

# **BimML 1.0 (BimDump 3)**

**Универсальный формат данных для представления  
информационных моделей объектов строительства**

## СОДЕРЖАНИЕ

<b>1. Общее описание формата</b>	<b>9</b>
1.1 Назначение формата	9
1.2 Расширения файлов	9
1.3 Форматирование файлов	10
<b>2. Взаимосвязанные сведения и материалы</b>	<b>11</b>
2.1 Виды взаимосвязи	11
2.2 Глобальные и локальные идентификаторы	12
2.3 Ключ-идентификаторы	12
2.4 Компонентная взаимосвязь	13
2.5 Атомарная взаимосвязь	14
2.6 Пространственная взаимосвязь	14
<b>3. Объектно-ориентированное представление данных</b>	<b>15</b>
3.1 Контейнеры	15
3.2 Структура данных	15
3.3 Праймы	17
3.4 Элементы	19
3.5 Примитивы	19
<b>4. Первичные объекты (праймы)</b>	<b>20</b>
4.1 Типы и виды прикладных объектов	20
4.2 Основные типы моделей	21
4.3 Системные объекты	22
4.4 Системные таблицы	23
<b>5. Прикладной программный интерфейс (API)</b>	<b>26</b>
<b>6. Модели с геометрическим содержанием</b>	<b>27</b>
6.1 Состав моделей	27
6.2 Элементы моделей	28
6.3 Геометрические примитивы	30
6.4 Атрибутивные примитивы	31
6.5 Скрипты и функции	33
6.6 Метрики и индексы	34
<b>7. Пространственные структурные модели</b>	<b>37</b>
7.1 Структурные пространственные модели (Me3D)	37
7.2 Главная модель (Mm3D)	39
7.3 Плоские трехмерные модели (Pm3D, Pz3D)	39
7.4 Наборы элементов моделей (Ms3D)	39
7.5 Геометрические примитивы 3D моделей	40
7.6 Внешняя геометрия gITF	45
<b>8. Пространственные параметрические модели</b>	<b>46</b>
8.1 Общее описание параметрических моделей	46
8.2 Пространственные параметрические модели (Mp3D)	46
8.3 Расчетные и агентские модели	46
8.4 Параметры моделей	47
8.5 Динамические таблицы	48
<b>9. Пространственные расчетные модели</b>	<b>51</b>
9.1 Общее описание расчетных моделей	51
9.2 Состав моделей	52
9.3 Типы и параметры расчетных элементов	62
9.4 Опоры и краевые условия	67

9.5	Нагрузки и воздействия .....	68
9.6	Начальные условия.....	72
9.7	Эпюры и диаграммы .....	73
9.8	Физические величины .....	74
9.9	Значащие цифры.....	78
9.10	Аналитические модели (Fa3D).....	78
9.11	Конечно-элементные модели (Fe3D) .....	78
9.12	Расчетные задания (TaMS) .....	79
9.13	Результаты расчетов (ReMS, RaMS).....	79
9.14	Комплексные модели (Se3D, Sa3D) .....	79
9.15	Валидация и верификация моделей .....	79
9.16	Референсные модели (Fr3D) .....	79
<b>10.</b>	<b>Пространственные агентские модели.....</b>	<b>79</b>
10.1	Общее описание агентских моделей.....	79
10.2	Анимационные агентские модели (Aa3D) .....	83
10.3	Расчетные агентские модели (As3D).....	84
<b>11.</b>	<b>Диагностические пространственные модели.....</b>	<b>84</b>
11.1	Общее описание диагностических моделей (De3D) .....	84
11.2	Датчики.....	84
11.3	Генераторы .....	87
11.4	Измерительные установки .....	88
11.5	Трандюсеры .....	91
11.6	Результаты измерений .....	91
11.7	События.....	92
<b>12.</b>	<b>Проекции, чертежи и абрисы .....</b>	<b>93</b>
12.1	Проекции (Pm2D).....	93
12.2	Чертежи (Pd2D) .....	94
12.3	Абрисы (Pa2D).....	95
12.4	Параметрические модели (Mp2D, Md2D, Ma2D).....	95
12.5	Геометрические примитивы 2D моделей.....	95
12.6	Внешняя геометрия SVG .....	97
<b>13.</b>	<b>Преобразования моделей.....</b>	<b>99</b>
13.1	Виды и Проекции .....	99
13.2	Камеры и снимки .....	99
13.3	Пространственные преобразования (Tm3D) .....	101
13.4	Проекционные преобразования (Tp3D, Tp2D).....	102
13.5	Чертежные преобразования (Td2D) .....	103
13.6	Аттрибутные фильтры (Fm3D).....	104
13.7	Замены .....	104
13.8	Пересечения и объединения.....	104
<b>14.</b>	<b>Текстовые сведения .....</b>	<b>104</b>
14.1	Инфодокумент (ITd).....	104
14.2	Инфонабор (ITs) .....	105
14.3	Шаблон (ITt) .....	105
14.4	Компоненты (ITc) .....	105
14.5	Элементы текстовых объектов .....	105
14.6	Информатизированный текст.....	106
14.7	Составной текст.....	106
14.8	Иллюстрации, схемы и чертежи .....	107
14.9	Свойства и данные текстовых объектов .....	107
14.10	Схемы инфодокументов .....	108
14.11	Стили отображения инфотекста .....	113
14.12	Макеты отображения инфотекста.....	115
14.13	Параметры интернализации инфотекста.....	116

14.14	Схемы данных .....	117
14.15	Внешнее представление HTML.....	118
14.16	Внешнее представление SVG.....	118
<b>15.</b>	<b>Шкалы и палитры.....</b>	<b>118</b>
15.1	Шкалы.....	118
15.2	Палитры .....	119
<b>16.</b>	<b>Работы и проекты .....</b>	<b>119</b>
16.1	Комплекты моделей (JPK).....	119

## СОДЕРЖАНИЕ (подробное)

<b>1.</b>	<b>Общее описание формата</b>	<b>9</b>
1.1	Назначение формата	9
1.2	Расширения файлов	9
1.3	Форматирование файлов	10
1.3.1	Форматирование представления структуры данных	10
1.3.2	Форматирование представления числовой информации	10
1.3.3	Форматизаторы	11
<b>2.</b>	<b>Взаимосвязанные сведения и материалы</b>	<b>11</b>
2.1	Виды взаимосвязи	11
2.2	Глобальные и локальные идентификаторы	12
2.3	Ключ-идентификаторы	12
2.4	Компонентная взаимосвязь	13
2.5	Атомарная взаимосвязь	14
2.6	Пространственная взаимосвязь	14
<b>3.</b>	<b>Объектно-ориентированное представление данных</b>	<b>15</b>
3.1	Контейнеры	15
3.2	Структура данных	15
3.2.1	Синтаксис	15
3.2.2	Иерархическая структура объектов	16
3.2.3	Классы и типы объектов	16
3.2.4	Множественные и уникальные объекты	16
3.2.5	Идентификаторы	16
3.2.6	Системные и прикладные объекты	17
3.3	Праймы	17
3.3.1	Имена праймов	17
3.3.2	Системные элементы	18
3.3.3	Значки и изображения праймов	18
3.4	Элементы	19
3.4.1	Имена элементов	19
3.5	Примитивы	19
3.5.1	Имена примитивов	19
<b>4.</b>	<b>Первичные объекты (праймы)</b>	<b>20</b>
4.1	Типы и виды прикладных объектов	20
4.1.1	Классификация прикладных объектов	20
4.1.2	Модели реальных объектов	20
4.2	Основные типы моделей	21
4.2.1	Структурные пространственные модели	21
4.2.2	Расчетные конечно-элементные модели	21
4.2.3	Расчетные агентские модели	21
4.2.4	Диагностические модели	22
4.2.5	Пространственно-временные модели	22
4.2.6	Плоские модели	22
4.2.7	Текстовые сведения и документы	22
4.3	Системные объекты	22
4.4	Системные таблицы	23
4.4.1	Таблицы пространственных данных	23
4.4.2	Таблицы атрибутов отображения	23
4.4.3	Таблицы характеристик	24
4.4.4	Таблицы свойств	25
4.4.5	Словари	26
4.4.6	Классификаторы	26
<b>5.</b>	<b>Прикладной программный интерфейс (API)</b>	<b>26</b>
<b>6.</b>	<b>Модели с геометрическим содержанием</b>	<b>27</b>
6.1	Состав моделей	27
6.2	Элементы моделей	28
6.2.1	Идентификаторы элементов	28
6.2.2	Наименование элементов	29

6.3	Геометрические примитивы .....	30
6.3.1	Позиционирование геометрических примитивов.....	30
6.3.2	Наименования геометрических примитивов.....	30
6.4	Атрибутивные примитивы.....	31
6.4.1	Имена свойств.....	31
6.4.2	Явные и извлекаемые значения свойств .....	31
6.4.3	Наборы свойств.....	32
6.4.4	Имена и структура наборов свойств .....	32
6.4.5	Адресация свойств.....	33
6.5	Скрипты и функции .....	33
6.6	Метрики и индексы .....	34
6.6.1	Геометрические метрики .....	34
6.6.2	Атрибутивные метрики.....	34
6.6.3	Структурные метрики.....	35
6.6.4	Метрические индексы .....	35
<b>7.</b>	<b>Пространственные структурные модели .....</b>	<b>37</b>
7.1	Структурные пространственные модели (Me3D) .....	37
7.1.1	Уровни геометрической проработки - LOD(G) .....	37
7.1.2	Уровни атрибутивной проработки - LOD(P) .....	37
7.1.3	Системные элементы и таблицы модели .....	37
7.1.4	Состав элементов модели.....	37
7.1.5	Элементы с геометрическим представлением .....	38
7.1.6	Элементы без геометрического представления.....	39
7.2	Главная модель (Mm3D).....	39
7.3	Плоские трехмерные модели (Pm3D, Pz3D).....	39
7.4	Наборы элементов моделей (Ms3D) .....	39
7.5	Геометрические примитивы 3D моделей.....	40
7.5.1	3D - Трехмерная геометрия - тела .....	40
7.5.2	2D - Двумерная геометрия – поверхности .....	43
7.5.3	1D - Одномерная геометрия - линии и кривые .....	45
7.5.4	0D - Точки .....	45
7.5.5	Текст .....	45
7.5.6	Ссылочная геометрия.....	45
7.6	Внешняя геометрия gTF .....	45
<b>8.</b>	<b>Пространственные параметрические модели.....</b>	<b>46</b>
8.1	Общее описание параметрических моделей.....	46
8.2	Пространственные параметрические модели (Mp3D).....	46
8.3	Расчетные и агентские модели.....	46
8.4	Параметры моделей .....	47
8.5	Динамические таблицы.....	48
8.5.1	Динамические таблицы для дискретных параметров .....	49
8.5.2	Динамические таблицы для переменных параметров .....	49
8.5.3	Таблица существования .....	50
<b>9.</b>	<b>Пространственные расчетные модели.....</b>	<b>51</b>
9.1	Общее описание расчетных моделей .....	51
9.2	Состав моделей.....	52
9.2.1	Объекты элементов модели.....	52
9.2.2	Таблицы базы данных модели .....	53
9.2.3	Словари описания физических величин .....	62
9.3	Типы и параметры расчетных элементов .....	62
9.3.1	Локальная система координат элементов .....	62
9.3.2	Конечные элементы.....	63
9.3.3	Типы и параметры материалов .....	66
9.3.4	Типы и параметры сечений .....	66
9.4	Опоры и краевые условия .....	67
9.4.1	Опоры .....	67
9.4.2	Краевые условия.....	68
9.5	Нагрузки и воздействия .....	68
9.5.1	Механические нагрузки.....	68
9.5.2	Мультифизические воздействия .....	71
9.6	Начальные условия.....	72
9.7	Эпюры и диаграммы .....	73
9.7.1	Диаграммы .....	73
9.7.2	Эпюры.....	73

9.8	Физические величины .....	74
9.8.1	Словарь системы физических величин - QuSysDict .....	74
9.8.2	Словарь кратных величин - QuMultDict .....	75
9.8.3	Словарь типов величин QuTypeDict .....	75
9.8.4	Обозначений величин .....	76
9.8.5	Словарь обозначений величин QuMarkupDict .....	77
9.9	Значение цифры .....	78
9.10	Аналитические модели (Fa3D) .....	78
9.11	Конечно-элементные модели (Fe3D) .....	78
9.12	Расчетные задания (TaMS) .....	79
9.13	Результаты расчетов (ReMS, RaMS) .....	79
9.14	Комплексные модели (Se3D, Sa3D) .....	79
9.15	Валидация и верификация моделей .....	79
9.15.1	Референсы и метрики .....	79
9.16	Референсные модели (Fr3D) .....	79
<b>10.</b>	<b>Пространственные агентские модели .....</b>	<b>79</b>
10.1	Общее описание агентских моделей .....	79
10.1.1	Описание моделей .....	79
10.1.2	Элементы агентских моделей .....	80
10.1.3	Примитивы агентских элементов модели .....	80
10.1.4	Наборы данных агентов .....	81
10.1.5	Треки .....	83
10.2	Анимационные агентские модели (Aa3D) .....	83
10.3	Расчетные агентские модели (As3D) .....	84
<b>11.</b>	<b>Диагностические пространственные модели .....</b>	<b>84</b>
11.1	Общее описание диагностических моделей (De3D) .....	84
11.2	Датчики .....	84
11.2.1	Типы (марки) датчиков .....	85
11.2.2	Датчики .....	86
11.3	Генераторы .....	87
11.3.1	Типы (марки) генераторов .....	87
11.3.2	Генераторы .....	88
11.4	Измерительные установки .....	88
11.4.1	Типы (марки) измерительных установок .....	89
11.4.2	Измерительные установки .....	91
11.5	Трандьюсеры .....	91
11.6	Результаты измерений .....	91
11.7	События .....	92
<b>12.</b>	<b>Проекция, чертежи и абрисы .....</b>	<b>93</b>
12.1	Проекция (Pm2D) .....	93
12.2	Чертежи (Pd2D) .....	94
12.2.1	Состав чертежей .....	94
12.2.2	Элементы оформления .....	94
12.2.3	Области чертежа .....	95
12.3	Абрисы (Pa2D) .....	95
12.4	Параметрические модели (Mp2D, Md2D, Ma2D) .....	95
12.5	Геометрические примитивы 2D моделей .....	95
12.6	Внешняя геометрия SVG .....	97
<b>13.</b>	<b>Преобразования моделей .....</b>	<b>99</b>
13.1	Виды и Проекция .....	99
13.2	Камеры и снимки .....	99
13.3	Пространственные преобразования (Tm3D) .....	101
13.4	Проекционные преобразования (Tp3D, Tp2D) .....	102
13.5	Чертежные преобразования (Td2D) .....	103
13.6	Аттрибутные фильтры (Fm3D) .....	104
13.7	Замены .....	104
13.8	Пересечения и объединения .....	104
<b>14.</b>	<b>Текстовые сведения .....</b>	<b>104</b>

14.1	Инфодокумент (ITd).....	104
14.2	Инфонабор (ITs) .....	105
14.3	Шаблон (ITt) .....	105
14.4	Компоненты (ITc) .....	105
14.5	Элементы текстовых объектов .....	105
14.6	Информатизированный текст.....	106
14.7	Составной текст.....	106
14.8	Иллюстрации, схемы и чертежи .....	107
14.9	Свойства и данные текстовых объектов .....	107
14.10	Схемы инфодокументов .....	108
14.10.1	Назначение схем.....	108
14.10.2	Типы элементов .....	108
14.10.3	Классы элементов.....	110
14.10.4	Объединение инфодокументов .....	111
14.10.5	Задание схем .....	112
14.11	Стили отображения инфотекста .....	113
14.12	Макеты отображения инфотекста.....	115
14.13	Параметры интернализации инфотекста.....	116
14.14	Схемы данных .....	117
14.15	Внешнее представление HTML.....	118
14.16	Внешнее представление SVG.....	118
<b>15.</b>	<b>Шкалы и палитры .....</b>	<b>118</b>
15.1	Шкалы.....	118
15.2	Палитры .....	119
<b>16.</b>	<b>Работы и проекты .....</b>	<b>119</b>
16.1	Комплекты моделей (JPK).....	119



# 1.Общее описание формата

## 1.1 Назначение формата

Формат **BimDump** является прикладным форматом обмена объектно-ориентированными данными об объектах строительства и застроенных земельных участках на основе формата JSON5.

**BimDump** - формат представления (сериализации) произвольных данных информационных моделей объектов строительства обозначениями типов и свойств объектов по различным стандартам и руководствам, например IFC 2, IFC 4, CityGML, gbXML и т.п. В формате **BimDump** используются основные геометрические представления объектов моделей, предусмотренные в форматах IFC и CityGML. Для эффективности отображения моделей и упрощения требований к программам просмотра, в **BimDump** не используются сложные геометрические представления, использующие какие-либо операции над объектами (пересечение, вырезание и т.п.)

Формат разработан для возможности экспорта из программ и программных комплексов для информационного моделирования строительных объектов максимально-возможного объема взаимосвязанной информации без каких-либо усечений и интерпретаций под системы классификации каких-либо отдельных форматов описания информационных моделей. Экспорт информации может осуществляться средствами прикладного программного интерфейса (API) программного комплекса с использованием классификации элементов модели, принятой в программном комплексе, с последующей обработкой и преобразованием в другие совместимые системы классификации элементов и их свойств с применением сторонних специализированных программ конверторов.

В формате **BimDump** предусмотрены структуры данных для реализации представления моделей с разными уровнями проработки (LOD) как геометрического представления, так и наборов свойств.

Название формата **BimDump** состоит из двух частей:

**Bim** (BIM, Building Information Modelling) - обозначает, что формат разработан для обмена информацией о преимущественно строительных объектах при использовании технологии информационного моделирования

**Dump** (выгрузка, свалка) - обозначает, что формат данных ориентирован на совместимость с внутренними структурами данных максимально широкого числа программных комплексов при экспорте данных с помощью плагинов, с последующей обработкой и преобразованием в системы классификации различных современных и возможно будущих промышленных форматов.

По аналогии с другими форматами описания моделей объектов, например таких как GML, CityGML, GeoSciML и т.п., формат **BimDump** можно назвать языком разметки (ML, Markup Language) и соответственно использовать название формата **BimML** (BIM Markup Language) - язык разметки для информационных моделей объектов строительства.

Для использования в отечественном документообороте формат **BimDump** может также обозначаться транслитерацией на русский язык - **БимДамп** или **БимМЛ**

Формат **BimWeb** - подмножество **BimDump**, ориентированной на использование в интернет-браузерах

## 1.2 Расширения файлов

- bimj3, bimml**           Расширение файлов BimDump с данными на языке JSON5
- bimjzip**           Расширение ZIP контейнеров BimDump.
- bimwzip**           Расширение ZIP контейнеров BimDump с внешней геометрией GLTF и SVG .
- bimuzip**           Расширение ZIP контейнеров BimDump с внешней геометрией U3D .
- bimml.pdf** (двойное расширение) - Расширение PDF контейнеров BimDump

## 1.3 Форматирование файлов

### 1.3.1 Форматирование представления структуры данных

Файлы БимДамп используются преимущественно для обмена данными между программными комплексами и вычислительными системами, но также могут использоваться для восприятия людьми при разработке и тестировании программного обеспечения.

Для удобства восприятия, в формате JSON5 предусмотрена возможность включения комментариев и различных символов форматирования, не влияющих на представление информации, но делающих отображение структуры информации более наглядным при просмотре файла в текстовых редакторах.

Обычно распространенные программные библиотеки автоматически добавляют символы форматизации для структурированного представления объектов и элементов массивов в стиле, индивидуальном для каждой такой библиотеки.

В насыщенных файлах данных JSON5 объем таких дополнительных не содержащих информацию символов может достигать до 15-25% от минимально-необходимого и достаточного объема.

### 1.3.2 Форматирование представления числовой информации

Другим аспектом ненужного увеличения объема файлов является представление чисел с незначащими цифрами.

Например, для описания геометрии большинства объектов строительства как правило достаточно задание координат точек объекта с точностью 0.1 мм.

Запись такой координаты в метрах будет выполняться с 4 знаками после запятой, например 1.0000 м.

При записи в различных единицах длины

0.5001 м - 6 символов

500.1 мм - 5 символов

5001 0.1 мм - 4 символов

0.5001000000000001 м - 18..20 символов, без округления до значащей цифры, с учетом вычислительной погрешности перевода машинного двоичного представления в десятичный вид.

Округление значений при выводе в файл до значащих цифр позволяет сократить объем файлов в 3-4 раза, а использование оптимальных единиц для представления свойств объектов позволяет сократить размер еще в полтора раза.

При разработке программного обеспечения при создании файлов БимДамп настоятельно рекомендуется использовать округление числовых значений до значащих цифр и по возможности использовать оптимальные для задачи единицы величин.

### 1.3.3 Форматизаторы

При разработке программного обеспечения для работы с форматом БимДамп целесообразно выполнять создание файлов JSON или JSON5 с использованием наиболее удобной или доступной библиотеки для выбранного языка программирования, а затем приводить созданные файлы в компактный вид путем удаления избыточных символов форматирования, округления числовых значений до значащих цифр, преобразования к оптимальным единицам величин и выполнения других подобных операций, с помощью специализированных программ.

Для рассмотрения структур данных файла в текстовых редакторах, также целесообразно переводить компактное представление в структурированный форматизированный вид с использованием аналогичных специализированных программ.

Такие программы называются **форматизаторами** БимДамп и разрабатываются совместно с данным описанием формата языка разметки.

## 2. Взаимосвязанные сведения и материалы

### 2.1 Виды взаимосвязи

Термин "взаимосвязанные сведения, документы и материалы" употребляется в определении информационной модели, приведенной в Градостроительном кодексе.

Под взаимосвязанностью можно подразумевать наличие каких-либо отношений соотношений между элементами информационной модели, и описание этих отношений в составе этой модели.

Отношения взаимосвязи между сведениями в информационной модели могут быть достаточно сложными и многообразными и быть различного типа. На данном уровне развития проблематики технологии информационного моделирования взаимосвязи между материалами информационной модели можно подразделить на следующие типы по уровню детализации взаимосвязываемых материалов и сведений:

**Компонентная взаимосвязь** - взаимосвязи между отдельными материалами информационной модели, представленными отдельными файлами или электронными документами. Примерами такой взаимосвязи может быть отношение между расчетной моделью и отчетной документацией по расчету, выполненному по этой расчетной модели. Или отношение между чертежом и трехмерной моделью, образом которой является данный чертеж. При компонентной взаимосвязи может быть не установлена взаимосвязь между отдельными объектами и элементами в составе взаимосвязанных материалов.

**Атомарная взаимосвязь** - взаимосвязь между отдельными объектами различных материалов и электронных документов. Примером такой взаимосвязи является отношение графического изображения какого-либо объекта трехмерной модели здания или сооружения на чертеже и соответствующего объекта трехмерной модели.

**Пространственная взаимосвязь** - взаимосвязи между системами координат материалов информационной модели, представленными трехмерными моделями и

масштабными чертежами и схемами. По сведениям о взаимосвязи систем координат, возможно установление атомарной взаимосвязи между объектами и элементами геометрически взаимосвязанных материалов информационной модели, расположенных в одинаковых областях пространства модели.

## 2.2 Глобальные и локальные идентификаторы

Для обеспечения взаимосвязи всем взаимосвязываемым объектам присваиваются уникальные идентификаторы, неповторяющиеся для рассматриваемого набора объектов.

Объекты, связанные с другим объектом, содержат свойство или другой атрибут с перечнем идентификаторов объекта, с которыми данный объект связан какими-либо отношениями.

Для компонентной взаимосвязи как правило используются **глобальные идентификаторы** (GUID, Globally Unique Identifier) - номера, выбираемые случайным образом из большого диапазона возможных чисел. Размер диапазона глобальных идентификаторов зависит от количества взаимосвязываемых объектов и должен быть достаточно большим, чтобы вероятность выделения функцией случайных чисел двум объектам одинакового идентификатора была меньше разумного значения для рассматриваемой области применения.

В настоящее время в компьютерных сетях и системах, в различных форматах и языках разметки в качестве GUID используется в виде числа в диапазоне до  $2^{128}$  (около  $3.4 * 10^{38}$ ). Такой GUID используется в том числе в форматах IFC и CityGML и подобных, и считается что этот диапазон обеспечивает уникальность идентификации всех элементов всех объектов строительства при современной практике проектирования и строительства на многолетнюю перспективу.

**Локальный идентификатор** — это идентификатор, обеспечивающий уникальность идентификации объектов в пределах сравнительно небольшого контейнера, содержащего объекты. Как правило локальный идентификатор подбирается таким образом, чтобы было удобно осуществлять поиск объекта в контейнере. Простейшим локальным идентификатором является порядковый номер объекта в какой-либо таблице или массиве данных.

При использовании для идентификации объекта в контейнере локальных идентификаторов, глобальная идентификация таких объектов может выполняться с использованием структуры, состоящей из глобального идентификатора контейнера и локального идентификатора объекта в контейнере. Такой метод идентификации объектов в данном документе будет называться **контейнерной идентификацией**.

## 2.3 Ключ-идентификаторы

**Ключ-идентификатор** (LKID, Local Key IDentifire) – локальный идентификатор, соответствующий хеш-коду глобального идентификатора, представляемый в воспринимаемой человеком форме, с низкой вероятностью повторений в пределах информационного контейнера разумного объема.

Ключ-идентификаторы могут быть разных типов в зависимости от используемой системы счисления, количества используемых разрядов чисел и параметров хэш-функции.

Кодировка типа локальных идентификаторов – два символа AN

A-код системы представления числа

D – десятичные числа (10 цифр)

H- шестнадцатеричные числа (шестнадцать цифр)

Q - числа, закодированные цифрами и заглавными латинскими символами A-Z.  
Количество цифр - 37. 0123456789 ABCDEFGHEF IJKLMNOPQR STUVXYZ

Z– числа, закодированные цифрами и заглавными латинскими символами A-Z.  
Количество цифр – 37. первой цифрой всегда должна быть латинская буква A-Z.  
Требование, чтобы начальным символом должна быть буква такого числа в идентификаторах имен объектов в форматах XML, SVG, JSON5 и подобных,

N-количество символов в хеш-коде.

Например H4 – шестнадцатеричный хешкод из 4 символов, примеры: 1A4F, 010E, 1234 и т.п.

Если для объекта с ключ-идентификатором задан GUID, то ключ-идентификатор должен быть равен хэш-функции GUID

Для преобразования глобальных идентификаторов GUID в ключ-идентификаторы разных типов в БимДамп используется следующая хэш-функция:

$$\underline{LKID = (GUID \% D1) \% D2}$$

$$\underline{\text{Если } LKID < \text{МинЗначение то } LKID = \text{МинЗначение} + LKID * Km}$$

Параметры функции принимаются по таблице:

	<b>LKID</b>	<b>минЗначение</b>	<b>МаксЗначение</b>	<b>D1</b>	<b>D2</b>	<b>Km</b>
D4	0	9999	12345678	9999		
H4	0	65 535	1234567890	65535		
Q4	0	1 874 160	123456789012345	1874160		
Z4	506 530	1 874 160	123456789012345	1874160	2	

## 2.4 Компонентная взаимосвязь

В формате БимДамп компонентная взаимосвязь реализуется путем включения в состав заголовка первичных объектов идентификатора объекта-прародителя и перечня родительских объектов, полученных последовательностью преобразований из прародительского объекта, и других взаимосвязанных объектов

Также в заголовке может содержаться информация о последовательности преобразований, что позволяет создавать в случае изменения прародительского объекта заново новый объект, являющийся актуальной версией рассматриваемого объекта.

В качестве примера можно привести модели чертежей для раздела проектной документации.

Для чертежей, выполненных с применением технологии информационного моделирования с обеспечением компонентной взаимосвязи, прародителями как правило являются два объекта - пространственная модель объекта строительства, и набор текстовых сведений, описывающих этот объект.

Чертежи как правило выполняются с применением трансформаций объектов, показанных ниже. Для наглядности трансформируемые **объекты** выделены жирным шрифтом :

1. Создаются чертежные **виды** пространственной **модели** - изображения планов, сечений, узлов и т.п.
2. Чертежные **виды** моделей масштабируются и размещаются в модели **чертежа**
3. В модель чертежа размещаются блоки текста из набора текстовых сведений (**инфотекста**)
4. Модель **чертежа** преобразуется в **Документацию** в формате, требуемом для представления заинтересованному лицу (заказчику, экспертной организации и т.п.)

В этом примере у объекта **Документация** два прародителя:

1. Модель
2. Инфотекст.

Для создания Документации из объектов-прародителей используются следующие типы преобразований:

1. Создание чертежных **видов** из **модели**
2. Создание **чертежа** из чертежных **видов** модели и **инфотекста** (Но если модель чертежа использует ссылки на внешние объекты чертежных видов и текстовых блоков, то использование такого преобразования не будет требоваться. Вместо преобразования будет использоваться структура внешних ссылок в чертеже)
3. Создание документации из чертежа

Если у нескольких объектов одинаковые прародители или родители, то такие объекты также являются взаимосвязанными через родительские объекты.

## 2.5 Атомарная взаимосвязь

Для обеспечения атомарной взаимосвязи элементов в составе первичных объектов в формате БимДамп используется контейнерная идентификация с использованием следующих правил создания моделей:

1. При преобразованиях объектов локальные идентификаторы элементов производного объекта не изменяются и соответствуют локальным идентификаторам исходных элементов
2. Элементы компонентно-взаимосвязанных объектов, имеющие одинаковые локальные идентификаторы внутри объекта, относятся к одному и тому же элементу моделируемого объекта реального мира.
3. Компонентно-взаимосвязанные объекты указываются в списке взаимосвязанных объектов рассматриваемого объекта, или определяются через отношения родства между объектами
4. При создании объектов из одного или более исходных объектов, обеспечивается соответствие локальных идентификаторов элементов для первого (основного) исходного объекта. Информация о соответствии локальным идентификаторам других исходных объектов указывается в свойствах (атрибутах) элементов.

## 2.6 Пространственная взаимосвязь

Пространственная взаимосвязь между трехмерными моделями обеспечивается компонентной взаимосвязью между моделями и использованием общей системы координат для представления моделей.

Пространственная взаимосвязь между элементами чертежей и соответствующих им элементов трехмерных моделей обеспечивается компонентной взаимосвязью между моделями и заданием информации о проекциях моделей, расположенных на чертеже и соответствующих проекционных преобразованиях.

## 3. Объектно-ориентированное представление данных

### 3.1 Контейнеры

Данные БимДамп содержатся в контейнере, состоящем из одного или нескольких файлов.

Контейнером как правило является zip-архив.

Для небольших информационных моделей допускается использование представления из одного отдельного файла, не упакованного в контейнер

Описание основного структурированного набора данных информационной модели выполняется в файле в формате JSON5. Связанные данные, которые целесообразно использовать в других форматах для обеспечения эффективности их чтения и обработки, размещаются в отдельных файлах в подпапках контейнера.

Контейнер БимДамп - папка, zip-архив или pdf-файл, содержащий файлы с моделями. .bimj и сопутствующими файлами (текстурами и т.п.).

Файлы моделей должны лежать в корневой папке.

Сопутствующие файлы, в том числе файлы моделей, которые используются как ссылки из других моделей, могут лежать в подпапках контейнера.

### 3.2 Структура данных

#### 3.2.1 Синтаксис

Набор данных представляется как набор отдельных объектов, содержащих данные.

Для представления данных используется язык разметки JSON5. Соответственно для представления данных необходимо использовать принципы и синтаксические ограничения этого языка разметки.

Основные аспекты языка разметки JSON5 приведены ниже:

5. Все данные являются объектами
6. Объекты содержат наборы данных, которые могут быть:
  - 6.1. атрибуты, которым присвоено значение
  - 6.2. массивы значений
7. Значениями атрибутов и массивов могут быть числа, текстовые строки и другие объекты
8. Все атрибуты и объекты имеют уникальные имена в наборе данных объекта, в состав которого они входят

9. Имена объектов и атрибутов состоят из латинских букв, цифр и знаков "\_" и "\$"
10. Имена объектов и атрибутов не могут начинаться с цифры

Дополнительно к синтаксису JSON5 для объектов в формате БимДамп накладываются дополнительные условия для возможности обеспечения быстрого извлечения данных и их обработки прикладными программами информационного моделирования:

1. Имена объектов содержат поля с метаданными о идентификаторе объекта и его типе

### 3.2.2 Иерархическая структура объектов

Данные БимДамп являются набором объектов, которые называются "первичными объектами" (primary objects) или "Праймами"

Базовые объекты состоят из набора вложенных объектов, которые называются "элементами первичных объектов" или просто "элементами"

Объекты, входящие в состав элементов, описывают геометрическую форму элементов, наборы свойств элементов и какие либо другие параметры элементов из набора типов, реализованных в данной спецификации формата БимДамп. Такие объекты называются "примитивами".

Таким образом все объекты данных в формате БимДамп можно представить четырехуровневой иерархической структурой:

2. БимДамп - файл JSON5
3. **Праймы** - первичные объекты
4. **Элементы** - вложенные объекты праймов
5. **Примитивы** - вложенные объекты элементов
6. Другие вложенные объекты

### 3.2.3 Классы и типы объектов

Все объекты имеют заданные в данном формате типы.

Тип объекта состоит из класса объекта и набора подтипов (атрибутов класса), зависящих от класса

### 3.2.4 Множественные и уникальные объекты

Объекты в наборе могут быть множественными и уникальными.

Уникальными объектами для набора объектов называются объекты заданного типа, которые могут быть в наборе только в единственном экземпляре. Данные объекты однозначно определяются и идентифицируются по своему типу

Множественными объектами для набора объектов называются объекты заданного типа, которые могут быть в наборе в нескольких экземплярах.

### 3.2.5 Идентификаторы

Все множественные объекты имеют уникальные идентификаторы.

Идентификаторы объектов-праймов задаются в формате Z4 в соответствии с разделом документа "Ключ-идентификаторы". Если для прайма задан также глобальный идентификатор, то идентификатор Z4 должен быть хеш-функцией глобального идентификатора.



Идентификаторы других множественных объектов могут быть заданы в различных форматах. Как правило идентификаторы элементов праймов и примитивов задаются в виде порядкового номера создания такого объекта, начиная с 1.

Праймы должны иметь уникальную идентификацию при использовании в составе наборов данных с несколькими миллионами объектов для возможности обеспечения уникальных ссылок для проектирования и строительства отдельных строительных объектов.

Хорошей практикой является назначение праймам также глобальных идентификаторов GUID в используемом для форматов IFC и аналогичных форматах виде - в виде в виде числа в диапазоне до  $2^{128}$ , что обеспечивает уникальность идентификации всех элементов объектов строительства при современной практике строительства на многолетнюю перспективу.

Идентификация элементов может быть осуществлена посредством указания идентификатора элемента в прайме. Для символьной записи в качестве разделителя можно использовать точку:

идентификаторПрайма.идентификаторЭлемента

Аналогично идентификация примитивов может быть осуществлена посредством указания идентификатора примитива в элементе. Соответствующая символьная запись:

идентификаторПрайма.идентификаторЭлемента.идентификаторПримитива

Соответствующие записи идентификаторов в нотации JSON5 будут иметь вид массива

[ идентификаторПрайма, идентификаторЭлемента ,идентификаторПримитива ]

Для глобальной идентификации праймов и других объектов в контейнере БимДамп следует указывать также глобальный или локальный идентификатор контейнера

## 3.2.6 Системные и прикладные объекты

Объекты могут быть двух видов - системные и прикладные.

Системные объекты - объекты, описывающие какие-либо системные свойства и параметры объекта и классифицирующие объект в соответствии с данной спецификацией.

К системным объектам относятся различные ссылочные таблицы, словари и подобная информация.

Прикладные объекты - объекты описания элементов пространственных моделей, чертежей и наборов свойств каких либо моделируемых объектов реального мира.

Элементы прикладных объектов могут ссылаться на элементы системных объектов, а системные объекты и их элементы не имеют ссылок на какие либо прикладные объекты.

## 3.3 Праймы

### 3.3.1 Имена праймов

Имена праймов имеют вид:

Идентификатор\$Класс\_списокАтрибутовКласса

Идентификатор - идентификатор прайма. Для уникальных объектов может отсутствовать

\$ - префикс класса

Класс - имя класса объекта

\_ - разделитель между именем класса и списком атрибутов класса. Если у класса нет атрибутов, то разделитель не ставится

СписокАтрибутовКласса - список атрибутов класса, разделенных знаком подчеркивание "\_". Список может отсутствовать, если у класса нет подтипов.

Примеры:

*12345\$M3M*

*\$vPen\_0\_1\_0*

*\$Point*

### 3.3.2 Системные элементы

Все прикладные праймы содержат системные элементы класса s\$, в которых указывается тип прайма и его основные параметры.

**s\$type** - массив, содержащий информацию о типе прайма - классе и атрибутах класса. Данная информация дублирует аналогичную информацию, содержащуюся в имени прайма.

Пример: *s\$type : [vPen, 0, 1, 1]*

**s\$id** - массив, содержащий идентифицирующую информацию о прайме - идентификатор, глобальный идентификатор, название.

Пример: *s\$id : [ 1234, 1fsrt234hdt1gwvd3dyt67e44fgf, Моя Модель ]*

**s\$header** - заголовок - объект, содержащий описание основных параметров прайма. Содержание информации в заголовке зависит от типа прайма

### 3.3.3 Значки и изображения праймов

Значок модели (icon) – растровый файл изображения модели небольшого размера. По умолчанию 100x100 пикселей. Расширение файла по умолчанию - .jpg

В zip-контейнерах БимДамп значки лежат в папке ICO. Название файла значка соответствует названию файла с объектом, если файл БимДамп содержит только один прикладной прайм, или имени прайма из файла, с добавлением расширения. Также значки могут присваиваться контейнерам (папкам) с моделями. Имя объекта от имени файла отделяется знаками ".@" (точка, @, точка).

Изображение объекта (img) - растровый файл изображения модели среднего размера. По умолчанию 640x480 пикселей. Расширение файла по умолчанию - .jpg

В zip-контейнерах с моделями изображения лежат в папке IMG. Название файла соответствует названию файла с объектом, или имени объекта с добавлением имени файла, с добавлением расширения изображения. Имя объекта от имени файла отделяется знаками ".@" (точка, @, точка).

## 3.4 Элементы

### 3.4.1 Имена элементов

Имена элементов содержат информацию о типе элемента и его идентификаторе.

Имена имеют вид:

КодИдентификатор

Идентификатор - идентификатор элементов. Для уникальных системных элементов объектов может отсутствовать

Код - код класса элемента, состоящий из двух букв или символов "\_" и "\$".

Коды классов прикладных элементов начинаются со служебных символов "\_" и "\$"

Коды классов системных элементов начинаются с буквы.

Примеры:

*To*

*Ti*

*TiABC*

*\_0*

*\_\$12345*

## 3.5 Примитивы

### 3.5.1 Имена примитивов

Имена примитивов содержат информацию о типе примитива и его идентификаторе.

Имена имеют вид:

КодАтрибутИдентификатор

Идентификатор - идентификатор примитива. Для уникальных системных примитивов может отсутствовать

Код - код класса примитива, состоящий из двух букв.

Атрибут - атрибут класса - служебный символ "\_" или "\$". Не обязательное поле в имени примитива, может отсутствовать.

Коды классов системных элементов начинаются с буквы.

Примеры:

*Ga*

*Ga12*

*Ga\_13*

## 4. Первичные объекты (праймы)

### 4.1 Типы и виды прикладных объектов

#### 4.1.1 Классификация прикладных объектов

Прикладные объекты, описываемые в формате БимДамп, можно разделить на три основные группы

**Модели реальных объектов** - модели существующих или проектируемых объектов строительства на каких-то этапах их жизненного цикла. Такие модели как правило являются пространственными (3D) или пространственно-временными (4D) моделями

**Модели представлений реальных объектов** - различные чертежи и документы, описывающие объекты строительства средствами инженерной графики и технической документации.

**Наборы сведений и данных о реальных объектах** - различные наборы текстовых и численных данных, используемые для выполнения расчетов и моделирования рассматриваемых объектов строительства, и для составления моделей, чертежей и документации

**Документы** - электронные документы

**Преобразования моделей** - наборы параметров, однозначно определяющие процедуры создания новых объектов (моделей, наборов, документов) из одной модели

**Преобразования сведений** - наборы параметров, однозначно определяющие процедуры создания новых объектов (наборов сведений, документов) из одной модели ...

**Операции** - наборы параметров, однозначно определяющие процедуры создания новых объектов из двух и более объектов.

#### 4.1.2 Модели реальных объектов

Модели реальных объектов как правило являются **пространственными моделями**.

В пространственных моделях все точки модели задаются тремя координатами. Размеры элементов модели соответствуют размерам моделируемых физических объектов в натуральную величину.

Пространственные модели могут быть следующих видов по внутренней структуре представления данных о моделируемом объекте:

**Элементные модели** - модели, содержащие набор элементов, для которых нет данных о взаимном расположении и соединении. Свойства на границах и точках сопряжения элементов отсутствуют.

**Элементно-узловые модели** - модели, содержащие набор элементов с данными о взаимном расположении и соединении. В местах соединения элементов могут быть заданы свойства.

По виду применения в архитектурной и инженерной практике модели можно подразделить на следующие виды:

**Структурные (описательные) модели** - модели, описывающие состав какого-либо реального объекта и его свойства в какой-либо момент времени или промежуток времени

**Расчетные модели** - модели, описывающие какие-либо явления и физические процессы, относящиеся к рассматриваемому реальному объекту

**Компоновочные модели** - модели, описывающие логику взаимного расположения и определения размеров и свойств элементов в зависимости от технологии изготовления элементов реального объекта и технологии возведения объекта. (аналоги - использование семейств для создания моделей в Revit)

**Диагностические модели** - модели, описывающие какие-либо явления и физические процессы на основании выполненных измерений, относящихся к состояниям рассматриваемого реального объекта в отдельные моменты и периоды реального времени.

По количеству вариантов отображения содержащейся в модели информации модели подразделяются следующим образом:

**Явные (неизменные) модели** - модели, однозначно описывающие состав какого либо реального объекта и его свойства в какой либо момент времени или промежуток времени

**Параметрические модели** - модель, содержащая набор параметров и варианты отображения модели в зависимости от значений параметров.

## 4.2 Основные типы моделей

### 4.2.1 Структурные пространственные модели

**Me3D** - трехмерные элементные модели.

**Mm3D** - "главные" элементные модели проектов

**Ms3D** - наборы элементов трехмерных элементные модели.

### 4.2.2 Расчетные конечно-элементные модели

**Fa3D** - расчетные аналитические модели.

**Fe3D** - расчетные конечно-элементные модели.

**Sa3D** - комплексные аналитические модели, содержащие результаты моделирования

**Se3D** - комплексные конечно-элементные модели, содержащие результаты моделирования

**Fr3D** - референтные модели и контрольные задачи.

### 4.2.3 Расчетные агентские модели

**Aa3D** - Агентские анимационные модели.

**As3D** - агентские расчетные модели.

#### 4.2.4 Диагностические модели

**De3D** - диагностические элементные модели.

#### 4.2.5 Пространственно-временные модели

Пространственно-временные структурные модели являются частным случаем пространственных параметрических моделей. Для таких параметрических моделей задается переменный параметр "время".

Для расчетных пространственных моделей понятие времени включено в состав понятий модели. В случае, если в модели моделируются процессы, использующие время, такая модель может быть обозначена как пространственно-временная.

Для диагностических пространственных моделей понятие времени также включено в состав понятий модели и является переменным параметром, то есть такая модель всегда является пространственно-временной.

#### 4.2.6 Плоские модели

В плоских моделях все точки модели лежат на плоскости и задаются двумя координатами относительно системы координат этой плоскости. Размеры элементов плоской модели соответствуют размерам изображаемых физических объектов в натуральную величину.

Все точки модели задаются двумя координатами. Для моделей, лежащих в горизонтальных и вертикальных плоскостях, точки могут задаваться ссылками на таблицу точек с трехмерными координатами, при этом ненужная координата не используется.

**Pm2D** - проекции - плоские двумерные модели, полученные проецированием (проекционным преобразованием) из трехмерных моделей. Для отображения модели может важен порядок отображения элементов, поскольку фигуры модели могут накладываться друг на друга. Для задания порядка отображения используется индекс глубины.

**Pd2D** - чертежи - плоские масштабные двумерные модели, представляющие собой проекции, пояснения и другие изображения, размещенные в пространстве печатного образа электронного документа.

#### 4.2.7 Текстовые сведения и документы

**ITd** – инфодокументы - структурированные наборы текстовых сведений и стилей их отображения в документы.

**ITs** – инфонаборы - наборы текстовых сведений.

### 4.3 Системные объекты

Имена системных объектов состоят из префикса "\$" и имени класса системного объекта. Классы системных объектов приведены ниже.

**Lib3D** - библиотека трехмерных прототипных объектов. (вначале ищутся прототипы в модели, потом в библиотеке)

**Lib2D** - библиотека двумерных прототипных объектов. (вначале ищутся прототипы в модели, потом в библиотеке)

**ScaleLib** - библиотека шкал

**PaletteLib** - библиотека палитр

**ScriptLib** - библиотека функций на JavaScript

## 4.4 Системные таблицы

В файле могут задаваться общие таблицы, на которые могут быть реализованы ссылки из разных моделей и объектов в файле БимДамп.

Также в общих объектах и таблицах задаются свойства, одинаковые для всех моделей, такие как размерности величин, этажи и т.п.

Имена системных таблиц состоят из префикса "\$" и имени класса системного объекта. Классы системных таблиц приведены ниже.

### 4.4.1 Таблицы пространственных данных

**Point3D** - Массив координат точек -  $[[x,y,z]]$

**Vector3D** - Массив векторов -  $[[x,y,z]]$

**Point2D** - Массив координат точек -  $[[x,y]]$

**Vector2D** - Массив векторов -  $[[x,y]]$

### 4.4.2 Таблицы атрибутов отображения

**Pen** – массив перьев. **Перо** – массив из 6 чисел – R G B A типЛинии толщина

**Brush** - массив кистей. **Перо** – массив из 6 чисел – R G B A типЗаливки r g b a m

R G B A - цвет и прозрачность фона

ТипЗаливки - 0 - заливка цветом, не 0 - номер штриховки

r g b a - цвет и прозрачность штриховки

m - масштаб отображения штриховки. По умолчанию 1.0

Цвет и прозрачность R G B A могут быть заданы равными нулю, в этом случае заливка считается условной. То есть графически заливка не отображается, но учитывается при трассировке лучей при поиске и выделении объектов по щелчку (клике) мышкой и в других подобных случаях.

**Font** – массив шрифтов.–  $[[ \text{СемействоШрифта, размер, гарнитура, жирность, выделение}, \dots ]]$

гарнитура - 1-обычный 2-курсив

жирность - 1-обычный, 2-жирный

выделение - 1-без выделения, 2-подчеркивание, 3-зачеркивание

**FontFamily** - таблица семейств шрифтов [[ название, TrueType, Вектор ],... ]

название - название семейства шрифта

TrueType - имя шрифта TrueType

Вектор - имя файла векторного шрифта в папке Fonts контейнера

**Texture** – массив текстур. Текстура –

тип текстуры ( S-стандартная, путь относительно папки текстур программы, L-локальная, путь относительно расположения данного файла),

путь к файлу текстуры,

номер кисти для случая упрощенного отображения (Необязательный параметр)

**Hatch** - массив штриховок.

Штриховка - массив записей, описывающих название штриховки и группы (семейства) штрихов и линий. Соответствуют формату описания штриховок в файлах .PAT программы AutoCAD.

Первая запись - массив текстовых строк. В первом элементе массива указывается название штриховки, далее идут строки с описанием и комментариями.

Например ["steel", "сталь конструкционная"]

Далее идут записи с параметрами группами штрихов:

[ уголНаклона, x, y, dx, dy, длинаШтриха, интервал ]

УголНаклона - угол наклона линии семейства штрихов

X y - координаты начала отсчета штрихов семейства

Dx dy - шаг смещения штрихов в семействе

длинаШтриха - длина штрихов в семействе. Если равно нулю, то рисуется бесконечная линия

интервал - интервал между штрихами в семействе

#### 4.4.3 Таблицы характеристик

**Material** – массив материалов - [ [ N, Pn, B, Pr, V, Pr ], ... ]

N – имяМатериала

Pn – номер пера

B - номер кисти

Pr- таблица свойств материала - массив [[ названиеСвойства, значениеСвойства ], ... ]

**Level** – таблица уровней (этажей) - [[номерУровня, "название уровня", "ID", низУровня, верхУровня]...]



Пример: [[-1, "Подвал", "23213", -3.45,0.0],[1,"Этаж 1", "23214", 0.0,4.25],[2,"Чердак", "23215",4.25, 6.3]]

**Block** - таблица блоков - [[номерБлока, "название блока","ID"...]

**MUnits** – массив размерностей и соответствующего множителя для координат геометрического представления объектов-моделей - [Dim, Mult,Int]

Dim – обозначение размерности. По умолчанию «М» - метры

Mult – число, на которое надо умножать координаты, заданные в файле, чтобы получить координату в указанной размерности. По умолчанию 1.0

Int-признак использования для координат целых чисел. 1-целые числа, 0-действительные числа (по умолчанию)

Рекомендуется использовать значения ["мм", 0.1,1] или ["м", 0.0001,1]

**DUnits** – массив размерностей и соответствующего множителя для координат геометрического представления объектов - "бумажных" документов - [Dim, Mult,Int]

Dim – обозначение размерности. По умолчанию «М» - метры

Mult – число, на которое надо умножать координаты, заданные в файле, чтобы получить координату в указанной размерности. По умолчанию 1.0

Int-признак использования для координат целых чисел. 1-целые числа, 0-действительные числа (по умолчанию)

Рекомендуется использовать значения ["мм", 0.1,1] или ["м", 0.0001,1]

**LOD** - информация об уровнях проработки [[ Название, Описание, G, P, P, ...], ...]

G - номер геометрической проработки элементов модели

P - номера наборов свойств элементов моделей, соответствующих данному уровню проработки

#### 4.4.4 Таблицы свойств

**PropertySetDefault** - массив массивов значений по умолчанию для наборов свойств - [[ номерНабора, имяНабора, узелНабора], ...]. ИмяНабора и УзелНабора - значения специальных свойств @name и @node. Данные значения используются, если в наборе свойств не заданы соответствующие значения.

**PropertyName** – массив названий свойств элементов моделей и других объектов - [ИмяСвойства , ...]

**PropertyValue** – массив наборов значений свойств - [ [значение, значение, ....] ]

**NumPropValue** – массив количества значений свойств в соответствующей таблице значений - [количествоЗначений,...]

**PropertyTab\_P\_N** - "внешняя" таблица параметров объектов. P - номер набора параметров (аналогичный P1, P2, P3... массивов параметров внутри объектов), N - идентификатор таблицы - номера таблицы.

В первой строке массива таблицы - массив названий параметров

В следующих строках массива таблицы - массив с именем элемента и соответствующими ему значениями параметров

Пример:

```
"Params_1_1 : [      [ имяПараметра1, имяПараметра2 ],  
                  [ _Элемент1, значениеПараметра1, значениеПараметра2 ],  
                  [ _Элемент2, значениеПараметра1, значениеПараметра2 ]  
                ]
```

Пример использования таблиц свойств с помощью примитива Prm

```
$PropertyName : ["MyParam1", "SecondParameter"],
```

```
$PropertyValue : ["MyValue", 123],
```

```
_001 { Prm : [[1,2]]}
```

```
_002 { Prm : [[1,2,0]]}
```

Будет соответствовать

```
_001 { Prm : ["MyParam1", 123]}
```

```
_002 { Prm : ["MyParam1", 2]}
```

#### 4.4.5 Словари

**Lang** - таблица языков. Содержит массив с обозначением и названием языка. Например [[ru, Русский], [en, English]]

**Dict\_N\_L** - словарь. N - номер или идентификатор словаря. L - номер языка словаря по таблице языков. Словарь содержит массив текстовых строк, на которые могут даваться ссылки из элементов, примитивов объектов в Файле.

**Dict\_0\_0** - словарь по умолчанию

#### 4.4.6 Классификаторы

**Type\_N** - перечень использованных типов элементов для системы классификации N. N - обозначение системы классификации. Например Revit2020, IFC2x3, CityGML3 и т.п.

## 5. Прикладной программный интерфейс (API)

Функции, заданные в моделях, преобразования моделей и других объектов, могут использовать прикладной программный интерфейс, предоставляемый программными комплексами, в которых осуществляется обработка или визуализация моделей.

Прикладной программный интерфейс может предоставлять доступ только к геометрическим параметрам и свойствам обрабатываемых моделей.

Для обеспечения однозначности представления информации прикладной программный интерфейс не должен предоставлять доступ:

- к любым данным, не содержащимся в обрабатываемой модели
- к свойствам и характеристикам программ

- к свойствам и характеристикам компьютеров
- к показаниям календарного и системного времени

Примерный состав задач, которые могут реализовываться прикладным программным интерфейсом

- данные о форме и геометрических параметрах элементов модели
- расстояния между элементами моделями
- наличие пересечений между элементами модели
- соответствие параметра модели какому-либо утверждению нормативного документа, представленного в электронном виде (перспектива)
- подобные задачи

Предполагается, что с практикой использования языков разметки для информационных моделей выработается общепринятый набор функций API, который будет являться стандартом де факто или будет оформлен в какой-нибудь документ по стандартизации.

## 6. Модели с геометрическим содержанием

### 6.1 Состав моделей

Все модели содержат в заголовке следующие свойства

#### **s\$header**

1. Descript - Массив строк описания модели
2. Source - источник - массив полей
  - 2.1. Prototype - имя (идентификатор) исходной (прародительской модели) модели
  - 2.2. Parent - имя (идентификатор) родительской модели
  - 2.3. TransformType - 1- View, 2 - Projection или 0 – Original (по умолчанию)
  - 2.4. Transform - имя преобразования
  - 2.5. Proc- описание программы или процедуры, выполнившей преобразование
3. Link - массив идентификаторов взаимосвязанных моделей, использующих одинаковое обозначение объектов. Массив накапливается при трансформациях исходной (прародительской) модели. В массив также могут добавляться сведения о других родственных моделях. Модели задаются в порядке убывания информативности - то есть наиболее полные модели, которые как правило являются прародителями других моделей, задаются в начале списка. Объем информативности при отсутствии других данных можно сопоставить с количеством элементов в модели.
4. Metric – метрика модели
  - 4.1. GM- максимальная геометрическая метрика – максимальная из геометрических представлений
  - 4.2. PM- метрика свойств модели – суммарная для всех наборов
  - 4.3. GA – массив массивов данных геометрических метрики
    - 4.3.1. Номер геометрического представления
    - 4.3.2. Количество линий
    - 4.3.3. Количество треугольников (сплошных и прозрачных)
    - 4.3.4. Количество прозрачных треугольников

- 4.4. PA – массив данных метрики свойств
  - 4.4.1. Номер набора свойств
  - 4.4.2. Количество свойств
- 5. is4D - признак пространственно-временной модели (параметрической модели, для которой заданно время)
- 6. Box – [x1, y1, z1, x2, y2, z2] – координаты крайних точек, описывающие вмещающий параллелепипед для модели
- 7. ExtGeom - массив параметров внешней геометрии на 2D/3D модели геометрического вида в внешнем файле для каждого уровня геометрического представления.
  - 7.1. Type - формат представления геометрии модели, например "U3D", "GLTF", SVG
  - 7.2. ModelFile - ссылка на файл, относительно корня zip-контейнера
  - 7.3. Scene - индекс или идентификатор сцены или другого подобъекта модели

Внешние модели как правило располагаются в папке \ExtGeom zip-контейнера модели

В заголовке также дополнительно содержится свойства, характерные для отдельных типов моделей

**s\$header** – дополнение для плоских модели P2D

1. Plane - плоскость в трехмерном пространстве, в которой расположена двумерная модель - массив из 9 координат - точка начала координат плоскости, вектор направления оси X плоскости, вектор направления оси Y плоскости. Вектор нормали определяется векторным произведением направлений X и Y

В составе моделей также задаются следующие объекты

**s\$ModInfo** - информация о модели

Program - название и версия программы (Например Revit 18, SketchUp 8,...)

Actor - название и версия скрипта создания дампа (Например Sitis BIM dump Python Plugin for Revit 27.02.2019)

ElemNum - массив количеств элементов в модели - общее, с геометрическим представлением, без геометрического представления, прототипы, группы

**s\$ModelHistory** – таблица с историей обработки модели.

[[дата\_время, названиеОбработчика, названиеЗадачи, названиеШаблонаОбработки, комментарийШаблонаОбработки], ...].

**s\$ModelProperty** - таблица свойств модели

Массив пар свойство-значение, аналогично свойствам объектов модели

## 6.2 Элементы моделей

### 6.2.1 Идентификаторы элементов

Идентификатором объекта могут быть:

целое десятичное число

краткое буквенно-числовое обозначение с буквами английского алфавита, начинающееся с буквы, не содержащее пробелов. Рекомендуется использовать локальный идентификатор LKID типа Z.

В некоторых случаях для обеспечения атомарной взаимосвязи элементов модели, идентификатор выполняется составным, с добавлением к базовому обозначению идентификатора числового индекса с использованием разделителя "\_".

Такие обозначения могут возникать, например для обозначения нескольких элементов чертежа, соответствующих одному и тому же объекту трехмерной модели, например на виде сверху и виде сбоку.

Примеры идентификаторов:

1

2

2\_1

A23RT3Z

A23RT3Z\_1

## 6.2.2 Наименование элементов

Имя прикладных элементов модели начинается с служебного символа, далее следует идентификатор элемента и могут содержаться дополнительные поля.

«\_» (одно подчеркивание) - объекты модели, имеющие геометрическое представление (координационные объекты, состоящие из 3D-2D-1D-0D геометрических примитивов)

«\$» - объекты модели, не имеющие геометрического представления

«\_\$» - прототипы координационных объектов модели

"\$\$" - группы объектов

"\$\_" - наборы свойств и величин

«\_\_» (два подчеркивания) - "специальные" элементы модели, обладающие какими-либо специфичными свойствами в зависимости от вида модели. В именах таких объектах кроме идентификатора также могут присутствовать дополнительные поля, содержащие какую-либо информацию о типе элемента.

Например в агентских моделях это объекты - агенты, в диагностических моделях это измерители, регистраторы, события и другие подобные элементы, в чертежах - элементы оформления (рамки, подписи и т.п.).

Примеры:

\_1

\$2

\_\$A23RT3Z

\_\$A23RT3Z\_1

## 6.3 Геометрические примитивы

### 6.3.1 Позиционирование геометрических примитивов

Позиционирование геометрических примитивов может быть двух типов:

**Тип 1 - координатный** - Положение элемента задается тремя числами - координатами X Y Z, направления задаются тремя компонентами вектора.

**Тип 2 - точечный** - Положение элемента задается порядковым номером записи (начиная с 1) в массиве координат точек Points, направление задается номером записи (начиная с 1) в массиве векторов Vectors

Для простых моделей без параметров, используются системные таблицы файла БимДамп:

\$Point3D - Массив координат точек -  $[[x,y,z]]$

\$Vector3D - Массив векторов -  $[[x,y,z]]$

Для параметрических моделей массивы задаются в самой модели. Массивы точечного позиционирования для параметрических моделей описаны в разделе "Параметрические модели"

### 6.3.2 Наименования геометрических примитивов

Геометрические примитивы представляются объектами JSON с именами следующей структуры:

1. код класса примитива - Два первых символа – латинские буквы.
2. код позиционирования - Третий символ. Если «\_» – то позиционирование точечное, иначе позиционирование координатное. Код позиционирования является опциональным и может отсутствовать. По умолчанию – тип позиционирования координатный.
3. Идентификатор (индекс) примитива - десятичное число, уникальное для всех примитивов в объекте (для обеспечения возможности преобразования типа позиционирования без изменения индекса). Индекс примитива является опциональным и может отсутствовать.

Геометрические примитивы, использующие точечный тип позиционирования, вместо тройки координат указание на номер точки в массиве Points3D, вместо пары координат - на массив Points2D, имеют в названии постфикс "\_".

В моделях могут одновременно присутствовать объекты с разными типами позиционирования

Например:

(примитивы без индексов):

Ga - тело, заданное триангулированными гранями с прямым указанием координат вершин граней,

Ga\_ - тело, заданное триангулированными гранями с указанием для вершин граней номеров соответствующих точек в массиве Points,

(примитивы с индексами):

Ga11 - тело, заданное триангулированными гранями с прямым указанием координат вершин граней,

Ga\_12 - тело, заданное триангулированными гранями с указанием для вершин граней номеров соответствующих точек в массиве Point3D,

## 6.4 Атрибутивные примитивы

### 6.4.1 Имена свойств

Имя свойста - текстовое значение с следующими ограничениями.

- имя не может быть целым числом
- разделителем между словами в тексте может быть только одиночный пробел
- в начале и конце текста пробелы должны отсутствовать
- в имени не должно быть невидимых знаков (табуляция, перевод каретки и т.п.)
- пробел " " и подчеркивание "\_" в именах свойств считаются эквивалентными знаками
- при сравнении, сортировке и поиске пробелы заменяются знаком подчеркивания "\_"
- при сравнении, поиске и сортировке знаки точки "." заменяются знаком подчеркивания "\_"

Эквивалентные (нормализованные) имена свойств - текстовые строки, в которых выполнены следующие преобразования

- пробелы заменены на знаки подчеркивания "\_".
- точки заменены на знаки подчеркивания "\_".

В одном наборе свойств не может быть свойств с одинаковыми или эквивалентными именами.

Пример - использование в одном наборе свойств "Моё свойство" и "Моё\_свойство" не допускается, так как в четвертой позиции имени стоят эквивалентные знаки.

Пример - использование в одном наборе свойств "Моё\_свойство\_1" и "Моё\_свойство.1" не допускается, так как в предпоследней позиции имени стоят эквивалентные знаки..

### 6.4.2 Явные и извлекаемые значения свойств

Значения свойства могут быть заданы явно, или могут извлекаться из значений других свойств элемента с помощью регулярных выражений или функций на языке JavaScript.

При явном задании для свойства указывается:

- имя свойства
- значение свойства.

При извлекаемом задании для свойства указывается

- имя свойства

- имя свойства того же элемента, значение которого будет использоваться в качестве аргумента функции или регулярного выражение

- регулярное выражение, или ссылка на функцию на языке JavaScript

Извлекаемые свойства могут использоваться для уменьшения размера представления информации в файлах БимДамп без уменьшения объема информации, и также позволяют упрощать ввод данных пользователями программ при создании моделей.

### 6.4.3 Наборы свойств

**P1 .. P99** – нумерованные наборы свойств.

Набор свойств - массив записей (массивов) , состоящих из двух или трех элементов.

#### **Записи из двух элементов :**

Записи из двух элементов задают имя параметра и его значение - [ параметр, значение]

Если параметр является целым числом, то значит в этом поле указан порядковый номер имени параметра в словаре имен параметров. Порядковые номера в словарях начинаются с 1.

Если значение является целым числом и отсутствует признак кодировки - число 0 в третьем элементе массива, то значит в этом поле указан порядковый номер значения параметра в словаре значений параметров

#### **Записи из трех элементов, третий элемент равен нулю:**

Если присутствует признак кодировки - число 0 в третьем элементе массива, то это значит в поле значения указано само значение, а не ссылка на словарь.

#### **Записи из трех элементов, третий элемент не равен нулю:**

Если в записи три поля, то они задают название параметра, ссылочный параметр и регулярное выражение, которое должно быть выполнено с использованием значения ссылочного параметра.

Если регулярное выражение начинается с зарезервированных слов, то оно имеет специальное значение:

`_script` идентификатор скрипта - ссылка на скрипт, содержащий функцию извлечения значения свойства из передаваемой строки.

**PN** - массив названий наборов свойств объекта, или индекс строки соответствующего названия в словаре

**PP** - массив массивов наборов свойств, является альтернативным представлением наборов P1 - P99

### 6.4.4 Имена и структура наборов свойств

Свойства, имена которых начинаются с символа `@` являются служебными и описывают свойства самого набора.

Служебные свойства:

`@name` - имя набора



@node - имя, обозначение или номер набора, в котором содержится данный набор

Наборы, у которых не задано свойство @node, или значение @node равно нулю, называются корневыми наборами.

Корневые наборы считаются расположенными в древовидной структуре данных в порядке возрастания номеров наборов.

Служебные свойства всех наборов свойств должны быть согласованы и однозначно представлять все свойства элемента в виде древовидной структуры.

### 6.4.5 Адресация свойств

В разных наборах свойств могут быть свойства с одинаковыми именами.

Также разные наборы свойств могут иметь одинаковые имена и обозначения с учетом вложенности в другие наборы.

Для однозначной адресации свойства используется путь - последовательность нормализованных (эквивалентных) имен наборов свойств разделенных знаком "точка", и нормализованного имени свойства,

Пример:

"Свойство №1" содержится в наборе "Набор 1.1"

"Набор 1.1" содержится в наборе "Набор 1"

"Набор 1" является корневым набором

Адресация свойства - "Набор\_1.Набор\_1\_1.Свойство\_№1"

Если адресация свойства не содержит знак точки, то есть задана без перечня наборов, в то свойство ищется по имени в наборах в порядке возрастания номеров наборов.

Также свойство может адресоваться заданием номера набора и нормализованного имени свойства, разделенных точкой.

Пример:

2.Площадь\_сечения\_m2

## 6.5 Скрипты и функции

В моделях и объектах, описывающих преобразования моделей могут содержаться функции на языке JavaScript, которые используются для вычислений геометрических характеристик и значений свойств элементов модели в зависимости от значений свойств других элементов модели.

Функции содержатся в системных объектах - библиотеках функций, которые могут быть в составе моделей и других объектов БимДамп, а также в системных библиотеках файла.

Функции представляют собой именованный объект, содержащий массив со строками, в которых содержится перечень параметров функции и запись функции на языке JavaScript.

Параметры функции указываются в первой строке массива, тело функции - в последующих.

Фигурные скобки в начале и конце тела функции, и точки с запятой в конце строк массива тела функции добавляются автоматически

Например объекту

```
APlusB : [ " a , b" , " var c= a + b", "return c" ]
```

Будет соответствовать функция

```
function APlusB ( a , b )  
{ var c= a + b;  
  "return c;  
}
```

Функции могут использовать функции прикладного программного интерфейса, предоставляемого программными комплексами, в которых осуществляется обработка моделей

## 6.6 Метрики и индексы

В данном разделе описываются данные для определения вычислительной сложности отображения моделей на устройствах вывода и для обработки свойств и параметров объектов модели

Весовая метрика – число, являющееся суммой числовых показателей, умноженных на соответствующие весовые коэффициенты показателя.

Логарифмическая весовая метрика - десятичный логарифм весовой метрики

### 6.6.1 Геометрические метрики

Весовые коэффициенты для геометрического представления

Обозначение	Параметр (показатель сложности геометрического вида модели)	Весовой коэффициент
L2	количество линий 2D	13
T2	количество треугольников 2D	20
L3	количество линий 3D	15
T3	количество треугольников 3D	20

Под количеством параметров понимается количество элементарных геометрических примитивов после всех преобразований и разбиений объектов (в том числе текстовых строк и штриховок) модели на треугольники и линии

Метрика геометрии модели:

$$MG = (\text{сумма}(\text{количествоПараметров} * \text{весовойКoeffициентПараметра})) * KG$$
$$KG = 1 / 10000 = 0.0001$$

### 6.6.2 Атрибутивные метрики

Весовые коэффициенты для представления структуры

Обозначение	Параметр (показатель количества свойств объектов)	Весовой коэффициент
P1	количество свойств элементов моделей	5
K1	количество объектов в выборках	1

Под количеством свойств понимается суммарное количество свойств, заданных для всех объектов модели в виде пар «свойство=значение»

Метрика свойств модели

$MP = (\text{сумма}(\text{количествоПараметров} * \text{весовойКоэффициентПараметра})) * KP$

$KP = 1 / 1000 = 0.0001$

### 6.6.3 Структурные метрики

Весовые коэффициенты для представления свойств

Обозначение	Параметр	Весовой коэффициент
E1	количество элементов модели с геометрическим представлением	1
E2	количество элементов модели без геометрического представления	1
E3	количество элементов-прототипов с геометрическим представлением	1
E4	количество элементов-прототипов без геометрического представления	1
G1	количество групп элементов	1

Метрика структуры модели

$MS = (\text{сумма}(\text{количествоПараметров} * \text{весовойКоэффициентПараметра})) * KS$

$KS = 1 / 100 = 0.01$

### 6.6.4 Метрические индексы

Метрические индексы файлов с моделями:

## **интегральные**

M01 - суммарная метрика всех моделей в файле

## **геометрические**

M20 - суммарная геометрическая метрика всех плоских моделей в файле

M21 - геометрическая метрика самой большой плоской модели или уровня детализации модели в файле

M30 - суммарная геометрическая метрика всех трехмерных моделей в файле

M31 - геометрическая метрика самой большой трехмерной модели (или уровня детализации модели) в файле

## **аттрибутивные**

M40 - суммарная метрика свойств всех моделей в файле

M41 - метрика свойств самой большой модели или уровня детализации модели в файле

M50 - суммарная метрика всех выборок всех моделей файле

M31 – самая большая метрика выборки в файле

Логарифмические метрические индексы - обозначаются буквой L вместо буквы M

Обозначение индексов в именах файлов:

[Am01@V.tc=B.m30.m31-C.m20.m21-T=D.m40.m41-F.m50.m51]

Где

A- тип метрики – M или L. Если не указано, то M – весовая метрика

V-код программы: BS-БимСтудия, BDF-БимДампФорматизатор STR-сайт sitistest.ru и т.п.  
...

te- время затраченное на преобразования для экспорта файла в заданном формате, секунды

tc – время затраченное на создание файла, секунды

B- количество трехмерных моделей

C – количество двумерных моделей. Если не задано, и также нет преобразований, то можно не указывать

T – Количество заданных преобразований из 3D в 2D. Если не задано, то можно не указывать.

D-максимальное количество уровней детализации в моделях

F-количество выборок

m01, ..., m51 – метрики. Если количество моделей для метрики соответствующей размерности равно нулю, то соответствующие нулевые метрики можно не указывать.

Примеры

[M123-3.100.70-2.15.10-2]

[123-0-1.123]

[123] - только одна трехмерная модель

## 7. Пространственные структурные модели

### 7.1 Структурные пространственные модели (Me3D)

#### 7.1.1 Уровни геометрической проработки - LOD(G)

В моделях Me3D может содержаться от 1 до 99 различных уровней геометрической проработки, обозначаемых соответствующими номерами

Элементы модели содержат подобъекты G1...G99, в которых задаются геометрические примитивы для соответствующих уровней геометрической детализации.

Уровни геометрической проработки являются альтернативными, то есть модель всегда отображается каким то одним заданным уровнем геометрической проработки, указанной в таблице уровней проработки модели

#### 7.1.2 Уровни атрибутивной проработки - LOD(P)

В моделях Me3D может содержаться от 1 до 99 наборов свойств, соответствующих различным уровням атрибутивной проработки, обозначаемых соответствующими номерами

Элементы модели содержат примитивы P1...P99, PP, PN, в которых задаются наборы свойств.

В отличие от уровней геометрической проработки, уровни атрибутивной проработки являются не альтернативными, а накопительными - в уровень проработки модели может входить несколько наборов свойств элементов модели

#### 7.1.3 Системные элементы и таблицы модели

**TypeBase** - массив с перечнем названий используемых классификаторов для обозначения типов элементов модели. Например - [ Revit2020, IFC2x3, CityGML3]. Типы элементов модели задаются по соответствующим системам классификации в примитиве TT.

**LinkBase** - массив с перечнем идентификаторов моделей, с которыми имеется атомарная взаимосвязь через указание идентификаторов взаимосвязанных элементов в примитиве RL.

**ScriptBase** - массив с объектами, задающими программные модули функций, используемых для вычисления геометрических размеров и значений свойств элементов модели

#### 7.1.4 Состав элементов модели

Все прикладные (несистемные) элементы моделей имеют (могут иметь) следующие примитивы

**TT** - массив типов объектов по разным системам классификации, указанными в системном объекте TypeBase. Если в массиве указано число, то это ссылка на соответствующую таблицу типов классификатора.

**LL** - массив с идентификаторами взаимосвязанных элементов в моделях, указанных в системном объекте LinkBase. Для каждой модели, заданной в LinkBase указывается соответствие одному элементу.

**TB** - массив принадлежности элемента - массив, содержащий номер этажа и номера блоков в таблице блоков

**P1 ... P99, PN, PP** - наборы свойств

### 7.1.5 Элементы с геометрическим представлением

Все элементы с геометрическим представлением имеют следующие примитивы:

**S** – стиль - массив из двух или трех чисел – номер кисти, номер пера, номер материала (если есть). Если номера равны нулю, то используется кисть или перо по умолчанию.

**B** – bounding box – оболочка – массив координат ограничивающего параллелепипеда из шести чисел: [x1,x2,y1,y2,z1,z2]

#### Индивидуальные элементы ( префикс )

Имена индивидуальных элементов состоят из префикса "\_" (подчеркивание) и идентификатора элемента.

Элементы могут иметь две опции задания геометрии - непосредственное задание геометрии и прототипное

При непосредственном задании геометрии элементы содержат объекты геометрических проработок G1..G9 с вложенными геометрическими примитивами.

При прототипном задании элементы содержат примитивы:

**E** - идентификатор элемента-прототипа

**T** - матрица преобразования прототипа.. Массив из трех или девяти элементов. Если в матрице три элемента - то это значения смещений по X,Y,Z геометрического представления прототипа . Если 9 элементов - то задана матрица 3x3 для трансформации геометрического представления прототипа

При использовании объекта прототипа для задания геометрии также используются и свойства объекта-прототипа.

#### Элементы-прототипы ( Префикс \$ )

Имена элементов-прототипов элементов состоят из префикса "\$\_" и идентификатора элемента.

Элементы-прототипы содержат объекты геометрических проработок G1..G9 с вложенными геометрическими примитивами.

## 7.1.6 Элементы без геометрического представления

### Индивидуальные элементы (Префикс \$)

Имена элементов-прототипов элементов состоят из префикса "\$" и идентификатора элемента.

Индивидуальные элементы без геометрического представления содержат только общие примитивы - типы, принадлежности, свойства.

### Группы элементов (Префикс \$\$)

Имена групп элементов состоят из префикса "\$\$" и идентификатора элемента.

Группы содержат примитивы

**L** - список элементов. ["Объект",...] - массив имен объектов, входящих в группу

**G** - список групп элементов. ["Группа",...] - массив имен групп объектов, входящих в группу

## 7.2 Главная модель (Mm3D)

Как правило в программных комплексах для информационного моделирования в строительстве создается набор взаимосвязанных объектов, который представляется в виде набора моделей, демонстрирующих какие либо аспекты проекта. Пространственная трехмерная структурная модель, представляющая набор взаимосвязанных объектов, называется "главной" моделью и ей задается тип модели Mm3D.

Внутренне содержание главной модели Mm3D полностью соответствует структурным моделям типа Me3D.

Предполагается, что модели Me3D какого-либо проекта программного комплекса информационного моделирования, являются какой-либо частью (подмоделью) главной модели проекта типа Mm3D

## 7.3 Плоские трехмерные модели (Pm3D, Pz3D)

Плоские трехмерные модели являются частным случаем пространственных структурных моделей, поэтому имеют такую же структуру элементов и набор примитивов.

Как правило плоские трехмерные модели создаются из объемных трехмерных моделей с помощью проекционных трансформаций.

## 7.4 Наборы элементов моделей (Ms3D)

Наборы элементов модели содержат выборки элементов родительской модели, заданной в заголовке объектом Parent.

Наборы содержат перечень отображаемых элементов родительской модели, присвоенные этим элементам значения и шкалы для отображения этих значений раскраской элементов.

Наборы содержат следующие примитивы.

**Filters** - массив имен фильтров, которыми выполнен набор (может отсутствовать или быть пустым)

**Element** - массив имен элементов модели

**Value** - массив значений, ассоциированных с элементами модели (может отсутствовать)

**BrushScale** - имя шкалы для кистей отображения элементов

**BrushStyle** - тип отображения цвета граней

**PenScale** - имя шкалы для перьев отображения элементов

**PenStyle** - тип отображения цвета ребер

Типы отображения цвета

1 - используется цвет шкалы

2 - используется сумма цвета элемента и цвета шкалы

3 - используется произведение цвета элемента и цвета шкалы

## 7.5 Геометрические примитивы 3D моделей

Все геометрические примитивы кодируются двумя символами.

Первый символ – заглавная латинская буква, обозначающая группу.

Второй символ – латинская буква, обозначающая примитив в группе.

Третьим символом может быть символ «\_», обозначающий типа позиционирования (задания координат).

(Точка в массивах представляется тройкой координат  $x$   $y$   $z$ , или номером точки в массиве точек, в зависимости от используемого типа позиционирования, задаваемого постфиксом "\_")

### 7.5.1 3D - Трехмерная геометрия - тела

1. **Ga, Ga\_** - замкнутые тела (имеющие объем), заданные триангулированными гранями.
  - 1.1. Структура Ga:  $[[[[x_1, y_1, z_1], [x_2, y_2, z_2], [x_3, y_3, z_3] ], \dots], \dots]$
  - 1.2. Структура Ga\_:  $[[[n_1, n_2, n_3], \dots], \dots]$
  - 1.3. Уровни вложенности:
    - 1.3.1. 1 – Полигоны
    - 1.3.2. 2 – Треугольники полигона
    - 1.3.3. 3 – Точки треугольника \ Номера точек треугольника
    - 1.3.4. 4 – Координаты точки \
2. **Gc, Gc\_** - замкнутые тела (имеющие объем), заданные цветными триангулированными гранями.
  - 2.1. Структура Gc:  $[[ \text{№ кисти1}, [[x_1, y_1, z_1], [x_2, y_2, z_2], [x_3, y_3, z_3]], \dots ], [ \text{№ кисти2}, [[\dots], [[\dots], \dots ], \dots ]]$
  - 2.2. Структура Gc\_:  $[[ \text{№ кисти1}, [n_1, n_2, n_3], [ \dots ], \dots ], [ \text{№ кисти2}, [n_4, n_5, n_6], [ \dots ], \dots ], \dots ]]$
  - 2.3. Уровни вложенности:
    - 2.3.1. 1 – Массивы полигонов одного цвета



- 2.3.2. 2 – Полигоны с указанной кистью
- 2.3.3. 3 – Точки полигона \ Номера точек полигона
- 2.3.4. 4 – Координаты точки \ -
3. **Gp, Gp\_** - замкнутые тела (имеющие объем), заданные нетриангулированными гранями без отверстий.
- 3.1. Структура Gp: [ [ [ [x<sub>1</sub>, y<sub>1</sub>, z<sub>1</sub>], ..., [x<sub>n</sub>, y<sub>n</sub>, z<sub>n</sub>] ], .. ] , .. ]
- 3.2. Структура Gp\_: [ [ [n<sub>1</sub>, ..., n<sub>m</sub>], ...], ... ]
- 3.3. Уровни вложенности (Gp \Gp\_):
- 3.3.1. 1 – Полигоны (границы)
- 3.3.2. 2 – Полигон границы
- 3.3.3. 3 – Точки полигона \ Номер точки полигона
- 3.3.4. 4 – Координаты точки \ -
4. **Gh, Gh\_** - замкнутые тела (имеющие объем), заданные нетриангулированными гранями, в которых могут быть отверстия.
- 4.1. Структура Gh: [ [ [ [ [x<sub>1</sub>, y<sub>1</sub>, z<sub>1</sub>], ..., [x<sub>n</sub>, y<sub>n</sub>, z<sub>n</sub>] ], .. ] , .. ] , .. ]
- 4.2. Структура Gh\_: [ [ [ [n<sub>1</sub>, ..., n<sub>m</sub>], ...], ... ] , .. ]
- 4.3. Уровни вложенности (Gh \Gh\_):
- 4.3.1. 1 – грани
- 4.3.2. 2 – грань
- 4.3.3. 3 - полигоны контуров. Первый контур - наружный, направление по часовой стрелке. Следующие контуры - контуры отверстий, направление против часовой стрелки
- 4.3.4. 4 – Точки полигона \ Номер точки полигона
- 4.3.5. 5 – Координаты точки \ -
5. **Gx, Gx\_** - замкнутые тела (имеющие объем), заданные произвольными гранями. Грани задаются массивами, первое значение в массиве – тип грани, следующие – в зависимости от типа грани.
- Структура: [ [ тип1, ... ] , [тип2, ...], ... ]
- 5.1. 1 – Триангулированная одноцветная грань
- 5.1.1. (1 элемент - тип, 2 по n - треугольники)
- 5.1.1.1. Структура: [1, [[x<sub>1</sub>, y<sub>1</sub>, z<sub>1</sub>], ..., [x<sub>3</sub>, y<sub>3</sub>, z<sub>3</sub>]], ..., [[x<sub>n</sub>, y<sub>n</sub>, z<sub>n</sub>], ..., [x<sub>3n</sub>, y<sub>3n</sub>, z<sub>3n</sub>]]
- 5.2. 2 – Нетриангулированная одноцветная грань (возможны отверстия)
- 5.2.1. (1 элемент - тип, 2 по n – полигоны контуров (первый полигон - наружный контур, направление по часовой стрелке. Следующие полигоны - контуры отверстий, направление против часовой стрелки), m – кол-во вершин полигона)
- 5.2.1.1. Структура: [2, [[x<sub>1</sub>, y<sub>1</sub>, z<sub>1</sub>], ..., [x<sub>m</sub>, y<sub>m</sub>, z<sub>m</sub>]], ..., [[x<sub>1n</sub>, y<sub>1n</sub>, z<sub>1n</sub>], ..., [x<sub>mn</sub>, y<sub>mn</sub>, z<sub>mn</sub>]]
- 5.3. 3 – Плиточная грань (текстурные плитки)
- 5.3.1. (1 элемент тип, 2 – текстура, 3 - полигоны, 4 - масштаб текстурных плиток)
- 5.3.1.1. Структура: [3, № текстуры, [[[x<sub>1</sub>, y<sub>1</sub>, z<sub>1</sub>], ..., [x<sub>m</sub>, y<sub>m</sub>, z<sub>m</sub>]], ..., [[x<sub>1n</sub>, y<sub>1n</sub>, z<sub>1n</sub>], ..., [x<sub>mn</sub>, y<sub>mn</sub>, z<sub>mn</sub>]]], [[[px<sub>1</sub>, py<sub>1</sub>, pz<sub>1</sub>], [v1x<sub>1</sub>, v1y<sub>1</sub>, v1z<sub>1</sub>], [v2x<sub>1</sub>, v2y<sub>1</sub>, v2z<sub>1</sub>]], ..., [[px<sub>n</sub>, py<sub>n</sub>, pz<sub>n</sub>], [v1x<sub>n</sub>, v1y<sub>n</sub>, v1z<sub>n</sub>], [v2x<sub>n</sub>, v2y<sub>n</sub>, v2z<sub>n</sub>]]]
- 5.3.1.2. **Масштаб текстурных плиток** – два ортогональных трехмерных вектора, лежащих в плоскости грани, задающих начальную точку, направление и растяжение текстуры. Если масштабные вектора не корректны, то грань закрашивается сплошным цветом. Задается для каждой грани.
- 5.4. 4 – текстурированная грань
- 5.4.1. (1 элемент тип, 2 – текстура, 3 - полигоны, 4 - текстурные координаты)

- 5.4.1.1. Структура: [4, № текстуры, [[[ $x_{1_1}, y_{1_1}, z_{1_1}$ ], ..., [ $x_{m_1}, y_{m_1}, z_{m_1}$ ]], ..., [[ $x_{1_n}, y_{1_n}, z_{1_n}$ ], ..., [ $x_{m_n}, y_{m_n}, z_{m_n}$ ]], [[[ $u_{1_1}, v_{1_1}$ ], ..., [ $u_{m_1}, v_{m_1}$ ]], ..., [[ $u_{1_n}, v_{1_n}$ ], ..., [ $u_{m_n}, v_{m_n}$ ]]]]]
6. **Ge, Ge** - тела, заданные экструзией - [тип экструзии, параметры экструзии]
- 6.1. Тип экструзии 1 - экструзия выпуклой триангулированной поверхности
- 6.1.1. Направление экструзии [ $x_1, y_1, z_1$ ],
- 6.1.2. Список точек наружного контура
- 6.2. Тип экструзии 2 - экструзия произвольной триангулированной поверхности (опорная плоскость задается точками наружного контура)
- 6.2.1. Направление экструзии [ $x_1, y_1, z_1$ ],
- 6.2.2. Контур
- 6.2.2.1. Наружный контур (список точек полигона)
- 6.2.2.2. Отверстие1 (список точек полигона)
- 6.2.2.3. ...
- 6.2.3. Тип списка точек
- 6.2.4. Список точек
- 6.2.4.1. ТипСамскаТочек=1 - список треугольников, задаваемых тремя точками.
- 6.2.4.2. ТипСпискаТочек=2 - список точек, каждые три точки последовательные точки задают треугольник поверхности
- 6.3. Тип экструзии 3 - экструзия произвольной плоской нетриангулированной поверхности (опорная плоскость задается точками наружного контура)
- 6.3.1. Направление экструзии [ $x_1, y_1, z_1$ ],
- 6.3.2. Наружный контур, список точек, направление по часовой стрелке
- 6.3.3. Список отверстий
- 6.3.4. Отверстие1, списков точек контура отверстия, направление против часовой стрелки
- 6.3.5. ...
- 6.4. Тип экструзии 11 - ортогональная экструзия выпуклой триангулированной поверхности
- 6.4.1. Высота экструзии
- 6.4.2. Список точек наружного контура
- 6.5. Тип экструзии 12 - ортогональная экструзия произвольной триангулированной поверхности (опорная плоскость задается точками наружного контура)
- 6.5.1. высота экструзии
- 6.5.2. Контур
- 6.5.2.1. Наружный контур (список точек полигона)
- 6.5.2.2. Отверстие1 (список точек полигона)
- 6.5.2.3. ...
- 6.5.3. Тип списка точек
- 6.5.4. Список точек
- 6.5.4.1. ТипСамскаТочек=1 - список треугольников, задаваемых тремя точками.
- 6.5.4.2. ТипСпискаТочек=2 - список точек, каждые три точки последовательные точки задают треугольник поверхности
- 6.6. Тип экструзии 13 - ортогональная экструзия произвольной плоской нетриангулированной поверхности (опорная плоскость задается точками наружного контура)
- 6.6.1. высота экструзии,
- 6.6.2. Наружный контур, список точек, направление по часовой стрелке
- 6.6.3. Список отверстий
- 6.6.4. Отверстие1, списков точек контура отверстия, направление против часовой стрелки
- 6.6.5. ...

7. **Gt, Gt\_** - тела, заданные типовыми операциями формообразования - [ тип, параметры] (ортогональные тела - тела, для которых опорная плоскость ортогональна оси тела, которая также является направлением экструзии)
  - 7.1. Тип 11-сплошной ортогональный стержень / тип 21- выпуклая ортогональная плита
    - 7.1.1. Опорная плоскость - точка, вектор X, вектор Y
    - 7.1.2. Высота стержня / Толщина плиты
    - 7.1.3. Наружный контур - список точек 2D на опорной плоскости, по часовой стрелке. Каждые три последовательные точки задают триангулированной треугольник поверхности плиты.
  - 7.2. Тип 12 -трубчатый ортогональный стержень / Тип 22- ортогональная плита
    - 7.2.1. Опорная плоскость - точка, вектор X, вектор Y
    - 7.2.2. Высота стержня / Толщина плиты
    - 7.2.3. Наружный контур - список точек 2D на опорной плоскости, по часовой стрелке
    - 7.2.4. Контуры отверстий - список точек 2D на опорной плоскости, против часовой стрелке
  - 7.3. Тип 13 - круглая ортогональная труба
    - 7.3.1. Опорная плоскость - точка, вектор X, вектор Y
    - 7.3.2. высота
    - 7.3.3. Радиус, толщина (если толщина=0, то сплошной стержень)
  - 7.4. Тип 31 - ортогональная стена без отверстий
    - 7.4.1. Опорная плоскость - точка, вектор X, вектор Y
    - 7.4.2. Высота, толщина, замкнутость (1-замкнуто, 0-незамкнуто)
    - 7.4.3. Список координат вершин осевой линии стены
  - 7.5. Тип 32 - ортогональная стена с прямоугольными отверстиями
    - 7.5.1. Опорная плоскость - точка, вектор X, вектор Y
    - 7.5.2. Высота, толщина, замкнутость (1-замкнуто, 0-незамкнуто)
    - 7.5.3. Список координат вершин осевой линии стены
    - 7.5.4. Список отверстий - Привязка начала отверстия (от первой вершины), низ отверстия, длина отверстия, высота отверстия

## 7.5.2 2D - Двумерная геометрия – поверхности

1. **Qa, Qa\_** - незамкнутые тела (не имеющие объема), поверхности, заданные треугольными сетками
  - 1.1. Структура аналогична структуре Ga,Ga\_
2. **Qc, Qc\_** - незамкнутые тела (не имеющие объема), поверхности, заданные замкнутыми полигонами указанного цвета
  - 2.1. Структура аналогична структуре Gc,Gc\_
3. **Qp, Qp\_** - незамкнутые тела (не имеющие объема), поверхности без отверстий, заданные замкнутыми полигонами
  - 3.1. Структура аналогична структуре Gp,Gp\_
4. **Qr, Qr\_** - незамкнутые тела (не имеющие объема), поверхности в которых могут быть отверстия, заданные замкнутыми полигонами
  - 4.1. Структура аналогична структуре Gh,Gh\_
5. **Qx, Qx\_** - незамкнутые тела (не имеющие объема), заданные произвольными гранями. Грани задаются массивами, первое значение в массиве – тип грани. Дальнейшие значения в массиве – в зависимости от типа грани.
  - 5.1. 1 – Триангулированная одноцветная грань:
    - 5.1.1. (1 элемент тип, 2 по n - треугольники)
      - 5.1.1.1. Структура: [1, [[x1<sub>1</sub>, y1<sub>1</sub>, z1<sub>1</sub>], ..., [x3<sub>1</sub>, y3<sub>1</sub>, z3<sub>1</sub>]], ..., [[x1<sub>n</sub>, y1<sub>n</sub>, z1<sub>n</sub>], ..., [x3<sub>n</sub>, y3<sub>n</sub>, z3<sub>n</sub>]]



### 7.5.3 1D - Одномерная геометрия - линии и кривые

1. **La, La\_** - линии, заданные полигонами
  - 1.1. Структура: [ [ [x<sub>1</sub>, y<sub>1</sub>, z<sub>1</sub>], ..., [x<sub>n</sub>, y<sub>n</sub>, z<sub>n</sub>], [...] ], ... ]
2. **Lc, Lc\_** - линии, заданные цветными полигонами.
  - 2.1. Структура: [ [ № пера1, [ [x<sub>1</sub>, y<sub>1</sub>, z<sub>1</sub>], ..., [x<sub>n</sub>, y<sub>n</sub>, z<sub>n</sub>], [...] ], ... ] , [ № пера2, ... ], ... ]
3. **Lt, Lt\_** - линии, заданные полигонами и кривыми. (первое значение в массиве значений - тип кривой)
  - 3.1. Типы кривых
    - 3.1.1. 1 - незамкнутый полигон
    - 3.1.2. 2 - замкнутый полигон
    - 3.1.3. 3- сплайн должно быть не менее 3 точек, иначе представляется незамкнутым полигоном
    - 3.1.4. 4-кубический сплайн (должно быть задано не менее 4 точек, иначе представляется сплайном)

### 7.5.4 0D - Точки

1. **Da, Da\_** - наборы точек
2. **Dt, Dt\_** - генерируемые наборы точек (например точки, равномерно распределенные в объеме или на плоскости). Тип генерации задается в начале списка величин.

### 7.5.5 Текст

1. **Tx, Tx\_** – текст – массив строк текста, местоположения и направления
  - 1.1. [ [ выравнивание, цвет, точкаНачала, векторВдоль, векторВверх, шрифтСтиль, "Текст"], [...], ... ]
    - 1.1.1. Выравнивание - 1-по левому краю, 2-по правому краю, 3-по центру и т.п.
    - 1.1.2. Цвет - RGBA
2. **Ts, Ts\_** – ссылочный текст
  - 2.1. [ [ выравнивание, цвет, точкаНачала, векторВдоль, векторВверх, шрифтСтиль, текстовыйНабор, текстовыйОбъект], [...], ... ]
    - 2.1.1. Выравнивание - 1-по левому краю, 2-по правому краю, 3-по центру и т.п.
    - 2.1.2. Цвет - RGBA

### 7.5.6 Ссылочная геометрия

1. **En** – ссылочная геометрия - идентификтор объекта в геометрической модели, указанной в манифесте модели
  - 1.1. Для GLTF - индекс узла (node) в модели GLTF

## 7.6 Внешняя геометрия glTF

Каждому уровню геометрической проработки модели БимДамп соответствует отдельная сцена в файле glTF, указанном в заголовке модели.

Для компонентного взаимосвязывания в объекте asset сцены указываются атрибуты:

BimML=1

GUID = глобальный идентификатор модели

ID = локальный ключ-идентификатор модели

ModelName = название модели

LODG = номер уровня геометрической проработки

Для обеспечения атомарного взаимосвязывания для узловых объектов (nodes) в структуре данных glTF, соответствующих геометрическому представлению элемента модели БимДамп, задается имя этого элемента.

Геометрические примитивы с штриховками при преобразовании в glTF задаются эквивалентным цветом или эквивалентными текстурами.

## 8. Пространственные параметрические модели

### 8.1 Общее описание параметрических моделей

Параметрические модели - модели, для которых форма, внешний вид и свойства зависят (могут зависеть) от значений параметров, ассоциированных с этой моделью.

Параметры могут быть двух видов:

**переменные параметры**

**дискретные параметры** (параметрами выбора)

Дискретные параметры могут использоваться для объединения в одной модели вариантов отображения внешнего вида модели, удобных для отдельных задач анализа информации, содержащейся в модели. в зависимости от значений параметров из заданного в модели набора возможных значений,

Переменные параметры позволяют сделать модель с размерностью более чем 3 за счет отображения состояний геометрического вида 3D в зависимости других величин.

Такие многомерные модели для отличия от 3D моделей можно условно обозначать как 3D+ модели.

Пространственные модели, состояние которых зависит от времени, которые часто обозначаются как 4D модели, могут представляться пространственной параметрической моделью с одним переменным параметром "время".

К параметрическим моделям также можно отнести и расчетные и агентские модели, также описанные в спецификации БимДамп.

### 8.2 Пространственные параметрические модели (Mr3D)

Пространственная параметрическая модель типа Mr3D - структурная пространственная модель, аналогичная типам M3D, Pz3D и т.п., для которой заданы варианты геометрического представления модели в зависимости от заданного в модели набора дискретных и/или переменных параметров

### 8.3 Расчетные и агентские модели

К параметрическим моделям также можно отнести и расчетные и агентские модели, также описанные в спецификации БимДамп.

Расчетные модели реализуют понятие "нагрузка" (воздействие), являющееся по существу дискретным параметром для отображения вида нагрузок и воздействий, а также понятие "время", являющимся переменным параметром

В агентские модели всегда также задано понятие "время", являющееся переменным параметром

Расчетные и агентские модели также относятся к виду моделей, обозначаемых как 3D+ модели

## 8.4 Параметры моделей

В зависимости от типа параметрической модели объекты модели могут использовать данные из различных таблиц.

Таблицы подразделяются на следующие типы

1. **Статические таблицы** - таблицы, не зависящие от параметров
2. **Динамические таблицы** - таблицы, зависящие от параметров. Динамические таблицы имеют префикс "v"
3. **Косвенные таблицы** - динамические таблицы, зависящие от параметров, и использующие наборы свойств. Косвенные таблицы имеют префикс "r"

Структура описания параметров моделей в составе заголовка модели:

1. isParametric - тип модели 1 - параметрическая модель 0 - обычная модель
2. Params - Структура описания параметров модели
  - 1.2. PType - описание изменяемых свойств модели. Коды для объектов в PType приведены ниже в отдельном перечне
    - 1.2.1. Object - существование объектов (0 или 1)
    - 1.2.2. Point - координаты
    - 1.2.3. Pen - перья
    - 1.2.4. Brush - кисти
    - 1.2.5. Prop1 - свойства модели
    - 1.2.6. Prop2 - свойства объектов
  - 1.3. Discrete
    - 1.3.1. Дискретный параметр. Имя параметра - порядковый номер, начиная с 1
      - 1.3.1.1. Name - название, например "время"
      - 1.3.1.2. Symbol - обозначение, например "T"
      - 1.3.1.3. Descript - описание
      - 1.3.1.4. Unit - название размерности единицы, например "секунды"
      - 1.3.1.5. Design - обозначение размерности единицы, например "с"
      - 1.3.1.6. Value - массив значений
        - 1.3.1.6.1. Значение параметра. Например "dX"
        - 1.3.1.6.2. Описание значения. Например "Смещение по оси X"
      - 1.3.1.7. DValue - номер значения по умолчанию. Если не задано, то 1
    - 1.4. Var - описание переменных параметров (variables)
      - 1.4.1. Переменная. Имя переменной - порядковый номер, начиная с 1
        - 1.4.1.1. Name - название, например "время"
        - 1.4.1.2. Symbol - обозначение, например "T"
        - 1.4.1.3. Descript - описание
        - 1.4.1.4. Unit - название размерности единицы, например "секунды"
        - 1.4.1.5. Design - обозначение размерности единицы, например "с"

- 1.4.1.6. Start - начальное значение диапазона переменной
  - 1.4.1.7. End - конечное значение диапазона переменной
  - 1.4.1.8. isCycle - 0 или 1 - признак цикличности. По умолчанию 0
  - 1.4.1.9. isProcess - 0 или 1 признак того, что переменный параметр моделирует какой-либо процесс. Для управления отображением параметров процессов программы просмотра могут выводить интерфейсные элементы проигрывателя процесса.
  - 1.4.1.10. DValue - значение по умолчанию. Если не задано, то начало диапазона Start
- 1.5. Events - события
- 1.5.1. Таблица событий для переменной. Имя таблицы - номер переменной.  
Таблица - массив
    - 1.5.1.1. Name - название события
    - 1.5.1.2. Symbol - обозначение события
    - 1.5.1.3. Start - начальное значение диапазона события
    - 1.5.1.4. End - конечное значение диапазона события
- 1.6. Scaling - масштабный коэффициент для показа объектов PType с кодом 2x
- 1.6.1. Default - значение по умолчанию
  - 1.6.2. Min - начало диапазона
  - 1.6.3. Max - конец диапазона
  - 1.6.4. Fix - массив фиксированных значений (альтернатива диапазону)

PType - массив кодов, задающих правила формирования значения отображения элементов параметрической модели с использованием статических и динамических таблиц

Коды объекта PType:

1. K1 - код использования статической таблицы
  - 1.1. 1 - статическая таблица не используется
  - 1.2. 2 - сумма значений статической таблицы и динамического значения, умноженного на весовой коэффициент W1.
  - 1.3. 3 - статическое значение умножается на динамическое значение и весовой коэффициент W1
2. W1 - весовой коэффициент. По умолчанию -1.0
3. K2 - код формирования динамического значения (При суммировании используются переменные таблицы как без сопутствующих дискретных параметров, так и для соответствующих значений дискретных параметров)
  - 3.1. 1 - сумма значений дискретных динамических таблиц и суммы значений переменных таблиц с весовым коэффициентом W2. При суммировании используются переменные таблицы для соответствующих значений дискретных параметров.
  - 3.2. 2 - сумма значений дискретных динамических таблиц, умноженная на сумму значений переменных таблиц и весовой коэффициент W2
4. W2 - весовой коэффициент. По умолчанию 1.0

Для формирования значений типизированных атрибутов отображения, таких как Кисть и Перо, используется тип из первой используемой таблицы. Например, если используется статическая таблица, то из статической таблицы. Если статическая таблица не используется, то из динамической таблицы первого дискретного параметра.

## 8.5 Динамические таблицы



## 8.5.1 Динамические таблицы для дискретных параметров

Имена динамических таблиц дискретных параметров имеют вид;

pdТаблица\_D

где

1. pd - префикс динамической таблицы
  2. Таблица - тип таблицы, например Point, Vector, Pen, Brush и т.п.
  3. D - номер дискретного параметра
- 
1. pdPoint3D\_D - Массив координат точек -  $[[x,y,z,...]]$  для каждого значения параметра
  2. pdVector3D\_D - Массив массивов векторов -  $[[x,y,z,...]]$  для каждого значения параметра
  3. pdPoint2D\_D - Массив массивов координат точек -  $[[x,y,...]]$  для каждого значения параметра
  4. pdVector2D\_D - Массив массивов векторов -  $[[x,y,...]]$  для каждого значения
  5. pdPen\_D – массив перьев для каждого значения. Перо – массив из 6 чисел –  $[[R, G, B, A, \text{типЛинии}, \text{толщина}, ...]]$
  6. pdBrush\_D - массив кистей. для каждого значения Кисть – массив из 9 чисел –  $[[R, G, B, A, \text{типЗаливки}, r, g, b, a, m...]]$

## 8.5.2 Динамические таблицы для переменных параметров

Имена динамических таблиц переменных параметров имеют вид;

rvТаблица\_V

или

rvТаблица\_V\_D\_N

где

1. rv - префикс динамической таблицы переменного параметра
2. Таблица - тип таблицы, например Point, Vector, Pen, Brush и т.п.
3. V - номер переменного параметра
4. D – номер сопутствующего дискретного параметра
5. N – номер значения сопутствующего дискретного параметра

В записях таблицы вместо одного набора значений используется последовательность значений свойства и соответствующего значения переменной.

Например  $[x, y, z, v, \quad x, y, z, v, \quad x, y, z, v, \quad x, y, z, v, \quad \dots ]$

Значение свойства в зависимости от значения определяется линейной интерполяцией.

Значения свойства до первого заданного значения переменной равны первому заданному значению.

Значения свойства после последнего заданного значения переменной равны последнему заданному значению заданному значению.

Если в свойстве используется какой либо номер типа, например типЛинии или типЗаливки, до данные типы указываются в начале последовательности и действуют на всю последовательность

Например для V, определенного в диапазоне от start=0 до end=10, для набора X Y Z

[ 12, 12, 12, 2, 18, 18, 18, 8]

V=1 x y z = 12, 12, 12

V=5 x y z = 15, 15, 15

V=9 x y z = 18, 18, 18

Вместо задания набора значений возможно задания ссылки на объект модели, содержащий набор. Наборы - объекты модели с префиксом "\$\_", содержащие массивы с последовательностью значений.

То есть если в массиве записи динамической таблицы только одна величина, то это ссылка на объект, содержащий набор значений, и если в записи две и более величины - то это набор значений.

1. vPoint3D\_V\_D\_N - Массив последовательности переменных координат точек - [[x,y,z, v ... ]]
2. vVector3D\_V\_D\_N - Массив последовательности переменных векторов - [[x,y,z, v, ...]]
3. vPoint2D\_V\_D\_N - Массив последовательности переменных координат точек - [[x,y, v, ...]]
4. vVector2D\_V\_D\_N - Массив последовательности переменных векторов - [[x,y,v, ...]]
5. vPen\_V\_D\_N – массив последовательности переменных перьев. [[ типЛинии, R, G, B, A, толщина, v, R, G, B, A, толщина, v ... ]]
6. vBrush\_V\_D\_N - массив последовательности переменных кистей. Для типа заливки 0 задается только цвет фона R G B A. [[ типЗаливки, R, G, B, A, v, R, G, B, A, v ... ]]  
Для типов заливки с штриховками задаются масштаб штриховки и последовательности двух цветов - цвет фона и цвет штриховки. [[ типЗаливки, m, R, G, B, A, r, g, b, a, v, R, G, B, A, r, g, b, a, v ... ]]

### 8.5.3 Таблица существования

Таблица существования - таблица, в которой для объектов модели задан признак существования E.

Если E =1, то объект существует, если E=0, то не существует.

Все объекты, не заданные в таблице существования, существуют.

Таблица существования для дискретных параметров:

pdObject\_D - массив имен объектов и признаков существования для значений параметра [[ имяОбъекта, E, ... ]]

Таблица существования для переменных параметров:

pvObject\_V - массив имен объектов и последовательностей признака существования [[ имяОбъекта, E, v, ... ]]

Признак существования объекта устанавливается для объекта в заданном значении V и действуют до следующего задания признака в следующем значении переменной

По умолчанию в начале диапазона переменного параметра все объекты существуют.

## 9. Пространственные расчетные модели

### 9.1 Общее описание расчетных моделей

Расчетные модели отличаются от структурных моделей в следующих аспектах:

1. в расчетных моделях значительно меньше типов используемых элементов
2. элементы модели имеют более простую геометрическую форму по сравнению с структурными моделями
3. расчетные модели содержат информацию о взаимном сопряжении элементов в узловых точках (узлах)
4. узлы модели могут иметь свойства

В связи с этими отличиями структура и состав элементов расчетных моделей в формате БимДамп отличается от структурных моделей.

Основой расчетной модели является база данных реляционного типа компонентов модели в виде набора таблиц с примитивами, содержащими основные численные свойства элементов модели.

Каждой записи базы данных, соответствующей какому либо компоненту модели, может сопоставляться объект в объектно-ориентированной форме описания, обогащающий стандартный набор числовых свойств из базы данных набором свойств и примитивов произвольной структуры.

Расчетные модели и связанные с расчетным моделированием данные подразделяются на следующие виды объектов:

**Конечно-элементные (дискретизированные) модели - Fe3D** - состоят из сети конечных элементов с назначенными расчетными свойствами для возможности выполнения моделирования в расчетных программных комплексах, и последующей обработкой постпроцессорами. Сеть конечных элементов как правило создается разбиением (дискретизацией) концептуальных моделей, но также может быть создана непосредственно в расчетных программных комплексах без использования аналитических моделей. В конечно-элементных моделях также может указываться задачи моделирования - перечень данных, которые нужно получить в результате расчета, параметры выполнения расчета.

**Аналитические (концептуальные) модели - Fa3D** - состоят из формообразующих элементов, соответствующих отдельным частям моделируемого реального объекта с одинаковыми (единообразными) для рассматриваемой расчетной задачи свойствами и моделями реакции на воздействия. В аналитических моделях также может указываться задачи моделирования (расчетные задания) - перечень данных, которые нужно получить в результате расчета, параметры дискретизации модели или необходимая точность расчета, другие подобные параметры.

**Референтные модели - Fr3D** - состоят из аналитической модели, расчетного задания и набора известных свойств модели, полученных в результате расчетов или измерений с известной погрешностью, и считающихся правильными и достоверными. Референтные модели используются в качестве контрольных примеров для выполнения валидации решателей и верификации методов расчета.

**Расчетные задания - TaMS** - состоят из описания конвейера расчетных процедур, используемых препроцессоров, решателей и постпроцессоров, параметров запуска расчетов в программных комплексах и т.п. данные

**Результаты отдельного расчета - ReMS** - состоят из наборов данных результатов расчета, полученных при расчете по заданными решателями конечно-элементной модели, и последующей постобработки рассчитанных данных.

**Результаты анализа - RaMS** - состоят из наборов данных результатов одного или нескольких расчетов, полученных при расчете заданными решателями группы конечно-элементных моделей, соответствующих аналитической модели, и последующей постобработки рассчитанных данных.

**Комплексные аналитические модели - Sa3D** - состоят из аналитической или референтной модели, расчетного задания и аппроксимированных результатов одного или нескольких конечно-элементных расчетов и их анализа.

**Комплексные конечно-элементные модели - Se3D** - состоят из конечно-элементной модели, расчетного задания и результатов расчета.

Для упрощения терминологии для описания различных типов элементов в расчетных моделях, элементы моделей, соответствующие конечным элементам расчетных комплексов далее называются **элементами**, а объекты аналитических моделей называются **аналитическими элементами**, или **протоэлементами**.

## 9.2 Состав моделей

### 9.2.1 Объекты элементов модели

Имя объектно-ориентированных элементов расчетных моделей имеет вид

**\_id\$class**

Где:

"\_" - префикс имени объекта

Id - уникальный идентификатор элемента в модели. Как правило десятичное число

"\$" - разделитель между идентификатором и именем класса

Class - имя класса элемента

Элементы имеют следующие примитивы:

**Class** - класс элемента

**Name** - имя

**Descr** - описание

**GUID** - глобальный идентификатор

**ListType** - Тип списка индексов элементов в базе

1 - перечень

2 -диапазон. В списке заданы начало и конец диапазона номеров

3 - перечень диапазонов

**List** - Список индексов элементов в базе

**P1** - свойства - массив пар имяСвойства-значениеСвойства

**Геометрические примитивы**

## 9.2.2 Таблицы базы данных модели

Классы системных таблиц с примитивами элементов и перечень параметров содержащихся в записях примитивов:

### Базовые примитивы пространства

**PointBase** - точки (рекомендуется вначале задавать точки узлов, потом другие точки, для более простой обработки массивов величин в узлах и результатов расчетов)

[[ x y z ]]

### Аналитические примитивы пространства (модели Fa3D)

**LineList** - линия

[[ типЛинии, списокТочек, ...]]

1. типЛинии

1.1. 1 - pline – полилиния, набор прямых отрезков (по умолчанию)

1.2. 2 - spline – кривая, сплайн через заданные точки, количество точек не менее 3, иначе - полилиния

1.3. 3 - cspline – кривая, кубический сплайн через заданные точки, количество точек не менее 4, иначе – сплайн

1.4. 4 - arch – дуги окружности. Количество точек не менее чем 3. Количество точек должно быть нечетным.

1.5. 5 - circle – окружность через 3 точки

1.6. 6 - archc – дуга окружности с центром в первой точке, и точкой начала и конца дуги

1.7. 7 - earch – эллиптическая дуга – центр, точка на главной оси, точки начала и конца дуги

1.8. 8 - ellipse – эллипс – центр, точка на главной оси, точка эллипса (не на главной оси)

**OutlineBase** - контур

[[типНаправления, списокЛиний,... ]]

1. ТипНаправления - код направления контура по направлению первой линии.

1.1. 1- по направлению первой линии (от первой точки к последней )

1.2. 2- против направления первой линии (от последней точки к первой)

**FacetBase** грани - элементы поверхностей. Первый контур задает внешний контур, последующие контуры задают контуры отверстий

[[[списокКонтуров,...]]]

**FacetDrawBase** триангулированные грани для отрисовки. Список треугольников (типСписка=1) или полоса треугольников (типСписка=2). Треугольники задаются тремя точками. Для полосы треугольников для каждого последующего треугольника задается одна новая точка, две точки берутся из предыдущего треугольника.

[[[ типСписка, списокТочекТреугольников,...]]]

**FieldBase** - объемные области

[[списокГраней,...]]

**NodPointBase** - "жесткие" узловые точки - точки, для которых при дискретизации модели создаются узловые точки конечных элементов

[[точка, типЖесткойТочки, объект]

1. ТипЖесткойТочки
  - 1.1. 1 - линия
  - 1.2. 2 - контур
  - 1.3. 3 - грань
  - 1.4. 4 - элементарное тело

### Примитивы времени

**TimeBase** - расчетное время - список моментов времени. Первый момент времени - начало расчетного времени, последний - окончание расчетного времени. Если время окончания не задано, меньше или равно времени начала, то расчетное время является бесконечным.

[ моментВремени, ]

**EventBase** - события

[[ моментВремени, продолжительность, типСобытия, списокПараметровСобытия... ]]

### Примитивы дискретизации

**DiscretBase** - таблица параметров дискретизации. список периодов дискретизации времени. Если последующий момент времени меньше предыдущего, то период времени завершается. Если шаг равен нулю, то дискретизация времени не определена

[[ типДискрета, параметрыДискрета, ... ]]

Типы Дискретов и их параметры

Дискретизация пространства:

11 - максимальный размер элемента

12 - максимальный размер элементов рядом с заданными точками. Параметры - перечень точек

### Дискретизация времени:

21 - шаги дискретизации для периодов времени. Задаются шаги для периодов расчетного времени в TimeBase. Последний заданный шаг применяется для последующих периодов. То есть если задан один шаг времени, то он применяется для всего времени

22 - периоды дискретизации. Параметры - шаг времени, начало периода дискретизации, и т.д. Последний заданный период дискретизации считается бесконечным

### Дискретизация аппроксимирования результатов расчета:

31 - максимальный размер элемента диаграммы. Параметры - перечень величин, для которых строятся диаграммы. Если не задано, то все рассчитанные величины

32 - максимальный размер элементов диаграммы рядом с заданными точками. Параметры - перечень величин, для которых строятся диаграммы. Если не задано, то все рассчитанные величины

33 - количество линий уровня в диаграммах рассчитанных величин. Параметры - перечень величин, для которых строятся диаграммы. Если не задано, то все рассчитанные величины

## Расчетные задания

### **TaskBase** - расчетные задачи

[ кодЗадачи, типЗадачи, , названиеЗадачи, описаниеЗадачи ]

Код задачи - произвольное текстовое обозначение без пробелов, для удобства организации ссылок, взаимосвязи и документирования. Код не должен начинаться с цифры

Типы Задач:

#### 1 - дискретизация

101 - дискретизация пространства (конечно-элементное разбиение)

102 - дискретизация времени

103 - дискретизация пространства и времени

#### 2- Механический расчет

200 - произвольный расчет

201 - статический расчет

202 - динамический расчет

203 - статический и динамический расчет

#### 3 - Мультифизический расчет

300 - произвольный расчет

- 301 - стационарный расчет
- 303 - нестационарный расчет
- 4 - Аппроксимация
- 400 - произвольная обработка
- 401 - создание эпюр и диаграмм
- 5 - Обработка сценариев воздействий
- 500 - произвольная обработка
- 501 - комбинации усилий
- 502 - комбинации перемещений
- 503 - комбинации усилий и перемещений
- 6 - конструирование
- 600 - произвольная обработка
- 7 - верификация
- 700 - произвольная обработка
- 8 - документирование
- 800 - произвольная обработка
- 9 - прочая обработка
- 900 - произвольная обработка

**SolvertBase** - описание решателей и обработчиков

[ кодРешателя, типЗадачи, названиеРешателя, списокПараметровРешателя ]

Код решателя - произвольное текстовое обозначение без пробелов, для удобства организации ссылок, взаимосвязи с расчетными комплексами и документирования. Код не должен начинаться с цифры

**JobBase** - расчетное задание (конвейер расчетных задач)

[Задача, Решатель, параметрыЗадачи ]

Задача - код задачи или её номер в таблице TaskBase

Решатель - код решателя или его номер в таблице SolverBase

Параметры задачи - список параметров в зависимости от типа задачи:

Дискретизация - список дискретов

Механический расчет - количество рассчитываемых моделей, список загрузений (если не задано, то все загрузения). Расчет выполняется для заданного количества последних полученных в расчетном задании конечно-элементных моделей. Если количество равно 0, то рассчитываются все модели.



Мультифизический расчет - количество рассчитываемых, моделей список сценариев (если не задано, то все сценарии). Расчет выполняется для заданного количества последних полученных в расчетном задании конечно-элементных моделей. Если количество равно 0, то рассчитываются все модели.

## **SimBase** - классификация модели для задач моделирования

[Группа, Тип ]

Группы от 1 до 99 – стандартные группы, заданные в данной спецификации. Группы с номерами более 100 – нестандартные группы. Их значение определяется пользователем и может использоваться в решателях и обработчиках.

### **Стандартные группы классификации**

10 – тип модели по уровню представления

100 геометрическая модель

200 концептуальная модель

300 расчетная модель

21 – размерность пространства

100 – 1D

200 – 2D

210 XY

220 XZ

230 YZ

300 – 3D

22-размерность времени

0 – нет времени (время не рассматривается)

1 – есть время

30 – тип модели по уровню детализации

500 аналитическая (прототипная) модель

900 дискредитированная (конечно-элементная) модель

40 – тип модели по представляемому физическому явлению

- 100 – мультифизика
- 200 – строительная механика
- 210 – строительная механика, статика
- 220 – строительная механика, динамика
- 300-гидродинамика
- 400-гидродинамика с моделями горения
- 500-движение людских потоков

## Вычислительные примитивы

### **FunctionBase** - функции

[[типПеременной, типФункции, списокПараметров...]]

1. ТипПеременной
  - 1.1. 0 - произвольная (для составных функций, типы 3,4)
  - 1.2. 1 - координата X
  - 1.3. 2 - координата Y
  - 1.4. 3 - координата Z
  - 1.5. 4 - время
2. ТипФункции
  - 2.1. 1 - Константа . Параметр – значениеФункции. По умолчанию 1.0
  - 2.2. 2 - Полиномиальная функция задаваемый парами «значениеПеременной значениеФункции»
  - 2.3. 3 – сумма функций. Параметры – список суммируемых функций и их весовых коэффициентов

СписокПараметров - последовательность пар значений величины и соответствующих значений переменной, в порядке возрастания переменной

## Протоэлементы (модели Fa3D)

### **BodyBase** - объемные элементы

[[номерМатериала списокОбластей... ]]

### **ShellBase** - оболочечные элементы

[[номерСечения НомерМатериала списокГраней... ]]

### **BarBase** - стержневые элементы

[[номерСечения НомерМатериала номерОриентации списокЛиний... ]]

НомерОриентации - ссылка на таблицу LocalBase. Для стержней с ориентацией по умолчанию НомерОриентации=0

### **LocalBase** - параметры ориентации элемента

[[типОриентации параметры... ]]

типОриентации=1 - ориентация локальной оси Z задана углом поворота. Параметр - угол поворота вокруг продольной оси относительно вертикальной плоскости

типОриентации=2 - ориентация локальной оси Z задана направлением на точку. Параметр - номер точки

### Конечные элементы и узлы (модели Fe3D)

**NodeBase** - узлы конечных элементов

[N] - номера (индексы) точек

**ElementBase**

[[ ТипФормыЭлемента, ТипМоделиЭлемента, списокТочек...]]

### Шарниры

**BarHingeBase** - шарниры в стержнях

[[номерСтержня, UX1, UY1, UZ1, UX2, UY2, UZ2]]

UX1 UY1 UZ1 – коды шарнирных соединений в начале стержня

UX2 UY2 UZ2 – коды шарнирных соединений в конце стержня

Значение кода - 1 - есть шарнир вокруг заданной оси, 0 - нет шарнира

**ElementHingeBase** - шарниры и смещения в вершинах конечных элементов

[[номерКонечногоЭлемента, номерВершины, dX, dY, dZ, UX, UY, UZ]]

dX, dY, dZ– коды линейных соединений в начале стержня - смещения по осям

UX UY UZ – коды шарнирных соединений в конце стержня - повороты вокруг осей

Значение кода - 1 - есть смещение или поворот, 0 - нет смещения или поворота

### Материалы

**MaterialBase**

[ типМатериала, стиль, списокПараметров... ]

**SectionBase**

[ типСечения, стиль, списокПараметров... ]

### Воздействия

**ActionBase** - воздействия

[ типВоздействия, списокПараметров]... ]

**ActionCaseBase** - сценарии воздействий

[ типВоздействия, списокПараметров... ]

## Опоры, закрепления, краевые условия

### **SupportBase**

#### Отображение

### **PenBase**

[ R, G, B, A, типЛинии, толщина ]

ТипЛинии соответствует типам линий системных таблиц

### **BrushBase**

[ R, G, B, A, типЗаливки ]

ТипЗаливки соответствует типам заливки системных таблиц

### **StyleBase**

[ типСтиля, списокПараметров... ]

[ списокКистей... ] наружная поверхность, внутренняя поверхность, тело, ...

[ списокПерьев... ] ребра, ...

1. ТипыСтиля
  - 1.1. 0 - произвольный стиль
  - 1.2. 1 - стиль материала
  - 1.3. 2 - стиль нагрузки
  - 1.4. 3 - стиль опоры
  - 1.5. 4 - стиль шаринра

## Результаты расчетов

### **SetBase** - наборы величин

[Обработчик, Группа, Решатель, погрешность, типНабораВеличин, типФорматаНабора, типПредставленияВеличин, типФункции, Время, списокКодовВеличин...]

1. Обработчик - номер этапа в таблице расчетного задания
2. Группа - номер группы наборов, или 0 если файл не входит в какую-либо группу
3. ТипНабораВеличин
  - 3.1. 1 - величины в узлах модели
  - 3.2. 2 - величины в элементах модели
  - 3.3. 3 - величины в вершинах элементов модели
  - 3.4. 4 - величины в узлах диаграмм и эпюр
4. ТипФорматаНабора
  - 4.1. 1 - набор задан в таблице ValueSet\_N
  - 4.2. 2 - набор задан в файле ИдентификаторМодели\_ValueSet\_N.csv. Величины разделены запятыми
  - 4.3. 3 - набор задан в файле ИдентификаторМодели\_ValueSet\_N.txt. Величины разделены пробелами
  - 4.4. 4 - набор задан в бинарном файле ИдентификаторМодели\_ValueSet\_N.bin.
5. ТипПредставленияВеличин
  - 5.1. 1 - в наборе находятся величины для заданного момента времени

5.2. 2 - в наборе находятся функции заданных величин от времени

**SetGroupBase** - группы наборов величин

[Обозначение, Контейнер, путь, Описание,]

1. Обозначение - произвольное численное или текстовое обозначение, не содержащее пробелов, точек, запятых. Обозначения групп, содержащихся в другой группе, не могут быть одинаковыми. Обозначение группы не может быть "0".
2. Контейнер - номер группы, в которую вложена данная группа, или 0 если нет вложенности.
3. Путь к папке, в которой расположены файлы группы с результатами расчетов. Если не задан, то используется путь из обозначений контейнеров группы
4. Описание - краткое текстовое описание группы

**ValueSet\_N** - набор величин, где N - номер (идентификатор) таблицы с величинами

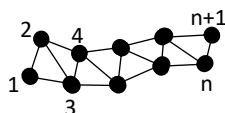
### Эпюры и диаграммы

**DNodBase** - наборы плоских координат узлов диаграмм, последовательность координат X Y узлов.

**DPointBase** - наборы пространственных координат узлов криволинейных диаграмм, последовательность координат X Y Z узлов

**DPointZBase** - наборы координат Z узлов криволинейных диаграмм, если координаты X Y пространственных точек соответствуют координатам узлов диаграммы

**DElemBase** - набор треугольных элементов диаграммы - перечень вершин, соответствующих наборам полос из треугольных элементов. Полоса - набор треугольников с общими ребрами.



У каждого последующего треугольника две вершины общие с предыдущим треугольником. То есть для следующего треугольника нужно задавать только новую вершину. В полосе задаются три вершины первого треугольника, и затем новые вершины для последующих треугольников. Полосы разделяются заданием 0 в последовательности.

Пример: набор полос 1 2 3 5 8 9 0 2 3 11 12

Будет соответствовать полосам

1 2 3 5 8 9

2 3 11 12

И соответствовать набору элементов

1 2 3      2 3 5      3 5 8      5 8 9

2 3 11      3 11 12

**DLineBase** - линии (линии уровня какой либо величины, направление склона или другие подобные линии) - набор полилиний, последовательностей узлов диаграммы, соединяемых отрезками. Полилинии разделяются заданием 0 в последовательности.

**DiagramBase** - таблица диаграмм.

ТипПротоэлемента - 1 - объемный, 2-оболочечный

Область - номер области в объемном элементе

Грань - номер грани в области или в оболочке. Если 0 - то вся область или оболочка

Узлы - номер набора узлов

Точки - номер набора точек. Если не задано, то 0

Z - номер набора координат Z. Если не задано, то 0

Преобразование - компоненты матрицы преобразования точек диаграммы в пространство модели

Величины - список наборов величин

**LevelBase** - таблица линий уровня.

Диаграмма - номер диаграммы

Набор - номер набора величин

кодВеличины - код величины

Величина - значение величины линии уровня

Линии - последовательность наборов задания линий уровня, разделяемых нулем

НаправлениеУвеличенияВеличины

1 - по часовой стрелке от линии увеличивается,

2 - по часовой стрелки от линии уменьшается.

списокЛиний - перечень номеров линий

### 9.2.3 Словари описания физических величин

В модели задаются словари описания расчетных величин, описанные в разделе "Физические величины":

**QuSysDict**

**QuMultDict**

**QuTypeDict**

**QuMarkupDict**

## 9.3 Типы и параметры расчетных элементов

### 9.3.1 Локальная система координат элементов

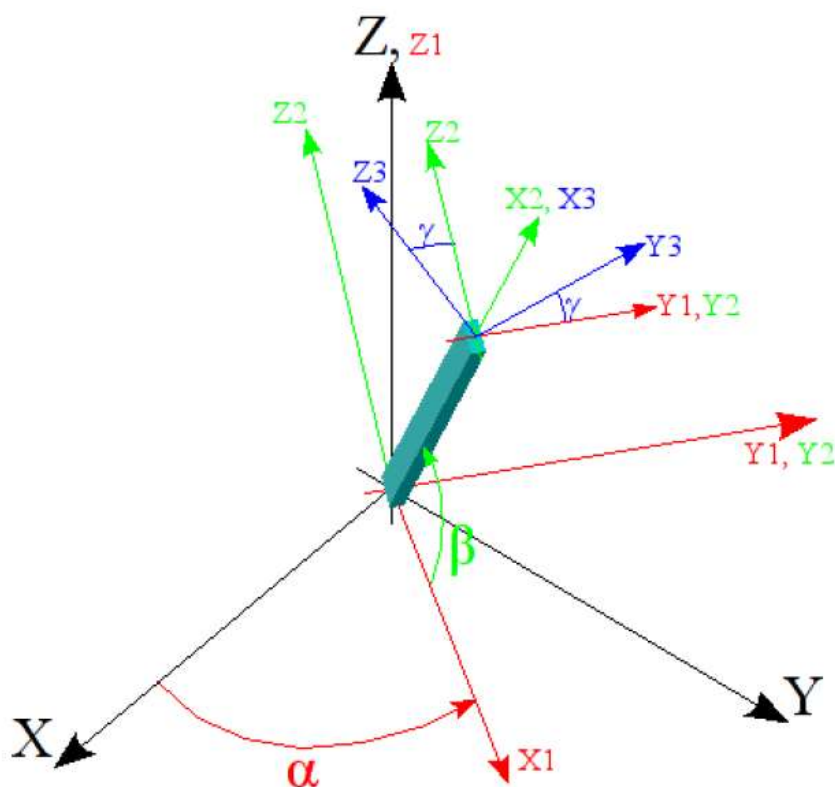
Локальная система координат элементов, описываемых тремя и более узлами, задается плоскостью, соответствующей расположению узлов, и нормалью к плоскости, направление которой соответствует порядком задания узлов.

Для стержневых элементов, задаваемых двумя узлами, по умолчанию положение локальной системы координат определяется следующими правилами:

Для вертикальных стержней - локальная ось  $Z$  ( $Z_2$  на рисунке) расположена параллельно глобальной  $X$  ( $X$  на рисунке)

Для не вертикальных стержней - локальная ось  $Z$  ( $Z_2$  на рисунке) расположена в вертикальной плоскости

Для стержневых элементов положение локальной оси ( $Z_3$  на рисунке) может быть уточнено путем задания угла поворота вокруг локальной продольной оси относительно расположения оси в вертикальной плоскости ( $Z_2$  на рисунке), или задания точки, лежащей в плоскости, описываемой векторами локальных осей  $X$  и  $Y$  ( $X_3$  и  $Z_3$  на рисунке)



### 9.3.2 Конечные элементы

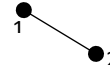
#### Произвольные конечные элементы

**100** - Произвольный конечный элемент.

Параметры конечного элемента заданы в свойствах соответствующего объекта

#### Линейные конечные элементы

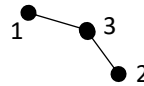
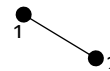
110 Прямой стержень - 2 узла. Положение локальных осей - по умолчанию



111 Прямой стержень - 2 узла. Локальная ось Z задается углом поворота

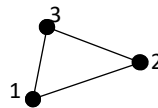


112 Прямой стержень - 2 узла. Дополнительный узел задает направление локальной системы координат

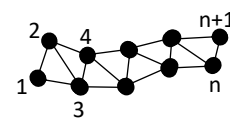


120 Криволинейный стержень, 3 узла

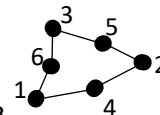
Плоские конечные элементы



210 Треугольник - 3 узла

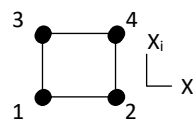


211 Полоса из треугольников (стрип) - 3 и более узлов



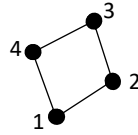
212 Криволинейная Треугольная оболочка - 6 узлов

213 Криволинейная изогнутая Треугольная оболочка - 7 узлов

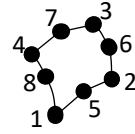


220 Прямоугольная оболочка - 4 узла

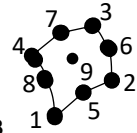




221 Четырехугольная оболочка - 4 узла

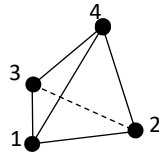


222 Криволинейная четырехугольная оболочка - 8 узлов

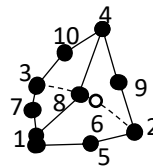


222 Криволинейная изогнутая четырехугольная оболочка - 9 узлов

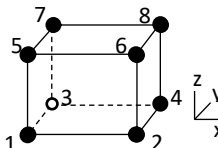
Объемные конечные элементы



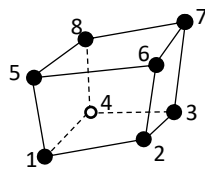
311 Пирамида - 4 узла



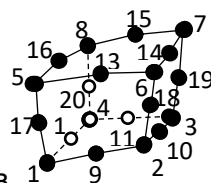
312 Криволинейная пирамида - 8 узлов



321 Прямоугольный параллелепипед - 8 узлов



322 Параллелепипед - 8 узлов



322 Параллелепипед - 16 узлов

### 9.3.3 Типы и параметры материалов

#### 1 - Произвольный материал.

Параметры материала заданы в свойствах объекта материала

#### 11 - Упругий изотропный материал

E - Модуль упругости

G - Модуль сдвига

D - Плотность

### 9.3.4 Типы и параметры сечений

#### 1 - Произвольное сечение

Параметры сечения заданы в свойствах объекта сечения

#### 110 - Набор величин

A - площадь сечения

$I_x$  - момент инерции относительно оси x сечения

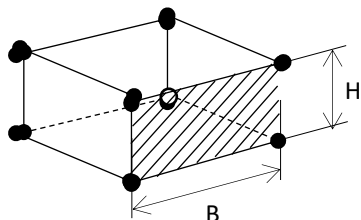
$I_y$  - момент инерции относительно оси y сечения

$I_{xy}$  - момент инерции относительно осей x и y сечения

#### 210 - Прямоугольное сечение

H - высота сечения

B - ширина сечения



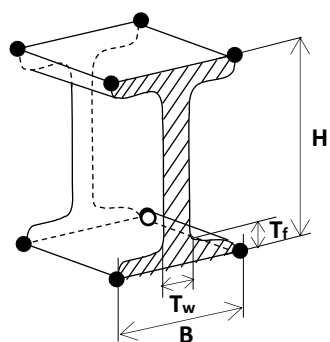
#### 310 - Двутавр

H - высота сечения

B - ширина сечения

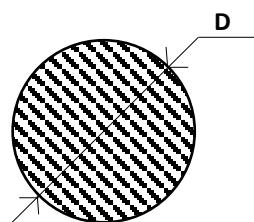
$T_f$  - толщина полки

$T_w$  - толщина стенки



#### 410 - Круг

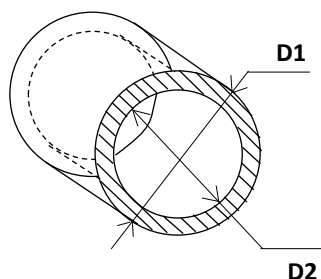
D - диаметр



#### 411 - Труба

D1 - наружный диаметр

D2 - внутренний диаметр



## 9.4 Опоры и краевые условия

### 9.4.1 Опоры

#### 11 - Узловые опоры

N номер точки узла

sX sY sZ – коды закреплений перемещений 1-есть закрепление, показывается вдоль оси, -1 – есть закрепление, показывается против оси, 0-нет закрепления, ничего не показывается

rX rY rZ – коды закреплений поворотов 1-есть закрепление, 0-нет закрепления

## 21 - Линейные опоры

L номер линии

sX sY sZ – коды закреплений перемещений 1-есть закрепление, показывается вдоль оси, -1 – есть закрепление, показывается против оси, 0-нет закрепления, ничего не показывается

rX rY rZ – коды закреплений поворотов 1-есть закрепление, 0-нет закрепления

## 9.4.2 Краевые условия

Краевые условия задаются аналогично начальным условиям, описанным в разделе "Начальные условия"

## 9.5 Нагрузки и воздействия

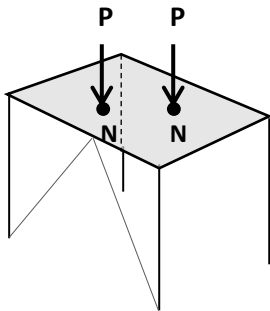
### 9.5.1 Механические нагрузки

#### 11 - Узловые силы

C - номер загрузки

N номер точки приложения нагрузки (узел или "жесткая" точка)

Px Py Pz - компоненты нагрузки по глобальным осям

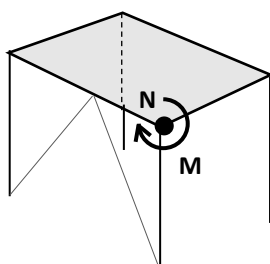


#### 12 - Узловые моменты

C - номер загрузки

N номер точки приложения нагрузки

Mx My Mz - компоненты нагрузки по глобальным осям



## 21 - Узловые массы

C - номер загрузки

N номер точки приложения нагрузки

M - масса

## 22 - Узловые направленные массы

C - номер загрузки

N номер точки приложения нагрузки

Mx My Mz - массы, направленные вдоль глобальных осей

$uM_x$   $uM_y$   $uM_z$  - моменты инерции вращения вокруг глобальных осей

## 31 - Сосредоточенные нагрузки на элемент произвольного типа

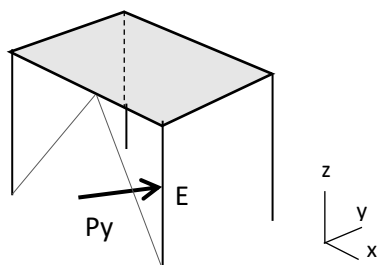
C - номер загрузки

T - тип элемента (1 - стержень, 2-оболочка, 3-тело)

E - номер элемента приложения нагрузки

X Y Z - координаты приложения нагрузки в глобальной системе

Px Py Pz - компоненты нагрузки в глобальной системе



## 41 - Смещения узлов

C - номер загрузки

N номер точки приложения нагрузки

dx dy dz - смещения узла

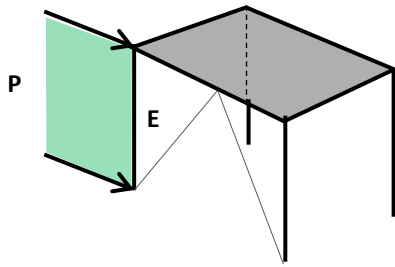
rx ry rz - повороты узла

## 111 - Линейные равномерные нагрузки на стержневой элемент

C - номер загрузки

E - номер элемента приложения нагрузки

Px Py Pz - компоненты нагрузки в глобальной системе



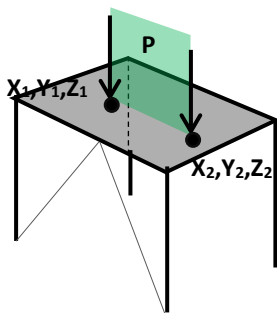
**112 - Линейные равномерные нагрузки (произвольные)**

C - номер загрузки

$P_x$   $P_y$   $P_z$  - компоненты нагрузки в глобальной системе

$X_1$   $Y_1$   $Z_1$  - координаты начала приложения нагрузки в глобальной системе

$X_2$   $Y_2$   $Z_2$  - координаты конца приложения нагрузки в глобальной системе



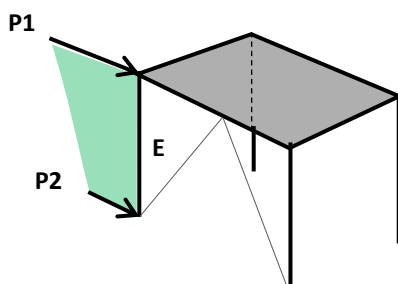
**121 - Линейные неравномерные (трапециевидные) нагрузки на стержневой элемент**

C - номер загрузки

E - номер элемента приложения нагрузки

$P_{x1}$   $P_{y1}$   $P_{z1}$  компоненты нагрузки в начальном узле в глобальной системе

$P_{x2}$   $P_{y2}$   $P_{z2}$  компоненты нагрузки в конечном узле в глобальной системе

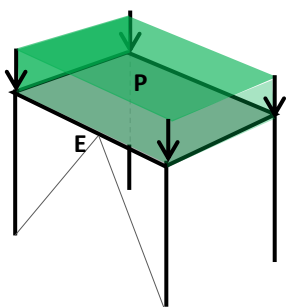


**211 - Равномерно распределенные по площади элемента нагрузки**

C - номер загрузки

E - номер оболочечного элемента приложения нагрузки

Px Py Pz - компоненты нагрузки в глобальной системе

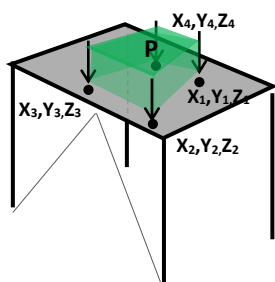


### 212 - Равномерно распределенные по площади нагрузки (произвольные)

C - номер загрузки

Px Py Pz - компоненты нагрузки в глобальной системе

x1 y1 z1 ... xn yn zn - координаты узловых точек нагрузки



## 9.5.2 Мультифизические воздействия

Дополнительные параметры воздействия (направление, сопутствующие величины, специфические параметры для конкретной расчетной задачи с учетом используемого решателя и т.п.) задаются в свойствах элементного объекта.

### 501 - равномерно распределенные воздействия на конечный элемент

номер\_загрузки

элемент

ТипВеличины

номерФункции

Величина

**511 - равномерно распределенные по длине воздействия на ребро**

номер\_загрузки

линия

ТипВеличины

номерФункции

Величина

**521 - равномерно распределенные по площади воздействия на поверхность**

номер\_загрузки

поверхность

ТипВеличины

номерФункции

Величина

**531 - равномерно распределенные по объему воздействия на тело**

номер\_загрузки

тело

ТипВеличины

номерФункции

Величина

## **9.6 Начальные условия**

**601 - Начальные условия для конечного элемента**

номер\_загрузки

элемент

ТипВеличины

номерФункции

Величина

**631 - Начальные условия для тела**

номер\_загрузки

тело

ТипВеличины



номерФункции

Величина

## 9.7 Эпюры и диаграммы

### 9.7.1 Диаграммы

Диаграммы - функции распределения каких-либо величин на плоской и или криволинейной поверхности, представляемые в виде сети треугольников, в вершинах которых заданы значения величин. Значения величин между вершинами определяется по интерполяции.

Узлы диаграммы задается в 2D координатах. Для криволинейной диаграммы также задаются трехмерные координаты поверхности.

Для диаграмм свойств элементов модели также задается функция преобразования из плоскости диаграммы в трехмерное пространство модели

Диаграммы используют следующие наборы данных

- Узлы (2D точки) диаграммы

- Точки (3D) криволинейной поверхности, соответствующие узлам диаграммы

- элементы диаграммы (треугольники)

- линии уровня

- полосы (наборы элементов между линиями уровня)

- наборы величин в узлах диаграммы

- параметрические функции величин в узлах диаграмм

- параметры преобразования из плоскости диаграммы в пространство модели

- параметры обратного преобразования из пространства модели в плоскость диаграммы

### 9.7.2 Эпюры

Эпюры - функции распределения каких-либо величин на прямом или криволинейном элменте, представляемые в виде последовательности направленных отрезков, в вершинах которых заданы значения величин. Значения величин между вершинами определяется по интерполяции.

Узлы эпюры задаются в 2D координатах. Для криволинейной диаграммы также задаются трехмерные координаты поверхности.

Для диаграмм свойств элементов модели также задается функция преобразования из плоскости диаграммы в трехмерное пространство модели

Диаграммы используют следующие наборы данных

- Узлы (2D точки) диаграммы

- Точки (3D) криволинейной поверхности, соответствующие узлам диаграммы

- элементы диаграммы (треугольники)

- линии уровня

полосы (наборы элементов между линиями уровня)

наборы величин в узлах диаграммы

параметрические функции величин в узлах диаграмм

параметры преобразования из плоскости диаграммы в пространство модели

параметры обратного преобразования из пространства модели в плоскость диаграммы

## 9.8 Физические величины

### 9.8.1 Словарь системы физических величин - QuSysDict

кодСистемнойВеличины    КодРазмерности    НазваниеВеличины    НазваниеЕдиницы    ОбозначениеЕдиницы    Описание  
ОбозначениеЕдиницы    описание

кодСистемной Величины	КодРазмерности	Название Величины	Название Единицы	Обозначение Единицы	Описание
100	БезРазмер	Z	1		
<b>//основные величины</b>					
111	Длина	L	l	метр	м
112	Масса	M	m	килограмм	кг
113	Время	t	t	секунда	с
114	СилаТока	I	I	ампер	A
115	Температура	T	T	кельвин	K
116	КоличествоВещества	N	n	моль	моль
117	СилаСвета	J	Iv	канделла	кд
<b>//дополнительные величины</b>					
221	цифроваяИнформация	B		байт	Б
<b>//производные величины</b>					
331	Угол	L/L	a	радиан	рад
332	ТелесныйУгол	L/L	omega	стерадиан	ср
341	Площадь	LL	S	квМетр	м <sup>2</sup>
342	Объем	LLL	V	кубМетр	м <sup>3</sup>
351	Плотность	M/LLL	ro	килограммНа КубМетр	кг/м <sup>3</sup>
352	ПоверхностнаяПлотность	M/LL	roA	килограммНа КвМетр	кг/м <sup>2</sup>
353	ЛинейнаяПлотность	M/L	roL	килограммНа Метр	кг/м
371	Скорость	L/t	v	метрВСекунд у	м/с
372	Ускорение	L/tt	a	метрНаСекун дуВКвадрате	м/с <sup>2</sup>
381	Сила	ML/tt	F	Ньютон	Н
382	Давление	M/Ltt	p	Паскаль	Па

383	МоментИнерции	MLL	I		кгм <sup>2</sup>
384	МоментСилы	MLL/tt	M		кгм <sup>2</sup> /с <sup>2</sup>
391	СилаНаЕдиницуДлины	M/tt	F	НьютонНаМе тр	Н/м
392	МоментСилыНаЕдиниц удлины	ML/tt	M		кгм/с <sup>2</sup>
393	СилаНаЕдиницуПлоща ди	M/ttL	F	НьютонНаКв Метр	Н/м <sup>2</sup>
394	МеханическоеНапряже ние	M/Ltt	p	Паскаль	Па
411	Энергия	MLL/tt	E	Джоуль	Дж
412	Мощность	MLL/ttt	P	Ватт	Вт
413	КоличествоТеплоты	MLL/tt	Q	Джоуль	Дж
414	ТепловойПоток	M/ttt	W	ВаттНаКвМет р	Вт/м <sup>2</sup>

## 9.8.2 Словарь кратных величин - QuMultDict

[кодСистемнойВеличины, КодКратности, НазваниеВеличины, НазваниеЕдиницы, КоэффициентА, коэффициентВ ]

Формула преобразования  $Q1 = Q * A + B$

В по умолчанию равно 0

кодСистемнойВеличины	КодКратности	НазваниеВеличины	НазваниеЕдиницы	Коэффициент А	Коэффициент В
111	00	М	м	1.0	
111	11	сантиметр	см	0.01	
111	12	миллиметр	мм	0.001	
111	13	микрон		0.000001	
111	21	километр	км	1000	
111	22	дюйм	дюйм	0.0254	
115	00	кельвин	К	1.0	
115	11	градусЦельсия	С	1.0	-273.5
115	12	градусФаренгейта	F	1.8	-460.3

## 9.8.3 Словарь типов величин QuTypeDict

Первые 3 цифры кода прикладной величины соответствуют кодСистемнойВеличины по словарю QySysDict

КодПрикладнойВеличины	масштабВеличины	описание
11101	00	Размер объекта
11102	00	Координата точек объекта
11103	12	Смещение точек объекта
11104	00	Дальность видимости
11301	00	Время процесса
11302	31	Время процесса

11501	11	Температура объекта
33101	00	Поворот
38101	00	Сила
38401	00	Изгибающий момент
38402	00	Крутящий момент
39101	00	Усилие на единицу длины
39201	00	Изгибающий момент на единицу длины
41401	21	Тепловой поток
35101	00	Плотность кислорода
35102	00	Плотность углекислого газа
35103	00	Плотность угарного газа
35104	00	Плотность хлористого водорода
35105	00	Плотность воздуха

## 9.8.4 Обозначений величин

Кодировка названия физических величин

Маска кодировки: **aeVVccccd**

1. **a** -код типа величины. Одна латинская прописная буква.
  - 1.1. s-скалярная величина.
  - 1.2. v-векторная величина в глобальной системе координат
  - 1.3. w-векторная величина в локальной системе координат элемента
  - 1.4. t-тензоры (будет описан в следующих версиях спецификации)
2. **e** -код характеристики величины. Одна латинская заглавная или прописная буква. Прописная буква используется для стандартных величин, задаваемых в программе и описанных в стандартном словаре программы. Заглавная буква используется для величин, задаваемых пользователем в пользовательском словаре в контейнере модели.
  - 2.1. U,u- величины для всех типов узлов и элементов
  - 2.2. N,n-величины для узлов
  - 2.3. E,e - величины для всех типов элементов
  - 2.4. L,l - величины для линейных элементов
  - 2.5. P,p - величины для плоских элементов
  - 2.6. V,v - величины для объемных элементов
  - 2.7. M,m - величины для моделей в целом
3. **ВВ** – обозначение величины – состоит из заглавных латинских букв. Число букв как правило не больше трех. Одинаковые векторных величин типа v и w должны иметь одинаковое обозначение.
4. **ссс**- список компонент векторной величины или элементов групп или тензоров. Последовательность символов x,y,z задает перечень компонентов векторов. Последовательность символов a- w задает перечень компонентов списков в алфавитной последовательности. "a" соответствует первому компоненту, "b"-второму, и так далее до "w"- двадцать третьему компоненту. Список может отсутствовать. Отсутствие списка компонентов обозначает «все компоненты»
5. **dd**-номер позиции в элементе, для которой задана величина-число от 0 до 99. Может отсутствовать. При отсутствии обозначает «весь элемент». Позиция как правило обозначает порядковый номер узла (вершины) элемента

Обозначение некоторых распространенных в статике конструкций величин

1. Величины для всех типов узлов и элементов
  - 1.1. suT – температура
  - 1.2. suTM-время
  - 1.3. vuGT – градиент температуры
2. Узловые величины
  - 2.1. vnC-координаты узла
  - 2.2. vnD – перемещение
  - 2.3. vnU– поворот
3. Величины в линейных (1D) элементах
  - 3.1. wIN – сила (z-продольная, x y – поперечные)
  - 3.2. wIM – изгибающий момент
4. Величины в плоских (2D) элементах
  - 4.1. wpN –продольное усилие на единицу длины элемента
  - 4.2. wPM – изгибающий момент на единицу
  - 4.3. wpQ – поперечные силы
  - 4.4. spS – сдвигающее усилие (Qxy)

### 9.8.5 Словарь обозначений величин QuMarkupDict

//Словарь обозначений величин - qustapp.txt

[ ОбозначениеВеличины , кодВеличины, описание

Компоненты векторной величины должны следовать непосредственно за описанием векторной величины. Порядок задания компонентов соответствует порядку расположения величин в наборах данных.

ОбозначениеВеличины	кодВеличины	описание
suT	11501	температура
suTM	11301	время
spQ	38101	сдвигающее усилие (Qxy)
vnC	11102	координаты узлов
vnCx	0	координата X
vnCy	0	координата Y
vnCz	0	координата Z
vnD	11103	смещение узлов
vnDx	0	смещение узла по X
vnDy	0	смещение узла по Y
vnDz	0	смещение узла по Z
vnU	33101	поворот узлов
vnUx	0	поворот узла вокруг X
vnUy	0	поворот узла вокруг Y
vnUz	0	поворот узла вокруг Z
wIN	38101	усилия в стержневых элементах
wINx	0	поперечная сила Qx в стержневых элементах
wINy	0	поперечная сила Qy в стержневых элементах
wINz	0	продольная сила N в стержневых элементах
wIM	38401	момент в стержневых элементах
wIMx	0	изгибающий момент в стержневых элементах
wIMy	0	изгибающий момент в стержневых элементах
wIMz	0	крутящий момент в стержневых элементах

wpN	39101	продольное усилие на единицу длины
wpNx	0	
wpNy	0	
wpM	39201	изгибающий момент на единицу длины
wpMx	0	изгибающий момент на единицу длины в плоских элементах
wpMy	0	изгибающий момент на единицу длины в плоских элементах
wpQ	38101	поперечные силы в плоских элементах
wpQx	0	поперечные силы в плоских элементах
wpQy	0	поперечные силы в плоских элементах

## 9.9 Значащие цифры

В числовых данных в наборах данных расчетных моделей в текстовом виде - в массивах JSON, в файлах типа CSV и TXT и т.п., должны быть указаны только значащие цифры.

Погрешности чисел в наборах данных быть указаны двумя значащими цифрами.

Данные положения приняты для результатов расчета в расчетных моделях по аналогии с требованиями приложения "Правила округления при обработке результатов измерений" ГОСТ 8.735-2011 "Методы обработки результатов измерений"

## 9.10 Аналитические модели (Fa3D)

Аналитические модели содержат структуры данных для описания прототипных элементов модели, закреплений и начальных условий, нагрузок и воздействий на прототипы.

Аналитические модели могут содержать расчетные задания и параметры преобразования в одну или несколько конечно-элементных моделей.

Аналитические модели как правило являются оригинальными расчетными моделями, то есть не образованными из других расчетных моделей.

Аналитические модели могут быть образованы из структурных моделей. Тогда в заголовке модели указываются соответствующие родительские модели и трансформация.

## 9.11 Конечно-элементные модели (Fe3D)

Конечно-элементные модели содержат структуры данных для описания конечных элементов модели, закреплений и начальных условий, нагрузок и воздействий на конечные элементы.

Конечно-элементные модели могут содержать расчетные задания.

Для конечно-элементных моделей, полученных из аналитических моделей, в заголовке модели аналитическая модель указывается родительской.

Если для получения модели использовалось расчетное задание родительской аналитической модели, то в качестве трансформации указывается данная аналитическая модель, и также после разделителя "\_" указывается номер этапа задания.

## 9.12 Расчетные задания (TaMS)

Расчетные задания содержат таблицу расчетных заданий.

Каждый этап расчетного задания является трансформацией для создаваемых моделей и результатов расчета.

Идентификатором таких трансформаций является идентификатор расчетного задания и номер этапа, с использованием разделителя "\_" (одно подчеркивание).

## 9.13 Результаты расчетов (ReMS, RaMS)

Результаты расчетов содержат наборы данных с результатами расчетов и копию таблицы расчетного задания.

В заголовке указывается родительская модель и соответствующая трансформация.

Результаты расчетов аналитических моделей группируются по расчетам полученных конечно-элементных моделей и используемым решателям

## 9.14 Комплексные модели (Se3D, Sa3D)

Комплексные модели содержат соответствующую аналитическую или конечно-элементную модель, расчетное задание, результаты расчета и/или результаты постобработки

## 9.15 Валидация и верификация моделей

### 9.15.1 Референсы и метрики

Валидация и верификация расчетных моделей

Референсы и метрики

## 9.16 Референсные модели (Fr3D)

.....

# 10. Пространственные агентские модели

## 10.1 Общее описание агентских моделей

### 10.1.1 Описание моделей

В агентских моделях содержатся элементы, которые могут взаимодействовать с другими элементами модели с течением времени или другого переменного параметра модели, и менять свое положение, форму или свойства по результатам такого взаимодействия.

Правила поведения и взаимодействия агентов задаются функциями на языке JavaScript

В данной спецификации описаны два вида агентских моделей:

**Расчетные агентские модели** (Agent Simulation Models)- модели, содержащие агентов с заданными правилами поведения. Такие модели могут проигрываться в программах агентского моделирования с записью в наборы данных модели передвижений, изменении формы и свойств элементов-агентов во времени моделирования. Расчетные модели имеют класс первичного объекта **As3D**

**Анимационные агентские модели** (Agent Animation Models)- модели, содержащие агентов с уже созданными наборами данных о передвижении, изменении формы и свойств элементов-агентов во времени моделирования. Анимационные модели имеют класс первичного **Aa3D**

### 10.1.2 Элементы агентских моделей

Агентские модели могут содержать обычные элементы с геометрическим представлением или без геометрического представления, не являющимися агентами

Имя элементов модели, являющиеся агентами, начинается с префикса «\_\_» (два подчеркивания) и имеет вид

\_\_идентификатор\_тип

«\_\_» (два подчеркивания) - префикс агента

Идентификатор - идентификатор элемента модели. Как правило число.

«\_\_» (два подчеркивания)- разделитель

Тип - обозначение типа агента. Обозначение типа может быть пустым.

В типах рекомендуется использовать кодировку типов, соответствующих примитиву AT элемента.

Например

\_\_1231

\_\_1232\_\_s

\_\_1232\_\_sv

\_\_1234\_\_svBoy

\_\_1235\_\_svGirl

\_\_1236\_\_tvChild

### 10.1.3 Примитивы агентских элементов модели

В составе элемента, описывающего агента модели, должны быть заданы примитивы с указанием типа и параметров агента

**AT** - объект, содержащий массив с кодами типизации агента



### 1 - код вида элемента

a - анимационный агент, не содержащий правил взаимодействия (по умолчанию)

s - расчетный агент, содержащий правила взаимодействия

t - агент прототип. Служит для создания анимационных и расчетных агентов. Для прототипов не создаются наборы данных

g - группа агентов. Содержит списки идентификаторов других объектов и групп.

### 2 - код геометрического представления

v - объемный агент. Геометрическое представление объекта является постоянной формой, заданной геометрическими примитивами, не зависящей от времени. Для показа агента задаются смещение и поворот от начального положения. (по умолчанию)

f - многофигурный объект. Для агента задано несколько форм, которые могут меняться

w - шагающий агент. Многофигурный агент, формы которого имитируют шаги

p - составной агент. Агент, форма которого состоит из нескольких отдельных форм, для каждой из которых может быть задано относительное смещение и поворот.

z - агент произвольного вида

### 3 - класс агента, произвольное обозначение задаваемое пользователем

Пример:

```
__1234__svBoy : { AT : [ s, v, Boy ] ; ... }
```

**E** - идентификатор элемента-прототипа

**T** - матрица преобразования прототипа для начального положения агента - для расчетных агентских моделей. Массив из трех или девяти элементов. Если в матрице три элемента - то это значения смещений по X,Y,Z геометрического представления прототипа. Если 9 элементов - то задана матрица 3x3 для трансформации геометрического представления прототипа

## 10.1.4 Наборы данных агентов

Наборы данных агентов содержат информацию о результатах моделирования и как правило являются сжатыми данными треков агентов или других форм представления результатов, полученных при выполнении моделирования в расчетных комплексах.

Пространственное положение

Набор данных агента о пространственном положении агента во времени - элемент модели с именем, имеющий вид "G\_\_идентификаторАгента". Например "G\_\_1231".

В наборе данных агента содержится массив функций компонентов пространственного положения (перемещений и поворотов) агента от времени.

Компоненты пространственного положения, содержащиеся в наборах, указываются в примитиве DP агента.

Объект **DP** содержит массив с данными

- тип функции
- номер набора свойств P1..P99
- список свойств из заданного набора.

Например DP : [ [ 1, 2, 1, 3 ], [ 1, 3, 2 ] ] ;

Будет обозначать, что в первой функции набора данных будут приведена линейная функция свойств 1 и 3 из набора P2, а во второй функции - линейная функция второго свойства из набора P3.

Стандартные компоненты перемещения агентов

- x y z - координаты положения
- dx dy dz - поворот относительно начального положения
- ux uy uz - поворот относительно начального положения
- t - компоненты матрицы преобразования 3x3 (9 чисел)

### Свойства

Набор данных агента о свойствах агента во времени - элемент модели с именем, имеющий вид "P\_\_идентификаторАгента". Например "P\_\_1231".

В наборе данных агента содержится массив функций компонентов пространственного положения (перемещений и поворотов) агента от времени.

Компоненты пространственного положения, содержащиеся в наборах, указываются в примитиве DG агента.

Объект **DG** содержит массив типов функции и списков компонентов, содержащихся в функциях набора, ассоциированного с агентом.

Например DG : [ [ 1, x, y ], [1, uz] ] ;

Будет обозначать, что в первой функции набора данных будут приведена линейная функция положения агента во времени, а во второй функции - линейная функция поворота агента относительно оси Z.

Стандартные компоненты перемещения агентов

x y z - координаты положения

dx dy dz - поворот относительно начального положения

ux uy uz - поворот относительно начального положения

t - компоненты матрицы преобразования 3x3 (9 чисел)

### 10.1.5 Треки

Треки содержат информацию о положении и свойствах агента на каждом шаге моделирования.

Трек агента - внешний файл типа CSV или TXT с именем, имеющий вид "Track\_идентификаторМодели\_идентификаторАгента". Например "AG\_111\_1231".

В файле CSV данные разделяются запятыми, в TXT данные разделяются пробелами.

Пробелы и запятые внутри строк, заключенных в кавычки, не рассматриваются как разделители.

В строках файла содержится информация в порядке возрастания номеров шагов моделирования:

Номер шага моделирования

Компоненты пространственного положения, указанные в объекте TF агента

Значения свойств, указанных в объекте TF агента

Значения моментов времени, соответствующих шагам моделирования указываются в файле с именем, имеющий вид "Track\_идентификаторМодели\_0". Например "AG\_111\_0".

Объект **TF** содержит массивы

- массив компонентов пространственного положения

- массивы наборов свойств - номер набора свойств P1..P99 и список свойств из заданного набора.

## 10.2 Анимационные агентские модели (Aa3D)

Анимационные агентские модели содержат элементы, описывающие агентов модели, и треки состояний и свойств этих элементов.

Также анимационные модели содержат элементы, не имеющие агентских свойств

## 10.3 Расчетные агентские модели (As3D)

.... Будут разработаны дополнительно

# 11. Диагностические пространственные модели

## 11.1 Общее описание диагностических моделей (De3D)

Диагностические модели содержат элементы пространственных структурных моделей с геометрическими и атрибутивными примитивами и дополнительно содержат специальные элементы

Элементы, описывающие компоненты измерительных систем - датчики, генераторы измерительных сигналов, измерительные установки

Результаты измерений

Описания событий, относящихся к диагностируемому объекту

Диагностические модели содержат словари описания физических величин:

QuSysDict

QuMultDict

QuTypeDict

QuMurkupDict

Словари описаны в соответствующем разделе описания расчетных моделей

## 11.2 Датчики

Имя элементов диагностической модели, являющиеся датчиком или прототипом датчика, начинается с префикса «\_\_» (два подчеркивания) и имеет вид

\_\_идентификатор\_вид

«\_\_» (два подчеркивания) - префикс специального элемента

Идентификатор - идентификатор элемента модели. Как правило число.

«\_\_» (два подчеркивания)- разделитель

Вид - обозначение вида элемента

s - обозначение датчика.

@s - обозначение прототипа датчика.

Например

\_\_1231\_\_s

\_\_1232\_\_@s

### 11.2.1 Типы (марки) датчиков

Под типом или маркой датчика подразумевается прототип, соответствующий понятию "тип средства измерения" в соответствии с документами по законодательной метрологии, или группа (производственная серия) датчиков с одинаковыми метрологическими характеристиками (полностью или частично).

Прототипы датчиков могут иметь прототипы и переопределять часть свойств и измерительных функций протоипа.

Прототипы датчиков имеют следующие примитивы:

**m@type** - массив с данными

код типа датчика

номер в реестре средств измерений

разработчик

название

описание

**m@sens** - массив описания первичных преобразователей (сенсоров)

L - обозначение величины

V - тип измеряемой первичным преобразователем физической величины

D - размерность величины

T - тип вычисления погрешности

1 - абсолютная погрешность

2 - относительная погрешность

3 - погрешность вычисляется функцией

E - погрешность (величина или идентификатор функции)

C - текстовая строка описание величины.

**m@calibr** - массив калибровочных коэффициентов измерительных функций

L - обозначение коэффициента

К - величина коэффициента по умолчанию.

**m@value** - массив описания измеряемых величин

L - обозначение величины. Если обозначение измеряемой величины равно обозначению величины, измеряемой первичным преобразователем, то измеренной величине датчика присваивается значение величины, измеренной первичным преобразователем с соответствующей погрешностью

V - тип измеряемой датчиком физической величины

D - размерность величины

F1 - идентификатор функции вычисления величины из показаний первичных преобразователей, калибровочных параметров, ранее вычисленных значений измеряемых величин

T - тип вычисления погрешности

1 - абсолютная погрешность

2 - относительная погрешность

3 - погрешность вычисляется функцией

E - погрешность (величина или идентификатор функции)

Измерительные функции содержатся в библиотеке функций модели - системном объекте диагностической модели **s@func**.

E - идентификатор элемента-прототипа (необязательный примитив)

## 11.2.2 Датчики

Датчики имеют следующие примитивы:

E - идентификатор элемента-прототипа

T - матрица преобразования прототипа в положение датчика. Если в матрице три элемента - то это значения смещений по X,Y,Z геометрического представления прототипа. Если 9 элементов - то задана матрица 3x3 для трансформации геометрического представления прототипа

L - обозначение (произвольная текстовая строка)

**calibr** - массив калибровочных коэффициентов измерительных функций

L - обозначение коэффициента

K - величина коэффициента.

Таблица датчиков

**SensorBase**

ID - идентификатор датчика (элемента диагностической модели)

## 11.3 Генераторы

Имя элементов диагностической модели, являющиеся генератором или прототипом генератора, начинается с префикса «\_\_» (два подчеркивания) и имеет вид

\_\_идентификатор\_вид

«\_\_» (два подчеркивания) - префикс специального элемента

Идентификатор - идентификатор элемента модели. Как правило число.

«\_\_» (два подчеркивания)- разделитель

Вид - обозначение вида элемента

g - обозначение генератора.

@g - обозначение прототипа генератора.

Например

\_\_1231\_\_g

\_\_1232\_\_@g

### 11.3.1 Типы (марки) генераторов

Прототипы генераторов воздействий на диагностируемый объект имеют следующие примитивы:

**m@gen** - массив описания первичных преобразователей (сенсоров)

L - обозначение величины

V - тип генерируемой физической величины

D - размерность величины

T - тип вычисления погрешности

1 - абсолютная погрешность

2 - относительная погрешность

3 - погрешность вычисляется функцией

E - погрешность (величина или идентификатор функции)

C - текстовая строка описание величины.

**m@calibr** - массив калибровочных коэффициентов функций генерации величин

L - обозначение коэффициента

К - величина коэффициента по умолчанию.

**m@value** - массив описания генерируемых величин

L - обозначение величины. Если обозначение измеряемой величины равно обозначению величины, измеряемой первичным преобразователем, то измеренной величине датчика присваивается значение величины, измеренной первичным преобразователем с соответствующей погрешностью

V - тип генерируемой датчиком физической величины

D - размерность величины

F1 - идентификатор функции вычисления величины из показаний других генерируемых величин

T - тип вычисления погрешности

1 - абсолютная погрешность

2 - относительная погрешность

3 - погрешность вычисляется функцией

E - погрешность (величина или идентификатор функции)

Измерительные функции содержатся в библиотеке функций модели - системном объекте диагностической модели **s@func**.

E - идентификатор элемента-прототипа (необязательный примитив)

### 11.3.2 Генераторы

Датчики имеют следующие примитивы:

E - идентификатор элемента-прототипа

T - матрица преобразования прототипа в положение датчика. Если в матрице три элемента - то это значения смещений по X,Y,Z геометрического представления прототипа. Если 9 элементов - то задана матрица 3x3 для трансформации геометрического представления прототипа

L - обозначение (произвольная текстовая строка)

**calibr** - массив калибровочных коэффициентов функций генерации

L - обозначение коэффициента

K - величина коэффициента.

## 11.4 Измерительные установки

Измерительные установки - набор датчиков и генераторов, установленных на фрагменте диагностируемого объекта, для измерения свойств объекта в месте установки.



Имя элементов диагностической модели, являющиеся измерительной установкой или прототипом установки, начинается с префикса «\_\_» (два подчеркивания) и имеет вид

`__идентификатор_вид`

«\_\_» (два подчеркивания) - префикс специального элемента

Идентификатор - идентификатор элемента модели. Как правило число.

«\_\_» (два подчеркивания)- разделитель

Вид - обозначение вида элемента

m - обозначение измерительной установки.

@m - обозначение прототипа измерительной установки.

Например

`__1231__m`

`__1232__@m`

### 11.4.1 Типы (марки) измерительных установок

Прототипы измерительных установок имеют следующие примитивы:

**m@detect** - массив описания датчиков

L - обозначение величины в измерительной установке

N - номер датчика (генератора) в установке (таблица датчиков задается в описании экземпляра установки)

SL - номер или обозначение величины в датчике (генераторе)

D - размерность величины

T - тип вычисления погрешности

1 - абсолютная погрешность

2 - относительная погрешность

3 - погрешность вычисляется функцией

E - погрешность (величина или идентификатор функции)

C - текстовая строка описание величины.

**m@param** - массив параметров измерительной установки

L - обозначение параметра измерительной установки

D - размерность величины параметра

P - значение параметра по умолчанию (может переопределяться для экземпляра вычислительной установки)

T - тип вычисления погрешности

1 - абсолютная погрешность

2 - относительная погрешность

3 - погрешность вычисляется функцией

E - погрешность (величина или идентификатор функции)

C - текстовая строка описание величины.

**m@value** - массив описания измеряемых и генерируемых величин

L - обозначение величины. Если обозначение измеряемой величины равно обозначению величины, измеряемой датчиком, то измеренной величине измерительной установки присваивается значение величины, измеренной датчиком с соответствующей погрешностью

V - тип измеряемой датчиком физической величины

D - размерность величины

F1 - идентификатор функции вычисления величины из показаний первичных преобразователей, калибровочных параметров, ранее вычисленных значений измеряемых величин

T - тип вычисления погрешности

1 - абсолютная погрешность

2 - относительная погрешность

3 - погрешность вычисляется функцией

E - погрешность (величина или идентификатор функции)

Измерительные функции содержатся в библиотеке функций модели - системном объекте диагностической модели **s@func**.

**E** - идентификатор элемента-прототипа (необязательный примитив)

Пример описания прототипа измерительной установки:

Измерительная установка - два тензометрических датчика, установленных на боковые поверхности стержня прямоугольного сечения шириной В и высотой Н.

Для измерений используются следующие обозначения, задаваемые в m\$detect:

e1 - относительное удлинение "e" датчика 1

e2 - относительное удлинение "e" датчика 2

Измерительная установка имеет следующие параметры, задаваемые в m\$param

E - модуль упругости материала

B - ширина сечения

H - высота сечения

Измеряемые установкой величины, заданные в m@value:

N - продольная сила

Функция F1 для вычисления значения продольной силы N будет иметь вид:

[ "e1, e2, B, H" , "var N = (e1+e2)\*B\*H/2" ]

## 11.4.2 Измерительные установки

Измерительные установки имеют следующие примитивы

**E** - идентификатор элемента-прототипа. Если прототип не задан, должны быть заданы примитивы m@detect, m@param, m@value.

**L** - обозначение (произвольная текстовая строка)

**Detect** - массив идентификаторов датчиков измерительной установки

**Param** - массив параметров измерительной установки

Пример описания измерительной установки

## 11.5 Трандьюсеры

Трандьюсеры - обобщающее обозначение датчиков, генераторов и измерительных установок.

Все трандьюсеры диагностической модели указываются в системной таблице модели TranduserBase.

### TranduserBase

ID - идентификатор элемента-трандьюсера - датчика, генератора или измерительной установки

## 11.6 Результаты измерений

Результаты измерений, как полученных непосредственно при измерениях, так и обработанных, описываются в системных таблицах диагностической модели и содержатся в наборах данных в объектах модели или во внешних файлах

**ProcBase** - системная таблица модели с описанием обработчиков наборов измеренных величин

[ кодОбработчика, названиеОбработчика, списокПараметровОбработки ]

Для измеренных данных без обработки - кодОбработчика равен "0".

**SetBase** - системная таблица с описанием наборов измеренных величин

[измеритель, Обработчик, типФорматаНабора, типПредставленияВремени, списокИзмеренныхВеличин...]

1. Измеритель - номер измерителя в таблице измерителей
2. ТипПредставленияВремени
  - 2.1. 1 - в первом элементе набора указывается дата, во втором элементе - время дня в секундах
  - 2.2. 2 - в первом элементе набора указывается время в секундах, дата не указывается
3. ТипФорматаНабора
  - 3.1. 1 - набор задан в таблицах ValueSet\_N\_i
  - 3.2. 2 - набор задан в файле ИдентификаторМодели\_ValueSet\_N\_i.csv. Величины разделены запятыми
  - 3.3. 3 - набор задан в файле ИдентификаторМодели\_ValueSet\_N\_i.txt. Величины разделены пробелами
  - 3.4. 4 - набор задан в бинарном файле ИдентификаторМодели\_ValueSet\_N\_i.bin.

**ValueSet\_N\_i** - наборы измеренных величин.

N - номер (идентификатор) таблицы (группы таблиц) с величинами,

i - индекс таблицы в группе.

В наборах измеренных величин содержатся массив записей, содержащих время измерения (генерации) и значения измеренных (сгенерированных) величин.

## 11.7 События

События - свойства моделей, описывающие какое либо качественное состояние диагностической модели в соответствующий момент или период времени.

События задаются в наборах событий.

**EventBase** - системная таблица с описанием наборов событий

[обозначениеНабора, типНабора, типФорматаНабора, описание ]

ТипФорматаНабора

- 1 - набор задан в таблицах EventSet\_N\_i
- 2 - набор задан в файле ИдентификаторМодели\_EventSet\_N\_i.csv. Величины разделены запятыми
- 3 - набор задан в файле ИдентификаторМодели\_EventSet\_N\_i.txt. Величины разделены пробелами
- 4 - набор задан в бинарном файле ИдентификаторМодели\_EventSet\_N\_i.bin.

**EventSet\_N\_i** - наборы событий.

N - номер (идентификатор) набора (группы наборов) событий,

i - индекс набора в группе.

В наборах событий содержится массив записей, содержащих:

Дата начала события

Время начала события

Дата окончания события (пусто если совпадает с датой начала)

Время окончания события (пусто если совпадает с временем начала)

Код типа события

Обозначение события

Идентификатор элемента (группы элементов), с которым связано событие

Краткое описание события

Развернутое описание события

## 12. Проекции, чертежи и абрисы

### 12.1 Проекции (Pm2D)

Проекции - плоские двухмерные модели, полученные с помощью преобразования из трехмерных пространственных (объемных или плоских) моделей.

Для проекций всегда однозначно определена и задана родительская модель, из которой получена проекция, и определена и задана соответствующая проекционная трансформация.

Имена элементов плоских проекций имеют кодировку с использованием префиксов, соответствующую кодировке пространственных моделей:

«\_» (одно подчеркивание) - объекты модели, имеющие геометрическое представление

«\$» - объекты модели, не имеющие геометрического представления

«\_ \$» - прототипы объектов модели с геометрическим представлением

"\$\$" - группы объектов

"\$\_" - наборы свойств и величин

В плоских моделях задается системный элемент

**TypeBase** - массив с перечнем названий используемых классификаторов

Аналогично пространственным моделям в элементах плоских моделях содержатся примитивы:

**G1..G99** - для описания уровней геометрической проработки

**P1...P99, PP, PN** - для описания наборов свойств и уровней атрибутивной проработки

**TT** - массив типов объектов по разным системам классификации. Если в массиве указано число, то это ссылка на соответствующую таблицу типов классификатора.

**TB** - массив принадлежности элемента - массив, содержащий номер этажа и номера блоков в таблице блоков

**S** – стиль - массив из двух или трех чисел – номер кисти, номер пера, номер материала (если есть). Если номера равны нулю, то используется кисть или перо по умолчанию.

**B** – bounding box – оболочка – массив координат ограничивающего прямоугольника из четырех чисел: [x1,x2,y1,y2]

**L** - список элементов. ["Объект",...] - массив имен объектов, входящих в группу

**G** - список групп элементов. ["Группа",...] - массив имен групп объектов, входящих в группу

**E** - идентификатор элемента-прототипа

**T** - матрица преобразования прототипа.- имеет меньшую размерность, чем в пространственных моделях. Матрица преобразования 2D - массив из двух или четырех элементов. Если в матрице два элемента - то это значения смещений по X,Y геометрического представления прототипа . Если 4 элементов - то задана матрица 2x2 для трансформации геометрического представления прототипа

## 12.2 Чертежи (Pd2D)

### 12.2.1 Состав чертежей

Чертежи являются плоскими двумерными моделями, содержащими следующие компоненты:

**Модельные компоненты** (имеют взаимосвязь с элементами родительских моделей)

- масштабные изображения проекций моделей (планы, сечения, узлы и т.п.)
- таблицы и спецификации с отображением свойств проекций модели
- текстовое отображение инфодокумента и элементов инфонаборов

**Компоненты оформления** (не имеют родительских моделей или не имеют взаимосвязи с элементами родительских моделей)

- графические элементы оформления
- поясняющие надписи

### 12.2.2 Элементы оформления

В чертежах элементы, являющиеся элементами оформления (рамки, надписи и т.п.), не являющиеся элементами проекций моделей или инфодокументов, (то есть не имеющие взаимосвязи с элементами родительских моделей), являются специальными элементами, их имена начинаются с префикса двойное подчеркивание "\_\_".

Элементы оформления содержат двумерные геометрические примитивы

### 12.2.3 Области чертежа

Логически-связанные компоненты чертежа группируются и располагаются в неперекрывающихся областях, которым соответствуют специальные элементы модели чертежа.

Имя объекта области имеет вид:

\_\_идентификатор\_\_vp

Где \_\_vp - обозначение типа специального объекта ( ViewPort )

В элементах областей содержатся примитивы:

Геометрические примитивы знака (марки) обозначения области

**B** – bounding box – оболочка – массив координат ограничивающего прямоугольника из четырех чисел: [x1,x2,y1,y2]

**BB** - массив координатами точек полигона границы области (если не задана граница прямоугольной оболочки области B) . Направление задания - по часовой стрелке.

**L** - список элементов. ["Объект",...] - массив имен элементов, входящих в группу

**P1...P99, PP, PN** - описание наборов свойств и уровней атрибутивной проработки

Свойства области могут содержать специальные свойства с указанием типа области, её названия, номера., с указанием проекции модели, трансформации и т.п. информации.

### 12.3 Абрисы (Pa2D)

Абрисы - произвольные плоские двумерные модели, не являющиеся производными из каких либо трехмерных моделей путем проецирования с последующими масштабными преобразованиями или другим аналогичным образом.

Абрисы составляются путем непосредственного рисования человеком, создаются программным обеспечением в качестве иллюстративной графики при визуализации каких либо данных.

### 12.4 Параметрические модели (Mp2D, Md2D, Ma2D)

Проекция и чертежи могут содержать дискретные и переменные параметры, от которых зависит их представление в графическом виде.

Состав и описание параметров двумерных параметрических моделей соответствует структурам данных, используемым для пространственных параметрических моделей.

Двумерные параметрические модели имеют следующие обозначения типов:

**Mp2D** - параметрические проекции

**Md2D** - параметрические чертежи

**Ma3D** - параметрические абрисы

### 12.5 Геометрические примитивы 2D моделей

Точка в массивах представляется парой координат  $x$   $y$ , или номером точки в массиве точек, в зависимости от используемого типа позиционирования, задаваемого постфиксом "\_".

Нормализованная геометрия - базовый минимальный набор команд, соответствующий набору примитивов, наиболее распространенных графическим форматам и библиотекам - SVG и т.п.

Если в примитивах перо или кисть задано равным 0, то соответствующие ребра или заливки не рисуются. То есть круг при задании нулевой кисти будет являться окружностью.

В именах примитивов может задаваться числовой индекс (см. раздел 6.3.2 "Геометрические примитивы). При задании числового индекса, он может интерпретироваться программами просмотра как порядковый номер для отрисовки примитивов в элементе модели, если не задан порядок рисования примитивов

## 1. Индекс глубины

1.1. **DI** – (depth index) – индекс глубины элемента. Целое число, задающее уровень расположения элемента на изображении. Элементы с БОльшим индексом глубины отрисовываются поверх объектов с меньшим индексом. По умолчанию - 0

1.2. **DO**- (drawing order) - порядок рисования примитивов - массив имен примитивов

## 2. Контур

2.1. **CC** - контур элемента - массив массивов с точками, задающими замкнутые полигоны. Первый элемент в полигоне - тип полигона

2.1.1. 1-наружный контур. Задается по часовой стрелке.

2.1.2. 0-контур отверстия внутри ранее заданного наружного контура. Контур отверстия задается против часовой стрелки.

## 3. Нормализованная двумерная геометрия

3.1. **pP, pP\_** - "пути", соответствующие нормализованному "path" стандарта SVG 1.2. В массиве задаются номера команд и координаты точек.

3.1.1. Структура pP: [[перо, кисть, [команда\_1, параметр\_команды\_11, параметр\_команды\_12, ..., команда\_2, параметр\_команды\_21, параметр\_команды\_22, ... ], ...], ...]

3.1.2. Структура pP\_: аналогично pP, но вместо пар координат  $x$   $y$  - номер точки  $n$

3.1.3. Номера команд "path" SVG:

3.1.3.1. 1 - L = lineto (линия до заданной точки)

3.1.3.2. 2 - L - = lineto (линия до заданной точки) (дублирование команды 1)

3.1.3.3. 3 – C = curveto (кубическая кривая Безье)

3.1.3.4. 4 - S = smooth curveto (продолжение кубической кривой Безье)

3.1.3.5. 5 - Q = quadratic Bézier curve (квадратичная кривая Безье)

3.1.3.6. 6 - T = smooth quadratic Bézier curveto (продолжение квадратичной кривой Безье)

3.1.3.7. 7 - M = moveto (перенос в заданную точку)

3.1.3.8. 8 - Z = closepath (замыкание пути)

3.1.4. Параметры команд:

3.1.4.1. M = [x, y], где (x, y) - точка, в которую перемещается путь.

3.1.4.2. L = [x, y], где (x, y) - точка, в которой заканчивается линия.

3.1.4.3. C = [x1 y1, x2 y2, x y], где (x,y) - точка, в которой заканчивается кривая, (x1,y1) - контрольная точка для начала кривой, (x2,y2) - контрольная точка для конца кривой.

3.1.4.4. S = [x2 y2, x y], где (x,y) - точка, в которой заканчивается кривая, (x2,y2) - контрольная точка для конца кривой.

3.1.4.5. Q = [x1 y1, x y], где (x,y) - точка, в которой заканчивается кривая, (x1,y1) - контрольная точка кривой.

3.1.4.6. T = [x, y], где (x,y) - точка, в которой заканчивается кривая.



- 3.1.4.7.  $Z$  = отсутствуют.
- 3.2. **pC, pC<sub>n</sub>** - круг.
- 3.2.1. Структура "pC": [[перо1, кисть1, [центр\_x\_11, центр\_y\_11, радиус\_11], [центр\_x\_12, центр\_y\_12, радиус\_12], ...], [перо2, кисть2, [центр\_x\_21, центр\_y\_21, радиус\_21], [центр\_x\_22, центр\_y\_22, радиус\_22], ...], ... ]
- 3.2.2. Структура pC<sub>n</sub>: аналогично pC, но вместо координат угла  $x$   $y$  – номер точки  $n$
- 3.3. **pR, pR<sub>n</sub>** - прямоугольник, ортогональный осям координат.
- 3.3.1. Структура pR: [[перо1, кисть1, [угол\_x\_11, угол\_y\_11, ширина\_11, высота\_11], [угол\_x\_12, угол\_y\_12, ширина\_12, высота\_12], ...], [перо2, кисть2, [угол\_x\_21, угол\_y\_21, ширина\_21, высота\_21], [угол\_x\_22, угол\_y\_22, ширина\_22, высота\_22], ...], ... ]
- 3.3.2. Структура pR<sub>n</sub>: аналогично pR, но вместо координат угла  $x$   $y$  – номер точки  $n$
- 3.4. **pT, pT<sub>n</sub>** - текст.
- 3.4.1. [ [ выравнивание, цвет, точкаНачала, шрифтСтиль, "Текст"], [...], ... ]
- 3.4.1.1. Выравнивание - 1-по левому краю, 2-по правому краю, 3-по центру и т.п.
- 3.4.1.2. Цвет - RGBA
- 3.5. **pt, pt<sub>n</sub>** - ссылочный текст.
- 3.5.1. [ [ выравнивание, цвет, точкаНачала, векторВдоль, шрифтСтиль, текстовыйНабор, текстовыйОбъект], [...], ... ]
- 3.5.1.1. Выравнивание - 1-по левому краю, 2-по правому краю, 3-по центру и т.п.
- 3.5.1.2. Цвет - RGBA
- 3.6. **pS, pS<sub>n</sub>** - горизонтальный текст.
- 3.6.1. [ [ выравнивание, цвет, точкаНачала, шрифтСтиль, "Текст"], [...], ... ]
- 3.6.1.1. Выравнивание - 1-по левому краю, 2-по правому краю, 3-по центру и т.п.
- 3.6.1.2. Цвет - RGBA
- 3.7. **ps, ps<sub>n</sub>** - горизонтальный ссылочный текст.
- 3.7.1. [ [ выравнивание, цвет, точкаНачала, шрифтСтиль, текстовыйНабор, текстовыйОбъект], [...], ... ]
- 3.7.1.1. Выравнивание - 1-по левому краю, 2-по правому краю, 3-по центру и т.п.
- 3.7.1.2. Цвет - RGBA
- 3.7.1.3.
- 3.8. **pZ, pZ<sub>n</sub>** - растровое изображение. Изображение при показе масштабируется для отображения в заданной области
- 3.8.1. [[точкаНачала, высота, ширина, уголПоворота, путьКФайлуРастровогоИзображения ] ]

## 12.6 Внешняя геометрия SVG

Каждому уровню геометрической проработки модели БимДамп соответствует отдельный файл SVG, указанный в заголовке модели.

Для компонентного взаимосвязывания в объекте <svg> файла указываются атрибуты:

BimML=1

GUID = глобальный идентификатор модели

LKID = локальный ключ-идентификатор модели

ModelName = название модели

LODG = номер уровня геометрической проработки

SF = масштабный фактор - коэффициент перевода координат SVG в координаты модели или пространства чертежа

Для обеспечения эффективности и интерактивности отображения графики в браузерах, структура организации данных в SVG отличается от структуры организации данных в БимДамп.

В БимДамп - список элементов различного типа, последовательность задания элементов роли не играет.

В SVG - иерархическая древовидная структура из вложенных групп элементов, отсортированных по индексу глубины, с заданием для групп различных атрибутов отображения и интерактивности.

Каждому элементу БимДамп соответствует группа SVG, вложенная в группу, задающую тип (вид) элемента БимДамп.

В любой группе SVG могут быть группы, предназначенные для задания общих атрибутов рисования и интерактивности вложенных элементов.

Группы, соответствующие элементам БимДамп, определяются по атрибутам, имеющим заданные значения:

Группа элементов модели БимДамп одного вида (типа) элементов

Атрибут **etype** - код типа элементов БимДамп в группе

**model** - обычные элементы, имеющие атомарную взаимосвязь с элементами связанных моделей (соответствуют префиксу имени элемента "\_")

**special** - специальные элементы, не имеющие атомарную взаимосвязь с элементами связанных моделей (соответствуют префиксу имени элемента "\_\_")

Элементы модели БимДамп

Атрибут **id** - идентификатор элемента БимДамп (как правило десятичное целое число, без префикса)

Дерево элементов SVG:

1. **<g etype=special ... атрибуты отрисовки и интерактивности>** - группа, содержащая специальные элементы БимДамп
  - 1.1. **<g ... атрибуты отрисовки>** - группы по атрибутам отрисовки
    - 1.1.1. **<g id=... атрибуты отрисовки>** - элементы бимдамп (специальные)
      - 1.1.1.1. **<g ... атрибуты отрисовки >** - группы по атрибутам отрисовки
        - 1.1.1.1.1. Элементы SVG, соответствующие геометрическим примитивам элементов БимДамп
2. **<g etype=model ...атрибуты отрисовки и интерактивности>** - группа, содержащая модельные элементы БимДамп
  - 2.1.1. **<g ... атрибуты отрисовки>** - группы по атрибутам отрисовки
    - 2.1.1.1. **<g id=... атрибуты отрисовки>** - элементы бимдамп (модельные)
      - 2.1.1.1.1. **<g ... атрибуты отрисовки >** - группы по атрибутам отрисовки
        - 2.1.1.1.1.1. Элементы SVG, соответствующие геометрическим примитивам

Порядок элементов в SVG должен формироваться с учетом индекса глубины в модели БимДамп - сначала идут объекты с меньшим индексом

Штриховки геометрических примитивов элементов моделей БимДамп должны преобразовываться в соответствующие объекты с использованием штриховок <hatch> формата SVG

## 13. Преобразования моделей

### 13.1 Виды и Проекции

**Виды модели** (view) - преобразования (transform) одной модели в другую модель, являющуюся представлением подмножества объектов исходной модели, при котором сохраняется взаимно-однозначное соответствие между объектами исходной и преобразованной модели. То есть в программах с возможностью редактирования моделей, при изменении свойств объекта модели изменяются свойства видов модели, и аналогично при изменении свойств объекта в каком-либо виде модели, меняются свойства в модели и других видах.

**Проекция модели** (projection) - преобразование одной модели в другую модель, при котором теряется взаимно-однозначное соответствие между объектами модели и объектами проекции, и/или их свойствами.

### 13.2 Камеры и снимки

Для заданий трансформаций пространственных моделей используются указанные ниже объекты, задающие параметры камер, параметры съемки и параметры снимков.

**CameraType** - тип камеры

**Name** – название типа камеры

**HFOV** –диапазон горизонтального угла - массив начального и конечного значения

**FOF** – диапазон вертикальный угол вида- массив начального и конечного значения

**FocLen** – диапазон расстояния до фокусной плоскости - массив начального и конечного значения

**Lens** - оптическое искажение – массив функций параметров k1,k2,p1,p2 класса vtkWarpLens от значений фокусного расстояния FocLen

**Raster** – массив возможных параметров растра снимков [[HPix, VPix]] – количество пикселей по горизонтали и вертикали. Для векторных преобразований – [0,0] или [0]

**CameraView** - параметры съемки

**Pos** – позиция камеры

**ProjDir** - направление оси камеры - Projection Direction

**ViewUp** – вектор, задающий поворот камеры относительно вертикальной оси

**FocLen**– расстояние до фокусной плоскости - Focal Length

**HFOV** –горизонтальный угол вида

**VFOF** – вертикальный угол вида

**Lens** - оптическое искажение – параметры [k1,k2,p1,p2] класса vtkWarpLens

**Info** – свойства, вычисляемые из параметров камеры:

**FocPoint** – фокальная точка

**H** –высота кадра

**W** -ширина кадра

**DistCor** [ .... ] – параметры для коррекции оптического искажения

**CameraShot** - параметры снимка

**CameraTypeName** - имя типа камеры

**CameraID** - идентификатор камеры

**Time** - время съемки

**Raster** –параметры раstra снимка [[HPix, VPix]] – количество пикселей по горизонтали и вертикали. Для векторных преобразований – [0,0] или [0]

**ImgType** - тип изображения

**ImgFile** - файл с изображением

**Pos** – позиция камеры

**ProjDir** - направление оси камеры - Projection Direction

**ViewUp** – вектор, задающий поворот камеры относительно вертикальной оси

**FocLen**– расстояние до фокусной плоскости - Focal Length

**HFOV** –горизонтальный угол вида

**VFOF** – вертикальный угол вида

**Lens** - оптическое искажение – параметры [k1,k2,p1,p2] класса vtkWarpLens

**Info** – свойства, вычисляемые из параметров камеры:

**FocPoint** – фокальная точка

**H** –высота кадра

**W** -ширина кадра

**DistCor** [ .... ] – параметры для коррекции оптического искажения

Для описания видеосъемки используются объекты VideoView и VideoShot.

**VideoView** - параметры подвижной съемки

**Duration** - продолжительность периода съемки

**Raster** –параметры раstra снимков [HPix, VPix] – количество пикселей по горизонтали и вертикали. Для векторных преобразований – [0,0] или [0]

Приведенные ниже параметры являются функциями от относительного времени - времени от начала съемки:

**Pos** – позиция камеры

**ProjDir** - направление оси камеры

**ViewUp** – вектор, задающий поворот камеры относительно вертикальной оси

**FocLen**– расстояние до фокусной плоскости

**HFOV** –горизонтальный угол вида

**VFOF** – вертикальный угол вида

**Lens** - оптическое искажение – параметры [k1,k2,p1,p2] класса `vtkWarpLens`

**VideoShot** - параметры видеозаписи

**CameraTypeName** - имя типа камеры

**CameraID** - идентификатор камеры

**Time** - время начала съемки

**Raster** –параметры раstra кадра [[HPix, VPix]] – количество пикселей по горизонтали и вертикали. Для векторных преобразований – [0,0] или [0]

**ImgType** - тип изображения

**Frame** - массив с моментами времени кадров

**FrameFolder** - папка с файлами кадров. Имя файла должно начинаться с идентификатора видеосъемки, за которым следует номер кадра. Разделитель полей - знак подчеркивания "\_".

**VideoFile** - путь к видеофайлу

**AudioFile** - путь к аудиофайлу

Функции от относительного времени съемки:

**Pos** – позиция камеры

**ProjDir** - направление оси камеры - Projection Direction

**ViewUp** – вектор, задающий поворот камеры относительно вертикальной оси

**FocLen**– расстояние до фокусной плоскости - Focal Length

**HFOV** –горизонтальный угол вида

**VFOF** – вертикальный угол вида

**Lens** - оптическое искажение – параметры [k1,k2,p1,p2]

### 13.3 Пространственные преобразования (Tm3D)

Параметры пространственного преобразования задаются в заголовке `s$header` преобразования

1. TType - тип преобразования
  - 1.1. 1 - Сечение - только вид среза объектов заданной плоскостью (глубина области преобразования принимается 0)
  - 1.2. 2 - Разрез - вид среза и контуров объектов, лежащих в области преобразования
  - 1.3. 3- вся модель
2. Plane - плоскость ( для разрезов, сечений и т.п.)
3. Origin - точка, задающая совместно с нормалью плоскость
4. Normal – направление нормали – задает направления вида или отсекаемую часть
5. Points - координаты трех точек, задающих плоскость. Порядок точек задает направление нормали. Данная опция является альтернативой задания плоскости через одну точку и нормаль
6. Type - расположение (справочная информация)
  - 6.1. 1 - горизонтальная плоскость
  - 6.2. 2 - вертикальная XZ
  - 6.3. 3 - вертикальная YZ
  - 6.4. 4 - наклонная
7. Depth - глубина области преобразования
8. Box - координаты точек, задающих область преобразования
9. Filters - массив идентификаторов фильтров
10. LODg - номер используемого уровня проработки геометрического представления (по умолчанию - 1)

## 13.4 Проекционные преобразования (Tr3D, Tr2D)

Преобразования задаются следующими свойствами в заголовке объекта s\$header:

**TargetType** - тип модели результата

**TransformType** - тип преобразования

1 - Сечение - только вид среза объектов заданной плоскостью (глубина области преобразования принимается 0)

2 - Разрез - вид среза и контуров объектов, лежащих в области преобразования

3 - Фасад - внешний вид объектов области преобразования со стороны заданной плоскости (плоскость не сечет никаких объектов в области преобразования) – (может быть представлен снимком далеко расположенной камерой)

4-Камера –снимок заданной камерой на заданную плоскость

5-РастроваяКамера - растровый снимок заданной камерой

**Plane** - плоскость 2D модели - массив свойств

**Loc** - расположение

1 - горизонтальная плоскость

2 - вертикальная XZ

3 - вертикальная YZ

4 - наклонная

**Points** - координаты точек, задающих плоскость.

**Normal** – направление нормали – задает направления вида или отсекаемую часть

**Depth** - глубина области преобразования

**Box** - координаты точек, задающих область преобразования

**CameraView** - параметры съемки

**Filters** - массив идентификаторов используемых фильтров [ фильтров

**LODg** - номер проработки геометрического представления (по умолчанию - 1)

## 13.5 Чертежные преобразования (Td2D)

Чертежные преобразования задают размещение проекций моделей на плоскости документа, и расположение поясняющих надписей и графических элементов.

В заголовке объекта задаются свойства:

**SheetSize** - размер области документа

Поясняющие надписи и графические элементы задаются элементами, содержащими графические примитивами

Области вставки на чертеж проекций модели задаются специальными объектами с именами вида:

### **\_\_идентификатор\_\_vp**

Область имеет следующие примитивы:

**Port** - область вставки - массив из 4 чисел, задающих координаты левого верхнего и правого нижнего углов области вставки. В области вставки располагаются обозначение (марка) области и масштабное изображение проекции модели

**Model** - список идентификаторов проекций, моделей и трансформаций. Если в списке только один идентификатор, то это должен быть идентификатор плоской модели проекции, вставляемой в область.

**Titul** - массив с списком строк, содержащих название области

**Scale** - массив из двух чисел - числителя и знаменателя масштаба, например [1, 100]. Если числитель равен 0, то масштаб вычисляется автоматически.

Области вставки на чертеж графических задаются специальными объектами с именами вида:

### **\_\_идентификатор\_\_dp**

Область имеет следующие примитивы:

**Port** - область вставки - массив из 4 чисел, задающих координаты левого верхнего и правого нижнего углов области вставки. В области вставки располагаются обозначение (марка) области и масштабное изображение проекции модели

**Ins** - идентификатор модели вставляемого графического изображения.

**Point** - массив с координатами точки вставки

## 13.6 Атрибутные фильтры (Fm3D)

Параметры фильтров задаются в заголовке `s$header` фильтра

Структура **s\$header** атрибутных фильтров (фильтров свойств)

**Title** - наименование

**Descript** - описание - массив строк

**Mask** - массив правил проверки элементов модели

Тип проверки объекта - 1-входит в набор, 2-не входит в набор

Тип маски объекта: 1-простая строка, 2-регулярное выражение

Значение типа объекта

**Prop** - Массив правил проверки свойств элементов модели

Тип проверки свойства - 1-определено, 2-не определено, 11-соответствует значению, 12-не соответствует значению 21-выборка значения

Тип маски свойства: 1-простая строка, 2-регулярное выражение

Название свойства

**Value** - Массив значений свойств

Тип маски значения свойства: 1-простая строка, 2-регулярное выражение

Значение свойства

## 13.7 Замены

....будет указано дополнительно.

## 13.8 Пересечения и объединения

.....будет указано дополнительно.

# 14. Текстовые сведения

## 14.1 Инфодокумент (ITd)

Инфодокумент - тип текстового объекта, представляющий собой структурированный набор текстовой информации и соответствующей ей данных, предназначенный для восприятия человеком через взаимнооднозначные представления в виде человекочитаемых документов и для автоматизированной обработки в информационных системах.



Инфодокументы включают следующие данные:

Схема инфодокумента - описание типов (классов) текстовых данных в документе и их иерархию в документе, описание структуры свойств инфодокумента

Стили - описание отображения данных в виде форматированного текста

Макет - описание формата представления данных в виде бумажного или электронного документа (размер и ориентация страниц документа и т.п.)

Свойства - набор структурированных численных, логических или текстовых данных

Инфотекст - набор текстовых сведений

## 14.2 Инфонабор (ITs)

Неструктурированный набор текстовых сведений называются инфонабором.

Инфонаборы предназначены для использования в качестве ссылочных объектов для создания инфодокументов, чертежей, моделей и других объектов информационного моделирования.

## 14.3 Шаблон (ITt)

Инфодокумент или часть инфодокумента, предназначенная для создания новых инфодокументов и инфонаборов, называется шаблоном инфодокумента.

Шаблон содержит схему инфодокумента и может содержать стили, макет, текстовые элементы, схемы данных, данные

Отдельные элементы шаблона могут быть заданы как неизменяемые (нередатируемые) и защищены указанием контрольных сумм (будет описано дополнительно).

## 14.4 Компоненты (ITc)

Набор частей инфодокумента, предназначенных для импорта в другие инфодокументы, называется компонентом инфодокумента.

Компонент может содержать схему инфодокумента, стили, макет, текстовые элементы, схемы данных, данные

## 14.5 Элементы текстовых объектов

Текстовые объекты состоят из набора отдельных элементов, содержащих имя элемента и инфотекст

Имя элемента состоит из префикса "\_" и идентификатора элемента.

Объектам может быть присвоено краткое обозначение, для возможности дополнительной идентификации объектов, удобной пользователям

В элементе содержится массив, содержащий:

номер класса элемента,

обозначение (метка) элемента. Пустая метка может задаваться числом 0 или пустой строкой "".

строки инфотекста.

Строки инфотекста при обработке объединяются в одну общую строку. Из получившейся строки перед разбором тегов удаляются все символы форматирования и алфавитно-цифровые символы (табуляция, перевод строки и т.п.), последовательности пробелов заменяются на одиночные пробелы.

Идентификатором как правило является число. Идентификатор должен быть уникальным в пределах документа. Число может записываться как в 10-ричном виде, так и в системах счисления с другими основаниями - 16 (HEX), 37 (Z37 по спецификации инфоконтейнера) и т.п. Как правило идентификатор - порядковый номер создания объекта в документе в 10-ричном виде.

## 14.6 Информатизированный текст

Информатизированный текст - форматированный текст, в котором атрибутами форматизации задаются не параметры отображения элементов текста в документе (размер шрифта, тип шрифта, курсив и т.п.), а информационные параметры утверждений автора, соответствующие элементам текста - например важные утверждения, очень важные, примечания, пояснения и т.п.

Утверждения в тексте выполняются заданием фрагменту текста признака типа утверждения. Фрагмент текста задается парой тегов - открывающим и закрывающим, аналогично контрольным словам формата RTF (Rich Text Format)

Теги начинаются с символа "\".

Открывающие теги имеют вид "имяТега".

имяТега соответствует типу (признаку) соответствующего утверждения

Закрывающие теги имеют вид "\^имяТега". Имя Тега в закрывающем теге необязательно. Оно может отсутствовать или быть задано не полностью, только первыми несколькими символами. Имя тега в закрывающем теге нужно для контроля вложенности - если при проверке форматирования текста или его обработке символы имени тега в закрывающем теге не соответствуют (не входят) в имя открывающего тега, то выдается предупреждающее сообщение в консоль или лог.

То есть для пары \name .... \^na не будет выводиться предупреждающее сообщение, а для \name .... \^ma при проверке будет выводиться предупреждение (строка "ma" не входит в строку "name")

Последовательности пробелов в информатизированном тексте при его отображении усекаются до одного пробела, как в HTML

Символ "\" в информатизированном тексте кодируется "\\".

## 14.7 Составной текст

Составной информатизированный текст - текст, в котором специальными тегами заданы ссылки на элементы из других инфодокументов или инфонаборов в контейнере БимДамп.

Ссылка на вставку инфотекста задается тегом **\insert**. В теге указывается идентификатор инфодокумента или инфонабора, и перечень идентификаторов вставляемых элементов.

Тип вставки задается ключами тега.

-class - вставляется инфотекст с заданными в нем классами и тегами. (по умолчанию)

-text - вставляется инфотекст с тегами. Применяется класс элемента, в который вставляется инфотекст.

-plain - вставляется только текст, без тегов и классов.

## 14.8 Иллюстрации, схемы и чертежи

Вставка графических материалов выполняется с использованием тега `\image`. В теге указывается идентификатор модели, изображение которой нужно вставить, или имя файла изображения.

Тип и параметры вставки задается ключами тега.

-h - высота поля вставляемого объекта

-w - ширина поля вставляемого объекта

Размеры поля задаются в мм для варианта вставки в документ с страницей формата A4 и текстовым наполнением шрифтом Arial размером 11. При выводе инфодокумента с макетами документов другого размера вставленные изображения при необходимости масштабируются. Если размеры поля не заданы, то применяются размеры поля, заданные в моделях. Для изображений применяются размеры, заданные по умолчанию в макетах инфодокумента.

Модели и файлы изображений должны содержаться в контейнере БимДамп.

## 14.9 Свойства и данные текстовых объектов

Свойства и данные текстовых объектов задаются в элементе `s$prop`.

Этот элемент содержит примитивы **P1...P99** с наборами свойств и их значений.

Наборы свойств могут содержать другие наборы свойств.

Для задания структуры наборов используются специальные свойства `@name` и `@node`

`@name` задает имя набора.

`@node` задает номер набора, в который входит набор данных

Один из наборов, не являющийся вложенным в другой набор, может не иметь специальное имя `"_@"`. Такой набор называется базовым набором документа.

Для вставки значений свойств в инфотексте используется специальный тег `\@`, не имеющий закрывающего тега. После этого тега приводится нормализованное имя свойства, которое в инфотексте заменяется его значением.

Нормализованное имя свойства может содержать путь к свойству - указание перечня нормализованных имен наборов, разделенных знаком "точка"

Для базового набора имя набора можно в пути можно не указывать

Пример1

пусть свойство Код в наборе Проект имеет значение "Проект-1".

Тогда инфотекст "Данный проект имеет обозначение \@ Проект.Код"

Будет отображаться "Данный проект имеет обозначение Проект-1"

Пример2

пусть свойство КодПроекта в базовом наборе имеет значение "Проект-2".

Тогда инфотекст "Другой проект имеет обозначение \@ КодПроекта"

Будет отображаться "Другой проект имеет обозначение Проект-2"

## 14.10 Схемы инфодокументов

### 14.10.1 Назначение схем

Схемы инфодокументов - формализованные машиночитаемые описания структуры документов.

Схемы используются для следующих задач:

1. Задания иерархической структуры информации в инфодокументе для возможности её наглядного отображения и форматирования
2. Объединения нескольких инфодокументов в один
3. Разделение инфодокумента на несколько отдельных взаимосвязанных документов
4. Выделение логической части документа в отдельный документ
5. преобразований инфодокументов в другие форматы
6. проверка соответствия структуры инфодокумента установленным требованиям

### 14.10.2 Типы элементов

В инфодокументах взаимосвязь между элементами представляется в виде древовидной структуры, в которой отдельные элементы могут быть вложены ( или с использованием другой терминологии, являться потомками) других элементов.

В качестве примера можно привести дерево (древовидную структуру) условной книги, состоящей из следующих условных текстовых элементов:

*Аннотация*

*Текст1*

*Оглавление*

*Текст2*

*Глава1*

*Текст3*

*Текст4*

*Глава2*

*Раздел1*

*Текст5*

*Текст6*

*Комментарий1*

*Примечание1*

*Текст7*

*Раздел2*

*Текст8*

## Комментарий2

Этой структуре соответствуют следующие типы элементов

*Аннотация*

*Оглавление*

*Глава*

*Раздел*

*Текст*

*Примечание*

*Комментарий*

Типы элементов в рассматриваемой структуре группируются с использованием следующих отношений смысловой вложенности элементов одних типов в элементы других типов:

Уровень иерархии 1 (самый высокий) - Аннотация Оглавление Глава

Уровень иерархии 2 - Раздел

Уровень иерархии 3 (самый низкий) - Текст Примечание Комментарий

Элемент считается вложенным в предшествующий элемент, если уровень иерархии их типа ниже чем уровень иерархии типа предыдущего элемента.

Таким образом, путем задания типов элементам в упорядоченном наборе (последовательности) текстовых фрагментов, и задании иерархии этих типов, можно однозначно представлять последовательность текстовых элементов в виде древовидной структуры, соответствующей представлению текста в книгах и документах.

Для изменения порядка расположения текстовых элементов в документе, можно задавать индекс документа - параметр, использующийся для упорядочивания (сортировки) элементов в дереве. Элементы с меньшим значением индекса располагаются на своем уровне вложенности в дереве раньше, чем элементы с большим значением индекса. Элементы с одинаковым значением индекса располагаются в порядке задания текстовых элементов в последовательности элементов.

Упорядочивание (сортировка) выполняется следующим образом - вначале создается "дерево" элементов исходя из иерархии типов элементов, затем сортируются "ветви" дерева с использованием индекса сортировки

Например, чтобы всегда выводить комментарии после примечаний и для расположения оглавления в конце книги можно задать следующие индексы упорядочивания:

*Аннотация* 10

*Оглавление* 20

*Глава* 10

*Раздел* 10

*Текст* 10

*Примечание* 10

*Комментарий* 20

Тогда древовидная структура документа будет иметь вид:

*Аннотация*  
    *Текст1*  
*Глава1*  
    *Текст3*  
    *Текст4*  
*Глава2*  
    *Раздел1*  
        *Текст5*  
        *Текст6*  
        *Примечание1*  
        *Комментарий1*  
        *Текст7*  
    *Раздел2*  
        *Текст8*  
        *Комментарий2*  
*Оглавление*  
    *Текст2*

Альтернативным способом задания упорядочивания элементов является задания порядка (упорядоченного перечня) классов в дереве документа, что соответствует заданию классам индекса сортировки, равного порядковому номеру класса в упорядоченном списке.

Для вышеописанного примера перечень классов будет иметь вид:

*Аннотация Глава Раздел Текст Примечание Комментарий Оглавление*

Использование метода упорядочивания с использованием порядка классов как правило проще для отдельных документов, но для инфодокументов, получаемых путем слияния (сборки) из набора отдельных инфодокументов с использованием различных схем инфодокументов, использование индексов упорядочивания предпочтительней.

### 14.10.3 Классы элементов

Класс элемента инфодокумента — это обобщающее понятие, соответствующее типу информации в элементе, с заданием параметров вложенности и упорядочивания элемента в структуре документа, и параметров, характеризующих количество и свойства элементов указанного типа.

Для классов инфоэлементов задаются следующие параметры:

**Имя класса** - уникально в схеме имя или маска имен, задаваемая регулярным выражением. Характеризует тип информации в элементе

**Описание** - описание класса. Приводится краткое описание вида информации, соответствующей элементам документа данного класса.

**Группа** - группа классов, которой принадлежит данный класс. Используется для группировки интерфейсных элементов в программах редактирования инфодокументов. Если класс не принадлежит никакой группе, то этот параметр равен 0.

**Уровень** - уровень иерархии

**Индекс** - индекс упорядочивания

**Минимум** - минимальное количество элементов данного класса в инфодокументе. Для необязательных элементов может быть равно 0.

**Максимум** - максимальное количество элементов данного класса в инфодокументе. Если максимальное количество не задано или равно 0, то количество элементов данного класса не ограничено. Если максимальное количество равно 1, то такой элемент также называется "уникальным" элементов для инфодокумента

**Наполнение** - минимальное количество вложенных элементов

**Вложенность** - тип соответствия элемента иерархической структуре схемы.

0-элемент может быть вложен в другой элемент с любым более высоким уровнем иерархии.

1-элемент должен быть вложен в другой элемент только на предыдущем уровне иерархии.

N (число больше 1) - элемент должен быть вложен в другой элемент, имеющий более высокий уровень иерархии, отличающийся не более чем на величину N

**Константа** - предопределенное значение текста инфоэлемента, которое может быть задано для уникальных классов. Если константа не задана, то текстовое значение элемента может быть любым.

Имена классов должны начинаться с буквы или с символов "\_@#&", не иметь пробелов, содержать буквы и цифры, знаки "\_@\$&-:" и знак точка ".".

Символ "%" является символом задания маски имен классов. Текст после символа "%" рассматривается как регулярное выражение. Пробелы, следующие после "%" до первого значащего символа, не включаются в регулярное выражение

Классы с маской в имени используются в следующих случаях:

При задании стиля для маски класса, стиль применяется для всех классов, соответствующих маске, как стиль по умолчанию для этого класса.

При проверке содержимого документа на соответствие схеме, все классы, соответствующие маске считаются правильными, при условии соответствия параметров класса заданным для класса маски параметрам, за исключением индекса упорядочивания. Значение индекса упорядочивания класса должно быть не меньше индекса упорядочивания класса маски

#### 14.10.4 Объединение инфодокументов

Схемы инфодокументов могут применяться для задания вида инфодокумента, полученного слиянием нескольких инфодокументов.

Например, пусть нужно составить документ по отчету по анализу какого либо объекта, выполняемого двумя специалистами по двум направлениям, которые условно обозначены как А и Б.

Отчет состоит из разделов - Описание, Анализ, Выводы

В каждом разделе вначале должен идти материалы по направлению А, затем по направлению Б.

Для составления такого инфодокумента документа выполнена схема, в которой заданы классы с указанными группами, уровнями иерархии, индексами сортировки, минимальным и максимальным количеством, наполнением, вложенностью, предопределенным значением

Описание	0	10	10	1	1	1	0	"Описание"
Анализ	0	10	20	1	1	1	0	"Анализ"
Выводы	0	10	30	1	1	1	0	"Выводы"
ТекстА	1	20	10	1	0	0	1	
ТекстБ	1	20	20	1	0	0	1	

При выполнении отчетных документов по своим направлениям, специалист по первому направлению должен использовать для текста класс ТекстА, а по второму - класс ТекстБ.

Тогда при слиянии документов будет получен документ требуемой структуры.

Также возможен другой подход, при котором специалисты создают документы с использованием одинаковой схемы:

Описание	0	10	10	1	1	1	0	"Описание"
Анализ	0	10	20	1	1	1	0	"Анализ"
Выводы	0	10	30	1	1	1	0	"Выводы"
Текст	1	20	10	1	0	0	1	

И при слиянии документов в общий документ класс Текст в документе Б преобразуется в класс ТекстБ с индексом сортировки 20. Для преобразования классов может использоваться таблица преобразования или какой-либо другой способ.

При слиянии также могут учитываться значения каких-либо свойств инфодокументов.

## 14.10.5 Задание схем

Схемы состоят из следующих структур описаний:

**s\$schema** - схема документа

1. ClassList - массив с массивами имен и параметров классов. Записи массива описания класса
  - 1.1. Имя класса.



- 1.2. Описание
- 1.3. Группа
- 1.4. Уровень иерархии
- 1.5. Индекс сортировки
- 1.6. Минимальное количество
- 1.7. Максимальное количество
- 1.8. Наполнение
- 1.9. Вложенность
- 1.10. Константа
2. `ClassDef` - номер в перечне имени класса по умолчанию. Если не задан или не правилен (больше чем размер перечня), то принимается равным 1.
3. `TagList` - массив с перечнем тегов инфотекста - [[ имяТега, кодXML, описание Тега],...]

Если в инфодокументе не задана схема, то для инфодокумента используется схема, заданная в системном объекте файла БимДамп.

Если в файле также нет схемы, то все классы элементов инфодокумента считаются располагающимися на первом уровне иерархии классов.

**s\$order** - последовательность элементов в документе - массив идентификаторов элементов. Если последовательность не задана, то порядок элементов соответствует порядку возрастания идентификаторов элементов.

## 14.11 Стили отображения инфотекста

Стили отображения инфотекста - сопоставление классов объектов инфодокумента и тегов выделений с тегами форматирования текста для отображения инфотекста на экран или на печать.

Стили инфотекста являются каскадными, по типу CSS для HTML. То есть стиль может основываться на каком-либо другом стиле, с заменой каких-либо отдельных параметров

В стилях также для классов в стилях также может быть задана автонумерация

Стили отображения задаются элементом **s\$style**.

Если в инфодокументе не заданы стили, то используются стили из системного объекта файла БимДамп

**s\$style**.- элемент для задания стилей

1. `Plain` - "основной" стиль - стиль по умолчанию. Содержит массив с перечнем величин стандартных свойств стиля класса.
2. `Class` - таблица стили отображения элементов в зависимости от их классов или уровня иерархии классов. Объект содержит массив, содержащий имя класса, номер "опорного" класса, описание класса, перечень значений стандартных свойств стиля класса. Затем могут идти последовательность номеров и значений расширенных свойств класса. Если для свойства задано относительное значение, то рассчитывается значение относительно свойства "опорного" класса
3. `Tag` - стили отображения информативных фрагментов текста. Объект содержит массив, содержащий имя тега, имя "опорного" тега, массива последовательности из

номера или названия свойства стиля и значения свойства. Свойства могут иметь относительные значения

4. Color - таблица цветов, используемых в стиле. Задается название цвета и соответствующие ему компоненты цвета R G B. Имя (название) цвета не должно начинаться с цифры и знаков "+", "-", "\*". Общепринятые (стандартные) имена цветов (белый, черный, серый, синий, голубой, розовый, красный и т.п.) определены по умолчанию и не могут быть переопределены. В таблице задаются все цвета, используемые в инфодокументе. В таблице могут быть заданы цвета, которые не используются в инфодокументе, но которые могут быть использованы при дальнейшем редактировании документа.
5. Font - массив шрифтов, используемых в стиле. Задаются название семейств шрифтов. Могут содержаться шрифты не содержащиеся в инфодокументе, которые могут быть использованы при дальнейшем редактировании документа

Стандартные свойства стилей классов и тегов (в скобках указаны значения по умолчанию.)  
- упорядоченный список

- 1 font - номер семейство шрифта (1, Arial)
- 2 size - размер (11)
- 3 color - номер цвета шрифта (1, черный)
- 4 ground - номер цвет фона (2, белый)
- 5 bold - жирность (0)
- 6 italic - курсив (0)
- 7 underline - подчеркивание (0)
- 8 strike - зачеркивание (0)
- 9 interval - интервал между строк (1)
- 10 indent - отступ (0)
- 11 ledge - выступ первой строки (0)
- 12 align - выравнивание (1) 1-по левому краю, 2- по центру, 3 - по правому краю, 4- по ширине
- 13 numbering - автонумерация (0)

Дополнительные свойства стилей классов и тегов (в скобках указаны значения по умолчанию.) - упорядоченный список

- 21 sup - верхний индекс (0)
- 22 sub - нижний индекс (0)
- ..... и т.д.

Логические величины обозначаются цифрами 0-нет и 1-да

Для величин свойств и компонентов цвета можно указывать "относительные" значения. знаки увеличения "+", уменьшения "-" и умножения "\*", изменения пропорционально

диапазону "%" (например используется для компонентов цвета, имеющих диапазон 0..255). Например +2, -2, \*1.2, %1.1. Для обеспечения отличия относительных значений от абсолютных, относительные значения можно указывать в скобках. Например "(+2)"

Цвет обозначается номером в таблице цветов. Для номеров цвета может применяться относительные значения

Для стилей тегов относительные значения применяются к текущему значению свойства, с учетом класса и других тегов.

Для стилей класса относительные значения применяются к соответствующему значению стиля "Опорного" класса, заданного в таблице стилей. Если опорный класс не задан, то относительные значения применяются относительно предыдущего заданного уровня иерархии класса, или стиля по умолчанию если не заданы стили уровня иерархии.

Пример:

В схеме инфоэлемента заданы следующие классы с уровнями иерархии - Раздел (10), Глава (20), Текст (30).

В стилях заданы стили для уровня 10, для классов Раздел и Глава не заданы.

Относительное значение свойства стиля для класса Текст будет применяться относительно свойства класса более верхнего уровня иерархии, для которого задана схема - уровня 10.

Если в схеме это свойство не задано - то относительно свойства основного стиля (стиля по умолчанию).

Если в схеме не задан основной стиль, то относительное значение будет применено относительно значения свойства по умолчанию.

## 14.12 Макеты отображения инфотекста

Макет отображения инфотекста - описание размера и оформления страниц печатного образа инфодокумента.

Макет отображения задается элементом **s\$layout**.

Если в инфодокументе не задан макет, то используется макет из системного объекта файла БимДамп

элемент **s\$layout** содержит элементы оформления страниц:

Элементы страниц начинаются с префиксов, задающих тип оформления страницы.

"\_номер" - в имени примитива задан номер страницы, с которой начинается заданное оформление

"\$идентификатор" - в свойствах примитива заданы классы, уровни или индексы инфоэлемента, с которых начинается заданное оформление. Может быть задан только один из способов выбора оформления, по классам, или по уровням, или по индексам.

\_\$идентификатор - прототип страницы

Элементы страниц содержат примитивы

**P1** - набор свойств страницы (в скобках указаны значения по умолчанию):

F - название формата. Например "A4" (A4)

L - расположение. 1-книжное 2-альбомное (1)

H - высота, мм

W - ширина, мм

ML - поле слева (20)

MR - поле справа (10)

MT - поле сверху (10)

MB - поле снизу (10)

IW - ширина поля вывода изображений по умолчанию (50)

IH - высота поля вывода изображений по умолчанию (30)

C - количество колонок (1)

**C** - массив имен классов

**L** - массив уровней иерархии

**I** - массив индексов упорядочивания

**T** - идентификатор прототипа

**примитивы оформления** - графические примитивы 2D графики

## 14.13 Параметры интернализации инфотекста

Параметры интернализации инфотекста - сопоставление тегов выделений с параметрами автоматического перевода текста.

Например, текст на английском языке содержит описание какого либо зарезервированного слова языка программирования. При переводе на русский язык будет требоваться перевести описание, а зарезервированное слово оставить в оригинальном написании. Для этого тегами интернализации можно указать, что зарезервированное слово является непереводаемым.

Пример строк и их автоматического перевода, когда тег \nt обозначает непереводаемое выражение.

1 - Operator **return** is used to return result of calculations

2 - Operator \nt **return** ^ is used to return result of calculations

1 - Оператор **возврат** используется для возвращения результата вычислений

2 - Оператор **return** используется для возвращения результата вычислений

**s\$lang**.- параметры интернализации инфодокумента

1. Tag - коды языка и коды переводимости. Объект содержит массив, содержащий имя тега, код языка и код переводимости.

Код языка может быть не задан. Коды могут задаваться не только для разговорных языков, но и для языков программирования и других машинных языков.

Код переводимости - 1 - подлежит переводу, 0 - не подлежит переводу. По умолчанию - 1

## 14.14 Схемы данных

Схемы данных, которые должны содержаться в инфодокументах и инфонаборах - формализованные машиночитаемые описания структур данных и их параметров.

Схемы используются для следующих задач:

1. проверка правильности структуры данных
2. проверка полноты представления данных

Схемы состоят из следующих структур описаний:

**s\$dataschema** - схема данных

1. S1..S99 - типы наборов данных
  - 1.1. Structure - массив параметров набора
    - 1.1.1. Описание - описание набора. Произвольная строка небольшой длины, используется для удобства восприятия пользователем. Используется в интерфейсах программ вместе с номером набора как подсказка или пояснение.
    - 1.1.2. Номер типа набора данных, в которые входит данный набор. Если 0 - то данный набор находится на первом уровне иерархии
    - 1.1.3. Минимальное количество наборов такого типа
    - 1.1.4. Максимальное количество наборов такого типа. Если 0 - то не ограничено.
  - 1.2. Name - массив имен наборов данного типа. Имена используются последовательно при создании наборов такого типа. При использовании всех имен из массива создаются имена наборов с добавлением порядкового номера после последнего имени
  - 1.3. XML - массив имен XML при преобразование схемы в XML-schema (XSD). Имена используются последовательно при создании наборов такого типа. При использовании всех имен в массиве создаются имена с добавлением порядкового номера после последнего имени
  - 1.4. Property - массив массивов описания параметров свойств в наборе.
    - 1.4.1. Код существования - 1 - обязательное 0 - опциональное
    - 1.4.2. Имя свойства
    - 1.4.3. XML - имя объекта при представлении свойства в XML-schema
    - 1.4.4. Тип
      - 1.4.4.1. T - текст
      - 1.4.4.2. I - целое число
      - 1.4.4.3. R - действительное число
      - 1.4.4.4. D - дата
      - 1.4.4.5. H - время
    - 1.4.5. Значение по умолчанию. Если не задано, то используется минимальное значение. Если минимальное значение не задано, то используются следующие значения.
      - 1.4.5.1. Текст - пусто

- 1.4.5.2. Целое число - 0
- 1.4.5.3. Действительное число - 0.0
- 1.4.5.4. Дата - 01.01.2000
- 1.4.5.5. Время 00:00
- 1.4.6. Минимальное значение (или количество символов)
- 1.4.7. Максимальное значение (или количество символов)
- 1.4.8. Формат - строка формата отображения свойства, соответствующая формату, используемому в JavaScript. По умолчанию - пусто (не задано)

## 14.15 Внешнее представление HTML

Внешнее представление HTML - представление инфодокумента с использованием заданных стилей отображения в формат HTML с использованием каскадных стилей

## 14.16 Внешнее представление SVG

Внешнее представление SVG - представление страниц печатного образа инфодокумента с использованием заданных стилей отображения и макета в формат SVG.

Для компонентного взаимосвязывания в объекте <svg> файла указываются атрибуты:

VimML=1

GUID = глобальный идентификатор инфодокумента

ID = локальный ключ-идентификатор инфодокумента

InfoDoc = название инфодокумента

Page=номер страницы

Инфоэлементам в файлах SVG соответствуют группы <g> с вложенными объектами геометрических примитивов.

Для обеспечения атомарного взаимосвязывания для этой группы задается атрибут id=ИмяЭлементаИнфодокумента с именем соответствующего элемента инфодокумента.

Штриховки геометрических примитивов макетов инфоэлементов должны преобразовываться в соответствующие объекты с использованием штриховок <hatch> формата SVG

# 15. Шкалы и палитры

## 15.1 Шкалы

Шкала – функция преобразования значений величин в значение цвета, соответствующего этой величине.

Имя объекта шкалы состоит из префикса "\_" и уникального идентификатора для библиотеки шкал. Идентификатором как правило является число.

**Type** = scale - обозначение типа объекта

**Code** - код назначения (типа) шкалы

**Name** - Название шкалы

**Comment** - Комментарий

**Range** - Диапазон шкалы [начальное значение, конечное значение]

**Rtype** - тип диапазона. 1-абсолютный. Используются значения диапазона. 2-относительный. Диапазон шкалы масштабируется для соответствия диапазону величин в рассматриваемом наборе

**Palette** - имя используемой палитры

## 15.2 Палитры

Палитра – функция преобразования значений условной величины в значение цвета, соответствующего этой величине.

Имя объекта палитры состоит из префикса "\_" и уникального идентификатора для библиотеки палитр. Идентификатором как правило является число.

**Type** = palette

**Code** - код назначения (типа) палитры

**Name** - название

**Comment** - описание

**Range** - диапазон палитры [x1,x2]

**Color** - функция раскраски – функция соотношения цвета rgba и величины в заданном диапазоне палитры) - массив компонентов цвета R G B A и соответствующих значений величины

**Stype** - тип функции раскраски

1 - полигональная – задаются цвета последовательности величин, компоненты цвета величины определяются интерполяцией между этими точками

2 - дискретная – задаются цвета последовательности величин, компоненты цвета величины определяются устанавливаются постоянным до следующего заданного значения

## 16. Работы и проекты

### 16.1 Комплекты моделей (JPK)

Комплект моделей - набор моделей и документов, соответствующий состоянию или выборке информации какой-либо работы или проекта, и описанию взаимосвязей в данных комплекта и взаимосвязей с другими комплектами.

2. Комплект содержит заголовков и элементы **s\$kit**, **s\$group**, **s\$status**, содержащие информацию об элементах набора.

Элемент **s\$kit** содержит таблицу (массив массивов ) описания с следующими полями

Идентификатор модели

Группа моделей (если группа не задана, то 0)

Тип модели

Название модели

Модель прародителя

Модель родителя

Трансформация

Список идентификаторов атомарно взаимосвязанных моделей

Элемент **s\$group** содержит таблицу (массив массивов ) с описанием древовидной структуры набора с следующими полями

Вложенность - номер группы, в которую входит данная группа. Если не задано, то 0.

Название

Описание

Элемент **s\$status** содержит таблицу (массив массивов ) описания статуса набора в проекте с следующими полями. В таблице может быть несколько строк с описанием статусов набора по различным критериям

Код статуса

Название статуса

Название критерия