Министерство образования и науки Российской Федерации

Федеральное агентство по образованию

#### САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМА-ЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ



## Д.И. Муромцев

# Системы инженерии знаний

## МЕТОДИЧЕСКОЕ ПОСОБИЕ



## Санкт-Петербург

2009

УДК [004.891 + 002.53:004.89] (075.8) Д.И. Муромцев. Системы инженерии знаний – СПб: СПб ГУ ИТМО, 2009. – 60 с.

В методическом пособии представлены лабораторные работы, позволяющие студентам овладеть основными навыками инженерии знаний с помощью систем LogicGem, FreeMind и SNePS. Рассматриваются основные аспекты разработки баз знаний, графические формы представления знаний, а также принципы формирования запросов к сетевым базам знаний.

Методическое пособие адресовано студентам высших учебных заведений, обучающихся по направлению 210202.65.08 «Проектирование и технология электронных средств» и по специальности 0900104.65 «Комплексная защита объектов информатизации».

Одобрено на заседании ученого совета факультета компьютерных технологий и управления Санкт-Петербургского государственного университета информационных технологий, механики и оптики, протокол № 10 от 19 мая 2009 года.



В 2007 году СПбГУ ИТМО стал победителем конкурса инновационных образовательных программ вузов России на 2007–2008 годы. Реализация инновационной образовательной программы «Инновационная система подготовки специалистов нового поколения в области информационных и оптических технологий» позволит выйти на качественно новый уровень подготовки выпускников и удовлетворить возрастающий спрос на специалистов в информационной, оптической и других высокотехнологичных отраслях экономики.

© Муромцев Д.И., 2009 © СПб ГУ ИТМО, 2009

## Оглавление

Введение	5
Разработка таблиц решений в системе LogicGem	7
Лабораторная работа № 1. Создание таблицы	8
Шаг 1. Создание таблицы решений	8
Шаг 2. Добавление описаний для условий	8
Шаг 3. Добавление описаний для действий	. 10
Шаг 4. Определение условий	. 10
Шаг 5. Определение действий	. 11
Шаг 6. Добавление количественных параметров	. 12
Шаг 7. Сохранение таблицы решений в файл	. 13
Лабораторная работа № 2. Создание логических структур	. 14
Шаг 1. Проверка на пропущенные правила	. 14
Шаг 2. Исправление таблицы решений	. 14
Шаг 3. Компиляция таблицы решений	. 15
Шаг 4. Автоматическая генерация продукционных правил	. 17
Разработка карт памяти (mindmaps) в системе FreeMind	. 19
Лабораторная работа № 1. Создание и сохранение новой карты	. 19
Шаг 1. Создание карты	. 19
Шаг 2. Добавление дочерних узлов	. 20
Шаг 3. Узлы с общим родителем	. 21
Шаг 4. Сохранение карты	. 25
Лабораторная работа № 2. Форматирование карты	. 26
Шаг 1. Изменение типа узла	. 26
Шаг 2. Изменение цвета узла	. 26
Шаг 3. Добавление облаков	. 28
Лабораторная работа № 3. Добавление ссылок на внешние ресурсы	. 29
Шаг 1. Подготовка к работе	. 29
Шаг 2. Добавление ссылок	. 31
Шаг 3. Добавление ссылки на документ или папку	. 32
Шаг 4. Добавление ссылок на веб-ресурсы	. 34
Шаг 5. Работа с картой	. 35
Лабораторная работа № 4. Использование ссылок на другие карты	. 36
Шаг 1. Подготовка к работе	. 36
Шаг 2. Создание ссылки на другую карту	. 37
Шаг 3. Переход по ссылке	. 37
Лабораторная работа № 5. Работа с картами памяти	. 38
Шаг 1. Загрузка карты	. 38
Шаг 2. Поиск на карте	. 38
Шаг 3. Сворачивание узлов	. 39
Лабораторная работа № 6. Импорт и экспорт карт памяти	. 40
Шаг 1. Загрузка карты	. 40
Шаг 2. Импорт ветви из другой карты	. 40
Шаг 3. Экспорт карты в различные форматы	.41
Шаг 4. Защита карты	. 42
Лаоораторная работа № 7. Различные режимы работы	. 43
шаг 1. Загрузка карты	. 43
шаг 2. Переключение режимов	. 43
Разраоотка семантических сетеи в системе SNePS	. 44
О программе SNePS	. 44

Типы узлов	
Пояснения по семантике выражений SNePS	
Лабораторная работа № 1. Добавление высказываний в базу знаний	
Шаг 1. Запуск системы	
Шаг 2. Начало работы	
Шаг 3. Определение типов отношений	47
Шаг 4. Описание объектов	47
Шаг 5. Проверка введенных данных	
Шаг 6. Добавление информации об экземплярах классов	
Шаг 7. Проверка обновленных данных	
Шаг 8. Ввод описаний	
Шаг 9. Просмотр семантической сети в виде графа	
Лабораторная работа № 2. Поиск в базе знаний	50
Шаг 1. Дополнение базы знаний	50
Шаг 2. Формирование простейшего вопроса	50
Шаг 3. Детализация вопроса	51
Лабораторная работа № 3. Построение узлов	
Шаг 1. Подготовка к работе	
Шаг 2. Добавление предположений	
Шаг 3. Просмотр графа	53
Шаг 4. Поиск введенных утверждений	53
Шаг 5. Поиск аналогичных объектов	
Шаг 6. Доопределение сети	
Шаг 7. Пример универсального поиска	
Лабораторная работа № 4. Работа с высказываниями	55
Шаг 1. Просмотр пространства убеждений	55
Шаг 2. Запуск рассуждений (вывода)	55
Шаг 3. Просмотр графа сети	
Шаг 4. Другие примеры вывода	
Лабораторная работа № 5. Работа с узлами	57
Шаг 1. Подготовка к работе	
Шаг 2. Доступ к узлу через переменную	
Шаг 3. Использование базовых узлов	
КАФЕДРА ПРОЕКТИРОВАНИЯ КОМПЬЮТЕРНЫХ СИСТЕМ Ошибка! За	кладка не
пределена.	

#### Введение

Инженерия знания (knowledge engineering) — достаточно молодое направление искусственного интеллекта, возникшее тогда, когда практические разработчики столкнулись с весьма нетривиальными проблемами трудности «добычи» и формализации знаний. В первых книгах по ИИ эти факты обычно только постулировались, в дальнейшем начались серьезные исследования по выявлению эффективных стратегий выявления знаний.

Так и появилась наука, посвященная теоретическим и практическим проблемам проектирования баз знаний — получению и структурированию знаний специалистов для последующей разработки интеллектуальных систем.

#### Инженерия знаний — это ветвь информатики, изучающая модели и методы извлечения, структурирования и формализации (представления) знаний для их обработки в интеллектуальных и информационных системах.

Традиционно при разработке систем, основанных на знаниях (knowledge based systems), отчетливо выделяют три фазы до-машинной обработки:

- Первая добыча (получение) знаний из источника (эксперты, Интернет, специальная литература). Трудоемкость этой фазы недооценена. Результат ее — огромное количество гетерогенных (разнохарактерных) противоречивых фрагментов знаний.
- Вторая концептуализация (структурирование) разрозненных фрагментов в единую модель. Результат часто слабоформализованное представление, называемое полем знаний.
- Третья формализация поля знаний при помощи специализированных языков представления знаний (ЯПЗ), в результате чего формируется база знаний, описанная на выбранном ЯПЗ.

Центральным понятием на стадиях получения и структурирования является так называемое *поле знаний*.

Поле знаний — это условное неформальное описание основных понятий и взаимосвязей между понятиями предметной области, выявленных из системы знаний эксперта., в виде графа, диаграммы, таблицы или текста.

Очевидно, что поле знаний — это один из способов «компрессии» знаний и их наглядного представления. Однако, поле знаний — это только эскиз будущей базы знаний, но не формальное описание базы.

Собственно формализация всегда выполняется с привлечением специализированных программных систем. В данном пособии рассмотрены три различных системы:

- 1. LogicGem использует табличную форму формализации знаний.
- 2. FreeMind знания представляются на концептуальном уровне в виде mind maps или ментальных карт памяти.
- 3. SNePS сетевое описание знаний с помощью LISP-подобных структур.

Более подробное теоретическое описание проблемы инженерии знаний можно найти в книге Гаврилова Т.А., Муромцев Д.И. Интеллектуальные технологии в менеджменте. СПб. Изд-во ВШМ СПбГУ, 2008.

## Разработка таблиц решений в системе LogicGem

Самая трудная операция в разработке программ или процессах документирования — это не программирование, а усилия по покрытию всех возможных комбинаций обстоятельств — ввод пользователем, перемещение базы данных или любые другие события/правила, с которыми может столкнуться программа/процесс.

Если любая комбинация ввода данных/событий не будет предусмотрена, то программа будет функционировать случайно или бизнес процессы будут незавершенны. Хуже всего, когда ошибки случаются во время использования программы или модификации процессов, обычно программистами или аналитиками, которые не понимают полностью взаимодействие между несколькими различными возможностями.

LogicGem предоставляет простой в использовании способ создания, редактирования, проверки и компиляции программной логики или процессов, представляя проблемы визуально и делая их простыми для понимания.

## Лабораторная работа № 1. Создание таблицы

## Шаг 1. Создание таблицы решений

🗰 LogicGem								
File Edit Table Logic (	Compiler Window Help							
D 🚅 🔒 👗 🛍 🛙		n 🖊				N -	YN NY 5	\$ 91 \$91
🕮 Worksheet - Works	sheet1						_	
Rule:								
		1	2 3	4	5 6	7	8	~
<u>—</u>								
		_						_
								<b>~</b>
A1								~
				1 1				
	Freque	ency:						~
	(	Cost						~
		<						>
Condition Entry: 1, 1	Com	pletion Rati	o: 8 : 0					

Для создания новой базы знаний выберите пункт New в меню File.

Рисунок 1. Чистая таблица решений

#### Шаг 2. Добавление описаний для условий

В верхнем левом поле таблицы находится колонка описания возможных событий. Двойным щелчком мыши кликните по полю напротив C1 и введите первое событие.

🎟 LogicGem											
File Edit Table Logic Compiler Window	Help										
🗅 🖆 🖬 👗 🛍 💼 🗐 🗐	<b>N</b>	1			:			N	YN	NY 5	91 \$91
Worksheet - Worksheet1										-	
Rule:											-
		1	2	3	4	5	6	7	8		<u> </u>
С1 Сегодня рабочий день?											
			_								- 1
											<b>~</b>
A1											~
*											
											~
	Frequency:	1									
	Cost:										~
		<									>
Condition Entry: 1, 1	Completic	n Ratio	:8:0								

Рисунок 2. Добавление события

Аналогичным образом введите второе событие напротив C2 и т.д. Для этого выделите, а затем щелкните дважды левой кнопкой мыши на строке, помеченной символом «\*».

🎟 LogicGem											
File Edit Table Logic Compiler Windo	w Help										
🗅 🛩 🖬 👗 🛍 🛍 🗐 🗐	n 🖌 🖌				<b>YN NY 5</b> \$ 91 \$91						
Uorksheet - Worksheet1											
Rule:					-						
	1	2 3	4 5	6 7	8 🔼						
С1 Сегодня рабочий день?											
С2 Сегодня праздник ?											
*											
A1					<u>~</u>						
*											
					<u>×</u>						
	Frequency:				<u>~</u>						
	Lost				✓						
	<				>						
Condition Entry: 1, 1	Completion Rat	io: 8 : 0			.::						

Рисунок 3. Таблица, содержащая несколько событий

## Шаг 3. Добавление описаний для действий

Колонка описания возможных действий находится в нижнем левом углу под колонкой описания событий. Напротив А1 вводим первое действие аналогично тому, как описано в шаге 2.

File Edit Table Logic Compiler Window Help  Worksheet - Worksheet1  Rule:  С Сегодня рабочий день? С2 Сегодня праздник? идет дождь?	EogicGem											
Image: Image	File Edit Table Logic Compiler Window Help											
Worksheet - Worksheet1         Image: Compare the second sec	🗅 🚔 📕 👗 🛍 🛍 💷 📰 🔛	1			:			N	YN	NY	\$ 9	i \$9-1
Rule:       1       2       3       4       5       6       7       8         С1       Сегодня рабочий день?       Сегодня праздник?       1 <td< td=""><td colspan="12">Worksheet - Worksheet1</td></td<>	Worksheet - Worksheet1											
1       2       3       4       5       6       7       8         С1       Сегодня рабочий день?       1       2       3       4       5       6       7       8         С2       Сегодня праздник?       1	Rule:											
С1 Сегодня рабочий день? С2 Сегодня праздник? идет дождь? ** А1 идти на работу 42 идти гулять посмотреть телевизор * Frequency: Cost		1	2	3	4	5	6	7	8			
*        А1     идти на работу       А2     идти гулять       А3     посмотреть телевизор       *	С1 Сегодня рабочий день? С2 Сегодня праздник? С3 идет дождь?											
А1 идти на работу А2 идти гулять посмотреть телевизор ж Frequency: Cost:	*											
Frequency: Cost	А1 идти на работу А2 идти гулять А3 посмотреть телевизор											
Frequency: Cost:												~
	Frequency: Cost:											
Action Stub: 1 Completion Ratio: 8 : 0												

Рисунок 4. Добавление действия

## Шаг 4. Определение условий

В правом верхнем углу напротив колонки событий вводим значения событий. у и N обозначают, что событие есть, n и Y что события нет. Ввести значения событий.

🗰 LogicGem											
File Edit Table Logic Compiler Window Help											
	1			:			N -	YN	NY 5\$	91 \$91	
Worksheet - Worksheet1											
Rule:											
	1	2	3	4	5	6	7	8		~	
С1 Сегодня рабочий день?	У	•	•	n							
С2 Сегодня праздник?	n	У	У	у						_	
С3 上 идет дождь?	l ·	У	n	У							
										_	
А1 🕨 идти на работу										~	
А2идти гулять											
АЗ посмотреть телевизор										_	
*										_	
										×	
Frequency	:	1									
Cost	:										
		'					1	- 1			
Condition Entry: 3, 4 Column Count: 1 Completion Ratio: 8 : 7											

Рисунок 5. Ввод значений для событий

## Шаг 5. Определение действий

В правом нижнем углу таблицы напротив описания действий находится колонка значений действий. Ввести значения действий: если действие необходимо выполнить то ставим 1, иначе оставляем поле пустым.

🗰 LogicGem											
File Edit Table Logic Compiler Window	Help										
🗅 📂 🖬 👗 🛍 🛍 🗐 🗐	n 🛛	1			:		:	N	YN	NY 5	91 \$91
Worksheet - Worksheet1											
Rule:											-
		1	2	3	4	5	6	7	8		<u> </u>
С1 Сегодня рабочий день?		У	•	•	n						
С2 Сегодня праздник?		n	У	у	У						
С3 идет дождь?		-	У	n	У						
											_
											. ✓
А1 🕨 идти на работу		1									~
А2идти гулять				1							
АЗ посмотреть телевизор			1		1						
_ <u>*</u> _			_		_						_
											✓
í	Frequency:							1			
	Cost:										
					· · · · ·			1	· ·		
		<									2
Action Entry: 1, 1 Column Count: 2 Completion Ratio: 8 : 7											

Рисунок 6. Ввод значений для действий

#### Шаг 6. Добавление количественных параметров

Внизу под таблицей находятся значения **Frequency** и **Cost**. **Frequency** (частота) – это вероятность того, что событие произойдет. Выставьте значение в процентах. Обе величины опциональны, их можно не выставлять.



Рисунок 7. Добавление параметра Frequency

Выставьте значение **Cost** (цена). Она может иметь разные единицы, например \$, % загрузки сервера и т.д.

🗰 LogicGem										
File Edit Table Logic Compiler Window Help										
	1			:			N <b>-</b>	YN	NY 55	9-1 \$9-1
Worksheet - Worksheet1										
Rule:										-
	1	2	3	4	5	6	7	8		~
С1 Сегодня рабочий день?	у	•	•	n						
С2 Сегодня праздник?	n	у	у	у						_
L3 ▶ идет дождь?	-	у	n	у						
		1								•
А1 🕨 идти на работу	1									~
А2идти гулять			1	-						
АЗ посмотреть телевизор		1		1						
Frequency:	60			15						~
Cost:	- 34		300							~
	<									
Cost: 3 Column Count: 2 Completio	on Rat	io: 8 : 7								

Рисунок 8. Добавление параметра Cost

## Шаг 7. Сохранение таблицы решений в файл

Сохраните файл, выберите пункт Save в меню File. Введите имя файла.

## Лабораторная работа № 2. Создание логических структур

Используя широкоформатную таблицу, LogicGem позволяет составить прототип продукционной базы знаний, генерировать английскую, испанскую и французскую, немецкую документацию и исходный код на языках Basic, C, C++, Fortran, FoxPro, Java, Pascal, PowerBuilder, SAS, Visual Basic, Visual Basic.NET, Visual C#.NET или Xbase.

## Шаг 1. Проверка на пропущенные правила.

Загрузите таблицу решений, созданную в процессе выполнения предыдущей лабораторной работы с помощью пункта **Open** меню **File**. Далее, в меню **Logic** выберите пункт **Missing**.

## Шаг 2. Исправление таблицы решений

В случае обнаружения пропущенных правил или противоречивости, программа выдаст сообщение соответствующего содержания. В этом случае введите недостающее правило.



Рисунок 9. Пример сообщения о пропущенных правилах

На рисунке 10 показана таблица с автоматически добавленными нерассмотренными вариантами правил.

🧰 LogicGem - [Worksheet - G:\Публикации\Пособия и методички\Системы инжен 🔳 🗖 🔀												
🧾 <u>F</u> ile <u>E</u> dit	: <u>T</u> able <u>L</u> ogic <u>C</u> ompiler <u>V</u>	<u>V</u> indow	Help									- 8 ×
0 🖻 🔒	X № € Ⅲ Ⅲ		ŝ	Ý	•••			: :	× :	:	YN	👬 💰 9
Rule:												-
		1	2	3	4	5	6	7	8			
C1	Сегодня рабочий день?	У		-	n	n	n					
C2	Сегодня праздник?	n	у	У	У	n	n					
C3 🕨	Идет дождь?	•	у	n	У	У	n					
*												
												~
A1		1						1		1		
	идти на работу			1								<u>^</u>
			1		1							
A3 🔽	смотреть телевизор											
												<b>~</b>
	Frequency											~
	Cost											~
		<										>
Rule Descript	ion 4 Column Count: 1	10	ompleti	ion Rat	io: 8 : 9	)						

Рисунок 10. Таблица решений после автоматической проверки

## Шаг 3. Компиляция таблицы решений

Если все возможные варианты учтены, нажмите кнопку **Compile**. Если ошибки все-таки остались, то программа выведет сообщение (рисунок 11) и окно компиляции с соответствующими комментариями (рисунок 12).

LogicGem	×
Columns 2 and 4 are ambiguo	us.
ОК	

#### Рисунок 11. Пример сообщения о противоречивости таблицы решений

Если обнаружено противоречие, то выделите один из указанных столбцов (в данном случае, например, 4) и удалите его с помощью клавиши Delete.



Рисунок 12. Окно компиляции

Перейдите в рабочее окно таблицы решений с помощью меню **Windows**, и внесите необходимые правки, например, как показано на рисунке 13. Если все возможные варианты учтены, нажмите кнопку **Compile**.

🗰 LogicGer	m - [Worksheet - G:\Ny(	блика	ции√	Іособ	ияи	метод	цички	\Сис	гемы	инжен	💽	
🔢 <u>F</u> ile <u>E</u> dit	<u>T</u> able Logic ⊆ompiler ⊻	<u>V</u> indow	<u>H</u> elp									- 🛛 ×
🗅 😅 日	X 🖻 🖪 🗐 🗐		ŝ	ý	•••			: :		: ×-	YN	🔹 🛐
Rule:												
		1	2	3	4	5	6	7	8			▲
C1	Сегодня рабочий день?	У	•	•	n	n	n					
	Сегодня праздник?	n	у	У	У	n	n					
	идет дождь?	·	у	n	У	У	n					
<u></u>								1		1		
												×
A1	идти на работу	1										~
A2 🕨	идти гулять			1			1					
A3	смотреть телевизор		1		1	1						
*												_
												~
	Frequencu						1	1	1	1		
	East											
	000.				1						_	<u> </u>
		<										>
Action Entry: 2	2, 6 Column Count: 1	C	ompleti	ion Rat	io: 8 : 9	9						

Рисунок 13. Исправленная таблица решений

#### Шаг 4. Автоматическая генерация продукционных правил

В секции **Options** справа окна компиляции поставьте галочки в полях:

- Rule Numbers
- Table
- Rule Discription

В списке Languages выберите язык. Пускай вас не удивляет тот факт, что русского языка в списке нет.

Выберите в секции Structure пункт Rule List. Нажмите кнопку Compile.



Рисунок 14. Окончательный результат компиляции таблицы решений

Таким образом, вы сможете посмотреть действия всех правил под их номерами.

## Разработка карт памяти (mindmaps) в системе FreeMind

Лабораторная работа № 1. Создание и сохранение новой карты

#### Шаг 1. Создание карты

Начнем создавать новую карту знаний, открыв меню **Файл-Новый**. На экране должен появиться овал с надписью «Новая ассоциативная карта». Это корневой узел (root node). Мы будем строить карту, добавляя записи к корневому узлу.

Корневой узел подкрашен серым цветом, что означает его текущее выделение. Кликните мышкой по овалу и переименуйте его название, например «Статья» и нажмите Enter.

💽 Acc	оциати	ивная ка	рта1*	- FreeMino	d - Режим	ı "MindMap"									<u>_     ×</u>
Файл	Измени	ить текст	Вид	Вставить	Формат	Перемещение	Дополнительно	Карты	Помощь						
	<b>ə</b> 📛	了 100	)%		•	] 🗋 😭 🗎	🛃 🕹 🤏 🧧	6 🗅 🗖	) 💡 i	b 💬	m	<b>~</b> =	■ 💠 🕳 [s	ansSerif	▼ 12
							стат	ья							
뜨															
Ред	актиров	ать Фор	мат '	Таблица											
	$\sim$	ሯ 🖻	<b>i</b>	AcadEre	əf	-	8 💌 b	i 🛛 🄇	፼   ≣	≣ ≣	IΞ	ΙΞ			
															<u>+</u>
Тек	ст <mark>HT</mark> M	IL Код													

Рисунок 15. Новая карта знаний

#### Шаг 2. Добавление дочерних узлов

Создадим теперь дочерние узлы для нашего корневого узла «Статья». Выберите опцию **Новый дочерний узел**, нажав правой кнопкой мыши по корневому узлу. Назовем дочерний узел «Кто читает».

💽 Acc	оциативная карта1*	- FreeMind -	Режим "MindMap"						
Файл	Изменить текст Вид	Вставить Ф	Формат Перемещение	Дополнительно	Карты Пом	ющь			
] 🕞 (	ə 📥 🕜 🔤		💌 📄 🔁 🖃	🛃 👟 🤏 💰	- 🗅 🛍 💡	i b 💬	ff 🕂 🖶 📼	💠 💳 SansSerif	▼ 12
				Стать	я Кто чи	тает			*
									▼ ▶
Реда	актировать Формат	Габлица							
	~ X 🖻 🛍 👭	AcadEref	•	8 🔽 b :	i 🛛 🤣 🛛	≣ ≡ ⊒	I≡ ≬⊟		1
									×
Тек	ст HTML Код								
Карта б	ыла автоматически сох	ранена (в фай	іл C:\Documents and Setti	ngs\Администрато	op\.freemind\Fl	M_unnamed528	024084741638	178.mm)	

Рисунок 16. Карта с добавленным узлом

Теперь аналогичным способом, нажав правой кнопкой мыши по корневому узлу, добавим еще три дочерних узла, которые будут соответственно называться «Иллюстрации», «Материалы», «Ссылки».

🕄 Ассоциативная карта1* - FreeMind - Режим "MindMap"	
Файл Изменить текст Вид Вставить Формат Перемещение Дополнительно Карты Помощь	
🕞 😌 🚑 🍸 100% 💽 📄 😭 😂 🗢 🍕 🖒 🖺 🔗 i b 💬 🎬 🖶 🛥 💠 SansSerif	12
	<b></b>
Материалы Ссылки Статья Кто читает Иллюстрации	
۲	<b>v</b>
Редактировать Формат Таблица	
∽ ∽ 🐰 🖻 🛍 🚧   AcadEref 💽 8 🔽 b i 型 🥹   ≣ ≣ 🗄 🗄	
	▼
Tekct HTML KOA	

Рисунок 17. Карта с несколькими дочерними узлами

#### Шаг 3. Узлы с общим родителем

Основа готова. Теперь создадим узлы имеющие, общего родителя (sibling nodes) (на одном уровне) с дочерним узлом «Кто читает».

Для этого нажимаем правой кнопкой мыши по дочернему узлу «Кто читает» и выбираем опцию **Новый дочерний узел**. Назовем его «Студенты».

🛐 Ассоциативная карта1* - FreeMind - Режим "MindMap"	
Файл Изменить текст Вид Вставить Формат Перемещение Дополнительно Карты Помощь	
🕞 😌 🚔 🏹 100% 💽 📄 😭 😂 😂 🛷 🍕 🖒 🎦 😌 i b 💬 🇱 🖶 🛥 💠 SansSerif	12
Материалы Ссылки статья Кточитает Студенты Иллюстрации	
	• •
	* *
TEKCT HTML KOA	
Карта была автоматически сохранена (в файл C:\Documents and Settings\Администратор\.freemind\FM_unnamed2550697347577461763.mm)	

Рисунок 18. Дочерний узел " Студенты "

Аналогичным способом добавим еще дочерние узлы «Преподаватели» и «Любопытствующие».

💽 Ассоциативная карта1* - FreeMind - Режим "MindMap"	
Файл Изменить текст Вид Вставить Формат Перемещение Дополнительно Карты Помощь	
🕞 🕤 📇 🏹 100% 💽 📄 😭 😂 🗢 🍕 🤚 🛅 😌 i b 🖓 🎬 🖶 🛥 💠 SansSe	rif 🔽 12
Студенты Материалы Ссылки Ссылки Иллюстрации	×
	<b></b>
Редактировать Формат Таблица   ∽ ∽ Ӽ № № № АсаdEref ▼ 8 ▼ b ź Щ 00 ] ≣ ≣ ≣ ≣ ≣ Текст НТМL Код	×

Рисунок 19. Несколько узлов с общим родителем

Теперь создадим узлы имеющие, общего родителя (sibling nodes) (на одном уровне) с дочерним узлом Иллюстрации.

Для этого нажимаем правой кнопкой мыши по дочернему узлу Иллюстрации и выбираем опцию Новый дочерний узел. Первый узел назовем Wiki source, второй Image search.

🖸 Ассоциативная карта1* - FreeMind - Режим "MindMap"	
Файл Изменить текст Вид Вставить Формат Перемещение Дополнительно Карты Помощь	
🕞 🕤 🚔 🏹 100% 💽 📄 😭 😂 😂 🛷 🍕 👆 🛅 😌 i b 💬 🇱 🖶 🛥 💠 SansSerif	▼ 12
Студенты Кто читает Преподаватели Материалы Ссылки статья Любопытствующие Иплюстрации Wiki source	
Image search	
	• •
Редактировать Формат Таблица	
▷ ♀ 🐰 🖻 🛍 🏘   AcadEref 💽 🔹 🕨 b i 🙂 🥹   ≣ ≣ 🗄 🗄	
	×
TEKCT HTML KOA	
Карта была автоматически сохранена (в файл C:\Documents and Settings\Администратор\.freemind\FM_unnamed5076778133343238679.mm)	

Рисунок 20. Узлы имеющие, общего родителя «Иллюстрации»

Следующим шагом создадим узлы имеющие, общего родителя (sibling nodes) (на одном уровне) с дочерним узлом «Материалы».

Для этого нажимаем правой кнопкой мыши по дочернему узлу Материалы и выбираем опцию Новый дочерний узел. Первый узел назовем Книги, второй Статьи, третий Исследования, четвертый Новости.

💽 Ассоциативная карта1* - FreeMind - Режим "MindMap"	
Файл Изменить текст Вид Вставить Формат Перемещение Дополнительно Карты Помощь	
$ \boxed{\bigcirc} \bigcirc \bigcirc \bigcirc i  \mathbf{b} \bigcirc \textcircled{m} \\ \blacksquare \bigcirc \blacksquare \bigcirc \blacksquare \textcircled{m} \\ \blacksquare \textcircled{m} @ \blacksquare \bigcirc \blacksquare \bigcirc \blacksquare \textcircled{m} @ \blacksquare \bigcirc \blacksquare \bigcirc \blacksquare \bigcirc \blacksquare \blacksquare \blacksquare \bigcirc \blacksquare \blacksquare \blacksquare \blacksquare \blacksquare \blacksquare \blacksquare \blacksquare \blacksquare$	<b>v</b> 12
Книги Студенты	
Исследования Материалы статья Любопытствующие	
Новости Wiki source	
Ссылки Image search	
٠	▼ ▶
Редактировать Формат Таблица	
∽ ∽ 🐰 🗈 🛍 🏟   AcadEref 💽 8 💌 b i 🖳 🥺   ≣ ≣ 🚍 🗄	
	* *
TEKCT HTML KOA	

Рисунок 21. Узлы имеющие, общего родителя «Материалы»

Следующим шагом создадим узлы имеющие, общего родителя (sibling nodes) (на одном уровне) с дочерним узлом «Ссылки».

Для этого нажимаем правой кнопкой мыши по дочернему узлу Материалы и выбираем опцию Новый дочерний узел. Первый узел назовем Релевантные статьи, второй Ресурсы, третий Новые слова.

💽 Ассоциативная карта1* - FreeMind - Режим "MindMap"	_ 🗆 🗵
Файл Изменить текст Вид Вставить Формат Перемещение Дополнительно Карты Помощь	
$\fbox{$\bigcirc$} \textcircled{$\bigcirc$} \textcircled{$\bigcirc$} \textcircled{$\land$} @ `` `` `` `` `` `` `` `` `` `` `` `` `$	<b>•</b> 12
Книги Статьи Исследования Новости Релевантные статьи Ресурсы Новые слова Ссылки Статья Студенты Преподаватели Статья Любопытствующие Иллюстрации Image search	Ă
Редактировать Формат Таблица	
∽ ∽ 🔏 🛍 🛍 🚧   AcadEref 🔹 💌 🛛 🔽 🙂 😥 🦉 🔄 🗄 🗄	
	*
TEKCT HTML KOA	

Рисунок 22. Узлы имеющие, общего родителя «Ссылки»

#### Шаг 4. Сохранение карты

Новая простейшая карта готова. Осталось ее сохранить. Для этого выбираем из меню **Файл-Сохранить как**.

💽 Ассоциативная карта1* - FreeMind - Режим "Min	ıdMap"	
Файл Изменить текст Вид Вставить Формат Пере	емещение Дополнительно Карты Помощь	
📄 Новый	Ctrl+N 💊 🐟 🦂 🖄 🚺 💡 i b 💬 🎬 🛟 📼 🛟 👄 SansSerif	- 12
📔 Открыть	Ctrl+O	<u> </u>
🔚 Сохранить	Ctrl+5	
🔛 Сохранить как	Ctrl+Shift+S	
Вернутьс Сохранить как рсии		
😑 Создать зашифрованную ассоциативнубю карту		_
Сохранить все	Студенты	
Закрыть	Ctrl+W Кто читает / Преподаватели	
Экспортировать	Калана Статья Мобопытствующие	
Импортировать	Иллюстрации Image search	
Параметры страницы		
📥 Напечатать	Ctrl+P	
Просмотреть		
Недавние файлы	<b>&gt;</b>	▼ ▶
Выйти	Ctrl+Q	
📘 🗠 🖙 👗 🛍 🛍 🏘 🗍 AcadEref		
		÷
Текст НТМL Код		

Рисунок 23. Сохранение карты

Выбираем путь сохранения, пишем название карты и нажимаем Save. Карта сохранена.

## Лабораторная работа № 2. Форматирование карты

Теперь немного приукрасим карту, сделанную нами в первой лабораторной работе и привлечем к ней внимание пользователей.

#### Шаг 1. Изменение типа узла

В FreeMind можно выбрать тип узла: линия или овал. По умолчанию у нас выбран тип Линия. Сделаем теперь тип узла Овал.

Для этого необходимо выделить нужный узел и нажав на правую кнопку мышки выбрать из контекстного меню **Формат-Тип** узла «овал».

💽 Ста	татья.mm* - FreeMind - Режим "MindMap" C:\Documents and Settings\Администратор\Рабочий стол\лабы\Статья.mm	
Файл	л Изменить текст Вид Вставить Формат Перемещение Дополнительно Карты Помощь	
0	🕤 😓 🏹 100% 💌 🔄 🖆 🖆 😂 🖉 🗢 🤜 💰 👆 🏠 💡 i b 💬 🎬 静 🛥 💠 🕳 SansSerif	▼ 12
	Книги	
	Статьи Материалы Студенты	
	Исследования Кто читает Преподаватели	
	Релевантные статьи Wiki source	
	Ресурсы Ссылки Иллюстрации Image search	
	Новые слова	
		•
II_		Þ
Ред	здактировать Формат Таблица	
	🗠 🖙 🔏 🏘 🛛 AcadEref 🛛 🔽 🛛 🔽 🖢 i 🙂 🥹 ) 三 三 三 三 三	
Тек	ekct HTML Kod	

Рисунок 24. Измененный тип узла

#### Шаг 2. Изменение цвета узла

Для большего привлечения к теме карты подкрасим корневой узел красным цветом.

Для этого правой кнопкой мыши нажмем на корневой узел и выберем из меню **Формат – Фоновый цвет узла** и выберем из палитры красный цвет.

	Статья.mm* - FreeMind - Режим "MindMap" C:\Documents and Settings\Администратор\Рабочий стол\лабы\Статья.mm	<u>_                                    </u>
Фа	айл Изменить текст Вид Вставить Формат Перемещение Дополнительно Карты Помощь	
	ତି 🙃 🚔 🏹 100% 🔽 💽 🖆 🖆 🗳 🔷 🍬 💰 🤚 💆 🦻 $i$ b 💬 🎬 🖶 🛥 💠 🗢 SansSerif	▼ 12
	Книги	*
	Статьи Исследования Новости Релевантные статьи Исследования Материалы Статья Статья Студенты Преподаватели Любопытствующие (Wiki source)	
	Ресурсы Новые слова	
		▼ ▶
F	Редактировать Формат Таблица	
	∽ ∝ 🐰 🗈 🔞 🏘   AcadEref 🛛 💌 🛛 🔽 🖢 i 型 🥝   書 書 ☷ ☷	
Ľ	TEKCT HTML KOA	

Рисунок 25. Изменение фона корневого узла

Аналогичным способом покрасим в синий цвет Дочерние узлы корневого узла Статья.

💽 Статья.mm* - FreeMind - Режим "MindMap" C:\Documents and Settings\Администратор\Рабочий стол\лабы\Статья.mm	
Файл Изменить текст Вид Вставить Формат Перемещение Дополнительно Карты Помощь	
$\boxed{\bigcirc} \bigcirc \bigcirc \bigcirc i \\ \boxed{\bigcirc} \bigcirc \bigcirc i \\ \boxed{\bigcirc} \bigcirc $	• 12
Книги Статьи Исследования Новости Релевантные статьи Ссылки Новые слова	
	▼ ▶
Редактировать Формат Таблица	
	* *
Герст Линском ј Карта была автоматически сохранена (в файл C:\Documents and Settings\Администратор\.freemind\FM_Статья.mm5046397844232045470.mm)	

Рисунок 26. Изменение фона дочерних узлов

## Шаг 3. Добавление облаков

Также для построения карт можно создавать «облака», которые охватывают узлы со сложной структурой. Часто облака используются для привлечения внимания, выделения информации на экране.

Для этого необходимо выделить нужный узел и нажав на правую кнопку мышки выбрать из контекстного меню **Вставить - Облако**.



Для облака можно выбрать цвет: меню Формат-Цвет облака.

Рисунок 27. Карта с облаками

На рисунке 27 приведен пример отформатированной карты с помощью различных свойств, которые представляет программа FreeMind для создания карт знаний. Для сохранения карты нажмите **Файл – Сохранить**.

## Лабораторная работа № 3. Добавление ссылок на внешние ресурсы

Узлы могут содержать связи на:

- веб страницы,
- локальные папки,
- исполняемые файлы,
- другие документы на Вашем локальном компьютере или в сети.

Построим карту памяти, которая бы помогала сотруднику в поиске и покупке комплектующих материалов для поддержки функционирования вычислительной сети, применив в ней различные ссылки.

## Шаг 1. Подготовка к работе

Создадим новую карту **Файл - Новый**. Изменим название корневого узла на «Комплектующие материалы».

💽 Ассоциативная карта1* - FreeMind - Режим "MindMap"	
Файл Изменить текст Вид Вставить Формат Перемещение Дополнительно Карты Помощь	
🛛 🕞 🕞 😓 🍞 100% 💽 🔄 🖆 🚰 🚰 營 🗢 🤜 🐗 🖒 🖍 💡 $i$ b 💬 🎬 🖶 🛥 💠 👄 SansSerif	• 12
Комплектующие материалы	
	v k
Редактировать Формат Таблица	
	*

Рисунок 28. Новая карта "Комплектующие материалы"

Создадим теперь дочерние узлы для нашего корневого узла Комплектующие материалы. Выберите опцию **Новый дочерний узел**, нажав правой кнопкой мыши по корневому узлу.

Назовем дочерние узлы:

- «Поиск» (сюда мы вставим ссылку на поисковую систему Яндекс, чтобы можно было находить интересующие комплектующие через Интернет);
- «Расчет стоимости» (сюда мы добавим ссылку на исполняемый файл, на Калькулятор, который пригодится для подсчета необходимых затрат на покупку комплектующих);
- «Что нужно купить» (сюда будут записываться конкретные названия комплектующих с указанием цен, после подсчета на Калькуляторе);
- «Документы» (здесь будет ссылка на локальную папку Мои документы, перейдя на которую пользователь сможет уже оформлять необходимые накладные на покупку оборудования и комплектующих).

💽 Ac	социативная карта1*	· - FreeMind - Peж	им "MindMap"							
Файл	Изменить текст Вид	Вставить Форма	т Перемещение	Дополнительно	Карты Г	Томощь				
] 🕞	🕤 📇 🍸 🛛 100%		🗕 🗋 🗋	🔡 👟 🧇 🤞	i 🗅 🚨	💡 i	<b>b</b> 💬	# 🔶 •	🗖 💠 💳 SansSe	erif 🗾 12
										<u> </u>
						_	_			
		<u>что нух</u> Д	кно купить Іокументы	омплектующие і	материаль		<u>Тоиск</u> Расчет сті	римости		
		-								
뿌	l									
Pez	актировать Формат	Таблица								
	n na 👗 🗈 🛍 🏕	AcadEref	•	8 <b>• b</b> ;	i 🛛 🥝	] ≣	= =	≡ }≡		
										×
Tei	ст HTML Код									

Рисунок 29. Добавление дочерних узлов

## Шаг 2. Добавление ссылок

Теперь необходимо сделать ссылки. Сначала добавим ссылку на калькулятор. Для этого выделим дочерний узел «Расчет стоимости» и выберем из меню Вставить-Указать ссылку (выбор файла). В открывшемся диалоговом окне указываем путь к исполняемому файлу calc.exe и нажимаем **Оpen**.



Рисунок 30. Добавление ссылки на calc.exe

На нашем дочернем узле Расчет стоимости появилась красная стрелка, которая показывает на ссылку. Нажав на нее, перед нами открывается калькулятор. Теперь можно считать.

🐼 Ко Файл	иплектующие мате Изменить текст Ви	ериалыЗ .mm* – FreeMind - Режин "MindMap" C:\Documents and Settings\Адиинистратор\Рабочий стоп\лабы\Конплектующие натериалыЗ .mm из Вставить Формат Переишение дополнительно Карты Поксшь	_IDI×
0		💌 🕒 😭 🖓 🐨 💊 🔹 🖄 😌 🖉 🌘 💬 🎬 😓 🛶 🔶 Sanssenf 🛛 💌 12	•
		Calculator       Image: Second s	*
🕁 🖉 😁 😧 🖉 🎝 🖧 🎝 🎝 🖉 🖉 🕲 🛞 🖓 🛥 🖒 🗘 🛥		<u>Чо нужно купить</u> Документы Комплектующие материалы Документы Поиск Документы Поиск Документы Поиск	
2	<b>.</b>		<u>&gt;</u>
務	Редактировать Фо	opmar Ta6nnua 1830 AAA  AcadEref ▼ 8 ▼ 15 / U 69   王 王 15 15	
leale.		no. 🖸 Konneersenuu 🕷 FreeMed - Mirro. 🕷 asiat - Mirroof 🕷 asiat - Mirroof 🕞 ABRV Jonon 😺 100 📑 Calculator 📰 🤉 🕫	// 🤲 🗐 🖨 🏚 13:49

Рисунок 31. Вызов калькулятора по ссылке

## Шаг 3. Добавление ссылки на документ или папку

Заполним узел «Что нужно купить». Например, картридж для принтера Samsung scx-4200 2 шт. Для этого правой кнопкой мыши нажмем на узел «Что нужно купить» и выберем **Новый дочерний узел**.

Теперь сделаем ссылку на документы, которые необходимы при покупке оборудования (накладные и т.п.). Все необходимые файлы с документами хранятся в папке «Мои документы». Сделаем ссылку на эту папку.

Для этого выделим узел «Документы» и выберем из меню Вставить-Указать ссылку (текстовое поле). В открывшемся диалоговом окне прописываем путь к папке «Мои документы» и нажимаем Ок.

C I	омплектую: іл Изменить	цие материалыЗ .mm* - Fr гекст Вид Вставить Форн	eeMind - Режим "MindMap" мат Перенешение Дополни	C:\Documents and Set тельно Карты Помош	tings\Администратор	Рабочий стол	і∖лабы\Комплектую	щие материалыЗ .mm		_0×
6	047	300%		) 📄 🖬 🛃 👟 🤜	- 	b 💬 🌐	🛟 🖴 💠 👄 Sans	Serif	- 12	-
	1200	2 шт								
0000						Кс	мплект	ующие ма	атериал	ы
) () () () () () () () () () () () () ()				докуме	нты					
(∂ )				Input Variant E://ho	ить ссылку вручную и документы/ Ж Cancel	×				
\ <u>Å</u> ☆	•									
	Редактиро	вать Формат Таблица								
1	100	🔏 🗈 🛍 🏘 🛛 AcadEre	f 🗾 8	💌 b i 😐 🥹						
	Текст Н	MLКод								
Капт	а была автома Іуск 🔯 Ком	тически сохоанена (в файо С: плектующие мат 📴 Fr	1Documents and Settinos)Алиин eeMind - Microsoft Word   66 Г	истратор) freemind)EM и Письмо от Муромцев Д	nnamed6473307651359288   🕎 na6a1 - Microsoft W	5526.mm) ord 🛛 🕅 ла	6a3 - Microsoft Word	work (E:)	RU ( <	× 🛃 🧙 😋 🍬 14:09

Рисунок 32. Ссылка на папку

На нашем узле Документы появилась красная стрелка, которая показывает на ссылку. Нажав на нее, перед нами открывается папка «Мои документы» с необходимым содержимым.



Рисунок 33. Переход к папке по ссылке

## Шаг 4. Добавление ссылок на веб-ресурсы

Для этого выделим узел «Поиск» и выберем из меню Вставить-Указать ссылку (текстовое поле). В открывшемся диалоговом окне прописываем путь и нажимаем Ок.



Рисунок 34. Добавление веб-ссылки

На узле «Поиск» появилась красная стрелка, которая показывает на ссылку. Нажав на нее, перед нами открывается окно с поисковиком.

<u>ख</u> к	омплектующие материалыЗ .mm* - FreeMind	Режим "MindMap" C:\Do	cuments and Settings	\Администратор\Рабо	чий стол\лабы\Комп	лектующие натериаль	43 .mm		_ [] ×
Фай	л Изненить текст Вид Вставить Формат Пер	<sup>е</sup> 🖉 Яндекс - Windows I	internet Explorer			للحا	괴죄		
	S = 7     150%		p://www.yandex.ru/	▼ 4:	🗲 🔀 ICQ Search	\$		- 12	*
		Ссылко 2 Настройка с Ссылко 2 Настройка с Ссылко 2 Настройка с Яндек	р://www.yandex.ru/ Избранное Сервис сылок с сотрудники с Поиск Карр	Cnpaeka Cnpaeka Cnpaeka Cnpaeka Newk Newkow Character Chara		- 18 - 03 - 1 & 3. Страница - 20 Серенс Видео Картинки е	× " ×	<b>•</b> 1 <sup>12</sup>	×
000000000000000000000000000000000000000	катридж для принтера Samsu	Найдётся в перель автоннять м Запоннять м Запоннять м автоннять м автоннать	сё Например , сё Например , Сайтоа Разплачение Сорт и ламх Разплачение Сорт и ламх Разбла и учеба Компьютеры	опимпийская коменда - Маркот мобливные телефоны Съявления о продаже новые Круг новые	Санкт- Петербург. 20 Погода 🔆 – 2 ночню – 2 завтра – 1 Котировки	расширанный п марта, пятница пробик 🌑 5 ба Двихенне плонее Карта Санкт. Петербурта Ф. 100%	→ Пои → Пои	ск нет стоимоя	сти
♪ ₹~ ∂ ∑ <b>? ┑ ::</b> ९ ♂ •: • <b>:</b> < <b>:</b> < : • : • : • : • : • : • : • : • : • :	Peger-reposate Gopean Ta6/reus Point & Ba (A Acadiref Taccr [HTML Kog]		b <i>i</i> ⊻ ⊕ ] ≡	æ æ  <del>III</del> III					
C:/Pri	noram Files (Internet, explorer/lexplore								
20 P	уск 🔯 Комплектующие 📴 FreeMind - Micros.	. С Письмо от Муром	💌 лаба1 - Microsoft	. 💌 лабаЗ - Microsoft	. 🥪 Work (E:)	internet explorer	🔄 🏉 Яндекс -	Windo 🔣 🕫	e 🤲 🛃 🥦 🗿 🖧 14:39

Рисунок 35. Переход по веб-ссылке

## Шаг 5. Работа с картой

Теперь найдем необходимый картридж в поисковике. Например, он будет стоить 88 у.е. Теперь, открыв калькулятор, умножаем 88\*2=176 у.е. и умножаем на курс у.е. 176\*33,42=5881,42, Мы узнали необходимую сумму на покупку двух картриджей. Теперь можно отредактировать узел «Картридж» для принтера Samsung scx-4200 2 шт., написав туда общую стоимость.

Измените с помощью кнопки **Редактировать** название узла «Картридж», как показано на рисунке 36.

3	Ko	мп	лек	тун	ощи	ем	атер	иал	ы3.	nm	* - F	ree	Min	ıd -	Pe	жи	м "Г	Mino	IM	ap'	' C:'	\Do	cur	ner	nts a	and	Set	ing	js∖/	٩дм	ини	стр	ато	p\F	або	чий	і сто	л\ла	бы∖к			×
Фа	йл	ŀ	1змен	ниті	ь тен	κст	Вид	B	тав	1ТЬ	Фо	омат	тΓ	Пер	еме	еще	ние	A	опо	олні	ител	пьно		Кар	ты	По	мощі	•														
	•	0	6	5	7	100	%					]	•		2	e	E	1		Ł	1	- 0	5	è		6	i	]	b	P	m	đ			• -	-	5ans9	ierif			- 12	2
K	ap	тр	идж	ДЛ	я пр	инт	epa :	San	nsun	gsı	cx-40	200	2 ш	JT.	51	881	,92	руб.	пе	Й	L	<u>юн</u> <u>Д</u>	уж оку	ме	нты	1ТЬ →	J	Ko	MR	лек	тую	цие	мат	гер	иали	ы	$\subset$	→ П _1 P	оиск асчет	стои	мост	и
																																									D	∙
F	ел	ак	тиро	ват	ъ	Door	ат	Таб	пица																																	
		) )	्य     मा	ж ML	Е	) (	3	<b>a</b> ]	Aca	dEre	f	_	_			_	•	8		•	] ]	b	i	U		3	E	ł	E :	=	E	4	:									

Рисунок 36. Работа с картой

# Лабораторная работа № 4. Использование ссылок на другие карты

Теперь создадим карту, в которой содержатся ссылки на другие карты знаний.

## Шаг 1. Подготовка к работе

Для начала создадим простую карту знаний «Живые существа» с дочерними узлами: «Животные», «Растения», «Грибы», «Бактерии».

<b>Q</b> A	ссоциативная карта1*	' - FreeMind - Режим '	"MindMap"			
Файл	и Изменить текст. Вид	Вставить Формат	Перемещение Дополнительн	о Карты Помощь	>	
] 🕞	🕤 📥 🍸 🛛 150%	-	] 🗋 😭 🗃 🛃 👟 🤏 🧯	🎸 🗅 🛅 💡 i	b 💬 🎬 🕂 🛏 💠 🤆	SansSerif 🗾 12
						<u> </u>
		Животн	ные Живые су	Пества	Грибы	
		Растен	ния	E	5актерии	
별						
Pe	дактировать Формат	Таблица				
	റ വം 👗 🗈 🛍 🖊	AcadEref	<b>v</b> 8 <b>v b</b>	i <u>u</u> 🤣 ] ≣	≡≡ Ё Ё	
						Ā
Te	KCT HTML KOA					

Рисунок 37. Новая карта «Живые существа»

Теперь необходимо сделать 4 небольшие карты знаний, которые давали бы развернутую классификацию каждого подкласса «Живые существа». На них мы и будем создавать ссылки с нашей карты «Живые существа».

Создадим карту «Бактерии» с дочерними узлами: «Тип I. Gracilicutes», «Тип II. Firmicutes», «Тип III. Tenericutes», «Тип IV. Mendosicutes».

Создадим карту «Грибы» с дочерними узлами: «Хитридиомицеты», «Гифохитриомицеты», «Оомицеты», «Зигомицеты», «Аскомицеты», «Базидиомицеты». Создадим карту «Животные» с дочерними узлами: «класс Рыбы», «класс Птицы», «класс Млекопитающие».

Создадим карту Растения с дочерними узлами: «Мхи», «Деревья», «Кустарники».

## Шаг 2. Создание ссылки на другую карту

Для создания ссылки на другие карты знаний (которые мы заранее заготовили: «Грибы», «Бактерии», «Животные», «Растения»), выбираем нужный узел, в меню Вставить - Указать ссылку (выбор файла) и указываем путь к нужной нам карте.

Так для дочернего узла «Грибы» создаем ссылку на карту с названием «Грибы», для узла «Бактерии» – ссылку на карту с названием «Бактерии», для узла «Животные» – ссылку на карту «Животные», и для узла «Растения» – ссылку на карту «Растения».



Рисунок 38. Карта с добавленными ссылками

## Шаг 3. Переход по ссылке

Чтобы перейти по ссылке необходимо нажать на нее, либо выделить узел и нажать Ctrl+Enter. Нажав на ссылку узла Животные карты, Живые существа, вы попадаете на карту Животные и аналогично на другие карты знаний.

## Лабораторная работа № 5. Работа с картами памяти

#### Шаг 1. Загрузка карты

Загрузите карту, созданную в работе 1 с помощью меню Файл – Открыть.



Рисунок 39. Загруженная карта

#### Шаг 2. Поиск на карте

Для поиска на карте памяти используйте комбинацию Ctrl+F или через меню Изменить текст-Искать. Найдем узел «Новости» исходящий из корневого узла «Статья». Выделяем узел «Статья», нажимаем Ctrl+F и пишем название узла «Новости».

Input			×
?	Искомый текст НОВОСТИ		
	ОК	Cancel	

Рисунок 40. Ввод названия узла для поиска



Нажимаем ОК, искомый нами узел подсвечивается серым цветом.

Рисунок 41. Подсветка серым найденного узла

## Шаг 3. Сворачивание узлов

Узлы могут сворачиваться (чтобы не занимать много места на экране). Для этого необходимо просто щелкнуть узлу и исходящие от него ветви свернуться. Повторный щелчок мыши по узлу разворачивает узел. Причем надо заметить, что корневой узел свернуть нельзя.



Рисунок 42. Карта со свернутыми узлами

## Лабораторная работа № 6. Импорт и экспорт карт памяти

Важными характеристиками любых пользовательских программных пакетов являются возможности импорта и экспорта документов. У FreeMind с этим все в порядке. Внутренний формат данной программы признается многими другими решениями для mind mapping — импорт документов FreeMind стал вполне стандартной опцией практически во всех приличных пакетах для «карт памяти». Что касается необычного импорта, FreeMind может загрузить в том числе файлы закладок браузера или даже структуру каталогов, находящихся на диске вашего компьютера.

#### Шаг 1. Загрузка карты

Загрузите карту, созданную в работе 1 с помощью меню Файл – Открыть.

#### Шаг 2. Импорт ветви из другой карты

Можно импортировать ветвь из другой карты знаний. Для этого выберите меню **Файл - Импортировать - Импортировать ветвь из (mm-) файла**: и укажите имя файла. Возьмем для примера файл «Животные», импортировав его к узлу «Книги».



Рисунок 43. Карта с импортированной ветвью

## Шаг 3. Экспорт карты в различные форматы

Возможности экспорта карт FreeMind достаточно широки. Можно экспортировать как карту целиком, так и какую-либо ее отдельную ветвь.

Готовую карту можно сохранить в виде изображения. Поддерживаются форматы SVG, PNG и JPEG. Ее можно сохранить в виде PDF-файла или документа в формате OpenOffice. Возможен экспорт в XHTML, причем в двух вариантах: простой или работающий на JavaScript.

Выберите меню Файл-Экспортировать-Как XHTML (использовать Java script).



Рисунок 44. Экспорт в ХНТМL

## Шаг 4. Защита карты

Сохраняемую карту также можно зашифровать, указав пароль: выберите меню Файл-Создать зашифрованную ассоциативную карту.

В появившемся диалоговом окне ввести пароль.

💽 Выберите пароль зашифрованного у	зла 🗙
Введите пароль.	
Введите пароль: П Повторите пароль:	
ОК Отменить	J

Рисунок 45. Ввод пароля для шифрования карты

Теперь карта защищена паролем.

## Лабораторная работа № 7. Различные режимы работы

FreeMind предлагает несколько режимов работы с картой. Режим выбирается из меню **Карты**:

- File (Alt+3)
- Mind Map (Alt+1)
- Browse (Alt+2)

#### Шаг 1. Загрузка карты

Загрузите карту, созданную в работе 1 с помощью меню Файл – Открыть.

#### Шаг 2. Переключение режимов

Основным является режим **Mind Map**, в котором и происходит создание и редактирование карты.

Просматривать сверстанную карту удобнее всего в режиме **Browse**. Здесь все сконцентрировано на демонстрации карты, что немного напоминает аналогичные режимы в презентационных пакетах типа Microsoft PowerPoint.

Еще один режим — File — превращает программу в своеобразный файловый менеджер, который демонстрирует структуру каталогов вашего компьютера в виде «карты памяти», элементами которой выступают файлы и папки. Доступна возможность быстрого поиска по именам файлов (на правую кнопку мыши узла «Искать»). К сожалению, на этом функциональность данного режима заканчивается — вы сможете только открыть файл или папку, но другие операции над ними здесь не работают.

## Разработка семантических сетей в системе SNePS

## О программе SNePS

SNePS (The Semantic Network Processing System) – система представления знаний, рассуждений и действий на основе логических, фреймовых и сетевых моделей.

*Семантическая сеть* – это взвешенный ориентированный граф, узлы которого представляют объекты, связи – парные отношения (например, связь R, идущая от узла n к m, означает: объект n имеет отношение R к объекту m).

SnePS использует пропозициональную модель знаний, т.е. база знаний состоит из набора утверждений о различных предметах и явлениях. Любое утверждение в сети обозначается узлом графа. Отношения между объектами (связи) могут быть частью синтаксической структуры узла, из которого они исходят.

Любой объект сети представлен уникальным узлом. Это обусловлено особенностью SnePS2: если добавляется узел, полностью повторяющий уже существующий в сети (с тем же набором связей), то вместо создания нового используется старый узел.

## Типы узлов

В сетях SnePS существует 4 вида узлов: базовый, переменный, молекулярный, структура.

Базовые узлы не имеют исходящих связей. Именем узла может быть любой символ Lisp. Создать базовый узел можно с помощью макрокоманды #. Такие узлы используются для обозначения неких сущностей: отношений «класс-экземпляр» (далее – концепт-экземпляр), объекта, класса, свойства и т.д. Подразумевается, что два узла не могут обозначать одну и ту же сущность. Стандартное имя узла: Вх, где х – некое число.

**Переменные** узлы также не имеют исходящих связей, обозначают произвольный концепт-экземпляр или высказывание (точно так же, как логические переменные). Макрокоманда для создания узла - \$. Стандартное имя узла: Vx, где x – некое число.

Атомарные (molecular) узлы и узлы-структуры могут иметь исходящие связи. Атомарными узлами обозначаются высказывания, а также правила. Атомарный узел, представляющий высказывание, может быть объявленным и необъявленным. Узлами-структурами принято обозначать произвольные высказывания или произвольно структурированные концептэкземпляры (аналогично логике предикатов).

#### Функции для объявления узлов:

- Структур и необъявленных атомарных узлов **build**
- Объявленных атомарных узлов assert
- ! объявить (команда постфиксная).
- Стандартное имя узла: Рх, где х некое число.
- Стандартное имя узла: Мх, где х некое число. (!Мх подтвержденный узел)

## Пояснения по семантике выражений SNePS

В первую очередь необходимо объявить связи (отношения), которые будут использованы далее. Можно объявить несколько связей сразу с помощью команды **define**, которая в качестве аргументов принимает имена отношений.

Например, зададим одновременно 2 отношения: student и tutor (define student tutor)

Чтобы удалить отношения, применяется команда undefine (undefine student tutor)

Assert используется для добавления высказывания. Так мы даем SNePS знать, что считаем данное высказывание истинным. assert принимает 2 аргумента: отношение и узел (или список узлов).

Например, выражение (assert student John) создает пропозициональный узел, который имеет связь student с узлом John.

Если пар отношение-узел несколько, то пропозициональный узел будет иметь по одной связи с каждым узлом.

Например, в результате выполнения (assert student John tutor Brown), созданный узел является объявленным. Идентификатор для узла создается автоматически. Восклицательный знак в имени узла показывает, что он объявлен.

## Лабораторная работа № 1. Добавление высказываний в базу знаний

Для успешной работы экспертной системы необходимо правильно заполнить базу знаний: внести в нее те высказывания, на основе которых будет приниматься решение.

В данной работе мы добавим в базу знаний информацию о трех животных (коте, рыбе, птице), их именах и способностях.

## Шаг 1. Запуск системы

Чтобы начать работать со SnePS необходимо установить дистрибутив, бесплатно предоставляемый разработчиками на сайте <u>http://www.cse.buffalo.edu/sneps/Downloads/.</u>

#### <u>Внимание! Для работы системы необходима установка Java!</u>

После конфигурации файлов (подробности можно узнать из файла readme), запускаем Sneps-2.7.0.bat.

В трэе (рядом с часами) появляется иконка SnePS, двойной щелчок по которой открывает консоль.

## Шаг 2. Начало работы

После того, как в начале строки появился символ «\*», можно вводить команды.

Команды записываются в круглых скобках и разделяются нажатием Enter.

Загрузка SNePS осуществляется командой (sneps)

Для очистки сети, удаления всех связей (отношения), которые были объявлены ранее, используется команда (resetnet t).

## Шаг 3. Определение типов отношений

Зададим все типы отношений, которые будут использованы в нашей базе знаний: (define member class object color)

В данном случае это: экземпляр класса, класс, объект, цвет.

## Шаг 4. Описание объектов

Теперь переходим к непосредственному описанию животных (после каждой строки нажимаем Enter):

(assert member Tom class cat)
(assert member Flownder class fish)
(assert member Tary class bird)

Тот является котом (или другими словами: Том – экземпляр класса кот), Flownder – рыбой, а Tary – птицей.

После ввода каждой команды (в данном случае – после ввода определенного отношения) SNePS указывает, в какой узел помещена введенная только что информация.

Так, после ввода (assert member Tom class cat) появится строчка (M1!)

Теперь М1 – это объявленный узел, который имеет исходящие связи **member** с узлом **Tom** и **class** с узлом **cat**.

#### Шаг 5. Проверка введенных данных

Проверить введенные данные можно, используя команду **dump:** (dump M1)

После того, как узел создан, к нему можно обращаться по имени, поэтому М1 используется без восклицательного знака.

#### Шаг 6. Добавление информации об экземплярах классов

Можно добавить, что кот, рыба и птица являются животными (т.е. в свою очередь являются экземплярами класса животные). После каждой строки нажимаем Enter:

```
(assert member cat class animal)
(assert member fish class animal)
(assert member bird class animal)
```

#### Шаг 7. Проверка обновленных данных

Проверим введенные нами данные с помощью команды **dump:** (dump animal)

SNePS выдает: (ANIMAL (CLASS- (M6! M5! M4!)))

(ANIMAL)

Т.е. экземпляра класса animal являются объявленные узлы M6!, M5! и M4!

Проверять вводимые данные можно как после их объявление (команда dump), так и непосредственно при вводе.

#### Шаг 8. Ввод описаний

Далее укажем, каков окрас животных. Для этого используется команда **describe**, аргументами которой является узел (узлы). Т.к. **assert**, в свою очередь, возвращает узел, то может служить аргументом для **describe**).

```
T.o., мы сразу просматриваем то, что вводим в базу знаний:
(describe(assert object Tom color black))
(describe(assert object Flownder color blue))
(describe(assert object Tary color yellow))
```

## Шаг 9. Просмотр семантической сети в виде графа

Однако существует более наглядный способ просмотреть созданную сеть. Команда **show** представляет введенные нами данные в виде графа, узлы которого можно перемещать для более удобного просмотра с помощью мыши:

(show)

Результат работы show представлен на рисунке 46.



Рисунок 46. Просмотр семантической сети

## Лабораторная работа № 2. Поиск в базе знаний

Как было продемонстрировано в Лабораторной работе №1, любое высказывание SnePS представляет в виде узла с исходящими связями (или связью).

Например, высказывание «Том – это кот» выглядит следующим образом:



Рисунок 47. высказывание «Том – это кот»

Из узла М1 исходит 2 связи, говорящих о том, что:

- 1) классом является «Кот»
- 2) членом класса является «Том».

Именно на этом принципе основывается поиск в базе знаний.

#### Шаг 1. Дополнение базы знаний

Дополним базу знаний, созданную в предыдущей лабораторной работе. Пусть в классе «Кот» появится еще один экземпляр, причем с белым окрасом:

(assert member Stu class cat)
(assert object Stu color yellow)

#### Шаг 2. Формирование простейшего вопроса

Спросим у SNePS, что подразумевается под словом «Животное». Иными словами, мы хотим найти все узлы, которые исходящими связями «Класс» соединены с узлом «Животное». Сделать это можно при помощи команды find:

(find class animal)

Если необходима более подробная информация, то следует воспользоваться комбинацией команд describe и find: (describe (find class cat))

Шаг 3. Детализация вопроса

Другой способ ответить на вопрос, что же такое «Животное» – это найти все экземпляры этого класса. В таком случае нам необходимы все узлы с входящей связью «Экземпляр», такие, чтобы у узла, из которого исходит эта связь, была бы еще исходящая связь «Класс» с узлом «Животное». Иначе говоря, этот промежуточный узел должен быть тем самым высказыванием, которое свидетельствует о связи экземпляр-класс.

Итак, нам нужно найти узлы с исходящей связью «Экземпляр». Это обозначается символом «-», т.е. в данном случае: member-.

Но этого недостаточно. Как было замечено выше, промежуточный узел должен иметь связь «Класс» с узлом «Животное». Дополняем: (memberclass).

Окончательно вволим: (find (member- class) animal)

В данном случае, аргументами find является пара связь-узел. Связью является отношение member- class (экземпляр класса), узлом – animal (тот класс, экземпляры которого необходимо найти).

Аналогично в базе знаний можно найти всех животных, окрашенных в желтый цвет (find (object- color) yellow)

Усложним задачу: осуществим поиск сразу по двум параметрам. Найдем всех котов черного цвета:

(find (member- class) cat (object- color) black)

## Лабораторная работа № 3. Построение узлов

В предыдущих лабораторных работах мы создавали узлы с помощью команды **assert**. При этом узел не только создавался, но и объявлялся: так мы давали SNePS понять, что есть некое высказывание, которое мы считаем правдой. В этом и состояло «объявление» узла. Бывают случаи, когда узел необходимо создать без объявления. Например, нам нужно добавить в базу знаний высказывание, которое считает правдой некоторое третье лицо. Наше отношение к этому высказыванию может быть таким же, противоположным или отсутствовать вообще.

Рассмотрим на примере поведение SnePS в такой ситуации

## Шаг 1. Подготовка к работе

Сначала создадим новую базу знаний. (resetnet t) (define member class ability agent act object) (assert member (Woody Willy) class bird)

Объявленным утверждением является тот факт, что Woody и Willy – птицы. Больше ничего об этом объекте SnePS неизвестно.

Если необходимо объявить сразу несколько экземпляров одного и того же класса, их достаточно перечислить, заключив в скобки (см. выше).

#### Шаг 2. Добавление предположений

Допустим, что некто Джон утверждает, что Woody умеет летать. Для того, чтобы сообщить об этом SNePS, используется конструкция **agent-act-object**:

Заметьте, что высказывание «Woody умеет летать» создано с помощью build, а не assert.

Таким образом, получается, что SnePS верит, что Джон верит в способность Woody летать, а не в саму способность Woody летать как таковую. В таких случаях говорят, что высказывание «Woody умеет летать» находится

в пространстве убеждений Джона, а в пространстве убеждений SnePS – «Джон верит, что Woody умеет летать.

## Шаг 3. Просмотр графа

Рассмотрим наглядно, как теперь выглядит база знаний: (show)



Рисунок 48. Граф семантической сети

Из рисунка 48 видно, что в отличии от узла, создаваемого с помощью **assert**, в данном случае у узла М2 появилась еще одна входящая связь, свидетельствующая о том, в чье пространство убеждений входит этот узел.

#### Шаг 4. Поиск введенных утверждений

Поскольку высказывание относительно способности Woody летать никак не оговаривалось для SNePS, выясним, какое утверждение было сохранено в базе знаний по поводу умения летать: (describe (find ability fly))

Аналогичный поиск можно осуществить командой **findassert**, с той лишь разницей, что поиск осуществляется только среди объявленных узлов: (describe (findassert ability fly))

Как видно, возвращенные значения отличаются.

## Шаг 5. Поиск аналогичных объектов

Найдем объекты, которые обладают способностью летать: (find (object- ability) fly)

A теперь найдем объекты, которые обладают способностью летать (по мнению SNePS): (find (object- ! ability) fly)

Сравним полученные результаты: в первом случае объект Woody был назван, т.к. узел такой действительно существует, во втором случае не найдено было ничего, т.к. утверждение «Woody может летать» не было объявлено (подтверждено) для SNePS, а только для Джона.

## Шаг 6. Доопределение сети

Определим, какие объекты, по мнению Джона способны летать.

Сначала выясним, какие узлы отвечают за знание SnePS о пространстве убеждений Джона (find agent John act believe)

Определим, какие объекты попадают в пространство убеждений Джона касательно способности к полету (или другими словами, какие узлы имеют входящие связи от узла убеждений Джона и исходящую связь к «fly») (find object- (find agent John act believe) ability fly)

Какие объекты, по мнению Джона, могут летать (с какими узлами у узла, полученного в предыдущем пункте, есть исходящие связи) (find object-(find object- (find agent John act believe)

```
ability fly))
```

#### Шаг 7. Пример универсального поиска

Возможен и более универсальный поиск, например, если узлы были описаны менее ясно. Применим следующую конструкцию: (find member ?x class bird)

Все найденные узлы сохраняются в переменной х. Префикс «\*» возвращает значения, находящиеся в х: \*х

## Лабораторная работа № 4. Работа с высказываниями

Кроме возможности всевозможного поиска в базе знаний, система SNePS способна строить рассуждения на основании заложенной информации.

## Шаг 1. Просмотр пространства убеждений

Воспользуемся базой знаний, созданной в предыдущей лабораторной работе.

Найдем все узлы, входящие в пространство убеждений SNePS: (describe \*nodes)

## Шаг 2. Запуск рассуждений (вывода)

Для получения информации об узлах сети применим рассуждение или вывод, запускаемый командой **deduce**. Она аналогична **assert**, **build** и **find**, в качестве аргументов принимает пару отношение-узел. Вместо имен узлов можно использовать переменные (но со знаком \$).

Пусть SnePS перечислит всех птиц, которые ему известны: (describe (deduce member \$x class bird)) Теперь, если применить команду **\*x**, узнаем содержимое этой переменной **\***x

## Шаг 3. Просмотр графа сети

Данные о классе «Птицы» записались в V1. Сеть приобрела вид, как на рисунке 4.

(show)



Рисунок 49. Граф с информацией о классе «Птицы», записанной в V1

## Шаг 4. Другие примеры вывода

Чтобы сделать вывод, является ли Woody птицей, введем: (describe (deduce member Woody class bird))

Аналогично определим, считает ли Джон, что Willy умеет летать: (describe (deduce agent John act believe object (build object Willy ability fly)))

Отсутствие ответа означает отрицание (в данном контексте – отсутствие данных у SNePS для ответа на вопрос).

## Лабораторная работа № 5. Работа с узлами

По умолчанию имена для узлов создаются SNePS по правилу xN, где x – тип узла, а N – порядковый номер. В базе знаний может содержаться не один десяток различных узлов, и запомнить их идентификаторы довольно сложно.

Поэтому, для простоты обращения к существующим узлам, существует возможность присваивания указателя в момент создания узла. Указателям можно присваивать имена, которые легче запомнить. Присваивание осуществляется инфиксным оператором «=».

## Шаг 1. Подготовка к работе

Очищаем сеть и все узлы и отношения, ранее созданные ранее. (resetnet t)

Объявляем типы связей для дальнейшего использования (define university faculty course name prerequisites)

Добавим в базу знаний информацию о группах студентов университета: (describe (assert university ITMO faculty KTU course 2 name «group 2157» ) = KTU2157)

Теперь переменная KTU2157 будет привязана к только что созданному узлу.

## Шаг 2. Доступ к узлу через переменную

Используя переменную, всегда можно получить доступ к самому узлу. Для этого применяется команда «\*». Например, укажем, что группа 3157 наследует информацию от предыдущей, 2157:

```
Найдем, от какого узла зависит KTU3157
(find prerequisites - *KTU3157)
```

```
A теперь – курсоры, зависящие от KTU2157
(find (name- prerequisites) *KTU2157)
```

## Шаг 3. Использование базовых узлов

Применять базовые узлы удобно, если мы хотим сообщить SNePS о некотором концепт-экземпляре, удовлетворяющем определенному описанию, но при этом не давать ему конкретного имени.

Базовые узлы не имеют исходящих связей, т.к. в них не содержится никакой информации о структуре сети.

Paccмотрим базовые узлы на примере. Определим типы отношений, которые будут присутствовать в нашей сети: (define agent act object member class)

Предположим, что нужно сообщить SNePS следующую информацию: собака укусила Джона. При этом не нужно указывать, какая именно собака (это неважно, или мы просто не знаем).

Выполним это действие в 2 этапа. Сначала скажем, что нечто является собакой. Для этого создаем константу Сколема (? Skolem constant), обозначающую собаку. Для этого используем символ #: (describe (assert member #dog class dog))

Действительно, если мы проверим командой **show**, по результатам последнего высказывания создался узел B1, который имеет только входящую связь (результат представлен на рисунке 50).



Рисунок 50. Пример базового узла

Теперь отметим, что собака укусила Джона. Для обращения к безымянному узлу используем оператор \*:

(describe (assert agent \*dog act bite object John)





В 2007 году СПбГУ ИТМО стал победителем конкурса инновационных образовательных программ вузов России на 2007–2008 годы. Реализация инновационной образовательной программы «Инновационная система подготовки специалистов нового поколения в области информационных и оптических технологий» позволит выйти на качественно новый уровень подготовки выпускников и удовлетворить возрастающий спрос на специалистов в информационной, оптической и других высокотехнологичных отраслях экономики.

## КАФЕДРА ПРОЕКТИРОВАНИЯ КОМПЬЮТЕРНЫХ СИСТЕМ

## ИСТОРИЯ КАФЕДРЫ

**1945-1966** РЛПУ (кафедра радиолокационных приборов и устройств). Решением Советского правительства в августе 1945 г. в ЛИТМО был открыт факультет электроприборостроения. Приказом по институту от 17 сентября 1945 г. на этом факультете была организо

вана кафедра радиолокационных приборов и устройств, которая стала готовить инженеров, специализирующихся в новых направлениях радиоэлектронной техники, таких как радиолокация, радиоуправление, теленаведение и др. Организатором и первым заведующим кафедрой был д.т.н., профессор С. И. Зилитинкевич (до 1951 г.). Выпускникам кафедры присваивалась квалификация инженер-радиомеханик, а с 1956 г. – радио-инженер (специальность 0705).

В разные годы кафедрой заведовали доцент Б.С. Мишин, доцент И.П. Захаров, доцент А.Н. Иванов.

**1966–1970 КиПРЭА** (кафедра конструирования и производства радиоэлектронной аппаратуры). Каждый учебный план специальности 0705 коренным образом отличался от предыдущих планов радиотехнической специальности своей четко выраженной конструкторско-технологической направленностью. Оканчивающим институт по этой специальности присваивалась квалификация инженер-конструктор-технолог РЭА.

Заведовал кафедрой доцент А.Н. Иванов.

**1970–1988 КиПЭВА** (кафедра конструирования и производства электронной вычислительной аппаратуры). Бурное развитие электронной вычислительной техники и внедрение ее во все отрасли народного хозяйства потребовали от отечественной радиоэлектронной промышленности решения новых ответственных задач. Кафедра стала готовить инженеров

по специальности 0648. Подготовка проводилась по двум направлениям – автоматизация конструирования ЭВА и технология микроэлектронных устройств ЭВА.

Заведовали кафедрой: д.т.н., проф. В.В. Новиков (до 1976 г.), затем проф. Г.А. Петухов.

**1988–1997 МАИ** (кафедра микроэлектроники и автоматизации проектирования). Кафедра выпускала инженеров-конструкторов-технологов по микроэлектронике и автоматизации проектирования вычислительных средств (специальность 2205). Выпускники этой кафедры имеют хорошую технологическую подготовку и успешно работают как в производстве полупроводниковых интегральных микросхем, так и при их проектировании, используя современные методы автоматизации проектирования. Инженеры специальности 2205 требуются микроэлектронной промышленности и предприятиям-разработчикам вычислительных систем.

Кафедрой с 1988 г. по 1992 г. руководил проф. С.А. Арустамов, затем снова проф. Г.А. Петухов.

С 1997 ПКС (кафедра проектирования компьютерных систем). Кафедра выпускает инженеров по специальности 210202 «Проектирование и технология электронно-вычислительных средств». Область профессиональной деятельности выпускников включает в себя проектирование, конструирование и технологию электронных средств, отвечающих целям их функционирования, требованиям надежности, проекта и условиям эксплуатации. Кроме того, кафедра готовит специалистов по защите информации, специальность 090104 «Комплексная защита объектов информатизации». Объектами профессиональной деятельности специ-алиста по защите информации являются методы, средства и системы обеспечения защиты информации на объектах информатизации.

С 1996 г. кафедрой заведует д.т.н., профессор Ю.А. Гатчин.

За время своего существования кафедра выпустила 4364 инженеров. На кафедре защищено 65 кандидатских и 7 докторских диссертаций.