

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Локтионова Оксана Геннадьевна

Должность: проректор по учебной работе

Дата подписания: 11.05.2023 11:17:26

Уникальный программный ключ:

0b817ca911e4024d9ca146ba0e51f0c20ba0e9430b64811ba66089

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Юго-Западный государственный университет»

(ЮЗГУ)

Кафедра «Машиностроительные технологии и оборудование»

УТВЕРЖДАЮ
Проректор по учебной работе

[Подпись]
« 10 » 05
О.Г. Локтионова
11.05.23



ОПИСАНИЕ СТРУКТУРЫ ОБЪЕКТОВ НА ЯЗЫКЕ EXPRESS

Методические указания к выполнению практической работы по дисциплине «Информационная поддержка жизненного цикла продукции» для студентов направления подготовки 15.03.01 «Машиностроение» направленность (профиль) «Технология, оборудование и автоматизация машиностроительных производств»

УДК 519.6

Составители: В.В. Куц, А.Г. Ивахненко

Рецензент

Кандидат технических наук, доцент кафедры
«Машиностроительные технологии и оборудования» А.Н. Гречухин

Описание структуры объектов на языке EXPRESS:
методические указания к выполнению практической работы по дисциплине «Информационная поддержка жизненного цикла продукции» по направлению подготовки 15.03.01 «Машиностроение» направленность (профиль) «Технология, оборудование и автоматизация машиностроительных производств» / Юго-Зап. гос. ун-т; сост.: В.В. Куц, А.Г. Ивахненко. Курск, 2023. 21 с. Библиогр.: с. 21.

Излагаются краткие сведения о структуре информационных моделей на языке EXPRESS. Приводятся варианты заданий для выполнения практических занятий по дисциплине «Информационная поддержка жизненного цикла продукции».

Предназначены для обучающихся по направлению подготовки 15.03.01 «Машиностроение» направленность (профиль) «Технология, оборудование и автоматизация машиностроительных производств».

Текст печатается в авторской редакции

Подписано в печать . Формат 60×84 1/16.
Усл. печ. л. . Уч. - изд. л. . Тираж 50 экз. Заказ 383
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

1 СТРУКТУРА МОДЕЛЕЙ НА ЯЗЫКЕ EXPRESS

Базовый для STEP-технологий язык Express описан в стандарте ISO 10303, том 11. Язык является объектно-ориентированным, имеет универсальный характер, его можно использовать для описания статических структур и их свойств в различных предметных областях, несмотря на то, что язык разрабатывался прежде всего в качестве средства представления моделей промышленных изделий на разных этапах их жизненного цикла.

Описание некоторого приложения на языке Express в рамках стандартов STEP называют Express моделью (**model**). В модели декларируются множества понятий и объектов, входящих в приложение, свойства и взаимосвязи объектов.

Модель состоит из одной или нескольких частей, называемых Express схемами (**schema**) или просто схемами, и обменного файла. Схема – раздел описания, являющийся областью определения данных. В ней вводятся необходимые типы данных. При описании свойств типов данных могут применяться средства процедурного описания – процедуры, функции, правила, константы. Обменный файл содержит конкретные экземпляры типов данных.

Описание схемы начинается с заголовка, состоящего из служебного слова **schema** и идентификатора – имени схемы. Далее следует содержательная часть – тело схемы. Описание заканчивается служебным словом **end_schema**:

```
SCHEMA <имя_схемы>;  
<тело_схемы>;  
END_SCHEMA;
```

Для установления интерфейса между двумя схемами вводятся спецификации интерфейса. Применяют два типа спецификаций – **use** и **reference**. Например:

```
SCHEMA s1;  
ENTITY par1;  
name: STRING;  
END_ENTITY;
```

```

END_SCHEMA;
SCHEMA s2; ( * в схеме s2 в качестве параметра x
используется name из s1.par1 *)
USE FROM s1.par1 (name AS x);
END_SCHEMA;

```

Ссылки типа **use** отличаются тем, что декларации сущностей из другой схемы используются в данной схеме как свои локальные, в то время как **reference** просто позволяет обращаться к декларациям другой сущности. Ограниченность **reference** выражается в том, что сущности из другой схемы можно использовать только в качестве типов атрибутов в сущностях данной схемы.

1.1. Типы данных в языке Express

В теле схемы декларируются типы данных (**Data Type**). Тип данных — это множество значений некоторой величины или множество объектов (набор экземпляров). В языке Express используются следующие типы данных:

- сущность (**Entity**),
- простой (**Simple Type**),
- агрегативный (**Aggregation Data Type**),
- определяемый (**Defined Data Type**),
- нечисловой (**Enumeration Data Type**);
- выделяемый (**Select Data Type**) типы.

Сущность — тип данных, представляющий набор концептуальных или реальных физических объектов с некоторыми общими свойствами. Сущности используют для описания объектов приложений. Свойства сущности выражают в виде атрибутов (**Attributes**). К характеристикам сущностей относятся также ограничения, накладываемые на значения атрибутов или на отношения между атрибутами.

Описание сущности начинается со служебного слова **ENTITY**, за которым следуют идентификатор сущности, описания ее

атрибутов и возможно также правил. Каждый из атрибутов представлен его идентификатором и типом:

```

ENTITY <имя_сущности>;
<идентификатор_атрибута>:<тип_атрибута>;
...
END_ENTITY;

```

Например, задание прямой линии (**line**) в виде двух инцидентных точек **p0** и **p1** (атрибутов типа **point**) выглядит следующим образом:

```

ENTITY line;
p0,p1: point;
END_ENTITY;

```

Атрибуты и переменные сами могут быть сущностями, так тип атрибутов **p0** и **p1** предыдущего примера декларируется, как сущность, атрибутами которой в случае пространства 3D являются геометрические координаты x, y, z :

```

ENTITY point;
x,y,z: REAL;
END_ENTITY;

```

Если свойство является необязательным для данной сущности, то его выражают так называемым необязательным (**optional**) атрибутом. В его описании перед типом атрибута добавляется служебное слово **OPTIONAL**:

```

<идентификатор_атрибута>:OPTIONAL тип_атрибута>;

```

Некоторые из атрибутов могут определяться через другие атрибуты. Тогда атрибуты, выражаемые через другие атрибуты, называют порожденными (**derived**), что отображается служебным

словом **derive** в декларации атрибута. Например, описание окружности, кроме обязательных атрибутов, которыми в нижеследующем примере выбраны радиус и центр окружности, может включать порожденный атрибут площадь круга:

```

ENTITY point;
x,y,z: REAL;
END_ENTITY;
ENTITY cycle;
center: point;
radius: REAL;
DERIVE
area: REAL := pi*radius**2;
END_ENTITY;

```

В этом примере использованы явные атрибуты **center**, **radius** и порожденный атрибут **area**.

Отметим, что между символами (***** и *****) записывается комментарий – произвольный текст по усмотрению автора модели. Если комментарий уместается в одной строчке, то достаточно перед его текстом поставить двойной дефис (**--**).

К простым типам данных относятся следующие типы:

- integer** – целые числа;
- real** – вещественные числа;
- number** – тип, объединяющий типы **integer** и **real**;
- logical** – его значениями могут быть **true**, **false** или **unknown** (неопределенность);
- Boolean** – с возможными значениями **true** или **false**;
- binary** – последовательность битов 1 или 0;
- string** – строка символов.

Для **binary** и **STRING** в круглых скобках можно указать максимально возможное число элементов множества, например, если строка **A** может включать до 24-х символов, то:

```

A: STRING(24);

```

Если строка должна содержать ровно 24 символа, то:

A: STRING(24) FIXED;

Если же ограничений на длину строки нет, то:

A: STRING;

Если переменная *x* имеет тип `binary`, то выражение `x[5:7]` означает биты с 5-го по 7-й в коде *x*.

Значения простых типов выражаются с помощью литералов. Литералы – это числа (целые, вещественные), двоичные коды, логические значения (**true**, **false**, **unknown**), фрагменты текста (строковый тип). Примеры записи литералов:

двоичный (начинается с знака %) **%100101110**

целое десятичное число **1052**

вещественный (обязательна десятичная точка) **34.e-3** или

0.034

строковый (занимает не более одной строки) **'first name'**.

Агрегативный тип данных – множество элементов некоторого типа.

Различают четыре разновидности агрегативных типов, сведения о которых приведены в табл. 1.

Таблица 1

Тип данных	Упорядоченность	Различие элементов
ARRAY	Да	Необязательно
BAG	Нет	Необязательно
LIST	Да	Обязательно
SET	Нет	Обязательно

При описании типа массив (**array**) после слова **array** в квадратных скобках указываются нижняя и верхняя границы индексов. Для остальных агрегативных типов записываются не граничные значения индекса, а нижняя и верхняя границы числа элементов. Например:

F1: ARRAY[2:8] of REAL;

(* массив F1 из 7-и элементов, элементы имеют тип REAL и нумеруются, начиная с индекса 2 *);

F2: LIST[1:?] of INTEGER;

(* множество F2 содержит, по крайней мере, один элемент типа INTEGER *) matr: ARRAY[1:10] of ARRAY[9:12] of atrac;

(* массив matr состоит из 10 четырехэлементных массивов, элементы типа atrac *)

Определяемый тип данных обычно вводится пользователем для улучшения читаемости модели. Нечисловой тип – тип данных, экземплярами которого являются нечисловые (предметные) переменные. Выделяемый тип соответствует поименованной совокупности других типов. Описание этих типов данных начинается со служебного слова **type**, за которым следует идентификатор типа и его определение. Пример описания определяемого типа:

```

TYPE volume = real;
END_TYPE;
ENTITY manual;
name: STRING;
v1,v2,v3: volume;
END_ENTITY;

```

Определение нечислового типа начинается со служебных слов **enumeration of**, после которых в скобках перечисляются элементы множества. Например:

```

TYPE color = ENUMERATION OF (red, green, blue);
END_TYPE;

```

Ссылка на значение **red** теперь возможна в виде **red** или **color.red**.

Выделяемый тип соответствует одному из некоторого списка уже введенных типов. Этот список записывается после служебного слова **select**. Ссылка на имя выделяемого типа означает, что выбирается один из типов совокупности:

```

TYPE a_c = SELECT (one, two, three);
END_TYPE;
...
proc: a_c; (* proc может быть объектом одного из типов
one, two, three *)

```

Способ описания констант очевиден из следующего фрагмента модели:

```

CONSTANT
year: INTEGER := 1995;
start: date := date(12,16,1982);
(* подразумевается, что при описании типа date указаны
три атрибута
    месяц, число, год *)
END_CONSTANT;

```

1.2. Супертипы и подтипы

Отношения агрегирования (типа целое-часть) или отношения обобщения (функция-вариант реализации), характерные для представления структур объектов в виде альтернативных (И-ИЛИ) деревьев, в языке Express выражаются в форме отношений между типами данных. Для этого введены понятия супертипа (**supertype**), как более общего типа, и подтипов (**subtypes**), как подчиненных типов.

Рассмотрим пример фрагмента И-ИЛИ-дерева, в котором имеется ИЛИ вершина **a1** и две подчиненные ей альтернативные вершины **b1** и **b2**. Общим атрибутом для **b1** и **b2** является **size** типа **real**, специфичный для **b1** атрибут – **vol** типа **real**, а специфичный

для **b2** атрибут **met** типа **string**. Этот фрагмент может быть описан следующим образом:

```

ENTITY a1
SUPERTYPE OF (ONEOF (b1,b2));
size: REAL;
END_ENTITY;
ENTITY b1
SUBTYPE OF (a1);
vol: REAL;
END_ENTITY;
ENTITY b2
SUBTYPE OF (a1);
met: STRING;
END_ENTITY;

```

Используются также следующие правила записи супертипов и подтипов: в случае, если **a1** есть И вершина, вместо **oneof** используется зарезервированное слово **and** (в более общем случае **andor**), т.е. вторая строчка примера будет выглядеть так: **supertype of (b1 and b2);**

если между подтипами нет взаимосвязи, выражаемой логической функцией (в частности, ИЛИ или И вершинами), то указание в **a1** факта, что это супертип, не требуется; достаточно упоминание о подчиненности подтипов в их декларациях в виде: **subtype of (a1);**

перед декларацией **supertype** записывается зарезервированное слово **abstract**, если вершине **a1** не соответствуют какие-либо экземпляры сущности, т.е. если **a1** введена только для указания общих для подтипов атрибутов;

у одного подтипа может быть больше одного супертипа; подтип наследует атрибуты всех своих супертипов; если в декларациях супертипов используются одинаковые идентификаторы атрибутов, то ссылка на них должна быть в виде составного идентификатора, например: **a1.size**.

Пример 1

```

ENTITY device
SUPERTYPE OF (ONEOF (transistor, diode));
(* device есть ИЛИ вершина И-ИЛИ-дерева с двумя
альтернативами transistor и diode *)
END_ENTITY;
ENTITY transistor
SUBTYPE OF (device);
b: REAL;
END_ENTITY;
ENTITY diode
SUBTYPE OF (device);
r: REAL;
END_ENTITY;

```

1.3. Процедуры и функции

Процедуры и функции в языке Express служат для описания процедурной части модели. Как и в алгоритмических языках, используется концепция формальных и фактических параметров. Описание процедуры начинается с служебного слова **PROCEDURE**, за которым следуют идентификатор процедуры и описание формальных параметров в круглых скобках. Пример описания заголовка процедуры:

```

PROCEDURE eq (x,y: REAL; n: INTEGER; VAR
RESULT: route);

```

Аналогично описываются функции, их отличает только описание в заголовке типа результата после закрывающей скобки:

```

FUNCTION log (a: REAL; m: INTEGER): REAL;

```

Локальные переменные, описанные в блоке действуют только в пределах данных функции или процедуры:

```

LOCAL

```

```

...

```

```

END_LOCAL;

```

Ряд функций и процедур относится к стандартным и потому не требует описания во вновь разрабатываемых моделях. Отметим следующие стандартные функции:

Abs – абсолютная величина;

Sqrt – корень квадратный;

Exp – экспонента;

Log, Log2, Log10 – логарифмы натуральный, двоичный, десятичный соответственно;

Sin, Cos, Tan, Acos, Asin, Atan – тригонометрические и обратные тригонометрические функции **sin, cos, tg, arccos, arcsin, arctg**.

В число стандартных входят также функции:

BLength – подсчет числа бит в двоичном коде;

HiBound – верхняя граница индекса у array или верхняя граница числа элементов у set, bag, list;

LoBound – то же в отношении нижних границ;

Length – подсчет числа символов в строке;

Odd – возвращает значение true, если аргумент – нечетное число;

SizeOf – возвращает число элементов в объекте агрегативного типа;

TypeOf – возвращает список типов, к которым принадлежит параметр этой функции;

Exists – возвращает значение true, если аргумент этой функции входит в число атрибутов соответствующей сущности, и др.

К стандартным процедурам относятся процедуры **Insert** и **Remove** – вставка или изъятие элемента в заданной позиции у объекта агрегативного типа соответственно.

При описании алгоритмов в телах процедур и функций могут использоваться операторы пустой (**Null**), присваивания (**Assignment**), выбора (**Case**), составной (**Compound Statement**), условный (**if..then..else**), цикла (**Repeat**), выхода из функции или процедуры (**Return**), перехода на конец цикла (**Skip**).

В выражениях используются операнды, знаки операций, вызовы функций. Так, для арифметических операций над числами

типа **real** применяются следующие знаки: * – умножение, / – деление, **DIV** – целочисленное деление, + – сложение, - – вычитание, ** – возведение в степень, **MOD** – деление по модулю.

Знаки логических операций:

not – отрицание,

and – конъюнкция,

or – дизъюнкция,

xor – исключающее ИЛИ.

В применении к величинам типа **logical** эти операции выполняются по правилам действий в трехзначном алфавите. Логическое выражение **a1 in a2** принимает значение **true**, если **a1** содержится в **a2**. Оператор **like** используется для посимвольного сравнения строк. Знаки отношений равно =, не равно <>, больше >, меньше <, больше или равно >=, меньше или равно <=. Для сравнения экземпляров сущностей используют операции "равно" и "неравно" со знаками :=: и :=<>: соответственно.

Операции над множествами (типами **bag** и **set**) – пересечение (**Intersection**), объединение (**Union**), разность (**Difference**). Их знаки суть * (умножение), + (плюс), - (минус) соответственно. Оператор **Query (A <* B | C)** возвращает подмножество тех элементов из агрегативного типа B, для которых выполняется условие C, здесь A – простая переменная, используемая в C.

Знак + (плюс) по отношению к операндам типа **binary** или **string** есть знак конкатенации.

В качестве формальных параметров процедур и функций, кроме типов данных, применяемых в других конструкциях языка и охарактеризованных выше, могут использоваться обобщенные типы: **generic**, **aggregate** и некоторые другие. Тип **generic** формального параметра означает, что соответствующий фактический параметр может иметь любой тип данных из числа предусмотренных при описании процедуры. Аналогично тип **aggregate** обобщает агрегативные типы данных – **array**, **bag**, **list**, **set**. Например:

Пример 2

```

FUNCTION add (a,b: GENERIC: intype): GENERIC:
intype;
LOCAL
nr: number;
vr: vector;
END_LOCAL;
IF ('number' IN TYPEOF (a)) AND ('number' IN TYPEOF
(b))
THEN nr := a+b;
(* функция typeof (a) возвращает тип аргумента а и
если этот тип есть number, то первый операнд
логического выражения равен true *)
RETURN (nr);
ELSE
IF ('this schema.vector' IN TYPEOF (a)) AND ('this
schema.vector' IN TYPEOF (b)) THEN
vr.i := a.i + b.i;
vr.j := a.j + b.j;
vr.k := a.k + b.k;
(* подразумевается, что декларация типа vector
была произведена в схеме с именем this schema *)
RETURN (vr);
END_IF;
END_IF;
END_FUNCTION;

```

1.4. Пример описания структуры объекта «Поезда»

```

SCHEMA поезда;
TYPE тип_колеи = ENUMERATION OF
(узкая, европейская, российская);
END_TYPE;

```

ENTITY поезд; номер_поезда :
INTEGER; локомотив : локомотив;
вагоны : **LIST [0 : ?] OF** вагон;
END_ENTITY;

ENTITY элемент_подвижного_состава;
база : тип_колеи;
масса : **REAL**;
END_ENTITY;

ENTITY локомотив
SUBTYPE OF (элемент_подвижного_состава);
мощность : **REAL**;
коэффициент_полезного_действия : **REAL**;
END_ENTITY;

ENTITY паровоз
SUBTYPE OF (локомотив);
запас_воды : **REAL**;
запас_угля : **REAL**;
расход_воды : **REAL**;
расход_угля : **REAL**;
END_ENTITY;

ENTITY тепловоз
SUBTYPE OF (локомотив);
запас_топлива : **REAL**;
расход_топлива : **REAL**;
END_ENTITY;

ENTITY электровоз
SUBTYPE OF (локомотив);
напряжение : **REAL**;
сила_тока : **REAL**;
END_ENTITY;

ENTITY вагон
SUBTYPE OF (элемент_подвижного_состава);
END_ENTITY;

ENTITY товарный_вагон
SUBTYPE OF (вагон);
END_ENTITY;

ENTITY почтовый_вагон
SUBTYPE OF (вагон);
END_ENTITY;

ENTITY пассажирский_вагон
SUBTYPE OF (вагон);
END_ENTITY;

ENTITY плацкартный_вагон
SUBTYPE OF (пассажирский_вагон);
END_ENTITY;

ENTITY общий_вагон
SUBTYPE OF (плацкартный_вагон);
END_ENTITY;

ENTITY купейный_вагон
SUBTYPE OF (пассажирский_вагон);
END_ENTITY;

ENTITY вагон_СВ
SUBTYPE OF (пассажирский_вагон);
END_ENTITY;

ENTITY вагон_межобластного_сообщения

SUBTYPE OF (пассажирский_вагон);
END_ENTITY;

ENTITY вагон_ресторан
SUBTYPE OF (вагон);
END_ENTITY;

END_SCHEMA;

2 ЗАДАНИЯ ДЛЯ ВЫПОЛНЕНИЯ

Вариант №1. Построить информационную модель «Студенты» на языке EXPRESS.

1. Создать сущность «человек», состоящую из следующих атрибутов: Фамилия, Имя, Отчество, пол, дата рождения, место рождения, адрес (индекс, город, ул., дом, кв.), паспорт (серия, номер, дата выдачи, кем выдан).

2. Создать сущность «Студент», которая является потомком сущности «человек» и обладает дополнительными атрибутами: наименование вуза, специальность, шифр специальности, группа, номер студенческого.

Вариант №2. Построить информационную модель «Упаковка бумаги» на языке EXPRESS.

1. Создать сущность «Бумага», состоящую из следующих атрибутов: Формат (например А1, А2, А3 и др.), размер бумаги (высота и ширина), плотность бумаги.

2. Создать сущность «Производитель бумаги», состоящую из следующих атрибутов: Наименование, телефон, факс, e-mail, юр. адрес (индекс, город, ул., дом, ком.), ИНН.

3. Создать сущность «Пачка бумаги», которая является потомком сущностей «Бумага» и обладает дополнительными атрибутами: наименование бумаги, кол. листов в пачке, цена, производитель бумаги.

Вариант №3. Построить информационную модель «Сотрудники» на языке EXPRESS.

1. Создать сущность «Человек», состоящую из следующих атрибутов: Фамилия, Имя, Отчество, пол, дата рождения, место рождения, адрес (индекс, город, ул., дом, кв.), паспорт (серия, номер, дата выдачи, кем выдан).

2. Создать сущность «Сотрудник», которая является потомком сущности «человек» и обладает дополнительными атрибутами: наименование организации, дата приема на работу, дата увольнения, должность, отдел, табельный номер, оклад.

Вариант №4. Построить информационную модель «Аудитория» на языке EXPRESS.

1. Создать сущность «комната», состоящую из следующих атрибутов: Номер комнаты, площадь, высота потолка, ответственный (Ф.И.О., должность).

2. Создать сущность «Лекционная аудитория», которая является потомком сущности «комната» и обладает дополнительными атрибутами: количество посадочных мест, тип доски (например: меловая, маркерная, смешанная) наличие проектора (есть или нет).

3. Создать сущность «Административная аудитория», которая является потомком сущности «комната» и обладает дополнительными атрибутами: количество человек в аудитории, наименование отдела.

Вариант № 5. Построить информационную модель «Носители информации» на языке EXPRESS.

1. Создать сущность «носитель», состоящую из следующих атрибутов: объем информации, фирма производитель.

2. Создать сущность «упаковка» состоящую из следующих атрибута: количество в упаковке.

3. Создать сущность «Магнитный носитель», которая является потомком сущностей «Носитель» и «Упаковка»

4. Создать сущность «CD disk», которая является потомком сущности «Магнитный носитель» и обладает дополнительными атрибутами: тип (R или RW), макс скорость.

5. Создать сущность «DVD disk», которая является потомком сущности «CD disk».

Вариант №6. Построить информационную модель «Мототранспорт» на языке EXPRESS.

1. Создать сущность «Колесо» состоящую из следующих атрибутов: Диаметр, ширина.

2. Создать сущность «Велосипед», состоящую из следующих атрибутов: колесо, количество колес, багажник (есть или нет), масса.

3. Создать сущность «Спортивный велосипед», которая является потомком сущности «Велосипед» и обладает дополнительными атрибутами: количество скоростей. Сделать проверку на то, что должно быть два колеса.

4. Создать сущность «Мотоцикл», которая является потомком сущности «Велосипед» и обладает дополнительными атрибутами: количество скоростей, мощность двигателя, количество посадочных мест. Сделать проверку на то, что должно быть два колеса, а количество посадочных мест не менее одного.

Вариант №7. Создайте модель на языке Express для ситуации: Фрагмент (fragment) 2D-изображения представлен окружностью (circle) или отрезком кривой (curve), кроме того, свойством фрагмента является цвет (colour). Окружность характеризуется радиусом (rad), отрезок кривой - тремя точками (P).

Вариант № 8. Создайте модель на языке Express для ситуации: Сущность Локальная вычислительная сеть (LAN) имеет атрибуты Метод доступа (access), Число узлов (numb_nodes) и наследует параметр Топология (topology) от сущности Структура (structure) и параметр rang от сущности Управление (control).

Вариант №9. Создайте модель на языке Express для ситуации:

Проект (project) состоит из пояснительной записки (text) и графического материала (graphics), атрибуты проекта - автор (author), название (titul), атрибуты записки - число страниц (pages) и аннотация (annot), атрибуты графики - число листов (numb) и форма представления (present). Число листов должно быть больше 9.

Вариант №10. Создайте модель на языке Express в виде двух схем для ситуации:

Первая схема - сущность person с атрибутами first_name, family типа string; атрибутами age, tabl типа integer; атрибутом sex типа enumeration со значениями man и woman;

Вторая схема - сущность specialist с атрибутом position типа subdivision (в свою очередь, имеющего атрибут ind типа string) и атрибутом name_sp типа employee со ссылкой на сущность person из первой схемы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ипатова, Э. Р. Методологии и технологии системного проектирования информационных систем : учебник / Э. Р. Ипатова, Ю. В. Ипатов. – 3-е изд., стер. – Москва : ФЛИНТА, 2021. – 256 с. : табл., схем. – (Информационные технологии). – Режим доступа: по подписке. –

URL: <https://biblioclub.ru/index.php?page=book&id=79551> (дата обращения: 12.04.2023). – Библиогр.: с. 95-96. – ISBN 978-5-89349-978-0. – Текст : электронный.

2. Схиртладзе, А. Г. Проектирование единого информационного пространства виртуальных предприятий : учебник : [16+] / А. Г. Схиртладзе, А. В. Скворцов, Д. А. Чмырь. – Изд. 2-е, стер. – Москва ; Берлин : Директ-Медиа, 2017. – 617 с. : ил., схем., табл. – Режим доступа: по подписке. –

URL: <https://biblioclub.ru/index.php?page=book&id=469047> (дата обращения: 12.04.2023). – Библиогр.: с. 606. – ISBN 978-5-4475-8634-8. – DOI 10.23681/469047. – Текст : электронный.

3. Эйхман, Т. П. Интегрированная информационная поддержка жизненного цикла наукоемких изделий в самолето- и вертолетостроении : учебное пособие : [16+] / Т. П. Эйхман, Н. В. Курлаев ; Новосибирский государственный технический университет. – Новосибирск : Новосибирский государственный технический университет, 2013. – 148 с. – Режим доступа: по подписке. –

URL: <https://biblioclub.ru/index.php?page=book&id=228916> (дата обращения: 12.04.2023). – ISBN 978-5-7782-2221-2. – Текст : электронный.

4. Совершенствование процесса изготовления сложных изделий с использованием PDM-систем : учебное пособие / В. Кузнецова, А. И. Сергеев, А. И. Сердюк, А. В. Попов ; Оренбургский государственный университет. – Оренбург : Оренбургский государственный университет, 2013. – 144 с. – Режим доступа: по подписке. –

URL: <https://biblioclub.ru/index.php?page=book&id=259356> (дата обращения: 12.04.2023). – Библиогр. в кн. – Текст : электронный.

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра «Машиностроительные технологии и оборудование»

УТВЕРЖДАЮ:
Проректор по учебной работе

« 10 » 03



**ОПИСАНИЕ СТРУКТУРЫ ОБЪЕКТОВ
НА ЯЗЫКЕ EXPRESS-G**

Методические указания к выполнению практической работы по дисциплине «Информационная поддержка жизненного цикла продукции» для студентов направления подготовки 15.03.01 «Машиностроение» направленность (профиль) «Технология, оборудование и автоматизация машиностроительных производств»

УДК 519.6

Составители: В.В. Куц, А.Г. Ивахненко

Рецензент

Кандидат технических наук, доцент кафедры
«Машиностроительные технологии и оборудования» А.Н. Гречухин

Описание структуры объектов на языке EXPRESS-G:
методические указания к выполнению практической работы по дисциплине «Информационная поддержка жизненного цикла продукции» по направлению подготовки 15.03.01 «Машиностроение» направленность (профиль) «Технология, оборудование и автоматизация машиностроительных производств» / Юго-Зап. гос. ун-т; сост.: В.В. Куц, А.Г. Ивахненко. Курск, 2023. 20 с. Библиогр.: с. 20.

Излагаются краткие сведения о структуре информационных моделей графических объектов на языке EXPRESS-G. Приводятся варианты заданий для выполнения практических занятий по дисциплине «Информационная поддержка жизненного цикла продукции».

Предназначены для обучающихся по направлению подготовки 15.03.01 «Машиностроение» направленность (профиль) «Технология, оборудование и автоматизация машиностроительных производств».

Текст печатается в авторской редакции

Подписано в печать . Формат 60×84 1/16.
Усл. печ. л. . Уч. - изд. л. . Тираж 50 экз. Заказ 384.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

1 СТРУКТУРА МОДЕЛЕЙ НА ЯЗЫКЕ EXPRESS-G

В языке Express-G схема представляется прямоугольником с разделительной горизонтальной линией, над этой линией записывается имя схемы, как это показано на рис. 1.

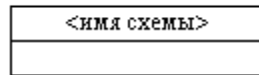


Рис. 1. Изображение схемы в языке Express-G

Сущности изображаются прямоугольниками, внутри прямоугольника записывается имя сущности (рис. 2).

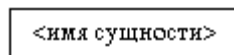


Рис. 2. Изображение сущности в языке Express-G

Изображение атрибутов в Express-G поясняет рис. 3, из которого, в частности, следует, что атрибут представлен прямоугольником, а связи "сущность-атрибут" или "сущность-сущность" отображаются линиями, причем в случае связи с optional атрибутом используется пунктирная линия. Направление связи обозначается окружностью на конце линии, ведущей к атрибуту. Имя атрибута записывается рядом с этой линией. В прямоугольнике атрибута записывается тип атрибута.

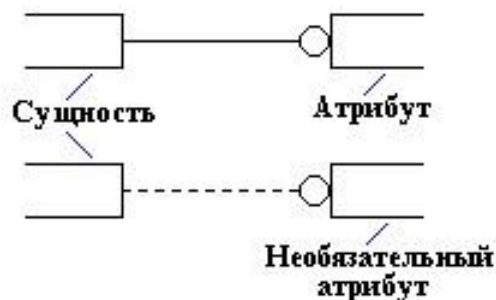


Рис. 3. Изображение атрибутов в языке Express-G

Изображения простых типов на языке Express-G показаны на рис. 4.

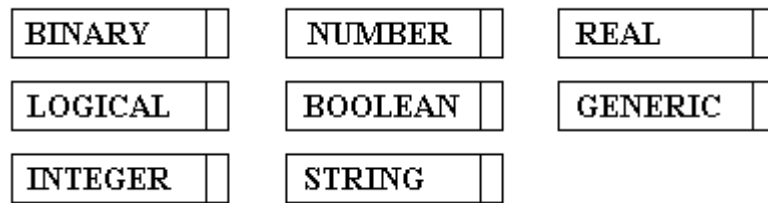


Рис. 4. Изображения простых типов в языке Express-G

Записи вида `array[2:8]` или `list[1:?]` в Express-G преобразуются в форму `A[2:8]` или `L[1:?]`, указываемую около линии атрибута агрегативного типа после имени этого атрибута. Так, первый из вышеприведенных примеров представлен на рис. 5.

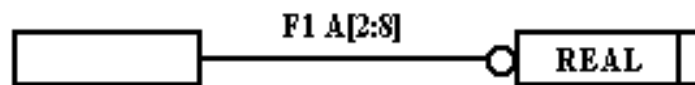


Рис. 5. Пример изображения агрегативного типа в языке Express-G

Графические изображения определяемых, нечисловых и выделяемых типов данных показаны на рис. 6. Внутри прямоугольников, ограничиваемых пунктирными линиями, записывается имя типа.

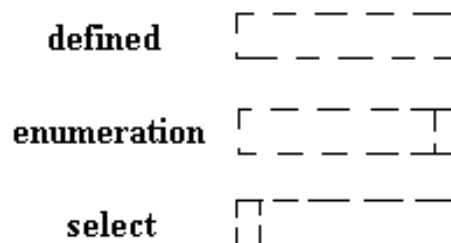


Рис. 6. Изображения определяемых, нечисловых и выделяемых типов данных в языке Express-G

Пример графического изображения супертипов и подтипов в языке Express-G приведен на рис. 7. (соединение показывают жирными линиями).

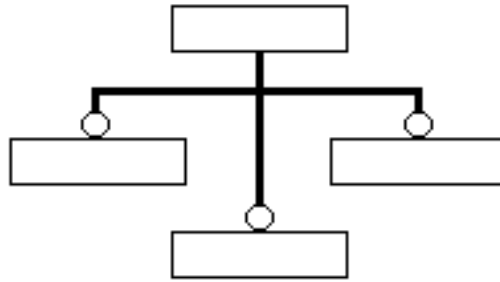


Рис. 7. Изображение супертипов и подтипов в языке Express-G

Пример 1.

Моделируется ситуация: Фрагмент (fragment) 2D-изображения представлен окружностью (cycle) или отрезком кривой (curve), кроме того, свойством фрагмента является цвет (colour). Окружность характеризуется радиусом (rad), отрезок кривой - тремя точками (P).

Полученная модель на языке Express имеет вид:

```

schema DIAG2;
type form = select(cycle,curve);
end_type;
type colour = enumeration of (red, blue, white);
end_type;
entity cycle;
rad: real;
end_entity;
entity curve;
P: ARRAY [1:3] OF point;
end_entity;
entity point;
X: ARRAY [1:2] OF REAL;
end_entity;
entity fragment;
figure: form; image: colour;
end_entity;
end_schema;

```

Модель на языке Express-G представлена на рис.8.

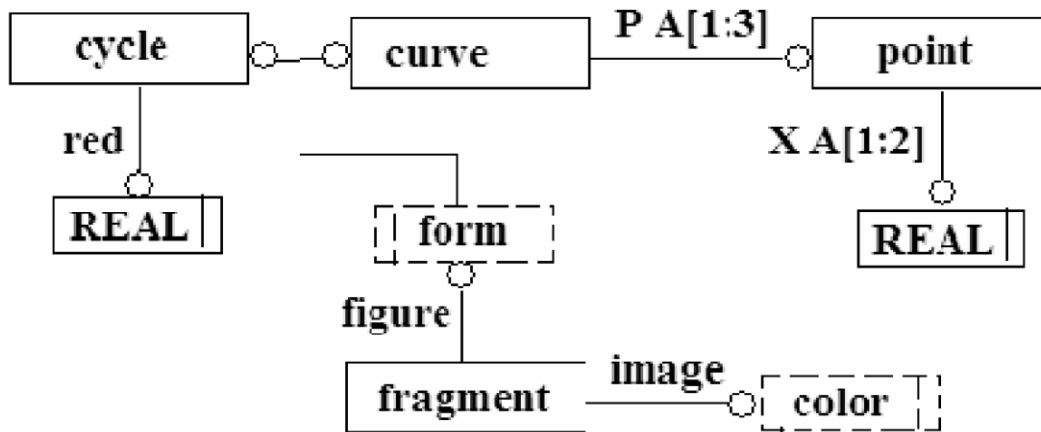


Рис. 8. Диаграмма примера 1 на языке Express-G

Пример 2

Дана модель на языке Express-G в виде схемы рис.9. Соответствующая ей модель на языке Express представлена ниже.

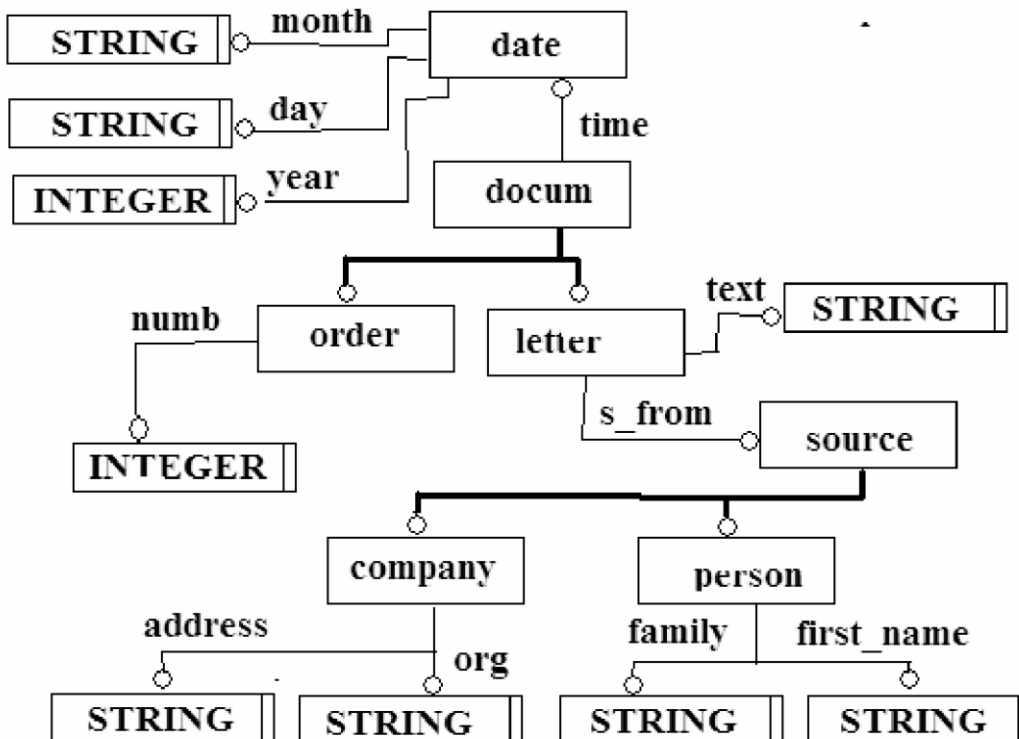


Рис. 9. Диаграмма примера 2 на языке Express-G

```

schema OFF;
entity docum;
time: date;
end_entity;

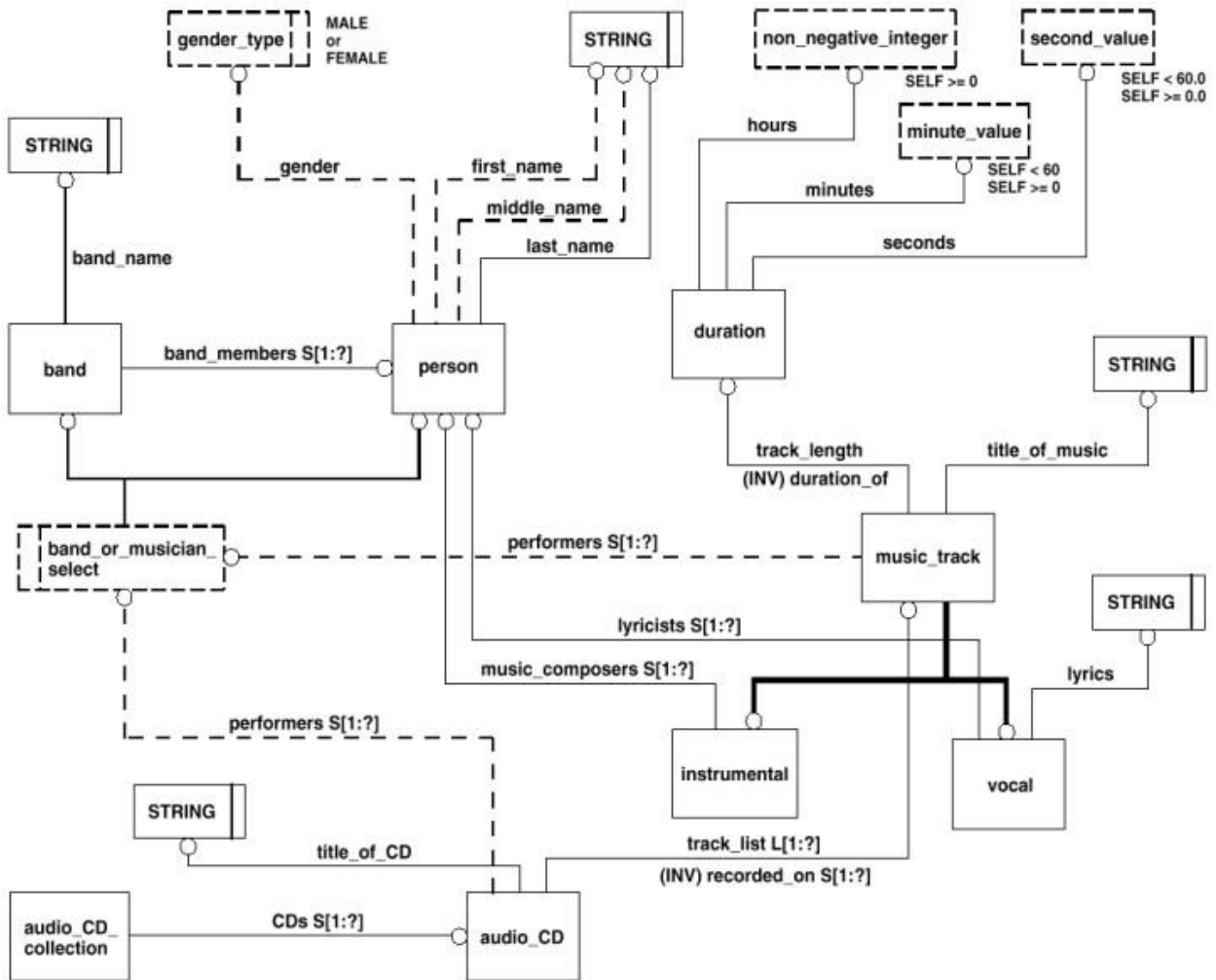
```

entity letter
subtype of
(docum);
text: **string**;
s_from: source;
end_entity;
40
entity order
subtype of (docum);
numb: **integer**;
end_entity;
entity source;
end_entity;
entity company **subtype of** (source);
address: **string**;
org: **string**;
end_entity;
entity person **subtype of** (source);
family: **string**;
first_name: **string**;
end_entity;
entity date;
month: **string**;
day: **integer**;
year: **integer**;
end_entity;
end_schema;

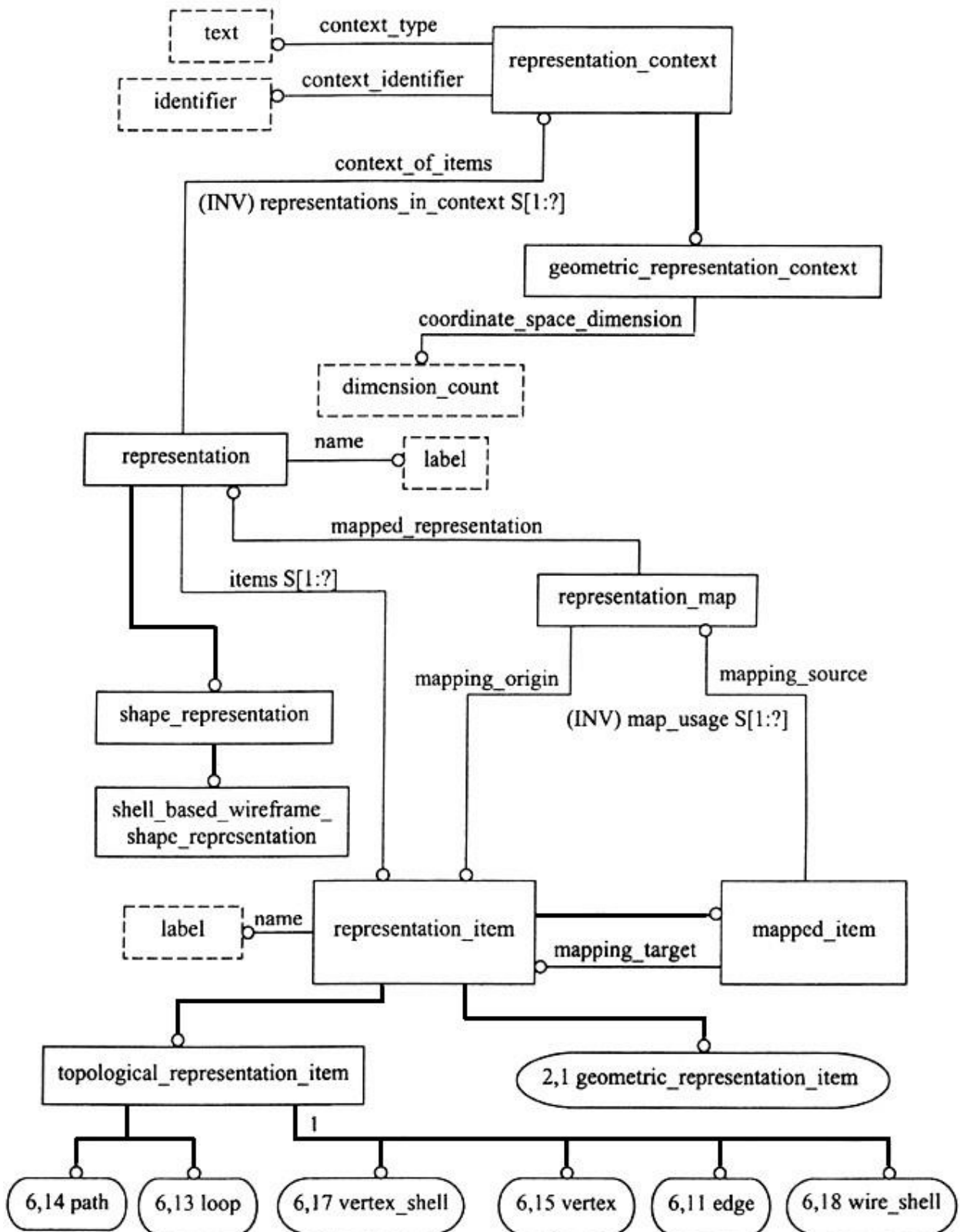
2 ЗАДАНИЯ ДЛЯ ВЫПОЛНЕНИЯ

По приведенным моделям в виде схем Express-G составить соответствующие им модели на языке Express.

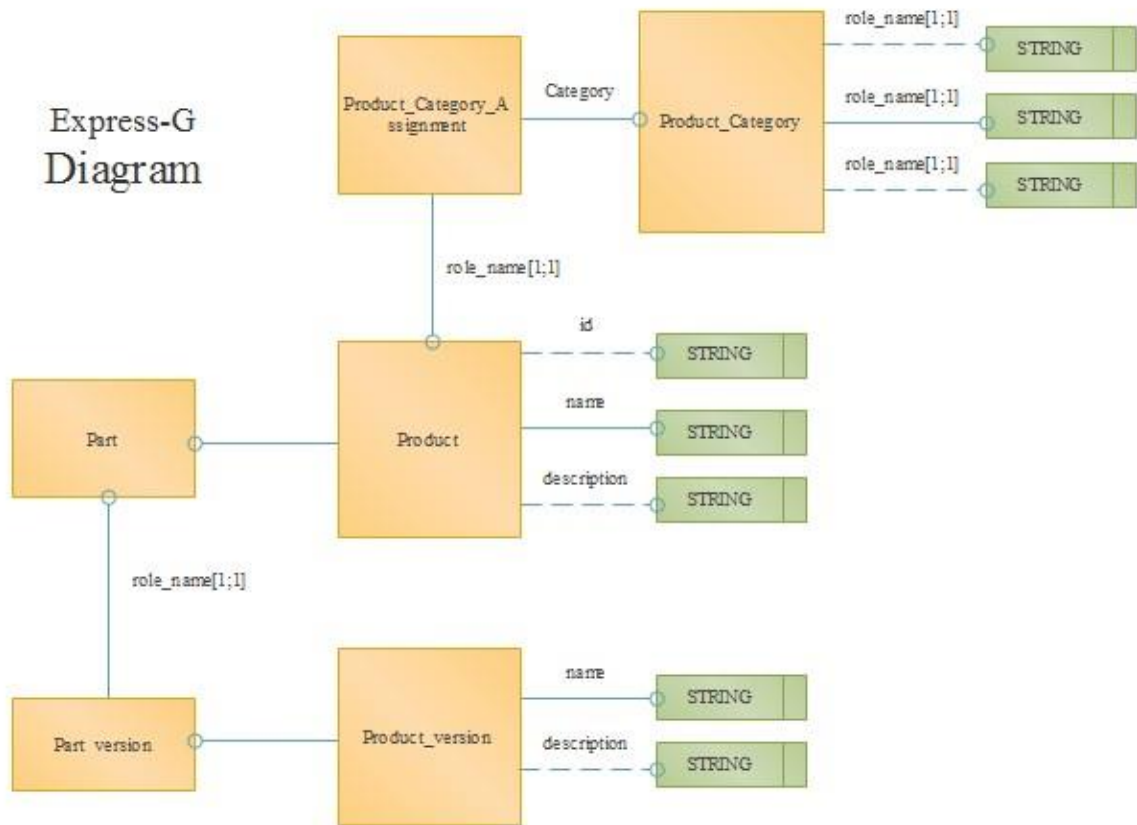
Вариант 1



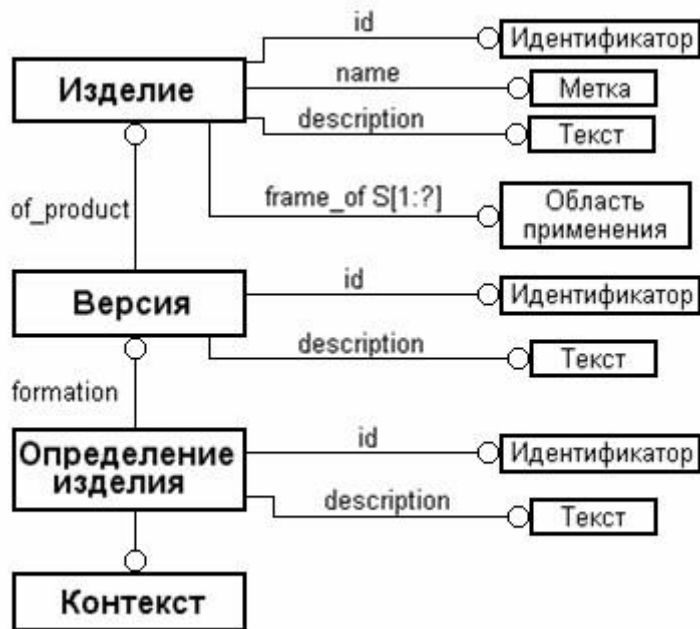
Вариант 2



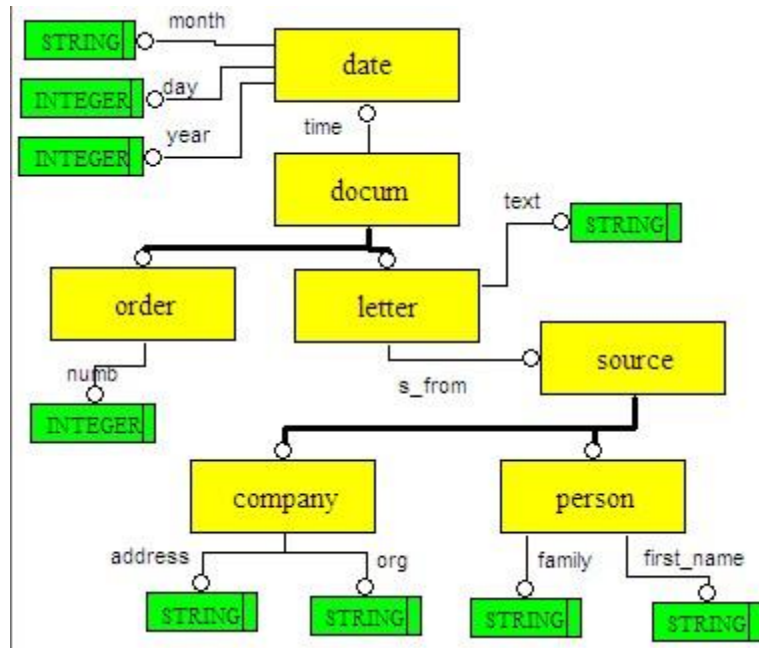
Вариант 3



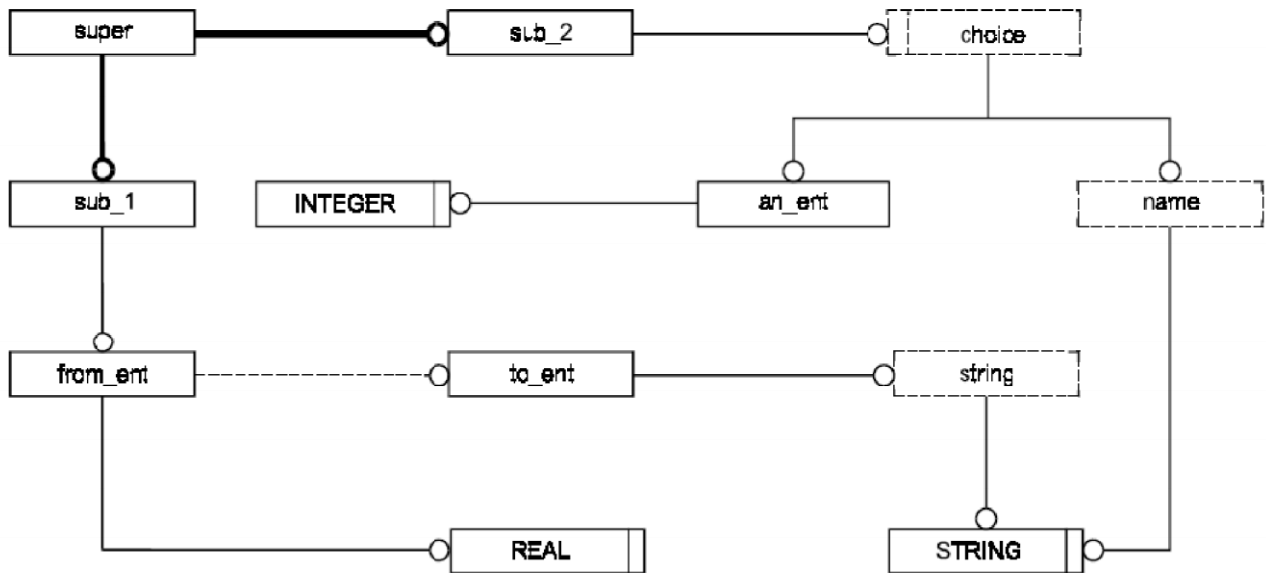
Вариант 4



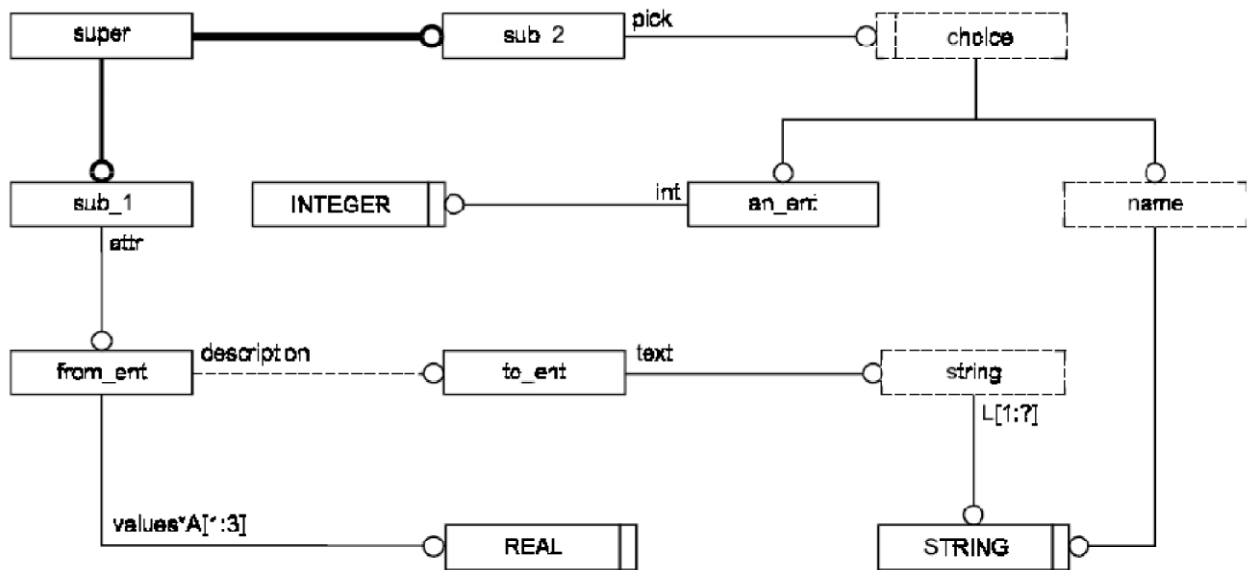
Вариант 5



Вариант 6



Вариант 7



По приведенным моделям на языке Express составить соответствующие им модели в виде схем Express-G.

Вариант 8

```

SCHEMA IZDELIE;
ENTITY date;
month : INTEGER;
day : INTEGER;
year : INTEGER;
END_ENTITY;
//изделие
ENTITY Izd;
Data_izg_izd : date;
Seb_izd : REAL;
D_izd : REAL;
Sh_izd : REAL;
V_izd : REAL;
END_ENTITY;
//корпус верхняя часть
ENTITY Korp_v;

```

```
SUBTYPE OF(Izd);
Data_izg_korp_v : date;
Mat_korp_v : STRING;
D_korp_v : REAL;
Sh_korp_v : REAL;
V_korp_v : REAL;
END_ENTITY;
//корпус нижняя часть
ENTITY Korp_n;
SUBTYPE OF(Izd);
Data_izg_korp_n : date;
Mat_korp_n : STRING;
D_korp_n : REAL;
Sh_korp_n : REAL;
V_korp_n : REAL;
END_ENTITY;
//Аккумулятор
ENTITY Akkum;
SUBTYPE OF(Korp_n);
M: INTEGER;
SA : REAL;
PA : STRING;
Data_postA : date;
END_ENTITY;
//USB
ENTITY USB;
SUBTYPE OF(Korp_n);
SU : REAL;
PU : STRING;
Data_postU : date;
END_ENTITY;
//Провода
ENTITY Provod;
SUBTYPE OF(Korp_n);
SP : REAL;
PP : STRING;
Data_postP : date;
```

```
END_ENTITY;  
//Вилка  
ENTITY Vilka;  
SUBTYPE OF(Korp_n);  
SV : REAL;  
PV : STRING;  
Data_postV : date;  
END_ENTITY;  
//Документы на изделие ENTITY Doc_izd;  
SUBTYPE OF(Izd);  
END_ENTITY;  
//3D модель изделия  
ENTITY 3D_izd;  
SUBTYPE OF(Doc_izd);  
A_3D_izd : STRING;  
Data_3D_izd : date;  
END_ENTITY;  
//Документы на верхнюю часть корпуса  
ENTITY Doc_korp_v;  
SUBTYPE OF(Korp_v);  
END_ENTITY;  
//3D модель верхней части корпуса  
ENTITY 3D_korp_v;  
SUBTYPE OF(Doc_korp_v);  
A_3D_korp_v : STRING;  
Data_3D_korp_v : date;  
END_ENTITY;  
//Чертеж верхней части корпуса  
ENTITY Chert_korp_v;  
SUBTYPE OF(Doc_korp_v);  
A_Chert_korp_v : STRING;  
Data_Chert_korp_v : date;  
END_ENTITY;  
//Документы на нижнюю часть корпуса  
ENTITY Doc_korp_n;  
SUBTYPE OF(Korp_v);  
END_ENTITY;
```

```

//3D модель нижней части корпуса
ENTITY 3D_korp_n;
SUBTYPE OF(Doc_korp_n);
A_3D_korp_n : STRING;
Data_3D_korp_n : date;
END_ENTITY;
//Чертеж нижней части корпуса
ENTITY Chert_korp_n;
SUBTYPE OF(Doc_korp_n);
A_Chert_korp_n : STRING;
Data_Chert_korp_n : date;
END_ENTITY;
//Маршрутная карта
ENTITY МК;
SUBTYPE OF(Doc_izd, Doc_korp_v, Doc_korp_n);
A_МК : STRING;
Data_МК : date;
END_ENTITY;
//ТП сборки
ENTITY TPS;
SUBTYPE OF(Doc_izd, Doc_korp_n);
A_TPS : STRING;
Data_TPS : date;
END_ENTITY;
//ТП изготовления корпуса
ENTITY TPI;
SUBTYPE OF(Doc_izd, Doc_korp_v, Doc_korp_n);
A_TPI : STRING;
Data_TPI : date;
END_ENTITY;
END_SCHEMA;

```

Вариант 9

```

SCHEMA ActorResource;
TYPE ActorSelect = SELECT (Organization, Person);
END_TYPE;

```

```

TYPE AddressTypeEnum = ENUMERATION OF (
END_TYPE;
TYPE Label = STRING(255);
END_TYPE;
TYPE ActorRole = Label;
END_TYPE;
ENTITY Address
ABSTRACT SUPERTYPE OF (ONEOF(PostalAddress,
TelecomAddress));
Purpose : AddressTypeEnum;
UserDefinedPurpose : OPTIONAL STRING;
INVERSE
OfPerson : SET OF Person FOR Addresses;
OfOrganization : SET OF Organization FOR Addresses;
WHERE
WR1 : (Purpose <> AddressTypeEnum.USERDEFINED) OR
((Purpose = AddressTypeEnum.USERDEFINED) AND
EXISTS(UserDefinedPurpose));
END_ENTITY;
ENTITY PostalAddress
SUBTYPE OF(Address);
AddressLines : LIST [1:?] OF Label;
END_ENTITY;
ENTITY TelecomAddress
SUBTYPE OF(Address);
TelephoneNumbers : OPTIONAL LIST [1:?] OF Label;
FacsimileNumbers : OPTIONAL LIST [1:?] OF Label;
ElectronicMailAddresses : OPTIONAL LIST [1:?] OF Label;
WWWUrls : OPTIONAL LIST [1:?] OF Label;
WHERE
WR1 : EXISTS (TelephoneNumbers) OR EXISTS (Facsimile-
Numbers) OR
EXISTS (ElectronicMailAddresses) OR EXISTS (WWWUrls);
END_ENTITY;
ENTITY Organization;
Id : INTEGER;
Name : Label;

```

```

Description : OPTIONAL STRING;
Roles : LIST [0:?] OF UNIQUE ActorRole;
Addresses : LIST [1:?] OF UNIQUE Address;
INVERSE
  IsRelatedBy : SET OF OrganizationRelationship FOR
RelatedOrganizations;
  Relates : SET OF OrganizationRelationship FOR RelatingOrga-
nization;
  Engages : SET OF Person FOR EngagedIn;
UNIQUE
  UR1 : Id;
END_ENTITY;
ENTITY OrganizationRelationship;
  Name : Label;
  Description : OPTIONAL STRING;
  RelatingOrganization : Organization;
  RelatedOrganizations : SET [1:?] OF Organization;
END_ENTITY;
ENTITY Person;
  Id : INTEGER;
  FamilyName : OPTIONAL Label;
  GivenName : OPTIONAL Label;
  MiddleNames : OPTIONAL LIST [1:?] OF Label;
  PrefixTitles : OPTIONAL LIST [1:?] OF Label;
  SuffixTitles : OPTIONAL LIST [1:?] OF Label;
  Roles : LIST [0:?] OF UNIQUE ActorRole;
  Addresses : OPTIONAL LIST [1:?] OF UNIQUE Address;
  EngagedIn : SET OF Organization;
UNIQUE
  UR1 : Id;
WHERE
  WR1 : EXISTS(FamilyName) OR EXISTS(GivenName);
END_ENTITY;
END_SCHEMA;

```

Вариант 10

```

SCHEMA person_organization_schema;
ENTITY address;
internal_location : optional label;
street_number : optional label;
street : optional label;
postal_box : optional label;
town : optional label;
region : optional label;
postal_code : optional label;
country : optional label;
facsimile_number : optional label;
telephone_number : optional label;
electronic_mail+address : optional label;
telex_number : optional label;
WHERE
wr1 : EXISTS(internal_location) OR EXISTS(street_number)
OR
EXISTS(street)
OR EXISTS(postal_box) OR EXISTS(town) OR
EXISTS(region)
OR EXISTS(postal_code)
OR EXISTS(country) OR EXISTS(facsimile_number) OR
EXISTS(telephone_number)
OR EXISTS(electronic_mail_address) OR EX-
ISTS(telex_number);
END_ENTITY;
ENTITY personal_address
SUBTYPE OF (address);
people : SET[1:?] OF person;
description : text;
END_ENTITY;
ENTITY person;
id : identifier;
last_name : OPTIONAL label;
first_name : OPTIONAL label;

```

```
middle_names : OPTIONAL LIST[1:?] OF label;  
prefix_titles : OPTIONAL LIST[1:?] OF label;  
suffix_titles : OPTIONAL LIST[1:?] OF label;  
UNIQUE  
ur1 : id;  
WHERE  
wr1 : exists(last_name) OR exists(first_name);  
END_ENTITY;  
END_SCHEMA;
```


БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ипатова, Э. Р. Методологии и технологии системного проектирования информационных систем : учебник / Э. Р. Ипатова, Ю. В. Ипатов. – 3-е изд., стер. – Москва : ФЛИНТА, 2021. – 256 с. : табл., схем. – (Информационные технологии). – Режим доступа: по подписке. –

URL: <https://biblioclub.ru/index.php?page=book&id=79551> (дата обращения: 12.04.2023). – Библиогр.: с. 95-96. – ISBN 978-5-89349-978-0. – Текст : электронный.

2. Схиртладзе, А. Г. Проектирование единого информационного пространства виртуальных предприятий : учебник : [16+] / А. Г. Схиртладзе, А. В. Скворцов, Д. А. Чмырь. – Изд. 2-е, стер. – Москва ; Берлин : Директ-Медиа, 2017. – 617 с. : ил., схем., табл. – Режим доступа: по подписке. –

URL: <https://biblioclub.ru/index.php?page=book&id=469047> (дата обращения: 12.04.2023). – Библиогр.: с. 606. – ISBN 978-5-4475-8634-8. – DOI 10.23681/469047. – Текст : электронный.

3. Эйхман, Т. П. Интегрированная информационная поддержка жизненного цикла наукоемких изделий в самолето- и вертолетостроении : учебное пособие : [16+] / Т. П. Эйхман, Н. В. Курлаев ; Новосибирский государственный технический университет. – Новосибирск : Новосибирский государственный технический университет, 2013. – 148 с. – Режим доступа: по подписке. –

URL: <https://biblioclub.ru/index.php?page=book&id=228916> (дата обращения: 12.04.2023). – ISBN 978-5-7782-2221-2. – Текст : электронный.

4. Совершенствование процесса изготовления сложных изделий с использованием PDM-систем : учебное пособие / В. Кузнецова, А. И. Сергеев, А. И. Сердюк, А. В. Попов ; Оренбургский государственный университет. – Оренбург : Оренбургский государственный университет, 2013. – 144 с. – Режим доступа: по подписке. –

URL: <https://biblioclub.ru/index.php?page=book&id=259356> (дата обращения: 12.04.2023). – Библиогр. в кн. – Текст : электронный.

УДК 519.6

Составители: В.В. Куц, А.Г. Ивахненко

Рецензент

Кандидат технических наук, доцент кафедры
«Машиностроительные технологии и оборудования» А.Н. Гречухин

Изучение структуры и создание символьного обменного файла по ГОСТ Р ИСО 10303-21-2022: методические указания к выполнению практической работы по дисциплине «Информационная поддержка жизненного цикла продукции» по направлению подготовки 15.03.01 «Машиностроение» направленность (профиль) «Технология, оборудование и автоматизация машиностроительных производств» / Юго-Зап. гос. ун-т; сост.: В.В. Куц, А.Г. Ивахненко. Курск, 2023. 8 с. Библиогр.: с. 8.

Излагаются краткие сведения о структуре символьного обменного файла по ГОСТ Р ИСО 10303-21-2022. Приводятся варианты заданий для выполнения практических занятий по дисциплине «Информационная поддержка жизненного цикла продукции».

Предназначены для обучающихся по направлению подготовки 15.03.01 «Машиностроение» направленность (профиль) «Технология, оборудование и автоматизация машиностроительных производств».

Текст печатается в авторской редакции

Подписано в печать . Формат 60×84 1/16.
Усл. печ. л. . Уч. - изд. л. . Тираж 50 экз. Заказ 385.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

В методических указаниях к выполнению практической работы «Описание структуры объектов на языке EXPRESS» изложены краткие сведения о структуре информационных моделей, которые следует использовать при выполнении настоящей практической работы. Далее рассмотрены сведения о структуре символьного обменного файла по ГОСТ Р ИСО 10303-21-2022, необходимые для выполнения задания.

1 КРАТКИЕ СВЕДЕНИЯ О СТРУКТУРЕ СИМВОЛЬНОГО ОБМЕННОГО ФАЙЛА

Стандарт ГОСТ Р ИСО 10303-21-2022 (п.1) устанавливает формат структуры обмена, использующий кодирование открытым текстом данных об изделии, для которого концептуальная модель определена в языке EXPRESS (ИСО 10303-11). Формат обмена пригоден для передачи данных об изделии между вычислительными системами и для распределения данных об изделии между несколькими вычислительными системами. Определено преобразование из языка EXPRESS в синтаксис структуры обмена. В синтаксис структуры обмена может быть преобразована любая EXPRESS-схема.

Для того чтобы облегчить синтаксический анализ с помощью программных средств, структура обмена описана однозначной, контекстно-свободной грамматикой (п. 4.1). Грамматика выражена в синтаксической нотации Вирта, которая описана в приложении В. Представление данных об изделии в структуре обмена определяется с использованием отображения из языка EXPRESS в синтаксис структуры обмена.

Структура обмена должна быть представлена в виде последовательного файла с использованием кодирования открытым текстом (п. 5.3). Структура обмена должна содержать заголовочную секцию и четыре необязательных секций данных: секция привязки, ссылочная секция, одна или более секций данных и одна или более секций подписи. Роль каждой секции описывается ниже в том же порядке, в котором они отображаются в структуре обмена. Заголовочная секция представляет данные, относящиеся к самой

структуре обмена. Структура заголовочной секции определена в п. 8 стандарта. В секции привязки предоставлены внешние имена для объектов и значений, на которые возможны ссылки из других структур обмена. Определение структуры секции привязки содержится в п. 9 стандарта. В секции ссылок предоставлены ссылки на объекты и значения, которые определены во внешних структурах обмена. Определение структуры секции ссылок содержится в п. 10 стандарта. Секция данных представляет данные, которые должны быть переданы. Структура секции данных определена в п. 11. Секция подписи обеспечивает достоверность передаваемых данных и подтверждает источник данных. Определение структуры секции подписи содержится в п. 14. Структура обмена определена с помощью WSN – синтаксической нотации Вирта (Wirth Syntax Notation), основы которой представлены в табл. 1.

Таблица 1

EXCHANGE_FILE	=	"ISO-10303-21;" HEADER_SECTION [ANCHOR_SECTION] [REFERENCE_SECTION] { DATA_SECTION } "END-ISO-10303-21;" { SIGNATURE_SECTION }.
HEADER_SECTION	=	"HEADER;" HEADER_ENTITY HEADER_ENTITY HEADER_ENTITY [HEADER_ENTITY_LIST] "ENDSEC;" .
HEADER_ENTITY_LIST	=	HEADER_ENTITY { HEADER_ENTITY } .
HEADER_ENTITY	=	KEYWORD "(" [PARAMETER_LIST] ")" ";" .
PARAMETER_LIST	=	PARAMETER { ";" PARAMETER } .
PARAMETER	=	TYPED_PARAMETER UNTYPED_PARAMETER OMITTED_PARAMETER .
TYPED_PARAMETER	=	KEYWORD "(" PARAMETER ")" .
UNTYPED_PARAMETER		"\$" INTEGER REAL STRING RHS_OCCURENCE_NAME ENUMERATION BINARY LIST .
LIST	=	"(" [PARAMETER { PARAMETER }] ")" .
DATA_SECTION	=	"DATA" ["(" PARAMETER_LIST ")"] ";" ENTITY_INSTANCE_LIST "ENDSEC;" .

Структура секции данных (**DATA_SECTION**) является основой для выполнения заданий.

2 ЗАДАНИЯ ДЛЯ ВЫПОЛНЕНИЯ

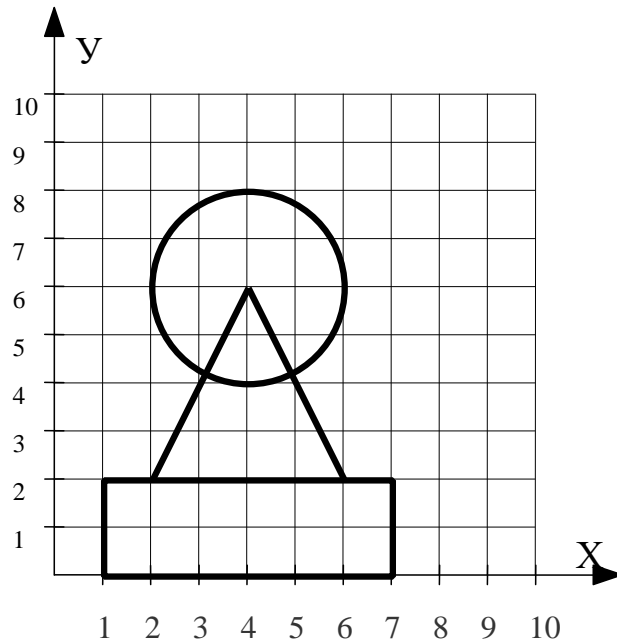
Создать символьный обменный файл по рисунку

Задание №1.

```

SCHEMA example_geometry;
ENTITY point;
x : REAL;
y : REAL;
END_ENTITY;
ENTITY line;
p0 : point;
p1 : point;
END_ENTITY;
ENTITY circle;
radius : REAL;
center : point;
END_ENTITY;
END_SCHEMA;

```

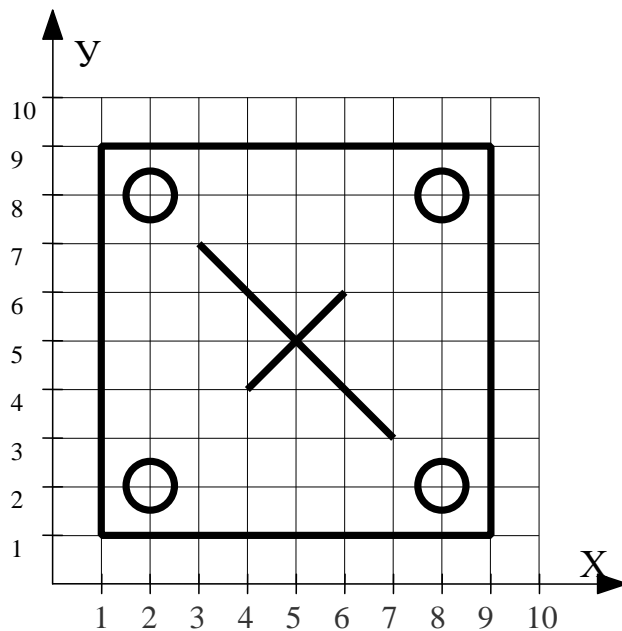


Задание №2.

```

SCHEMA example_geometry;
ENTITY точка;
x : REAL;
y : REAL;
END_ENTITY;
ENTITY линия;
x1,y1 : REAL;
x2,y2 : REAL;
END_ENTITY;
ENTITY окружность;
center : точка;
radius : REAL;
END_ENTITY;
END_SCHEMA;

```

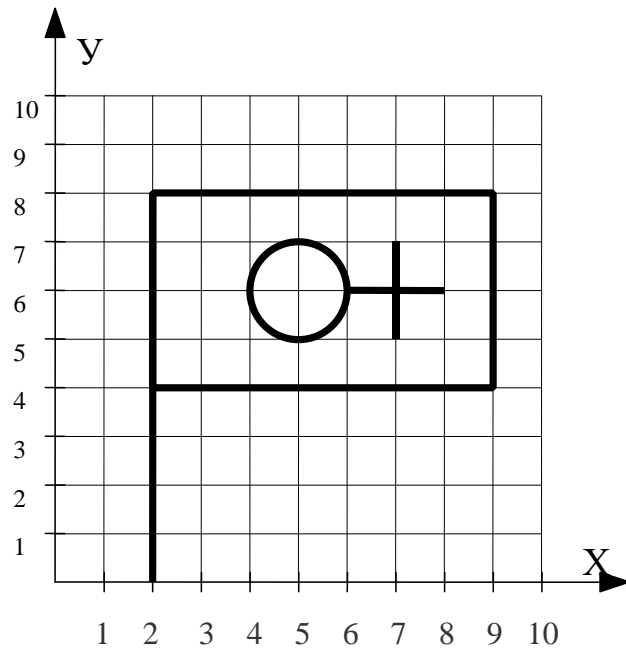


Задание №3.

```

SCHEMA example_geometry;
ENTITY точка;
x : REAL;
y : REAL;
END_ENTITY;
ENTITY линия;
x1,y1 : REAL;
x2,y2 : REAL;
END_ENTITY;
ENTITY окружность;
Xc,Yc : REAL;
radius : REAL;
END_ENTITY;
END_SCHEMA;

```

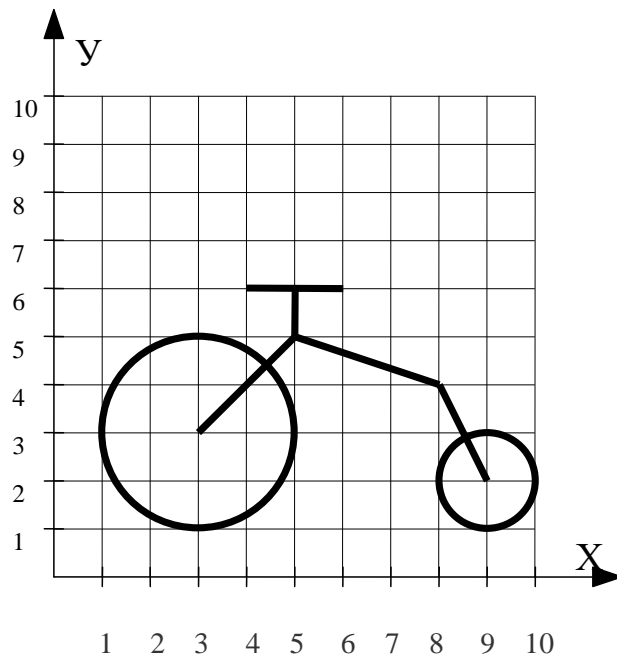


Задание №4.

```

SCHEMA example_geometry;
ENTITY point;
x : REAL;
y : REAL;
END_ENTITY;
ENTITY line;
p0 : point;
p1 : point;
END_ENTITY;
ENTITY circle;
radius : REAL;
Xc, Yc : REAL;
END_ENTITY;
END_SCHEMA;

```

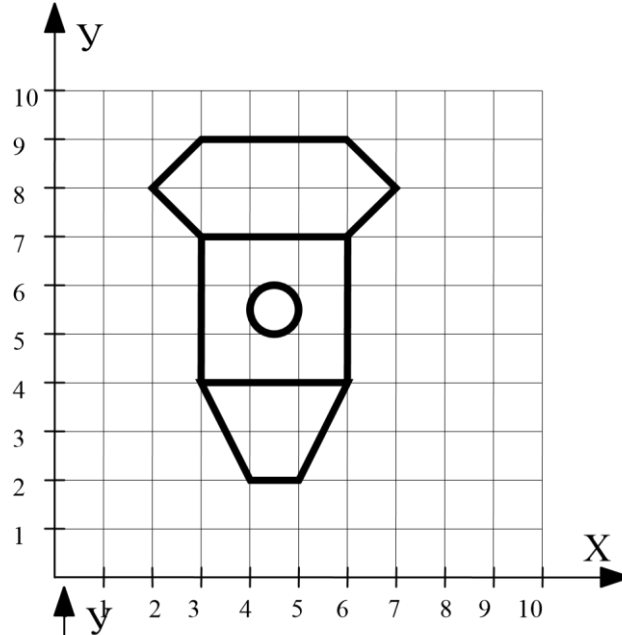


Задание №5.

```

SCHEMA example_geometry;
ENTITY point;
x : REAL;
y : REAL;
END_ENTITY;
ENTITY line;
p0 : point;
p1 : point;
END_ENTITY;
ENTITY circle;
radius : REAL;
Xc, Yc : REAL;
END_ENTITY;
END_SCHEMA;

```

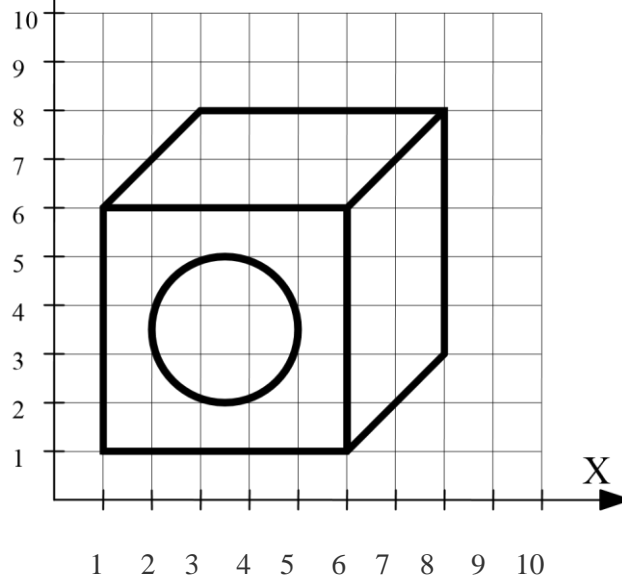


Задание №6.

```

SCHEMA example_geometry;
ENTITY point;
x : REAL;
y : REAL;
END_ENTITY;
ENTITY line;
p0 : point;
p1 : point;
END_ENTITY;
ENTITY circle;
radius : REAL;
Xc, Yc : REAL;
END_ENTITY;
END_SCHEMA;

```



БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ипатова, Э. Р. Методологии и технологии системного проектирования информационных систем : учебник / Э. Р. Ипатова, Ю. В. Ипатов. – 3-е изд., стер. – Москва : ФЛИНТА, 2021. – 256 с. : табл., схем. – (Информационные технологии). – Режим доступа: по подписке. –

URL: <https://biblioclub.ru/index.php?page=book&id=79551> (дата обращения: 12.04.2023). – Библиогр.: с. 95-96. – ISBN 978-5-89349-978-0. – Текст : электронный.

2. Схиртладзе, А. Г. Проектирование единого информационного пространства виртуальных предприятий : учебник : [16+] / А. Г. Схиртладзе, А. В. Скворцов, Д. А. Чмырь. – Изд. 2-е, стер. – Москва ; Берлин : Директ-Медиа, 2017. – 617 с. : ил., схем., табл. – Режим доступа: по подписке. –

URL: <https://biblioclub.ru/index.php?page=book&id=469047> (дата обращения: 12.04.2023). – Библиогр.: с. 606. – ISBN 978-5-4475-8634-8. – DOI 10.23681/469047. – Текст : электронный.

3. ГОСТ Р ИСО 10303-21 – 2022. Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 21. Методы реализации. Кодирование открытым текстом структуры обмена. М.: Российский институт стандартизации, 2022. – 84 с.

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра «Машиностроительные технологии и оборудование»

УТВЕРЖДАЮ
Проректор по учебной работе

« 10 » 05
О.Г. Локтинова
10.2023



**АНАЛИЗ СТРУКТУРЫ СИМВОЛЬНОГО ОБМЕННОГО
ФАЙЛА ПО ГОСТ Р ИСО 10303-21-2022**

Методические указания к выполнению практической работы по дисциплине «Информационная поддержка жизненного цикла продукции» для студентов направления подготовки 15.03.01 «Машиностроение» направленность (профиль) «Технология, оборудование и автоматизация машиностроительных производств»

УДК 519.6

Составители: В.В. Куц, А.Г. Ивахненко

Рецензент

Кандидат технических наук, доцент кафедры
«Машиностроительные технологии и оборудования» А.Н. Гречухин

Анализ структуры символьного обменного файла по ГОСТ Р ИСО 10303-21-2022: методические указания к выполнению практической работы по дисциплине «Информационная поддержка жизненного цикла продукции» по направлению подготовки 15.03.01 «Машиностроение» направленность (профиль) «Технология, оборудование и автоматизация машиностроительных производств» / Юго-Зап. гос. ун-т; сост.: В.В. Куц, А.Г. Ивахненко. Курск, 2023. 8 с. Библиогр.: с. 8.

Приведены ссылки, содержащие сведения о структуре символьного обменного файла по ГОСТ Р ИСО 10303-21-2022. Приводятся варианты заданий для выполнения практических занятий по дисциплине «Информационная поддержка жизненного цикла продукции».

Предназначены для обучающихся по направлению подготовки 15.03.01 «Машиностроение» направленность (профиль) «Технология, оборудование и автоматизация машиностроительных производств».

Текст печатается в авторской редакции

Подписано в печать . Формат 60×84 1/16.
Усл. печ. л. . Уч. - изд. л. . Тираж 50 экз. Заказ 386.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

В методических указаниях к выполнению практической работы «Описание структуры объектов на языке EXPRESS» изложены краткие сведения о структуре информационных моделей, которые следует использовать при выполнении настоящей практической работы. Сведения о структуре символьного обменного файла по ГОСТ Р ИСО 10303-21-2022, необходимые для выполнения задания приведены в методических указаниях к выполнению практической работы «Изучение структуры и создание символьного обменного файла по ГОСТ Р ИСО 10303-21-2022», которые также следует использовать при выполнении задания.

ЗАДАНИЯ ДЛЯ ВЫПОЛНЕНИЯ

По приведенной структуре символьного обменного файла и информации из секции данных (DATA_SECTION) построить графический объект.

Задание №1.

```
SCHEMA example_geometry;  
ENTITY point;  
x : REAL;  
y : REAL;  
END_ENTITY;  
ENTITY line;  
p0 : point;  
p1 : point;  
END_ENTITY;  
ENTITY circle;  
radius : REAL;  
center : point;  
END_ENTITY;  
END_SCHEMA;
```

```
data;  
#1=point(5.0, 2.0);  
#2=point(5.0, 5.5);  
#3=point(5.0, 8.0);  
#4=line(#7, #10);  
#5=line(#7, #11);
```

```

#4=line(#10, #11);
#6=circle(2.0, #1);
#7=point(5.0, 10.0);
#8=circle(1.0, #3);
#9=circle(1.5, #2);
#10=point(4.0, 9.0);
#11=point(6.0, 9.0);
endsec;

```

Задание №2.

```

SCHEMA example_geometry;
ENTITY точка;
x : REAL;
y : REAL;
END_ENTITY;
ENTITY линия;
x1,y1 : REAL;
x2,y2 : REAL;
END_ENTITY;
ENTITY окружность;
center : точка;
diametr : REAL;
END_ENTITY;
END_SCHEMA;

```

```

data;
#1=линия(6.0, 3.0, 6.0, 5.0);
#2=линия(4.0, 0.0, 4.0, 5.0);
#3=окружность(#15, 2.0);
#4=линия(7.0, 3.0, 7.0, 5.0);
#5=окружность(точка(4.0, 7.0), 4.0);
#6=линия(6.0, 5.0, 7.0, 5.0);
#7=линия(3.0, 4.0, 6.0, 4.0);
#8=линия(6.0, 3.0, 7.0, 3.0);
#15= точка(2.0, 4.0);
endsec;

```

Задание №3.

```

SCHEMA example_geometry;
ENTITY точка;
x : REAL;
y : REAL;
END_ENTITY;
ENTITY линия;

```

```

x1,y1 : REAL;
x2,y2 : REAL;
END_ENTITY;
ENTITY окружность;
Xc,Yc : REAL;
radius : REAL;
END_ENTITY;
END_SCHEMA;

```

```

data;
#1=линия(7.0, 6.0, 9.0, 3.0);
#2=линия(4.0, 7.0, 6.0, 5.0);
#3=окружность(2.0, 4.0, 1.0);
#4=линия(7.0, 3.0, 9.0, 3.0);
#5=линия(2.0, 5.0, 2.0, 7.0);
#6=линия(6.0, 5.0, 7.0, 5.0);
#7=линия(7.0, 3.0, 7.0, 6.0);
#8=линия(3.0, 4.0, 7.0, 4.0);
#9=линия(2.0, 7.0, 4.0, 7.0);
endsec;

```

Задание №4.

```

SCHEMA example_geometry;
ENTITY point;
x : REAL;
y : REAL;
END_ENTITY;
ENTITY line;
p0 : point;
p1 : point;
END_ENTITY;
ENTITY circle;
adius : REAL;
Xc, Yc : REAL;
END_ENTITY;
END_SCHEMA;

```

```

data;
#1=line(point(5.0, 1.0), #15);
#2=circle(1.0, 1.0, 8.0);
#3=line(point(7.0, 4.0), point(7.0, 7.0));
#4=circle(1.0, 4.0, 8.0);
#5=circle(1.0, 7.0, 8.0);
#6=line(#16, point(7.0, 4.0));
#7=line(point(5.0, 0.0), point(5.0, 1.0));

```

```

#8=line(#15, point(6.0 , 4.0));
#9=line(#15, point(7.0 , 6.0));
#10=line(#16, point(1.0 , 7.0));
#11=line(point(1.0, 7.0), point(7.0, 7.0));
#15= point(9.0 , 3.0);
#16= point(1.0 , 4.0);
endsec;

```

Задание №5.

```

SCHEMA example_geometry;
ENTITY точка;
x : REAL;
y : REAL;
END_ENTITY;
ENTITY линия;
x1,y1 : REAL;
p1 : точка;
END_ENTITY;
ENTITY окружность;
center : точка;
radius : REAL;
END_ENTITY;
END_SCHEMA;

```

```

data;
#1=линия(2.0, 0.0, точка(2.0, 8.0));
#2=линия(2.0, 8.0, точка(9.0, 5.0));
#3=линия(2.0, 2.0, точка(9.0, 5.0));
#4=линия(6.0, 4.0, точка(6.0, 6.0));
#5=окружность(точка(4.0, 5.0), 1.0);
#6=линия(5.0, 5.0, точка(7.0, 5.0));
endsec;

```

Задание №6.

```

SCHEMA example_geometry;
ENTITY точка;
x : REAL;
y : REAL;
END_ENTITY;
ENTITY линия;
x1,y1 : REAL;
x2,y2 : REAL;
END_ENTITY;
ENTITY окружность;

```

```
centr : точка;  
radius : REAL;  
END_ENTITY;  
END_SCHEMA;
```

```
data;  
#1=точка(8.5, 3.5);  
#2=окружность(#1, 0.5);  
#3=линия(1.0, 2.0, 1.0,5.0);  
#4=линия(2.0, 6.0, 3.0, 5.0);  
#5=линия(5.0, 5.0, 6.0, 6.0);  
#6=линия(9.0, 5.0, 10.0, 5.0);
```


БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ипатова, Э. Р. Методологии и технологии системного проектирования информационных систем : учебник / Э. Р. Ипатова, Ю. В. Ипатов. – 3-е изд., стер. – Москва : ФЛИНТА, 2021. – 256 с. : табл., схем. – (Информационные технологии). – Режим доступа: по подписке. –

URL: <https://biblioclub.ru/index.php?page=book&id=79551> (дата обращения: 12.04.2023). – Библиогр.: с. 95-96. – ISBN 978-5-89349-978-0. – Текст : электронный.

2. Схиртладзе, А. Г. Проектирование единого информационного пространства виртуальных предприятий : учебник : [16+] / А. Г. Схиртладзе, А. В. Скворцов, Д. А. Чмырь. – Изд. 2-е, стер. – Москва ; Берлин : Директ-Медиа, 2017. – 617 с. : ил., схем., табл. – Режим доступа: по подписке. –

URL: <https://biblioclub.ru/index.php?page=book&id=469047> (дата обращения: 12.04.2023). – Библиогр.: с. 606. – ISBN 978-5-4475-8634-8. – DOI 10.23681/469047. – Текст : электронный.

3. ГОСТ Р ИСО 10303-21 – 2022. Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 21. Методы реализации. Кодирование открытым текстом структуры обмена. М.: Российский институт стандартизации, 2022. – 84 с.