2.1. Подпункт 1	5
1.2.2.2. Подпункт 2	5
2. Оформление перечислений	6
2.1. Пример одноуровневого нумерованного перечисления	6
2.2. Пример одноуровневого ненумерованного перечисления	6
2.3. Пример вложенного перечисления	6
3. Оформление иллюстраций	8
4. Оформление формул	9
4.1. Простые примеры	9
4.1.1. Формула без присвоения порядкового номера	9
4.1.2. Формула с автоприсвоением порядкового номера	9
5. Оформление таблиц	10
5.1. Простая маленькая таблица	10
5.2. Широкая таблица с длинными заголовками колонок	10
5.3. Часто встречающаяся в документации таблица	11
6. Оформление листингов исходного кода	13
Перечень использованных источников	20
Перечень терминов	21
Перечень сокращений	22
Приложение 1. Пример приложения с номером и без разделов	23

Приложение 2. Пример приложения с номером и своими разделами	25
1. Первый раздел приложения	25
1.1. Пример формул в приложении	25
1.2. Пример рисунков в приложении	25
1.3. Пример таблицы в приложении	26
2. Второй раздел приложения	27
2.1. Подраздел 1	27
2.2. Подраздел 2	27
2.2.1. Пункт 1	27
2.2.2. Пункт 2	27
2.2.2.1. Подпункт 1	27
2.2.2.2. Подпункт 2	27

2. ОФОРМЛЕНИЕ ПЕРЕЧИСЛЕНИЙ

В данном разделе приводится пример оформления перечислений по п. 2.1.6 ГОСТ 19.106 [1].

2.1. Пример одноуровневого нумерованного перечисления

2.2. Пример одноуровневого ненумерованного перечисления

2.3. Пример вложенного перечисления

При таком перечислении 1 уровень делается нумерованным, 2 уровень — ненумерованным. Уровень 3 и далее — не рекомендуются.

3. ОФОРМЛЕНИЕ ИЛЛЮСТРАЦИЙ

В данном разделе приводится пример оформления иллюстраций по п. 2.3 ГОСТ 19.106 [1].

Иллюстрации, если их в документе более одной, нумеруют арабскими цифрами в пределах всего документа. В приложениях иллюстрации нумеруются в пределах каждого приложения аналогично как в основной части документа.



Рис. 1. Тестовое изображение «Лена»



Рис. 2. Тестовое изображение «Лена», вставленное еще раз для примера нумерации иллюстраций и уменьшенное в 2 раза

В тексте документа возможно вставлять ссылки на иллюстрации, например так:
см. рис. 1 или см. рис. 2.

4. ОФОРМЛЕНИЕ ФОРМУЛ

В данном разделе приводится пример оформления формул по п. 2.4 ГОСТ 19.106 [1].

4.1. Простые примеры

4.1.1. Формула без присвоения порядкового номера

Пример формулы, вставляемой в тексте без присвоения порядкового номера: формула квадратного многочлена: $f(x) = ax^2 + bx + c$, где a — первый (старший) коэффициент, b — второй (средний) коэффициент, c — свободный член.

4.1.2. Формула с автоприсвоением порядкового номера

Пример формул с присвоением порядкового номера и без удаления пробелов:

$$x = y + a \tag{1}$$

$$z = a + x \tag{2}$$

где: x — коэффициент 1;

z — коэффициент 2;

y — параметр 1;

a — параметр 2.

Пример ссылки на формулу: см. формулу (1).

Пример формул с присвоением порядкового номера и с удалением пробелов:

$$x = y + a \tag{3}$$

$$z = a + x \tag{4}$$

где: x — коэффициент 1;

z — коэффициент 2;

y — параметр 1;

a — параметр 2.

5. ОФОРМЛЕНИЕ ТАБЛИЦ

В данном разделе приводится пример иерархии вложенности по п. 2.6 ГОСТ 19.106 [1].

Оформление всегда следует вести при помощи класса `longtable`, поскольку это дает возможность переноса длинной таблицы на следующую страницу.

5.1. Простая маленькая таблица

Простой пример маленькой таблицы с маленькими колонками, выровненными по центру, без каких-либо переполнений.

Таблица 1 – Пример маленькой таблицы

колонка 1	колонка 2	колонка 3
111	222	333

5.2. Широкая таблица с длинными заголовками колонок

Пример таблицы с длинными заголовками колонок

Таблица 2 – Пример таблицы с возможно очень длинным заголовком, который будет перенесен на вторую строчку

колонка 1 с очень длинным заголовком, просто капец каким длинным	колонка 2 по центру	колонка 3 по правому краю
Содержание колонки 1	Содержание колонки 2	Содержание колонки 3, возможно очень длинное содержание, которое нормально отображается с переносом по словам

Пример оформления ссылки на таблицу: см. таблицу 2.

5.3. Часто встречающаяся в документации таблица

Пример таблицы, часто встречающийся в программной документации:

Таблица 3 – Пример таблицы

Номер слова	Наименование информации	Усл. об.	Размерн.	Пределы изменения	Примеч.
1	Контрольное слово	CW_	б/р	–	uint
2	Контрольное слово	CW_	б/р	–	uint
3	Контрольное слово	CW_	б/р	–	uint
4	Контрольное слово	CW_	б/р	–	uint
5	Контрольное слово	CW_	б/р	–	uint
6	Контрольное слово	CW_	б/р	–	uint
7	Контрольное слово	CW_	б/р	–	uint
8	Контрольное слово	CW_	б/р	–	uint
9	Контрольное слово	CW_	б/р	–	uint
10	Контрольное слово	CW_	б/р	–	uint
11	Контрольное слово	CW_	б/р	–	uint
12	Контрольное слово	CW_	б/р	–	uint
13	Контрольное слово	CW_	б/р	–	uint
14	Контрольное слово	CW_	б/р	–	uint
15	Контрольное слово	CW_	б/р	–	uint
16	Контрольное слово	CW_	б/р	–	uint
17	Контрольное слово	CW_	б/р	–	uint
18	Контрольное слово	CW_	б/р	–	uint
19	Контрольное слово	CW_	б/р	–	uint
20	Контрольное слово	CW_	б/р	–	uint
20	Контрольное слово	CW_	б/р	–	uint
20	Контрольное слово	CW_	б/р	–	uint
20	Контрольное слово	CW_	б/р	–	uint
20	Контрольное слово	CW_	б/р	–	uint
20	Контрольное слово	CW_	б/р	–	uint
20	Контрольное слово	CW_	б/р	–	uint
20	Контрольное слово	CW_	б/р	–	uint
20	Контрольное слово	CW_	б/р	–	uint
20	Контрольное слово	CW_	б/р	–	uint
20	Контрольное слово	CW_	б/р	–	uint
20	Контрольное слово	CW_	б/р	–	uint
20	Контрольное слово	CW_	б/р	–	uint
20	Контрольное слово	CW_	б/р	–	uint
20	Контрольное слово	CW_	б/р	–	uint
20	Контрольное слово	CW_	б/р	–	uint

см. далее

Продолжение таблицы 3

Номер слова	Наименование информации	Усл. об.	Размерн.	Пределы изменения	Примеч.
20	Контрольное слово	CW_	б/р	–	uint
20	Контрольное слово	CW_	б/р	–	uint
20	Контрольное слово	CW_	б/р	–	uint

Пример оформления ссылки на таблицу: см. таблицу 3.

6. ОФОРМЛЕНИЕ ЛИСТИНГОВ ИСХОДНОГО КОДА

В данном разделе приводится пример оформления листингов исходного кода.

```
//-----
/// \file sparse.h
/// \brief Работа с разреженными матрицами
/// \details Перевод матрицы из плотного представления в COO (Coordinate
list), CSR (Compressed Sparse Row Yale
format), CSC (Compressed Sparse Column Yale format) вид.
/// \date 27.05.22 - создан
/// \author Соболев А.А.
/// \addtogroup spml
/// \{
///

#ifndef SPML_SPARSE_H
#define SPML_SPARSE_H

// System includes:
#include <limits>
#include <armadillo>
#include <algorithm>

// SPML includes:
#include <compare.h>

namespace SPML /// Специальная библиотека программных модулей (СБ ПМ)
{
namespace Sparse /// Разреженные матрицы
{
//-----

/// \brief Структура хранения матрицы в COO формате (Coordinate list)
/// \details Матрица A[n,m] (n – число строк, m – число столбцов)
///
struct CMATRIXCOO
{
std::vector<double> coo_val; //;< Вектор ненулевых элементов матрицы
A[n,m] (n – число строк, m – число столбцов), размер nnz
std::vector<int> coo_row; //;< Индексы строк ненулевых элементов
std::vector<int> coo_col; //;< Индексы столбцов ненулевых элементов
};

/// \brief Структура хранения матрицы в CSR формате (Compressed Sparse Row
Yale format)
/// \details Матрица A[n,m] (n – число строк, m – число столбцов)
///
struct CMATRIXCSR
{
```

14
01234.56789-01 ПРИМЕР 03

```
std::vector<double> csr_val; ///< Вектор ненулевых элементов матрицы
    A[n,m] (n – число строк, m – число столбцов), размер nnz
std::vector<int> csr_kk; ///< Вектор индексов колонок ненулевых элементов,
    размер равен количеству ненулевых элементов nnz
std::vector<int> csr_first; ///< Вектор начальных смещений в векторе CSR,
    размер n+1
};

///
/// \brief Структура хранения матрицы в CSC формате (Compressed Sparse
Column Yale format)
/// \details Матрица A[n,m] (n – число строк, m – число столбцов)
///
struct CMatrixCSC
{
    std::vector<double> csc_val; ///< Вектор ненулевых элементов матрицы
        A[n,m] (n – число строк, m – число столбцов), размер nnz
    std::vector<int> csc_kk; ///< Вектор индексов колонок ненулевых элементов,
        размер равен количеству ненулевых элементов nnz
    std::vector<int> csc_first; ///< вектор начальных смещений в векторе CSC,
        размер m+1
};

//-----
///
/// \brief Преобразование плотной матрицы в COO формат (Coordinated list)
/// \param[in] A – исходная матрица, размер [n,m] (n – число строк, m –
    число столбцов)
/// \param[out] coo_val – вектор ненулевых элементов матрицы A, размер nnz
/// \param[out] coo_row – индексы строк ненулевых элементов, размер nnz
/// \param[out] coo_col – индексы столбцов ненулевых элементов, размер nnz
///
void MatrixDenseToCOO( const arma::mat &A, std::vector<double> &coo_val,
    std::vector<int> &coo_row,
    std::vector<int> &coo_col );

///
/// \brief Преобразование плотной матрицы в COO формат (Coordinated list)
/// \param[in] A – исходная матрица, размер [n,m] (n – число строк, m –
    число столбцов)
/// \param[out] COO – структура хранения матрицы в COO формате (Coordinate
    list)
///
void MatrixDenseToCOO( const arma::mat &A, CMatrixCOO &COO );

///
/// \brief Преобразование матрицы из COO формата (Coordinated list) в
    плотную матрицу
/// \param[in] coo_val – вектор ненулевых элементов матрицы A, размер
        равен количеству ненулевых элементов nnz
/// \param[in] coo_row – индексы строк ненулевых элементов, размер nnz
/// \param[in] coo_col – индексы столбцов ненулевых элементов, размер nnz
```

```
/// \param[out] A - плотная матрица, размер [n,m] (n - число строк, m -  
///   число столбцов)  
///  
void MatrixCOOtoDense( const std::vector<double> &coo_val, const  
    std::vector<int> &coo_row,  
const std::vector<int> &coo_col, arma::mat &A );  
  
///  
/// \brief Преобразование матрицы из COO формата (Coordinated list) в  
///   плотную матрицу  
/// \param[in] COO - структура хранения матрицы в COO формате (Coordinate  
///   list)  
/// \param[out] A - плотная матрица, размер [n,m] (n - число строк, m -  
///   число столбцов)  
///  
void MatrixCOOtoDense( const CMatrixCOO &COO, arma::mat &A );  
  
-----  
///  
/// \brief Преобразование плотной матрицы в CSR формат (Compressed Sparse  
///   Row Yale format)  
/// \details Данный способ хранения эффективен, если кол-во ненулевых  
///   элементов  $NNZ < (m * (n-1) - 1) / 2$   
/// \param[in] A - исходная матрица, размер [n,m] (n - число строк, m -  
///   число столбцов)  
/// \param[out] csr_val - вектор ненулевых элементов матрицы A, размер  
///   равен количеству ненулевых элементов nnz  
/// \param[out] csr_kk - вектор индексов колонок ненулевых элементов,  
///   размер равен количеству ненулевых элементов nnz  
/// \param[out] csr_first - вектор начальных смещений в векторе CSR,  
///   размер n+1  
///  
void MatrixDenseToCSR( const arma::mat &A, std::vector<double> &csr_val,  
    std::vector<int> &csr_kk,  
std::vector<int> &csr_first );  
  
///  
/// \brief Преобразование плотной матрицы в CSR формат (Compressed Sparse  
///   Row Yale format)  
/// \param[in] A - исходная матрица, размер [n,m] (n - число строк, m -  
///   число столбцов)  
/// \param[out] CSR - структура хранения матрицы в CSR формате (Compressed  
///   Sparse Row Yale format)  
///  
void MatrixDenseToCSR( const arma::mat &A, CMatrixCSR &CSR );  
  
///  
/// \brief Преобразование матрицы из CSR формата (Compressed Sparse Row  
///   Yale format) в плотную матрицу  
/// \param[in] csr_val - вектор ненулевых элементов матрицы A, размер  
///   равен количеству ненулевых элементов nnz  
/// \param[in] csr_kk - вектор индексов колонок ненулевых элементов,  
///   размер равен количеству ненулевых элементов nnz
```

```
/// \param[in] csr_first - вектор начальных смещений в векторе CSR, размер n+1
/// \param[out] A - плотная матрица, размер [n,m] (n - число строк, m - число столбцов)
///
void MatrixCSRtoDense( const std::vector<double> &csr_val, const
    std::vector<int> &csr_kk,
const std::vector<int> &csr_first, arma::mat &A );

///
/// \brief Преобразование матрицы из CSR формата (Compressed Sparse Row Yale format) в плотную матрицу
/// \param[in] CSR - структура хранения матрицы в CSR формате (Compressed Sparse Row Yale format)
/// \param[out] A - плотная матрица, размер [n,m] (n - число строк, m - число столбцов)
///
void MatrixCSRtoDense( const CMatrixCSR &CSR, arma::mat &A );

-----
///

/// \brief Преобразование плотной матрицы в CSC формат (Compressed Sparse Column Yale format)
/// \details Данный способ хранения эффективен, если кол-во ненулевых элементов NNZ<(m*(n-1)-1)/2
/// \param[in] A - исходная матрица, размер [n,m] (n - число строк, m - число столбцов)
/// \param[out] csc_val - вектор ненулевых элементов матрицы A, размер равен количеству ненулевых элементов nnz
/// \param[out] csc_kk - вектор индексов строк ненулевых элементов, размер равен количеству ненулевых элементов nnz
/// \param[out] csc_first - вектор начальных смещений в векторе CSR, размер m+1
///
void MatrixDenseToCSC( const arma::mat &A, std::vector<double> &csc_val,
    std::vector<int> &csc_kk,
std::vector<int> &csc_first );

///

/// \brief Преобразование плотной матрицы в CSC формат (Compressed Sparse Column Yale format)
/// \param[in] A - исходная матрица, размер [n,m] (n - число строк, m - число столбцов)
/// \param[out] CSC - структура хранения матрицы в CSC формате (Compressed Sparse Column Yale format)
///
void MatrixDenseToCSC( const arma::mat &A, CMatrixCSC &CSC );

///

/// \brief Преобразование матрицы из CSC формата (Compressed Sparse Column Yale format) в плотную матрицу
/// \param[in] csc_val - вектор ненулевых элементов матрицы A, размер равен количеству ненулевых элементов nnz
```

```
/// \param[in] csc_kk – вектор индексов строк ненулевых элементов, размер
/// равен количеству ненулевых элементов nnz
/// \param[in] csc_first – вектор начальных смещений в векторе CSR, размер
/// m+1
/// \param[out] A – плотная матрица, размер [n,m] (n – число строк, m –
/// число столбцов)
///
void MatrixCSCtoDense( const std::vector<double> &csc_val, const
    std::vector<int> &csc_kk,
const std::vector<int> &csc_first, arma::mat &A );

///
/// \brief Преобразование матрицы из CSC формата (Compressed Sparse Column
/// Yale format) в плотную матрицу
/// \param[in] CSC – структура хранения матрицы в CSC формате (Compressed
/// Sparse Column Yale format)
/// \param[out] A – плотная матрица, размер [n,m] (n – число строк, m –
/// число столбцов)
///
void MatrixCSCtoDense( const CMatrixCSC &CSC, arma::mat &A );

-----
///
/// \brief Преобразование матрицы в COO формате (Coordinate list) в CSR
/// формат (Compressed Sparse Row Yale format)
/// \param[in] coo_val – вектор ненулевых элементов матрицы A, размер
/// равен количеству ненулевых элементов nnz
/// \param[in] coo_row – индексы строк ненулевых элементов, размер nnz
/// \param[in] coo_col – индексы столбцов ненулевых элементов, размер nnz
/// \param[in] csr_val – вектор ненулевых элементов матрицы A, размер
/// равен количеству ненулевых элементов nnz
/// \param[in] csr_kk – вектор индексов колонок ненулевых элементов,
/// размер равен количеству ненулевых элементов nnz
/// \param[in] csr_first – вектор начальных смещений в векторе CSR, размер
/// n+1
/// \param[in] sorted – признак отсортированности COO входа по строкам
/// (даёт ускорение в ~2 раза)
///
void MatrixCOOtocsR( const std::vector<double> &coo_val, const
    std::vector<int> &coo_row,
const std::vector<int> &coo_col, std::vector<double> &csr_val,
    std::vector<int> &csr_kk,
std::vector<int> &csr_first, bool sorted = false );

///
/// \brief Преобразование матрицы в COO формате (Coordinate list) в CSR
/// формат (Compressed Sparse Row Yale format)
/// \param[in] COO – матрица в COO формате (Coordinate list)
/// \param[out] CSR – матрица в CSR формате (Compressed Sparse Row Yale
/// format)
/// \param[in] sorted – признак отсортированности COO входа по строкам
/// (даёт ускорение в ~2 раза)
///
```

```
void MatrixCOOtoCSR( const CMatrixCOO &COO, CMatrixCSR &CSR, bool sorted =
    false );

////
/// \brief Преобразование матрицы в COO формате (Coordinate list) в CSC
/// формат (Compressed Sparse Column Yale format)
/// \param[in] coo_val – вектор ненулевых элементов матрицы A, размер
/// равен количеству ненулевых элементов nnz
/// \param[in] coo_row – индексы строк ненулевых элементов, размер nnz
/// \param[in] coo_col – индексы столбцов ненулевых элементов, размер nnz
/// \param[in] csc_val – вектор ненулевых элементов матрицы A, размер
/// равен количеству ненулевых элементов nnz
/// \param[in] csc_kk – вектор индексов строк ненулевых элементов, размер
/// равен количеству ненулевых элементов nnz
/// \param[in] csc_first – вектор начальных смещений в векторе CSR, размер
/// m+1
/// \param[in] sorted – признак отсортированности COO входа по столбцам
/// (даёт ускорение в ~2 раза)
///
void MatrixCOOtoCSC( const std::vector<double> &coo_val, const
    std::vector<int> &coo_row,
const std::vector<int> &coo_col, std::vector<double> &csc_val,
    std::vector<int> &csc_kk,
std::vector<int> &csc_first, bool sorted = false );

////
/// \brief Преобразование матрицы в COO формате (Coordinate list) в CSC
/// формат (Compressed Sparse Column Yale format)
/// \param[in] COO – матрица в COO формате (Coordinate list)
/// \param[out] CSC – матрица в CSR формате (Compressed Sparse Column Yale
/// format)
/// \param[in] sorted – признак отсортированности COO входа по столбцам
/// (даёт ускорение в ~2 раза)
///
void MatrixCOOtoCSC( const CMatrixCOO &COO, CMatrixCSC &CSC, bool sorted =
    false );

//-----
////
/// \brief Ключ элемента Aij матрицы A в COO формате (Coordinate list)
/// \attention Оператор < перегружен для случая построчного хранения
///
struct CKeyCOO
{
public:
////
/// \brief i – индекс строки
////
int i() const { return i_; }

////
/// \brief j – индекс столбца
////
```

```
int j() const { return j_; }

///
/// \brief Конструктор по умолчанию
///
CKeyCOO() : i_( 0 ), j_( 0 ) {}

///
/// \brief Параметрический конструктор
/// \param[in] i - индекс строки
/// \param[in] j - индекс столбца
///
CKeyCOO( int i, int j ) : i_( i ), j_( j ) {}

bool operator <( CKeyCOO const& other ) const
{
    if( ( this->i_ < other.i_ ) || ( ( this->i_ == other.i_ ) && ( this->j_ <
        other.j_ ) ) ) {
        return true;
    }
    return false;
}

private:
int i_; ///< Индекс строки
int j_; ///< Индекс столбца
};

} // end namespace Sparse
} // end namespace SPML
#endif // SPML_SPARSE_H
/// \}
```

ПЕРЕЧЕНЬ ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19.106-78. ЕСПД. Требования к программным документам, выполненным печатным способом. — М. : Стандартинформ, 2005.

ПЕРЕЧЕНЬ ТЕРМИНОВ

КЦ – спичка.

Цак – колокольчик в нос.

Пепелац – межзвездный корабль.

Гравицаппа – деталь для пепелаца, позволяющая совершать межзвездные перелеты.

Кю – допустимое в обществе ругательство.

Ку – все остальные слова.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

ГОСТ – государственный стандарт.

ЕСПД – единая система программной документации.

ПРИЛОЖЕНИЕ 1
Обязательное

ПРИМЕР ПРИЛОЖЕНИЯ С НОМЕРОМ И БЕЗ РАЗДЕЛОВ

Текст нумерованного приложения. Формулы, рисунки, таблицы нумеруются заново в каждом приложении. Если приложение в документе одно, оно не нумеруется и это учитывает шаблон.

Пример формул в приложении:

$$x = y + a \quad (1)$$

$$z = a + x \quad (2)$$

где: x – коэффициент 1;

z – коэффициент 2;

y – параметр 1;

a – параметр 2.

Пример ссылки на формулу приложения без номера: см. формулу ??.

Пример рисунков в приложении:



Рис. 1. Тестовое изображение «Лена»



Рис. 2. Тестовое изображение «Лена»

Пример ссылки на рисунки приложения без номера: см. рис. 1.

Пример таблицы в приложении:

Таблица 1 – Пример таблицы

Номер слова	Наименование информации	Усл. об.	Размерн.	Пределы изменения	Примеч.
1	Контрольное слово	CW_	б/р	–	uint
2	Контрольное слово	CW_	б/р	–	uint
3	Контрольное слово	CW_	б/р	–	uint
4	Контрольное слово	CW_	б/р	–	uint

Пример оформления ссылки на таблицу приложения без номера: см. таблицу 1.

ПРИЛОЖЕНИЕ 2
Обязательное

ПРИМЕР ПРИЛОЖЕНИЯ С НОМЕРОМ И СВОИМИ РАЗДЕЛАМИ

Текст нумерованного приложения. Формулы, рисунки, таблицы нумеруются заново в каждом приложении. Если приложение в документе одно, оно не нумеруется и это учитывает шаблон.

1. ПЕРВЫЙ РАЗДЕЛ ПРИЛОЖЕНИЯ

Пример начала раздела приложения.

1.1. Пример формул в приложении

Пример формул в приложении:

$$x = y + a \quad (1)$$

$$z = a + x \quad (2)$$

где: x – коэффициент 1;

z – коэффициент 2;

y – параметр 1;

a – параметр 2.

Пример ссылки на формулу приложения без номера: см. формулу 1.

1.2. Пример рисунков в приложении

Пример рисунков в приложении:



Рис. 1. Тестовое изображение «Лена»



Рис. 2. Тестовое изображение «Лена»

Пример ссылки на рисунки приложения без номера: см. рис. 1.

1.3. Пример таблицы в приложении

Пример таблицы в приложении:

Таблица 1 – Пример таблицы

Номер слова	Наименование информации	Усл. об.	Размерн.	Пределы изменения	Примеч.
1	Контрольное слово	CW_	б/р	–	uint
2	Контрольное слово	CW_	б/р	–	uint
3	Контрольное слово	CW_	б/р	–	uint
4	Контрольное слово	CW_	б/р	–	uint

Пример оформления ссылки на таблицу приложения без номера: см. таблицу 1.

2. ВТОРОЙ РАЗДЕЛ ПРИЛОЖЕНИЯ

2.1. Подраздел 1

Текст
текст текст текст текст текст текст текст текст текст текст текст текст текст.

2.2. Подраздел 2

Текст
текст текст текст текст текст текст текст текст текст текст текст текст.

2.2.1. Пункт 1

Текст
текст текст текст текст текст текст текст текст текст текст текст текст.

2.2.2. Пункт 2

Текст
текст текст текст текст текст текст текст текст текст текст текст.

2.2.2.1. Подпункт 1

Текст
текст текст текст текст текст текст текст текст текст текст текст.

2.2.2.2. Подпункт 2

Текст
текст текст текст текст текст текст текст текст текст текст текст.

Лист регистрации изменений