



Автоматический синтез диаграмм классов языка UML на основе ассоциативных отношений предметной области

Специальность: 05.13.11 – Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

ДИССЕРТАЦИЯ

на соискание ученой степени кандидата технических наук

Соискатель:

Бикмуллина Ильсияр
Ильдаровна

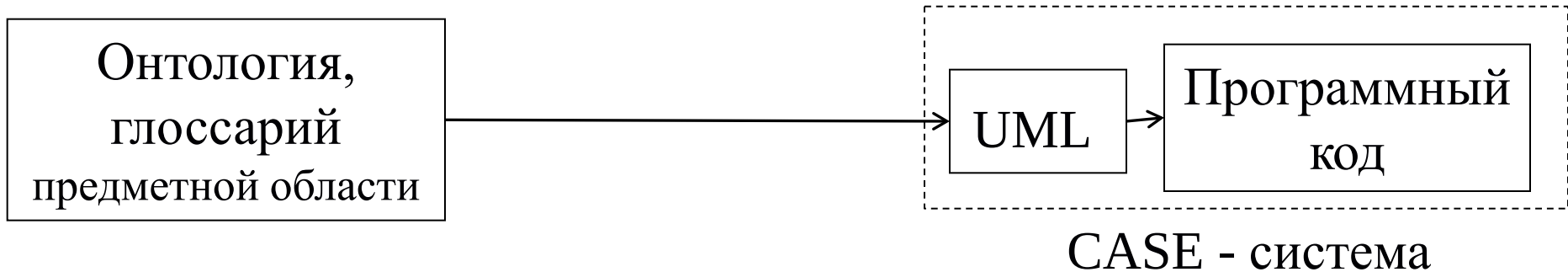
Научный руководитель:

Барков Игорь Александрович
д.т.н., доцент, каф. АСОИУ
КНИТУ-КАИ

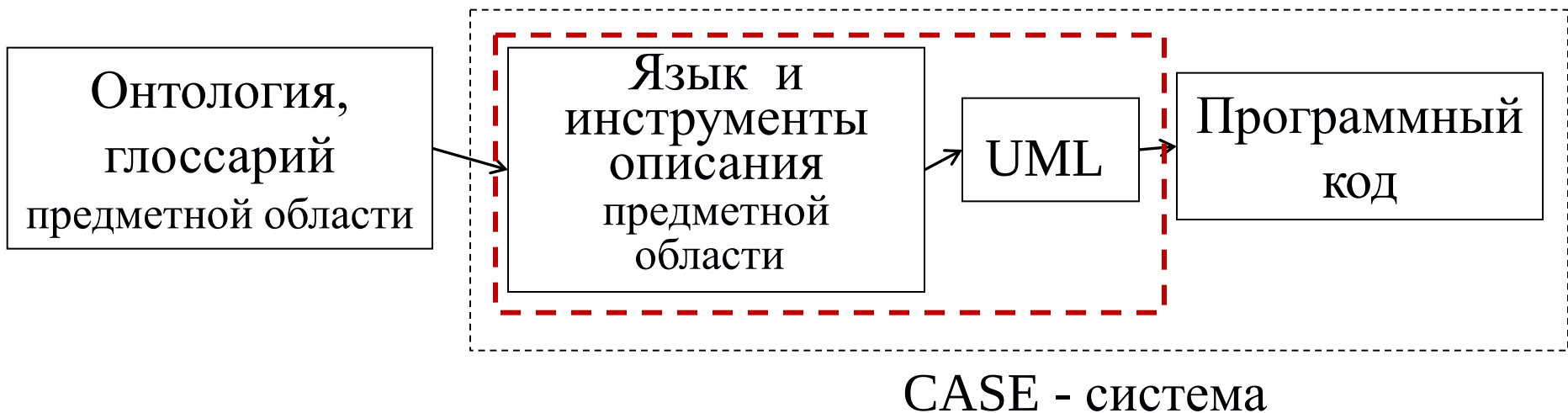
Казань, 2017

Проблема исследования

Технология разработки программ на основе UML



Предлагаемая технология разработки программ



Область исследования

Вариантов использования:



Классов:

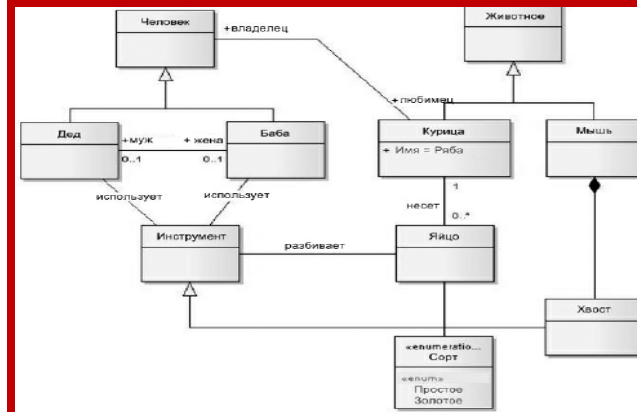
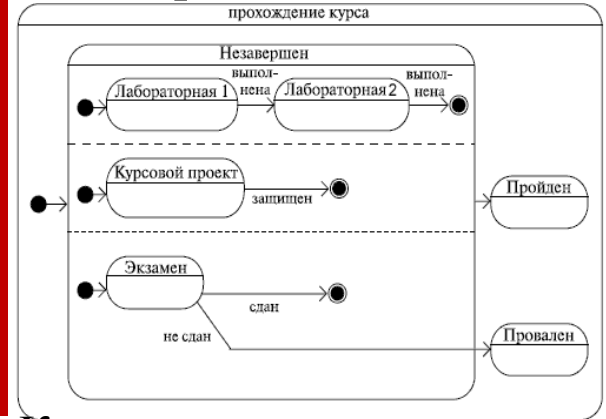


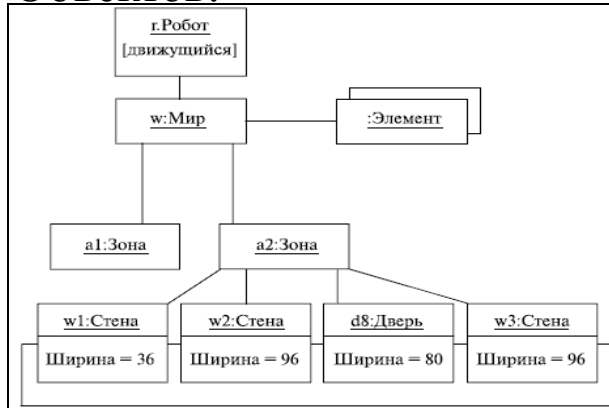
Диаграмма состояний:



Последовательности:



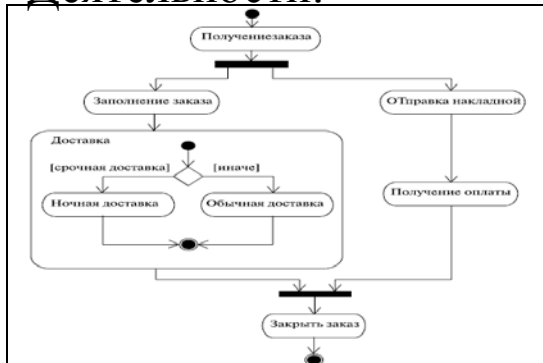
Объектов:



Кооперации:



Деятельности:



Компонентов:



Развертывания:



Общая характеристика исследования

Объект исследования: автоматический синтез диаграмм классов программных систем.

Предмет исследования: повышение уровня входного языка описания предметной области на основе ассоциативных отношений предметной области с последующим синтезом структурных отношений программной системы на основе семантических моделей.

Цель диссертационной работы: снижение сложности входного языка описания структурных отношений ПрО до уровня доступности прикладным экспертам; повышение качества и снижение трудоемкости проектной проработки программной системы за счет автоматизации синтеза диаграмм классов языка UML и расширение возможности многовариантного анализа и выбора структурных моделей программных систем.

Задачи исследования:

1. *Анализ существующих методов* синтеза структурных моделей программных систем.
2. *Разработка методики моделирования ассоциативных* отношений предметной области.
3. *Разработка методики синтеза* вариантных моделей диаграмм классов с неиерархическими связями на основе семантического анализа ассоциативных отношений предметной области.
4. *Разработка архитектуры* автоматизированной системы синтеза структурных моделей программных систем(на примере диаграммы классов).
5. *Экспериментальное исследование* разработанных методик и алгоритмов.

Научная новизна

Разработаны:

1. методика инвариантного к предметной области **описания** ассоциативных отношений на основе сигнатур, снижающая сложность языка описания и позволяющая включить в процесс разработки специалистов различных предметных областей. Данная методика отделяет ассоциативные отношения предметной области от всего множества структурных отношений.

2. методика автоматического **синтеза** диаграмм классов программной системы. Методика использует семантические методы для описания свойств диаграмм классов, с помощью которых осуществляется выделение классов, структурных отношений из ассоциативных отношений предметной области. В методике предложено отдельно смоделировать семантику диаграмм классов, увязав её с ассоциативными отношениями предметной области.

- Практическая ценность работы** заключается в разработке программного комплекса автоматического синтеза диаграмм классов, который позволяет:
- снизить трудоёмкость проектирования структурных моделей программных систем.
 - повысить качество программного проекта за счёт изменения характера труда программиста в направлении от ручной проработки диаграмм UML к выбору варианта из множеств автоматически полученных вариантов.
 - стимулировать разработчика на предварительное продумывание структуры прорабатываемой системы.
 - более активное вовлечение прикладных экспертов в процесс программного проектирования.

Результаты, выносимые на защиту

1. Базовый язык описания ассоциативных отношений предметной области.
2. Методика описания предметной области как профессионального расширения базового языка описания ассоциативных отношений .
3. Методика автоматического синтеза структурных моделей (диаграмм классов) программных систем, основанная на семантическом анализе ассоциативных отношений предметной области.

Апробация работы

Доклады на:

19 международных, **3** всероссийских конференциях,

Лучшая научная работа: конкурс при СПбГПУ по направлению «Системный анализ, проектирование, управление и обработка информации»(С-П, 2010).

Дипломы:

I степени: **XIX, XX** Международных молодежных конференциях «Туполевские чтения» (Казань, 2011-2012).

II степени по **XVI, XVII, XVIII** Международных молодежных конференциях «Туполевские чтения» (Казань, 2008-2010).

III степени на Республиканском конкурсе научных работ среди обучающихся на соискание премии им. Н.И. Лобачевского (Казань, 2014).

Публикации: всего **34** статей

Казань 2008-2016; *Нижнекамск* 2010; *С-П* 2010-2011; *Уфа* 2010; *Саратов* 2013-2014; *Воронеж* 2014-2015; *Курск* 2014-2015; *Минск* 2015-2016; *Самара* 2015-2016; *Екатеринбург* 2016, *Москва* 2016,

из них **4** статьи в рекомендуемых **ВАК** изданиях, **1** свидетельство о государственной регистрации программ для ЭВМ.

Глава 1. Анализ процесса проектирования программного обеспечения

Анализ логики задач проектирования программных систем

К языкам представления знаний в разных системах на сегодняшний день относятся: UML, Conceptual Graph, Frame-logic, Knowledge Interchange Format, диаграммы BPMN, IDEFx, ARIS, AnyLogic, схемы ДРАКОН, SysML, RDF Schema, OWL, BPEL4WS и т.д..

На данный момент существуют подходы на базе UML:

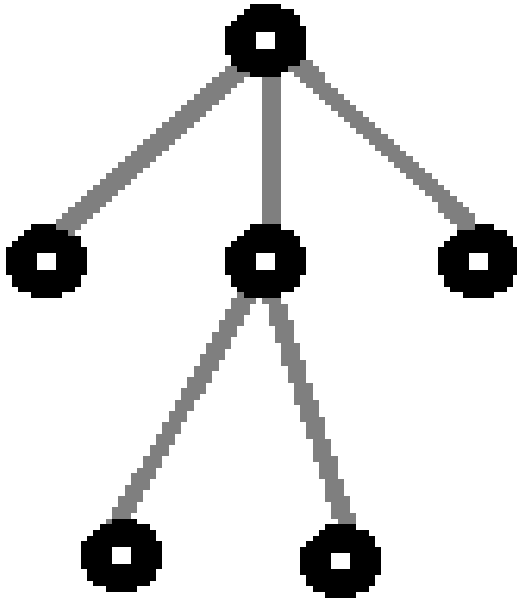
- ✓ Подход Шлеер-Меллора (Sally Shlaer and Stephen Mellor).
- ✓ Подход Гради Буча (GradyBooch).
- ✓ Подход Джеймса Рамбо (Object Modelling Technique James Rumbaugh - OMT).
- ✓ Подход Ивара Якобсона (Object-Oriented Software Engineering Ivar Jacobson - OOSE).

CASE средства кодогенерации

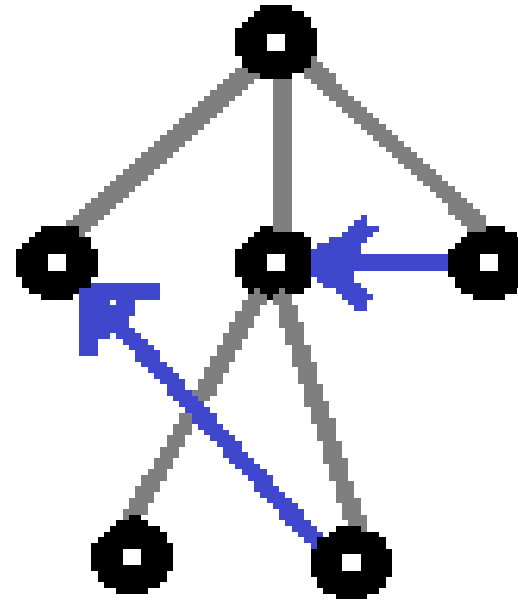
на основе моделей UML

CASE средства	Фирма	Поддержка кодогенерации	Прямое проектир-е	Обратное проектир-е	Уровень кодогенерации
Rational Rose Professional	IBM	Да, но	Да	Да	<u>не создает</u> 100 % код
	Rational Rose			(C,C++)	
Rational Rose RealTime	IBM Rational Rose	Да, но	Да (C,C++)	Да (C,C++)	<u>100 %</u> исполняемый код
Rational Rose Enterprise	IBM Rational Rose	Да, но	Да	Да	<u>не создает</u> 100 % код
MS Visual Studio 6.0 и выше	IBM Rational Rose	Да, но	Да (C++, VB, Java)	Да (C++, VB, Java)	<u>не создает</u> 100 % <u>код</u>
Visual Component Manager	IBM Rational Rose	Да, но	Да	Да	<u>не создает</u> 100 % <u>код</u>

Анализ структурных моделей UML диаграмм



а) синтез древовидных
структур **изучен**



б) синтез структур с
неиерархическими (по Бучу)
отношениями **изучен недостаточно**

Методика автоматического синтеза диаграмм классов

1 этап. Описание предметной области экспертами как множества профессиональных модулей систем понятий предметной области, объединенных ассоциативными отношениями.

2 этап. Назначения главного понятия.

3 этап. Создания сети понятий от главного понятия как транзитивного замыкания отношения ассоциации.

4 этап. Синтез классов.

5 этап. Синтез структурных отношений диаграммы классов UML.

6 этап. Визуализация диаграмм UML.

Глава 2. Моделирование ассоциативных отношений предметной области в задачах синтеза структурных моделей программных систем

Предложено для моделирования предметной области использовать **понятия**.
Понятие – форма мышления, отражающая существенные свойства, связи и отношения предметов и явлений.

Понятие 
→ термин
→ смысл

ЗАДАЧА. Разработка методики моделирования понятий предметной области и ассоциативных отношений.

Дано: предметная область.

Требуется: Построить описание системы понятий предметной области и ассоциативных отношений между понятиями.

Формальная постановка задачи:

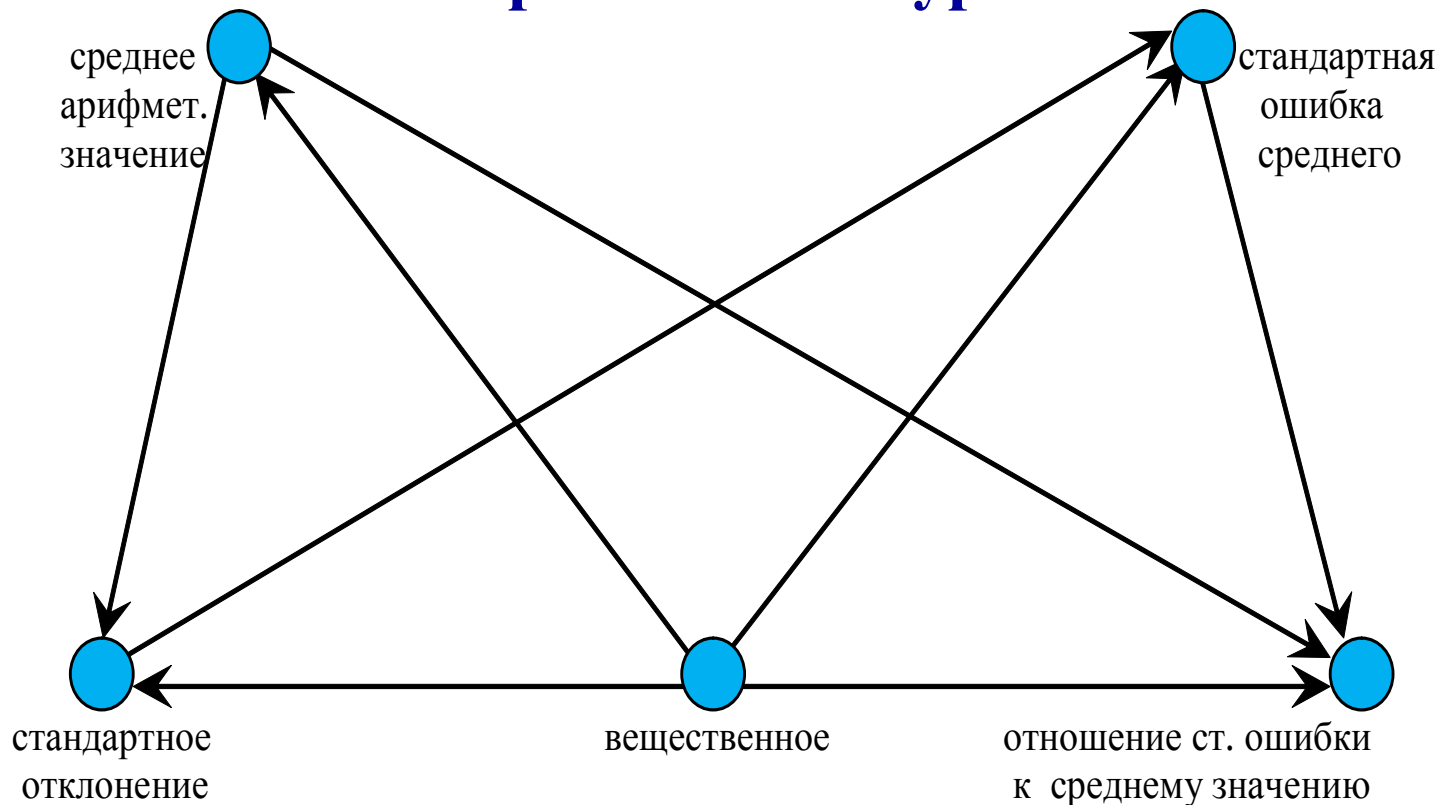
Пусть C - понятие, тогда **сигнатура понятия**

$$\Sigma(C) = (S, \Omega) , \text{ где}$$

S - множество основ - обозначений для всех используемых в понятии множеств данных;

Ω - множество идентификаторов для всех базовых операций над понятиями, которое представляет собой S – индексированное семейство множеств.

Расширение сигнатуры



A' - это *расширение* A , если $S^A \subseteq S^{A'}$ и $\Omega^A \subseteq \Omega^{A'}$.

Σ -алгебра A называется *Σ -редуктом* Σ' -алгебры A' , если

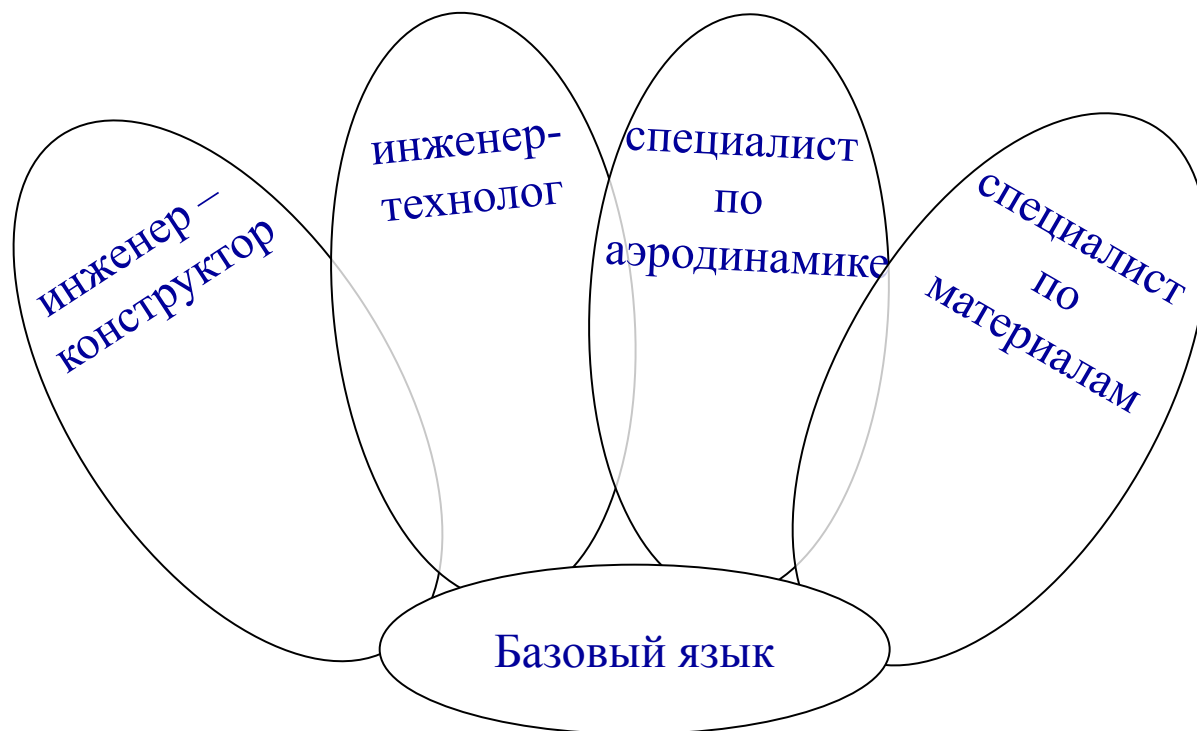
$S^A \subseteq S^{A'}$, $\alpha_{us}^A(\omega_{us}) \subseteq \alpha_{us}^{A'}(\omega_{us})$ для всех $s \in S$, $\omega \in \Omega$,

где us – индекс имени операции, α_{us} - индексированное семейство функций.

Совокупность основ и операций, добавление которых превращает алгебру A в алгебру A' - это *A'' -приращение* алгебры A .

Средства реализации профессионального взаимодействия экспертов предметных областей

В качестве инструмента создания профессионального расширения базового языка предложено использовать понятия совместно с правилами структурной сочетаемости понятий (ассоциативных отношений).



Математическая модель синтеза диаграмм классов

1) Правила построения системы понятий $T_{сист} = (\Sigma_{сист}, E_{сист})$, где

$$\Sigma_{сист} = \langle S_{сист}, \Omega_{сист} \rangle$$

$$E_{сист} = E_{пэ} \cup E_{ден} \cup E_{абст}$$

↓
Множество систем понятий

2) Правила построения сети понятий

$$T_{сети} = (\Sigma_{сети} \cup \Sigma_{сист}, E_{сети} \cup E_{акс})$$

$$\Sigma_{сети} = \langle S_{сети}, \Omega_{сети} \rangle$$

↓
Множества сетей понятий

3) Правила построения диаграмм классов

$$T_{диаг} = (\Sigma_{диаг} \cup \Sigma_{сети} \cup \Sigma_{сист}, E_{диаг})$$

$$\Sigma_{диаг} = \langle S_{диаг}, \Omega_{диаг} \rangle$$

Множество диаграмм классов

$E_{сист}$ — ограничения накладываемые на язык описания системы понятий.

$E_{пэ}$ — правила построения ассоциативных отношений на основе семантической сочетаемости понятий, характеризуют семантику предметной области (эксперт).

$E_{ден}$ - правила построения языка понятий (системой). $E_{абст}$ — правила взаимодействия с другими понятиями (системой). $E_{сети}$ — правила корректности сети понятий (системой). $E_{акс}$ — аксиомы сети понятий (системой). $E_{диаг}$ — правила построения диаграммы классов (разработчиками CASE системы).

SL/Prof - язык спецификации экспертного исходного описания предметной области (входной язык)

$$G(L) = \{T, N, P, S\},$$

T – алфавит определяемого языка, терминальный алфавит,
N – вспомогательный алфавит, нетерминальный алфавит,
P – множество правил вывода,
S – аксиома грамматики, $S \in N$.

Далее приведена часть правил вывода языка **P**:

Спецификация предметной области : { Спецификация профессиональной темы } .

*Спецификация профессиональной темы : **Раздел** , Наименование раздела профессиональной темы, [**Используемые** , { Наименование раздела профессиональной темы через , } ,] { Спецификация понятия } **Конец** , Наименование раздела профессиональной темы.*

Наименование раздела профессиональной темы : Составной идентификатор с строчной буквы.

*Спецификация понятия : **Понятие** , Наименование понятия , Спецификация параметров и операций понятия , [Спецификация свойства через ,] **Конец** , Наименование понятия .*

Наименование понятия : Составной идентификатор с строчной буквы .

Описание предметной области

Описание предметной области представляет собой систему понятий:

1) Информационная единица описания есть **понятие**.

2) Описание сигнатур понятий

Понятие <наименование_понятия>

Описание сигнатур параметров

<Наименование переменной>:{<тип_переменной>}

Описание сигнатур операций

Операции

[<Наименование операции> (<Наименование переменной на основе которой формируются данные>)] : [<тип>]

[<Наименование операции> (<Откуда берутся данные> : {вектор [<Размер вектора>]} из <тип подхватываемых данных>)} : [<Наименование заполняемой переменной, её тип>]]

[<Наименование операции> (<Наименование переменной на основе которой формируются данные>)] : [<Наименование заполняемой переменной, её тип>]

Описание сигнатур свойств(ограничений)

Свойство [<Ограничения>]

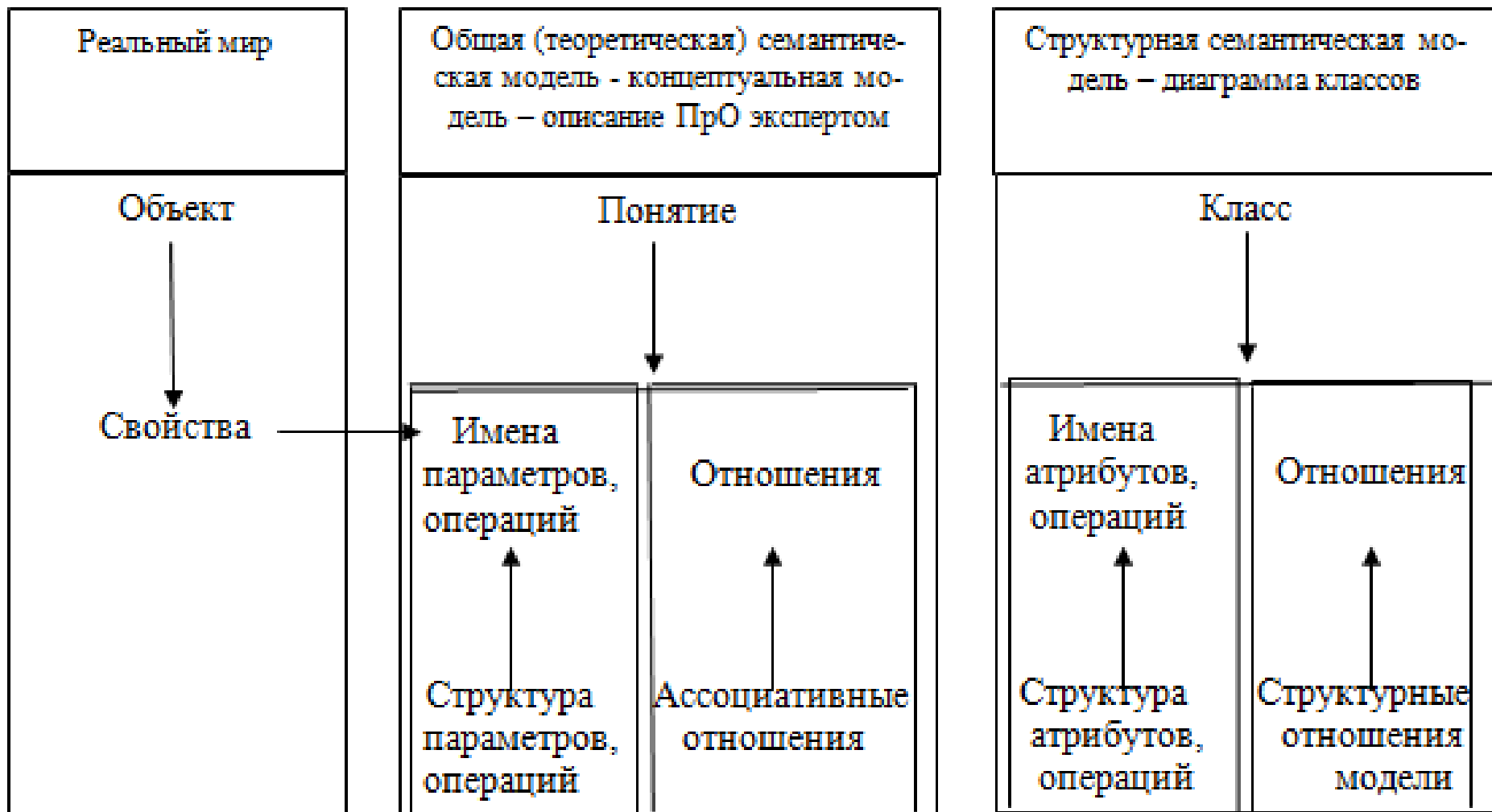
3) Правила синтаксиса входного языка:

Наименования понятий ведется с маленькой буквы.

Наименования параметров обозначается с заглавной буквы....

Глава 3. Разработка методики синтеза диаграмм классов

Система синтеза структурных моделей



Синтез сигнатуры класса

1) Информационная единица диаграммы классов является **класс**.

2) Синтаксис сигнатуры **класса**

<видимость> <Имя_класса>:[<Родительский_класс>]

Описание сигнатур **атрибута**

<видимость> <Имя>:<тип> = <начальное_значение>

Описание сигнатур **операций**

<видимость><Имя>(<список_параметров>):
[<Наименование_параметра_возвращаемого_значения>]:
<тип_возвращаемого_значения>

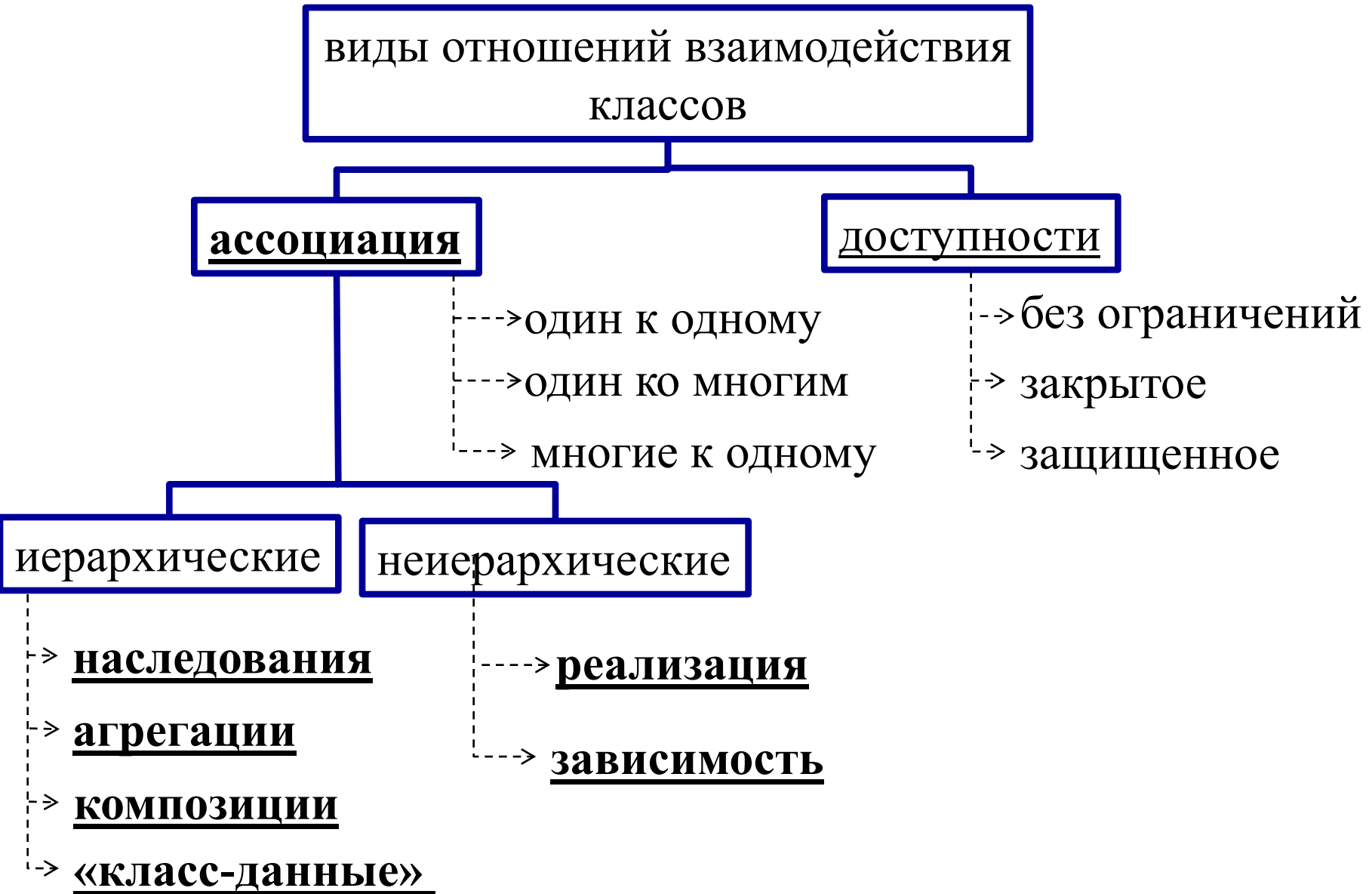
Описание сигнатур **свойств(ограничений)**

[<свойства атрибута>], [<свойства операций>]

3) Синтаксические правила оформления класса (входят в *Едиаг*):

- Наименование класса пишется с большой буквы, где пробелы заменяются на подчеркивания.
- Имя абстрактного класса пишется курсивом.
- Наименования атрибутов обозначаются с заглавной буквы.
- ...

Построение системы бинарных отношений



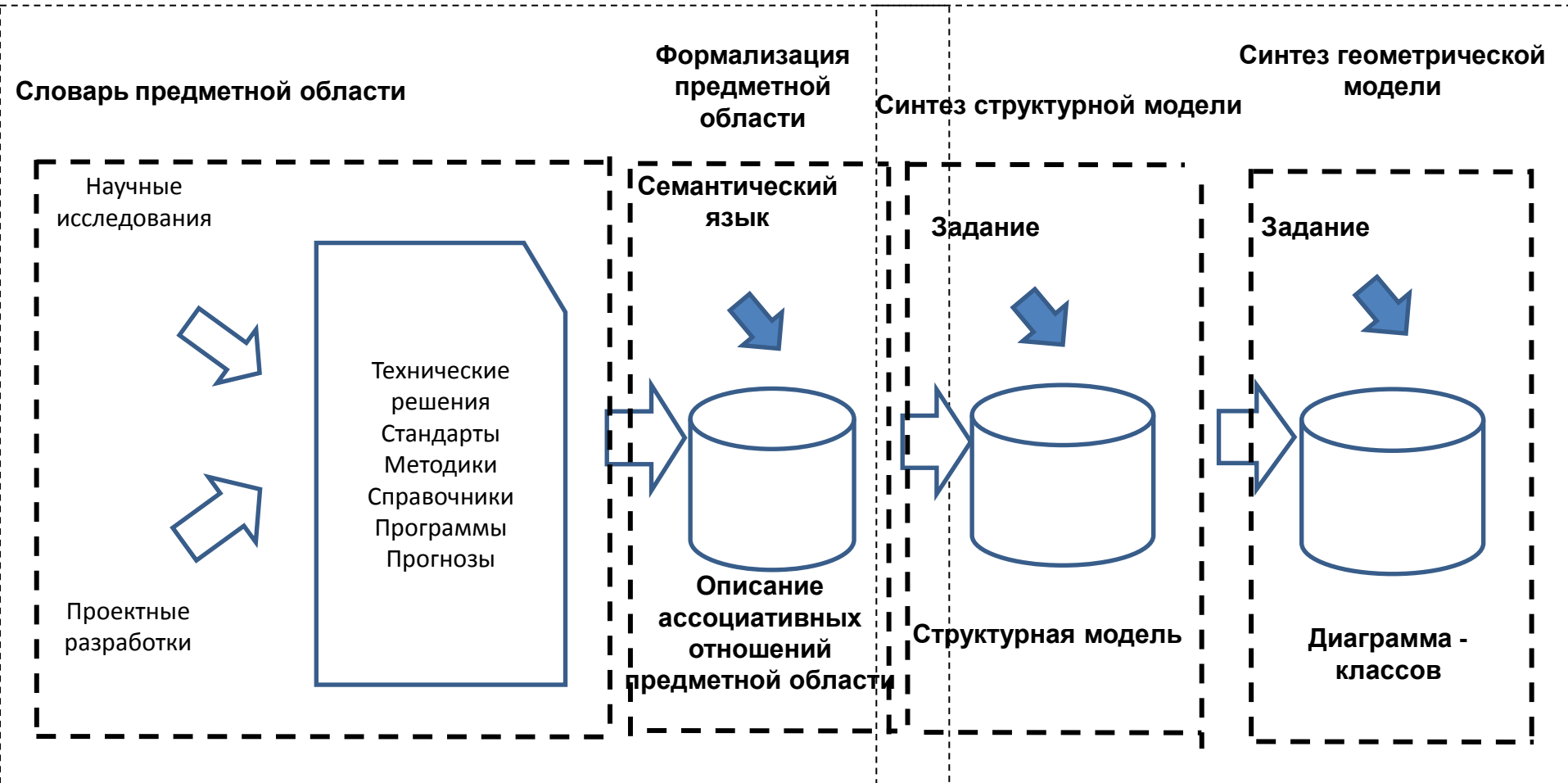
Правила генерация классов и отношений

1. $\forall x \forall y (K(x) \& K(y) \& O_1(x,y) \& O_2(x,y) \Rightarrow (O_1=O_2));$
2. $\forall x \forall n_1 \forall n_2 (K(x) \& Atr(x,n_1) \& Atr(x,n_2) \Rightarrow (n_1 \neq n_2));$
3. $\forall x \forall y \in x (K(x) \& K(y) \Rightarrow H(y, x));$
4. $\forall x \forall y \in x \forall z \in x (K(x) \& K(y) \& K(z) \& (y \neq z) \Rightarrow Ag(y,x) \& Ag(z,x));$
5. $\forall x \forall y (K(x) \& K(y) \& SS(x) \& \neg SS(y) \Rightarrow Ag(y,x));$
6. $\forall x \forall y (K(x) \& K(y) \& Ag(y,x) \Rightarrow Cut(y));$
7. $\forall x \forall y (K(x) \& K(y) \& SS(x) \& SS(y) \Rightarrow Ko(y,x));$
8. $\forall x \forall y (K(x) \& K(y) \& Ko(y,x) \& Cut(x) \Rightarrow Acut(y));$
9. $\forall x \forall y (K(x) \& K(y) \& Ag(y,x) \& Cut(x) \& Acut(y) \Rightarrow Ko(y,x)).$

...

где $K(x)$: «построен класс x »; $O_i(x, y)$: «построено i -е отношение от класса x к классу y »; $Atr(x, n)$: «атрибут класса x имеет имя n »; $H(y, x)$: «существует отношение наследования от класса (потомка) y к классу (с функцией родителя) x »; $Ag(y, x)$: «существует отношение агрегации от класса y с характеристикой «часть» к классу x с характеристикой «целое»»; $SS(x)$: «существует стандартный компонент системы класса x »; $Ko(y, x)$: «существует отношение композиции от класса y с характеристикой «часть» к классу x с характеристикой «целое»»; $Cut(y)$: «ликвидация экземпляров класса y решается явным образом»; $Acut(y)$: «автоматически удаляется класс y ».

Глава 4. Разработка информационной технологии автоматического синтеза структурных моделей программных систем и экспериментальные исследования



Информационная технология синтеза диаграмм классов (продолжение)

1.1. Заполнение экспертом исходного описания предметной области на основании ограничений *Есист*,

Конвертор_1[транслятор]:

1.2. Синтаксический анализ (на основе контекстно-свободной грамматики),

2. Выделение главного понятия

3. Переформирование описания предметной области в форме удобной для разработчика по *Есети*,

Конвертор_2[транслятор]:

4. Выделение классов из понятий по *Едиаг* и построение классов,

5.1. Назначение вместо отношения ассоциации возможные виды отношений *Едиаг*,

5.2. Трансляция в позитивно образованную формулу (входной язык: программы автоматического доказательства теорем «ProblemSolver»),

5.3. Исполнение позитивно образованной формулы в «Problem Solver» и построение вариантов диаграммы классов.

6. Построение визуальной модели диаграммы классов.

Программная реализация

Определение подраздела проекта

<-- Cancel читать. Выйти из

Название проекта: Совет

Наименование основного проекта: Университет

Выбранный подраздел проекта: Деканат

1. Выделите ниже название модификации проекта:

ПроектА
Ок
Октябрь22
Деканат

2. Выберите ниже эксперта.

Создать
Удалить

КТН

Откр

По выбранному разделу

Раздел Понятие Помощь

Главное понятие: Иерархия понятий

Создать понятие ...Удаление понятия

Создать понятие не затирая предыдущие данные

Список понятий:

Понятие	Раздел
---------	--------

Прими совет

1 шаг. Опишите первое понятие. Для этого:
или нажмите первую кнопку (наверху),
или в меню перейдите на ссылку Понятие->Создать понятие.

2 шаг. Выберите главное понятие.

За дополнительными сведениями обратитесь к автору.

Описание понятия

Имя понятия: аСОНУ

Родовое понятие: значение

КАИ

Источники: Изображение

Дополнительное описание: Описание

Смотреть

Атрибуты

	Наименование	Логический
1	Количество	Числовой(целое)

Внешние

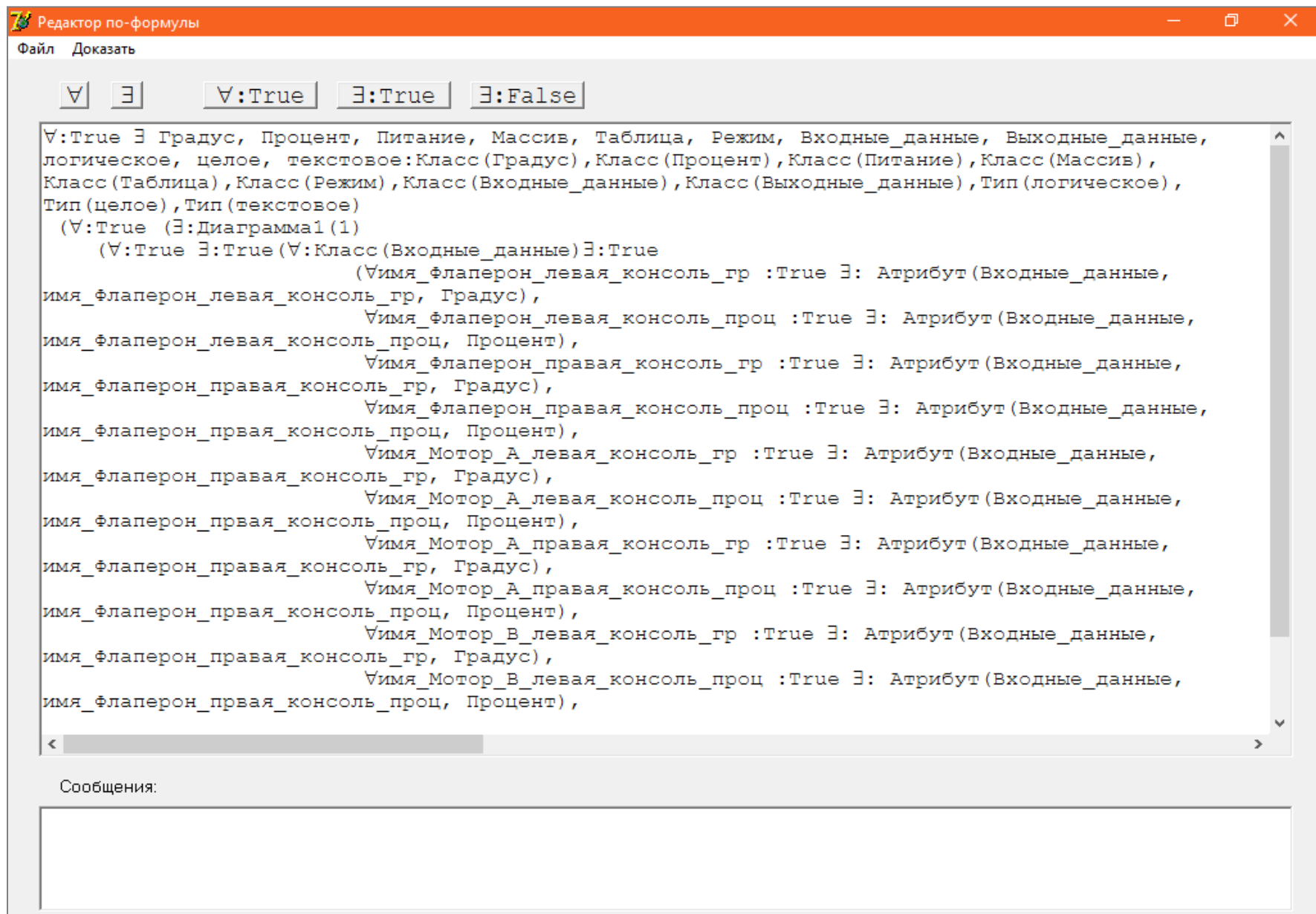
	Кафедры	Текстовый
--	---------	-----------

Сообщения

Сохранение прошло успешно - можете приступить к созданию нового понятия или выйти

Сохранить понятие

Программная реализация (продолжение)



Пример:

Разработка информационной системы управления
силовой установки БПЛА

для проверки работоспособности БПЛА и проведения
натурных испытаний,

используя систему автоматического синтеза
диаграммы классов

1 этап процесса. Описания предметной области экспертами в виде системы понятий

Понятие входные_данные

Флаперон_левая_консоль_гр: градус

Мотор_А_левая_консоль_гр: градус

Сервопривод_А_правая_гр: градус

Элерон_левая_консоль_гр: градус

Угол_поворота_консоль_гр: градус

...

Общий_шаг: логическое

Циклический_шаг: логическое

Синхронизация_моторов_левой: режим

Синхронизация_моторов_правой: режим

Поворотное_устройство: питание

Моторы: питание

Операции

Заполнить_таблицу(

Мотор_А_правая_консоль_гр, ...): таблица.

Выбор_шага (Общий_шаг, Циклический_шаг,

Нулевой_шаг): логический

Понятие градус

Значение: целое

Свойство $0 < \text{Значение} < 120$

Понятие процент

Значение: целое

Свойство $0 < \text{Значение} < 120$

Понятие режим

Подключено: логическое

Не_подключено : логическое

Режимный_объект : текстовое

Операции

Заполнить_объект(Х:текстовое):

Режимный_объект

Выбрать_режим(Подключено,

Не_подключено, входные данные:

Режимный_объект): логическое

2 этап процесса. Назначения главного понятия

Понятие питание

Подключено: логическое

Не_подключено : логическое

Режимный_объект: текстовое

Операции

Выбрать_режим(либо Подключено
(Режимный_объект) либо
Не_подключено (Режимный_объект)):
целое

Понятие таблица

Количество_параметров : целое

Данные : вектор

[Количество_параметров] из целое

Операции

...

Заполнить_вектор(Данные : вектор
[Количество_параметров] из целое):
Данные

Понятие выходные_данные

Настройка_порта_скорость : целое

All_params : входные_данные

...

Операции Выдать(All_params,
Настройка_порта_имя, ...) :
входные_данные

Понятие массив

Поле : текстовое

Размер_массива: целое

М : вектор [Размер_массива] из целое

Операции

Заполнить_поле(Х:текстовое): Поле

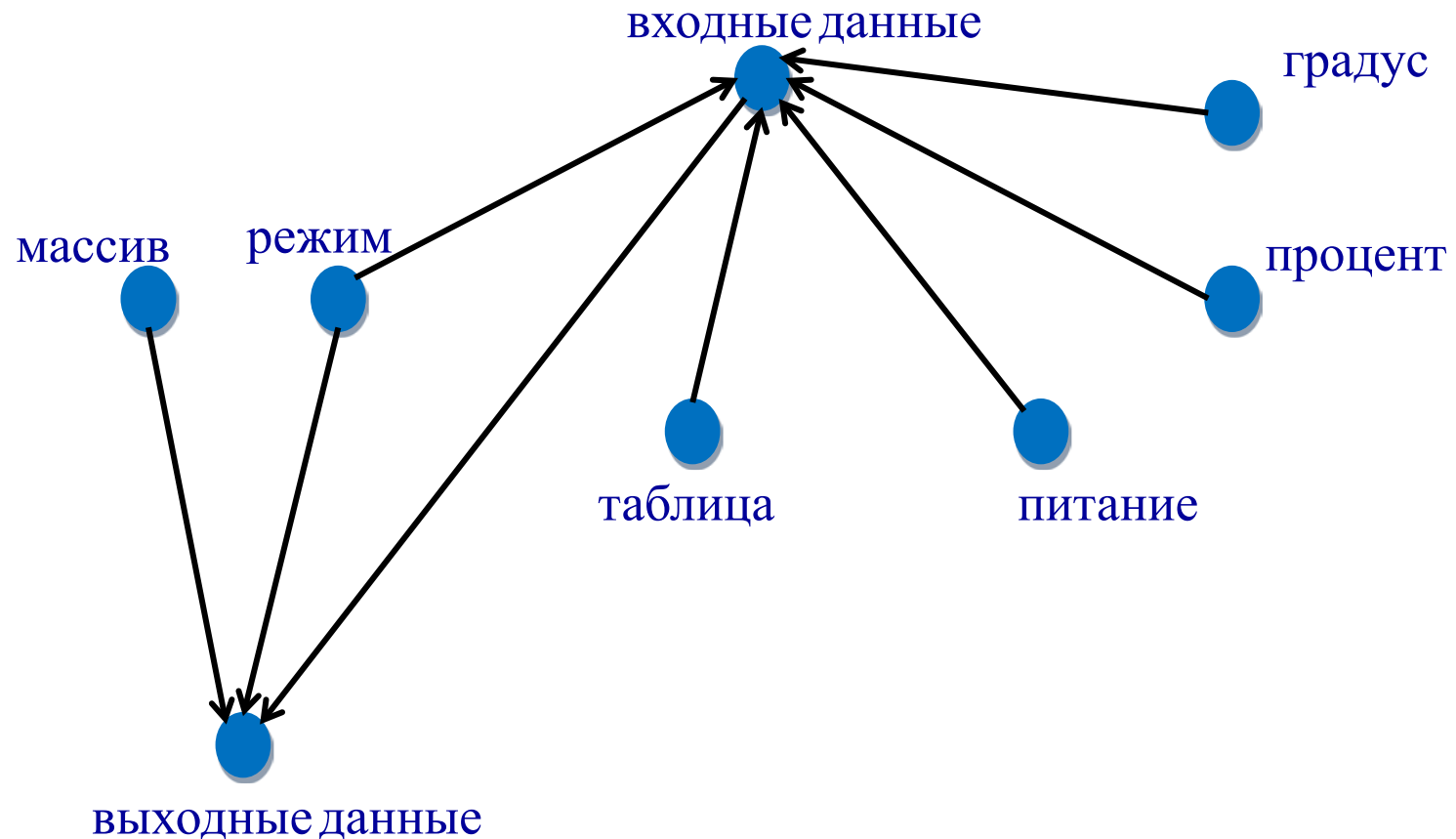
Подсчет_количества(Поле):

Размер_массива

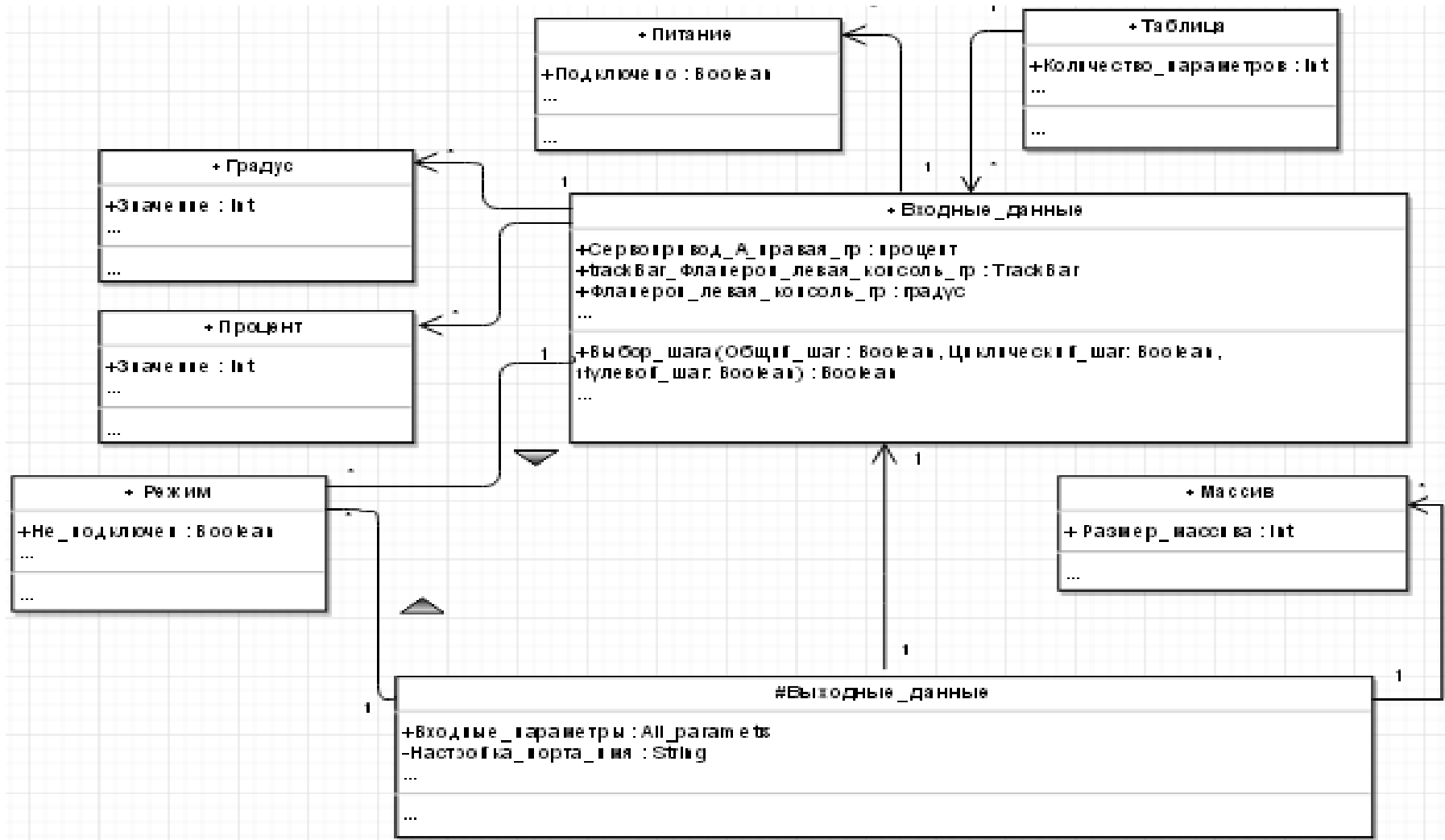
Заполнить_массив(Поле:вектор [Р
азмер_массива] из целое): М

Этап процесса. Создания сети понятий от главного понятия

Модель абстрактных структурных зависимостей



4 этап процесса. Синтез из понятия классов и назначение между классами отношений ассоциации



5 этап процесса. Спецификация предметной области в ПО-форме

$\forall$:True $\exists$ Градус, Процент, Питание, Массив, Таблица, Режим, Входные_данные, Выходные_данные, логическое, целое, текстовое:Класс(Градус),Класс(Процент),Класс(Питание),Класс(Массив), Класс(Таблица),Класс(Режим),Класс(Входные_данные),Класс(Выходные_данные),Тип(логическое), Тип(целое),Тип(текстовое)

($\forall$:True ($\exists$:Диаграмма1(1)

($\forall$:True $\exists$:True ($\forall$: Класс(Входные_данные) $\exists$:True

($\forall$ имя_ Флаперон_левая_консоль_гр :True $\exists$: Атрибут(Входные_данные, имя_ Флаперон_левая_консоль_гр, Градус) ,

$\forall$ имя_Общий_шаг :True $\exists$: Атрибут(Входные_данные, имя_Общий_шаг, логическое) , ..., $\forall$ имя_Заполнить_таблицу :True $\exists$: Функция(Входные_данные, имя_Заполнить_таблицу),

Сигнатура_вых(имя_Заполнить_таблицу, Таблица) ,
Сигнатура_вх1(имя_Заполнить_таблицу, имя_Мотор_А_правая_консоль_гр, Входные_данные) , ...,

$\forall$: Класс(Таблица) $\exists$:True

($\forall$ имя_Количество_параметров :True $\exists$: Атрибут(Таблица, имя_Количество_параметров, целое),...),

$\forall$:True ($\exists$:Отношение_агрегации(Таблица, Входные_данные),

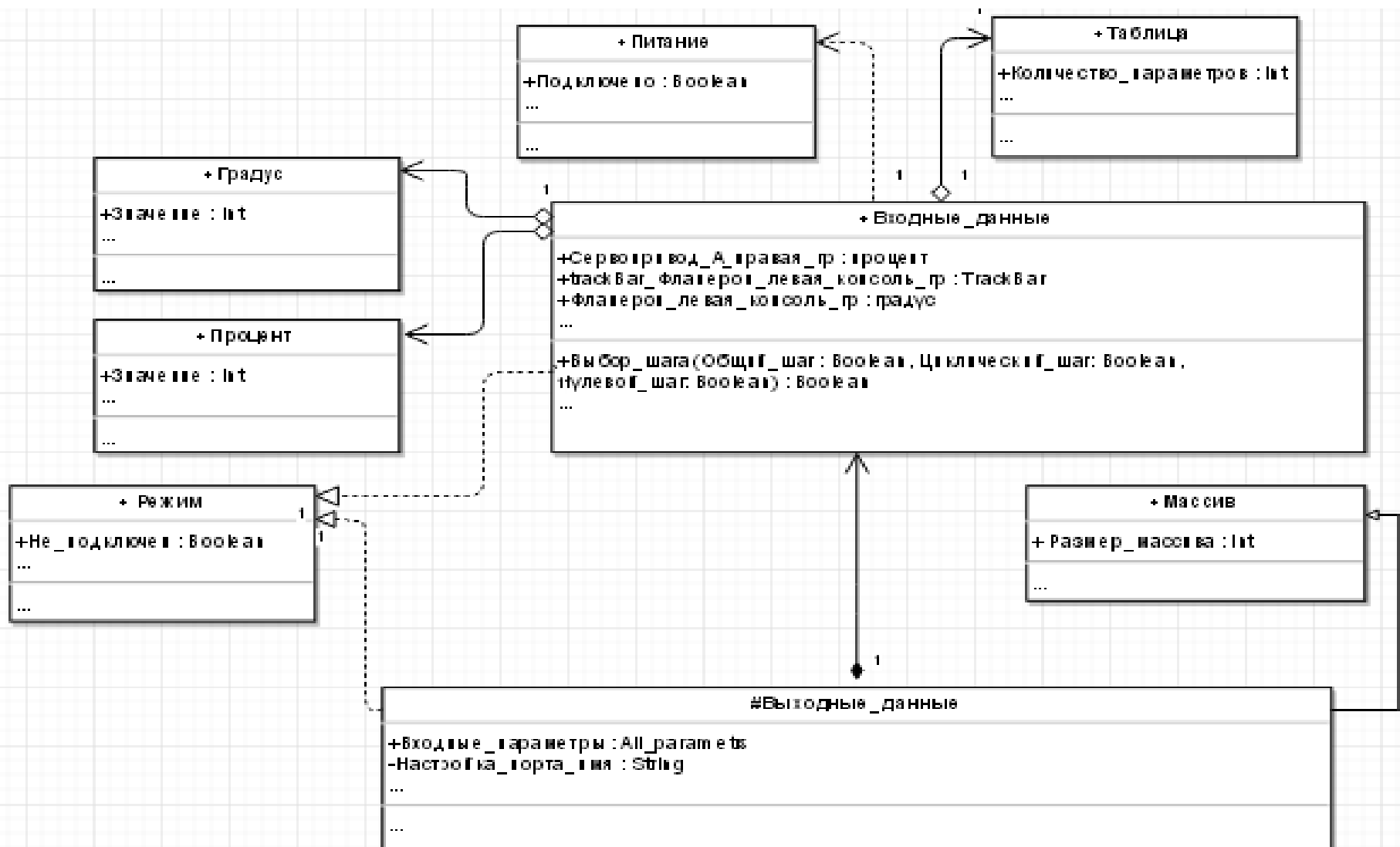
$\exists$:Отношение_наследования(Входные_данные, Таблица) ,

$\exists$:Отношение_реализация(Входные_данные, Таблица)),

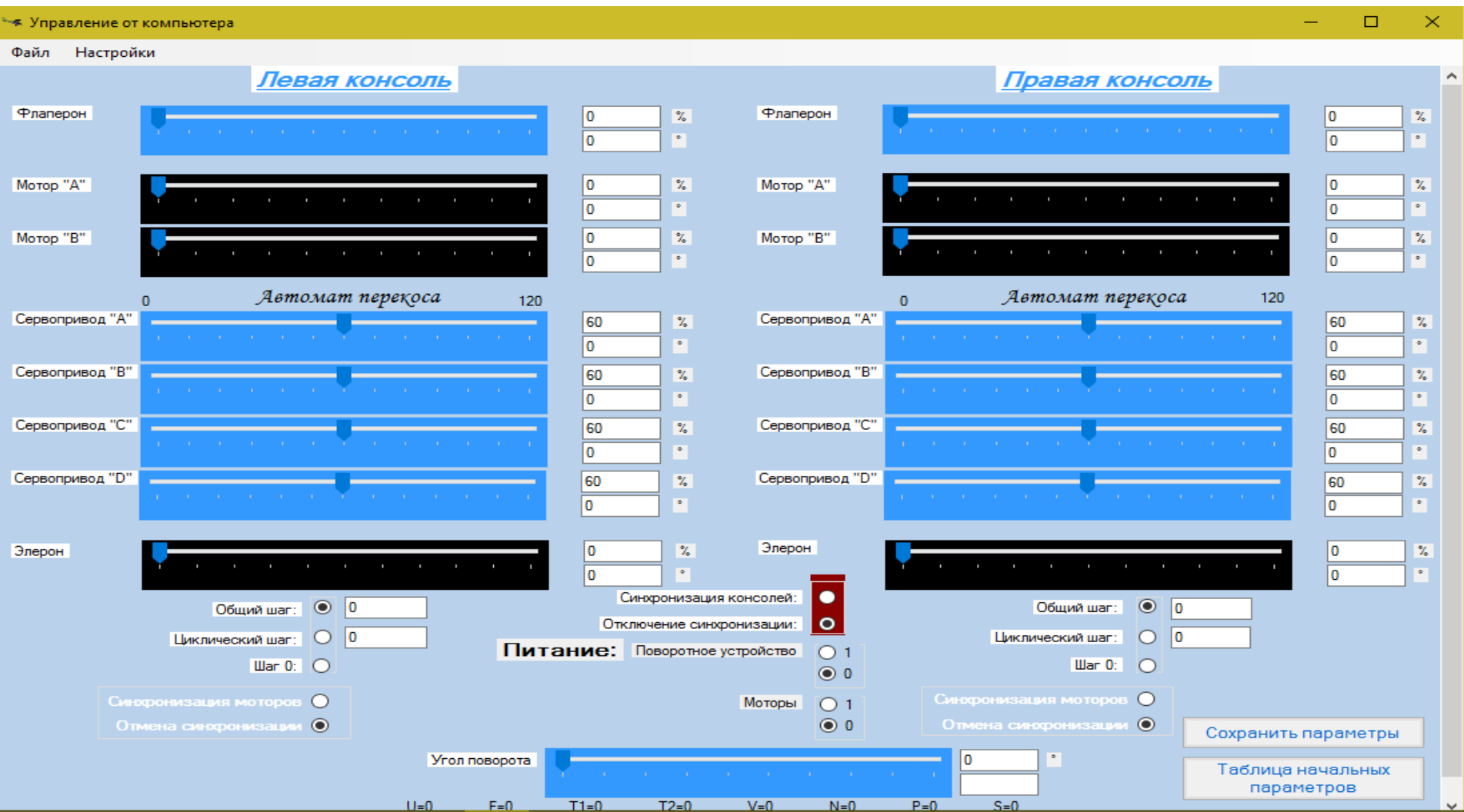
$\forall$: Класс(Градус) $\exists$:True ($\forall$ имя_Значение :True $\exists$:Атрибут(Градус, имя_Значение, целое)),... **31**

6 этап процесса. Определение видов отношений

Диаграмма классов UML



Разработка ИС управления силовой установкой БПЛА для проверки её работоспособности и проведения натурных испытаний



Силовая установка БПЛА для проверки её работоспособности и проведения натурных испытаний



Заключение

1. На основе анализа сделан вывод о том, что технология создания программных систем используется недостаточно эффективно по следующим факторам: высокая трудоемкость разработки вариантных моделей; низкая производительность труда разработчиков программных систем; сложность входного языка практически не позволяют подключить экспертов предметной области к процессу создания программной системы.
2. Предложено в исходном описании структурных моделей предметной области использовать только ассоциативные отношения между понятиями предметной области, что позволило упростить входной язык описания структурных отношений предметной области.
3. Предложена методика и язык инвариантного к предметной области описания ассоциативных отношений на множестве понятий предметной области на основе сигнатур, снижающие сложность языка описания и позволяющие включить в процесс разработки специалистов различных предметных областей.

Заключение (продолжение)

4. Предложено моделировать семантику структурных моделей программной системы отдельными средствами.
5. Предложена методика синтеза диаграмм классов программных систем, позволяющая в автоматическом режиме получать вариантные структурные модели диаграмм классов по описанию предметной области. Методика позволяет однократно смоделировать в CASE-системе семантику диаграмм классов, а описание структурных отношений предметной области передать экспертам.
6. Разработана информационная технология и программное обеспечение синтеза диаграмм классов языка UML, проведены на реальных данных эксперименты, которые полностью подтвердили теоретические результаты работы.



Спасибо за внимание!

