

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Оренбургский государственный университет»

Кафедра программного обеспечения вычислительной техники  
и автоматизированных систем

Н. А. Соловьев, Е.Н. Чернопрудова

# **РАЗРАБОТКА СРЕДСТВ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ**

Рекомендовано к изданию Редакционно-издательским советом федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет» в качестве методических указаний для студентов, обучающихся по программам высшего образования по направлениям подготовки 09.03.04 Программная инженерия и 09.03.01 Информатика и вычислительная техника

Оренбург  
2015

УДК 004.4(076.5)  
ББК 32.973.26-018.2я7  
С 60

Рецензент – доктор технических наук, профессор А.М. Пищухин

С 60

**Соловьев, Н.А.**

Разработка средств поддержки принятия решений: методические указания к лабораторному практикуму / Н.А. Соловьев, Е.Н. Чернопрудова; Оренбургский гос. ун-т – Оренбург: ОГУ, 2015. – 57 с.

Методические рекомендации к лабораторному практикуму предназначены для студентов, обучающихся по программам бакалавриата направлений: 09.03.01 – Информатика и вычислительная техника, профиль – Программное обеспечение средств вычислительной техники и автоматизированных систем и 09.03.04 – Программная инженерия, профиль «Разработка программно-информационных систем» при изучении дисциплин «Основы теории принятия решений», «Системы поддержки принятия решений», «Исследование операций».

УДК 004.4(076.8)  
ББК 32.973-018.2я7

© Соловьев Н.А., Чернопрудова Е.Н., 2015  
© ОГУ, 2015

## Содержание

Введение.....	5
1 Концептуальные основы систем поддержки принятия решений.....	8
2 Лабораторная работа №1 Построение модели потоков данных предметной области принятия решений с использованием CASE – средств.....	11
2.1 Инструментальные средства BPwin для разработки модели потоков данных.....	11
2.2 Методика выполнения задания.....	17
Вопросы коллоквиума.....	22
3 Лабораторная работа № 2 Построение функциональной модели предметной области принятия решений с использованием CASE – средств.....	24
3.1 Инструментальные средства BPwin для разработки функциональной модели предметной области.....	24
3.2 Методика выполнения задания.....	26
Вопросы коллоквиума .....	30
4 Лабораторная работа № 3 Разработка средств СППР в условиях определенности.....	31
4.1 Основы принятия решений в условиях определенности.....	31
4.2 Методика выполнения задания.....	37
Вопросы коллоквиума .....	38
5 Лабораторная работа № 4 Разработка средств СППР в условиях стохастической неопределенности.....	40
5.1 Основы принятия решений в условиях стохастической неопределенности .....	40
5.2 Методика выполнения задания.....	47
Вопросы коллоквиума .....	47

6 Лабораторная работа № 5 Разработка средств СППР в статистически неопределенных ситуациях.....	48
6.1 Основы принятия решений в статистически неопределенных условиях.....	48
6.2 Методика выполнения задания.....	51
Вопросы коллоквиума .....	51
Заключение	53
Список использованных источников.....	54
Приложение А Варианты задания на лабораторные работы №3-5	55
Приложение Б Пример представления алгоритма средства принятия решений.....	56
Приложение В Пример составления списка использованных источников для системного анализа предметной области разработки	57

## Введение

Лабораторный практикум имеет целью выработку у студентов умений и навыков проектирования и разработки объектов профессиональной деятельности с использованием современных инструментальных средств программирования. Объектом профессиональной деятельности бакалавра по профилю «Программное обеспечение средств вычислительной техники и автоматизированных систем» является автоматизированная система, которая относится к классу MIS (Management Information Systems – информационно-управляющие системы). Основу MIS составляют программные средства системы поддержки принятия решений (СППР), предназначенные для сбора, хранения и обработки информации в целях управления.

Принятие решения в большинстве случаев заключается в разработке модели объекта управления, генерации возможных альтернативных решений, их оценке и выборе лучшего варианта.

В сложных и ответственных условиях лицо, принимающее решение (ЛПР), обращается к опытным специалистам (экспертам) за подтверждением решения. Такие обращения определяют процесс поддержки принятия решения.

Замена эксперта программным средством является основной задачей средства (системы) поддержки принятия решений.

Неопределенность является неотъемлемой частью процессов принятия решений. При выборе варианта приходится учитывать большое число неопределенных и противоречивых факторов.

Лабораторный практикум представляет собой перечень лабораторных работ, имеющих целью выработки у студентов умений и навыков проектирования и разработки СППР с использованием современных инструментальных средств.

Обобщенная методика выполнения лабораторных работ включает:

– краткое описание теории, раскрывающей понятия и важнейшие особенности принятия решений в конкретной предметной области при

заданных условиях, на основе которых определяются методы решения задач;

- реализация программного средства на базе современной методологии программирования с учетом специфики методов (методик, алгоритмов) поддержки принятия решений применительно к автоматизированной системе.

Все лабораторные работы выполняются в соответствии с вариантом индивидуального задания, которое определяет преподаватель из перечня Приложения А (в отдельных случаях вариант предлагает студент по предметной области будущей выпускной квалификационной работы).

Для выполнения всех лабораторных работ предлагается единый порядок, предусматривающий следующие этапы:

- уяснение цели работы и формулировка инженерных задач для её достижения;
- ознакомление с инструментарием, используемым для описания модели (диаграммы), и разработка алгоритма программного средства;
- разработка и реализация программного средства (модели);
- тестирование разработанного средства (компонента автоматизированной системы – аналитического приложения);
- сохранение результата;
- оформление отчета по лабораторной работе.

Отчет по лабораторной работе оформляется в электронной форме и состоит из следующих разделов:

- 1) тема лабораторной работы;
- 2) цель работы;
- 3) индивидуальное задание;
- 4) описание модели (алгоритма), программы и оценки результатов решения задачи (тестирование);
- 5) выводы по работе.

Программное средство является компонентом автоматизированной системы, который реализует один из методов теории принятия решений.

Лабораторная работа является видом учебного занятия, способствующим

закреплению теоретических знаний, формированию у студентов умений и навыков решения задач по дисциплине, развитию творческого мышления и инициативы. Лабораторная работа проводится в компьютерном классе под руководством преподавателя.

# 1 Концептуальные основы систем поддержки принятия решений

Принятие решения в большинстве случаев заключается в генерации возможных альтернативных решений, их оценке и выборе лучшего варианта. В сложных и ответственных моментах лицо, принимающее решение, (ЛПР) обращается к опытным и знающим людям (экспертам) за подтверждением своего решения. Такие обращения представляют собой процесс поддержки принятия решения.

Основными функциями эксперта являются:

- оказание помощи ЛПР при анализе исходной информации (оценке сложившейся обстановки и ограничений, накладываемых внешней средой);
- выявление и ранжирование приоритетов, учет неопределенности в оценках ЛПР и формирование его предпочтений;
- генерация возможных решений (формирование списка альтернатив);
- оценка возможных альтернатив, исходя из предпочтений ЛПР, и ограничений, накладываемых внешней средой;
- анализ возможных последствий принимаемых решений;
- выбор лучшего, с точки зрения ЛПР, варианта.

При выборе варианта эксперту приходится учитывать большое число неопределенных и противоречивых факторов.

Неопределенность является неотъемлемой частью процессов принятия решений и включает:

- неопределенность, связанная с неполнотой знаний о проблеме, по которой должно быть принято решение;
- неопределенность, связанная с невозможностью полного учета реакции окружающей среды на принимаемые решения;
- неопределенность, связанная с нечетким определением ЛПР своих целей.

Противоречивость возникает из-за неоднозначности оценки ситуаций,



ошибки в выборе приоритетов, что, в конечном итоге, осложняет процесс принятия решений.

Исследования показывают, что ЛПР без дополнительной аналитической поддержки, как правило, используют упрощенные, а иногда и противоречивые правила выбора решения.

Формализация методов анализа и генерации решений, их оценка и согласование являются достаточно сложными задачами. Их решение становятся возможными на основе средств вычислительной техники и во многом зависят от возможностей программных средств, реализующих методы и способы интеллектуальной поддержки принимаемых решений.

Процесс принятия решений может протекать по двум основным схемам: интуитивно-эмпирической (основанной на сравнении проблемной ситуации с ранее встречавшимися, схожими ситуациями) и формально-эвристической (основанной на построении и исследовании модели проблемной ситуации). Независимо от схемы процесса принятия решений информационное обеспечение управления является одним из решающих факторов принятия эффективных решений. Обычно под информационным обеспечением управления понимают совокупность информационных ресурсов, средств, методов и технологий, способствующих эффективному проведению всего процесса управления, в том числе разработке и реализации управленческих решений.

При построении модели проблемной ситуации исследуют структуру процессов принятия решений, которая определяется такими элементами, как состояние исходных данных задачи, модель ситуации принятия решения, ограничения, варианты решений и их последствия, внешние факторы объективного и субъективного характера. Совокупность перечисленных элементов образует определенную среду (систему) принятия решений. Такая система называется системой поддержки принятия решений (СППР). Другими словами, СППР – система, обеспечивающая лицо, принимающее решение, необходимыми для принятия решения данными, знаниями, выводами и/или

рекомендациями.

Ориентация на современные информационные технологии позволяет выделить новый класс СППР — информационно-аналитические системы поддержки принятия решений (ИА СППР). ИА СППР — это класс человеко-машинных систем, предназначенных для оказания помощи ЛПР в их профессиональной деятельности по использованию данных, знаний и моделей при подготовке и принятии обоснованных решений.

Особенности автоматизированных СППР наиболее ярко проявляются в рамках следующих классификационных признаков: концептуальные модели, решаемые задачи, области применения.

Рассматривая существующие концептуальные модели СППР, выделяют информационный, интеллектуальный и инструментальный подходы.

В рамках информационного подхода СППР относят к классу автоматизированных информационных систем, основное назначение которых — «улучшить деятельность работников умственного труда (knowledge workers) в организациях путем применения информационной технологии». Главными компонентами этой модели являются: интерфейс «пользователь – система», база данных и база моделей.

В рамках интеллектуального подхода СППР, основанная на знаниях, призвана не оказать помощь ЛПР в решении стоящей перед ним проблемы, а заменить человека при решении проблемы.

При инструментальном подходе, в зависимости от специфики решаемых задач и используемых технологических средств, выделяют три уровня систем: прикладные, генераторы и инструментальные.

Прикладные СППР служат для поддержки решения отдельных прикладных задач в конкретных ситуациях.

Генераторы представляют собой пакеты программных средств поиска и выдачи данных, моделирования и т. д., которые используются разработчиками прикладных СППР.

Инструментальные СППР соответствуют высшему уровню

технологичности, которые предоставляют в распоряжение разработчиков мощные комплексы средств для создания специализированных систем.

## **2 Лабораторная работа №1. Разработка модели потоков данных предметной области принятия решений с использованием CASE – средств**

Учебные вопросы:

2.1 Инструментальные средства VPwin для разработки модели потоков данных

2.2 Методика выполнения задания.

### **2.1 Инструментальные средства VPwin для разработки модели потоков данных**

VPwin – CASE-средство разработки программного обеспечения автоматизированных систем верхнего уровня, поддерживающее методологии IDEF0 (функциональная модель), IDEF3 (WorkFlow Diagram) и DFD (DataFlow Diagram).

В VPwin для построения диаграмм потоков данных используется нотации Гейна-Сарсона. Диаграммы потоков данных (Data flow diagramming, DFD) описывают документооборот и обработку информации. DFD можно использовать как дополнение к модели IDEF0 для более наглядного отображения текущих операций документооборота в корпоративных системах обработки информации. Диаграммы, построенные по методологии DFD, описывают функции обработки информации (работы), документы, объекты, сотрудников или отделы, которые участвуют в обработке информации, и база данных для хранения информации. Для данной методологии, в отличие от IDEF0, для стрелок нет понятия вход, выход, управление или механизм. Стрелки в нотации DFD показывают, как объекты перемещаются от одной функции к другой.

В соответствии с методологией DFD диаграмм применяют компоненты в

нотации Йордона - Де Марко, условные обозначения которых представлены в таблице 2.1.

Таблица 2.1 – Нотации компонентов DFD диаграммы

Условное обозначение	Пояснение
Внешняя сущность	Внешняя сущность обозначает материальный объект или физическое лицо, представляющее собой источник или приемник информации, например, оператор, поставщики, клиенты, склад. Внешняя сущность находится за пределами границ анализируемой автоматизированной системы (АС), но может быть перенесена внутрь анализируемой системы (часть процессов АС может быть представлена как внешняя сущность).
Функции (работы)	Обозначает функции системы (процессы), преобразующие входы в выходы. При построении модели сложной АС процессы представляются в общем виде на контекстной диаграмме как единое целое. Имя процесса должно быть в виде предложения или словосочетания с активным недвусмысленным глаголом в неопределенной форме (рассчитать, определить, вычислить и т.д.), за которым следуют существительные в винительном падеже, например «вывести информацию о пациенте».
База (хранилище) данных	Хранилище данных представляет собой абстрактное устройство для хранения информации, которую можно в любой момент поместить в хранилище данных и извлечь.
Стрелки (потоки данных)	Стрелки описывают движение объектов из одной части системы в другую. Возможно применение двунаправленных стрелок для описания диалогов типа «запрос-ответ» между отображаемыми объектами.

При отображении на диаграмме бизнес-процессов необходимо учитывать, что данная диаграмма показывает потоки материальных и информационных потоков и не показывает временную последовательность работ, но в некоторых случаях временная последовательность совпадает с направлением движения потоков в бизнес-процессе.

При интерпретации DFD-диаграммы используются следующие правила:

- для каждого информационного потока и хранилища определяются связанные с ними элементы данных;
- каждому элементу данных присваивается имя с указанием типа данных и формата (эта информация является исходной при построении модели «сущность-связь»).

Требования к оформлению функций:

- каждая функция должна иметь идентификатор;
- названия работы необходимо формулировать согласно следующему формуле:

*Название работы = Действие + Объект, над которым действие осуществляется*

Например, если необходимо показать работу, связанную с действием по продаже продукции, то ее можно назвать «Продажа продукции». Название должно быть кратким (2-3 слова). Для более детального описания работы необходимо сделать её подробное описание в глоссарии.

Требования к оформлению потока данных:

- название потока необходимо формулировать согласно следующей формуле:

*Название потока = Объект, представляющий поток + Статус объекта*

Если, например, речь идет о продукции, которую отгрузили клиенту, то поток можно назвать <Продукция, отгруженная> или <Продукция, отгруженная клиенту>. В данном случае <Продукция> это объект, представляющий поток, а <отгруженная клиенту> — статус объекта.

- название должно быть по возможности кратким (2-3 слова).

При необходимости детализации контекстной диаграммы должны выполняться следующие правила:

- *правило балансировки* — при детализации процесса дочерняя диаграмма в качестве внешних источников/приемников данных может иметь только те компоненты (подсистемы, процессы, внешние сущности, накопители данных), с которыми имеет информационную связь соответствующий процесс на родительской диаграмме;

- *правило нумерации* — при детализации процессов должна поддерживаться их иерархическая нумерация.

- *правило семи* — для того, чтобы диаграмма легко читалась, количество функций на диаграмме не должно быть больше семи.

Для подробной детализации и однозначного понимания функций и потоков данных используют миниспецификацию (документ, детально описывающий логику процесса), которая должна формулировать его основные функции таким образом, чтобы в дальнейшем специалист, выполняющий реализацию проекта, смог выполнить их или разработать соответствующую программу.

Миниспецификация содержит номер процесса, списки входных и выходных данных, тело процесса – подробный алгоритм функции, преобразующий входные потоки данных в выходные. Решение о завершении детализации процесса и использовании миниспецификации принимается аналитиком исходя из следующих критериев:

- наличие у процесса небольшого количества входных и выходных потоков данных (2-3 потока);

- возможность описания процесса в виде последовательного алгоритма;

- процесс выполняет единственную логическую функцию преобразования входной информации в выходную;

- описать логику процесса можно в виде миниспецификации небольшого объема (не более 20-30 строк).

После построения модели информационных потоков системы её

необходимо верифицировать, т.е. проверить на полноту и согласованность. Модель считается полной, если все ее объекты (подсистемы, процессы, потоки данных) подробно описаны и детализированы. Модель считается согласованной, если для всех потоков данных и накопителей данных выполняется правило сохранения информации: все поступающие куда-либо данные должны быть считаны, а все считываемые данные должны быть записаны.

Анализ информационных процессов (документов) предприятия является основой для построения модели потоков данных и функциональной модели предметной области.

В качестве примера рассматривается постановка задачи на разработку автоматизированной информационной системы определения состояния здоровья пациентов клинической больницы.

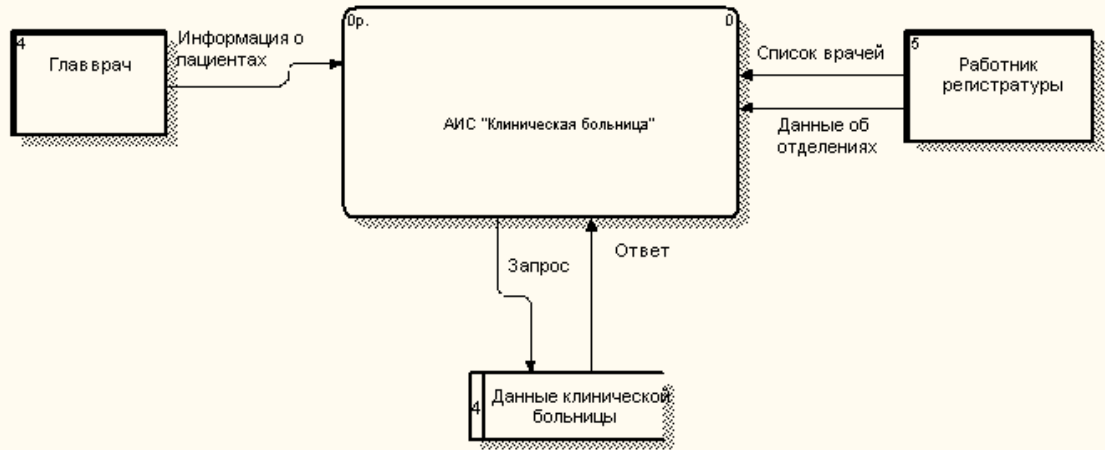
*Постановка задачи.* Клиническая больница. На каждого вновь поступившего больного заводится карточка медицинской статистики: Ф.И.О. больного, пол, возраст, предварительный диагноз, как поступил больной (направление поликлиники, доставлен скорой помощью и т.п.), дата поступления, прочее описание: примерный рост, цвет волос, особые приметы, примерный возраст, номер палаты, в которую положен больной. Информация о больном может быть неполной (пациент не может ответить на вопросы). За время лечения в больнице больной может быть переведен в разные палаты, необходимо знать дату перевода, номер и телефон палаты. После окончания лечения фиксируется дата выписки и причина выписки, либо другой исход (полное излечение, направлен в санаторий и т.п.).

Модель потоков данных автоматизированной системы «Клиническая больница» представлена контекстной диаграммой по методологии DFD на рисунке 1.1 и детализацией диаграммы второго уровня на рисунке 2.2.

На диаграмме отображены объекты (компоненты) системы управления и потоки данных, отражающие обобщенную совокупность обрабатываемых документов (информационные процессы).



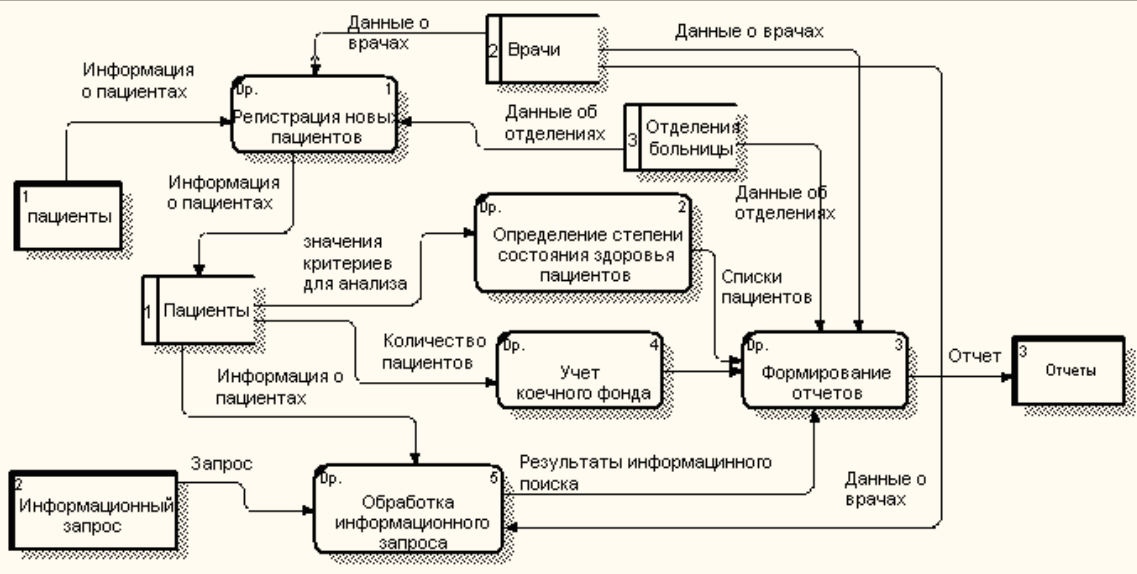
USED AT:	AUTHOR: Иванов С.Б. PROJECT: АИС "Клиническая больница"	DATE: 22.12.2014 REV: 22.12.2014	WORKING DRAFT RECOMMENDED PUBLICATION	READER	DATE	CONTEXT: TOP
NOTES: 1 2 3 4 5 6 7 8 9 10						



ODE: A-0	TITLE: АИС "Клиническая больница"	NUMBER:
----------	-----------------------------------	---------

Рисунок 2.1 – Контекстная диаграмма потоков данных методологии DFD

USED AT:	AUTHOR: Иванов С.Б. PROJECT: АИС "Клиническая больница"	DATE: 22.12.2014 REV: 22.12.2014	WORKING DRAFT RECOMMENDED PUBLICATION	READER	DATE	CONTEXT: A-0
NOTES: 1 2 3 4 5 6 7 8 9 10						



NODE: A 1	TITLE:	NUMBER:
-----------	--------	---------

Рисунок 2.2 – Детализация диаграммы второго уровня потоков данных

## 2.2 Методика выполнения задания

**Цель работы:** Построение модели потоков данных автоматизированной системы предметной области по методологии DFD Case-средством BPwin.

### Последовательность выполнения:

При создании новой модели необходимо воспользоваться пунктом меню File/New, в результате чего будет выведено диалоговое окно (рисунок 2.3), в котором программа предлагает ввести наименование модели, нотацию для отображения контекстной диаграммы, а также предлагает открыть существующую модель с диска или с системы управления моделями для групповой разработки ModelMart. После ввода имени и выбора нотации автоматически создается контекстная диаграмма изображающей систему в целом.

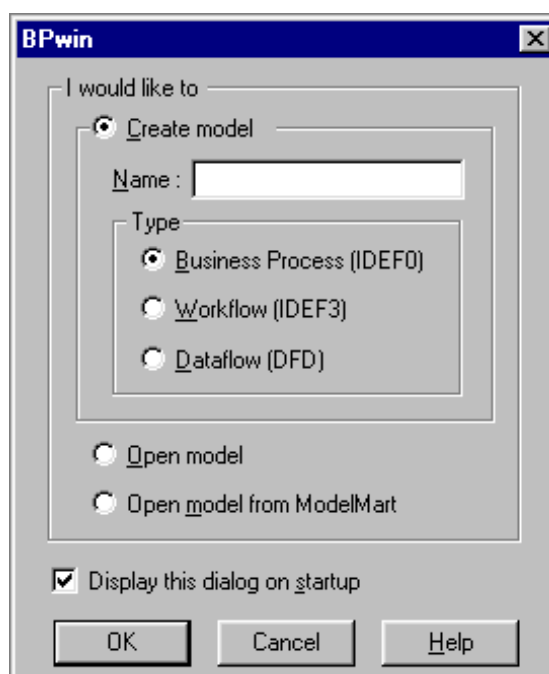


Рисунок 2.3 –Окно создания модели

Для описания других аспектов контекстной диаграммы, а также свойств модели служит диалог Model Properties Editor, который может быть вызван с

помощью пункта меню Edit / ModelProperties (рисунок 2.4).

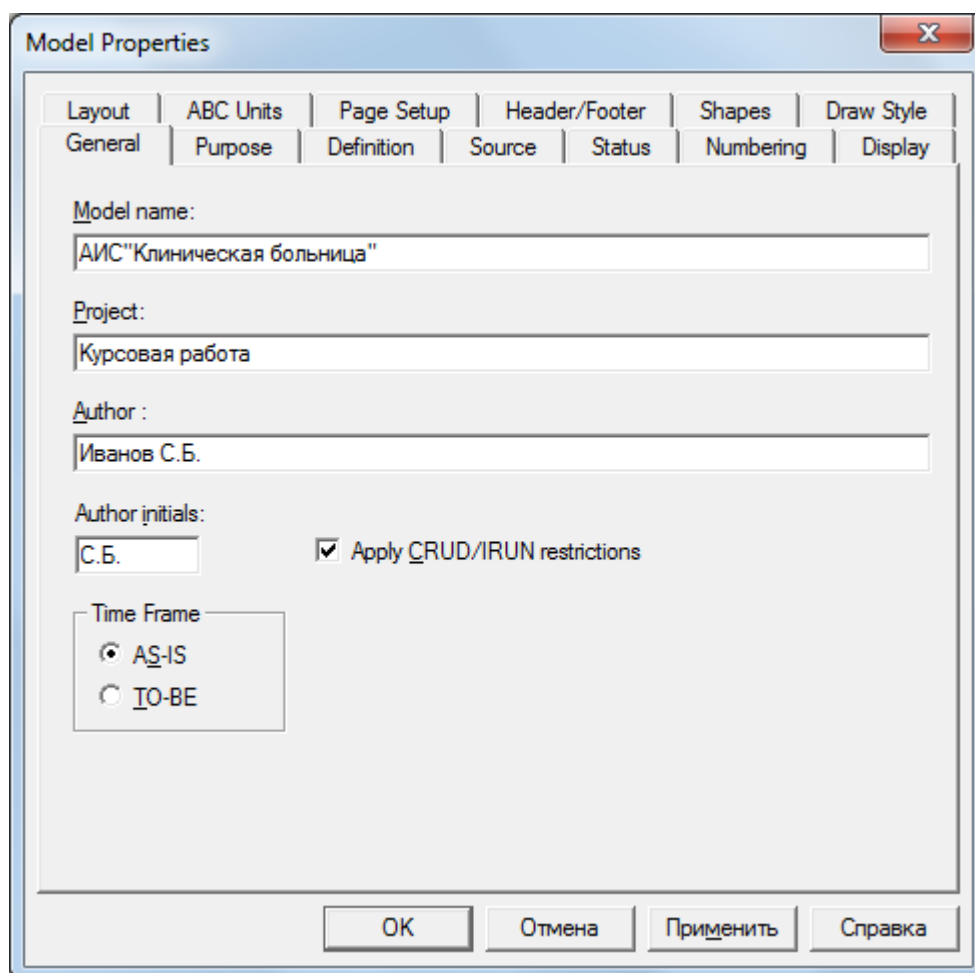


Рисунок 2.4 – Окно диалога *Model Properties Editor*

В данном диалоговом окне представлены несколько закладок: General, Purpose, Definition, Source, Status, Presentation, Display, Layout, ABC Units.

Закладка **General** предназначена для определения основных свойств модели. В ней представлены поля для ввода информации о наименовании модели, наименовании проекта, в который входит модель, имени и инициалов автора модели, а также тип модели: AS IS показывает систему в ее текущем состоянии или TO BE, которая показывает как система будет работать в будущем.

Закладка **Purpose** предназначена для ввода информации о назначении модели. На ней представлены два поля ввода информации: **Purpose** – для ввода информации о цели модели, объяснение того, почему система находится под

анализом, что показывает модель, и что пользователь может делать с моделью;  
**Viewpoint** – для ввода информации об источнике информации о системе.

Закладка **Definition** содержит два поля ввода **Definition** и **Scope**, которые предназначены соответственно для описания модели и ввода информации об уровне подробности и детальности модели. Закладка **Source** предназначена для ввода информации об источнике данных для построения модели, где указывается имя, должность или документ, в котором содержится информация для описания модели.

Стрелки на контекстной диаграмме служат для описания взаимодействия системы с окружающим миром. Они могут начинаться у границы диаграммы и заканчиваться у работы, или наоборот. Такие стрелки называются граничными. Для внесения граничной стрелки входа на контекстной диаграмме необходимо:

- 1) нажать на символе стрелки в палитре инструментов;
- 2) перенести курсор к левой стороне экрана, пока не появится начальная штриховая полоска;
- 3) нажать один раз на полоску (откуда выходит стрелка) и еще раз на левую часть работы со стороны входа (где заканчивается стрелка);
- 4) вернуться в палитру инструментов и выбрать опцию редактирования стрелки;
- 5) дважды нажать на линию стрелки, во всплывающем меню выбрать **Name Editor** и добавить имя стрелки.

Имена вновь внесенных стрелок автоматически заносятся в словарь (**Arrow Dictionary**). Словарь стрелок редактируется при помощи специального редактора **Arrow Dictionary Editor** (рисунок 2.5)., в котором определяется стрелка и вносится относящийся к ней комментарий. Словарь стрелок можно распечатать в виде отчета (меню **Report / Arrow Report...**) и получить тем самым толковый словарь терминов предметной области, использующихся в модели.

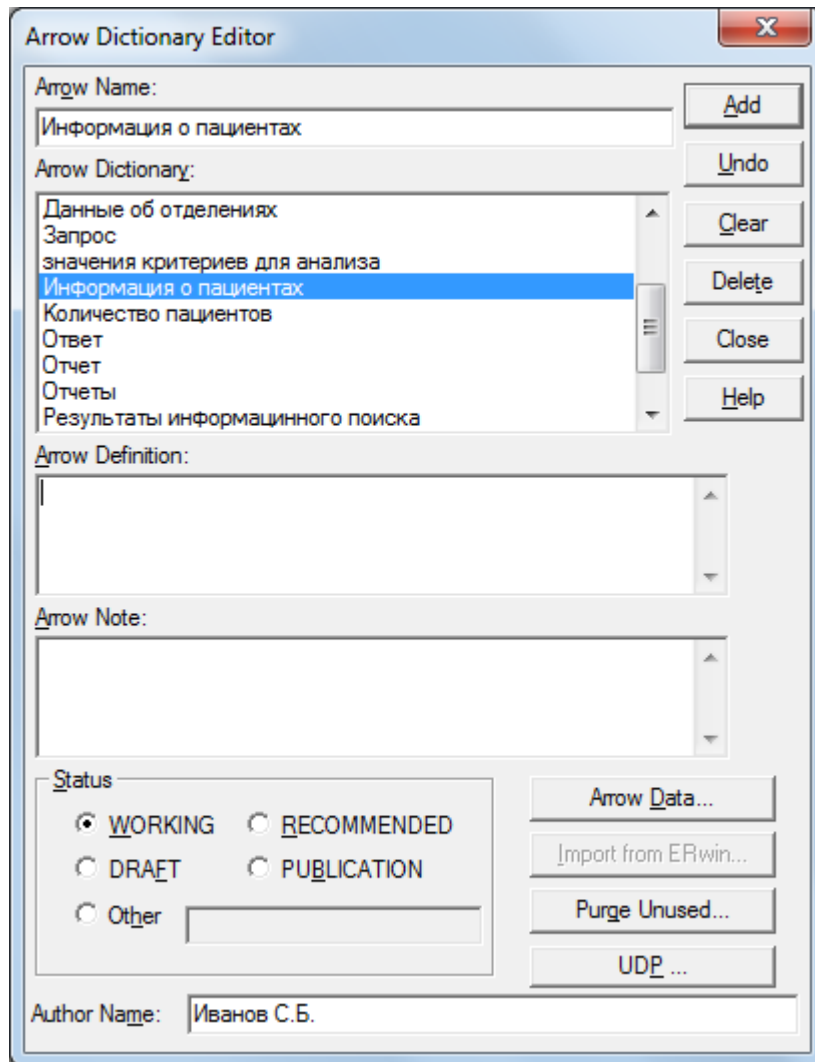



Рисунок 2.5 – Диалог *Arrow Dictionary Editor*

Для декомпозиции контекстной диаграммы необходимо выбрать декомпозируемый процесс и нажать кнопку перехода на нижний уровень , после чего в появившемся диалоговом окне *Activity Box Count* (рисунок 2.6) необходимо указать количество работ на диаграмме декомпозиции (в дальнейшем можно будет добавить недостающие работы или удалить лишние) и нотацию диаграммы.

При декомпозиции диаграммы DFD работы расположены в порядке доминирования (по степени важности или в порядке очередности выполнения).

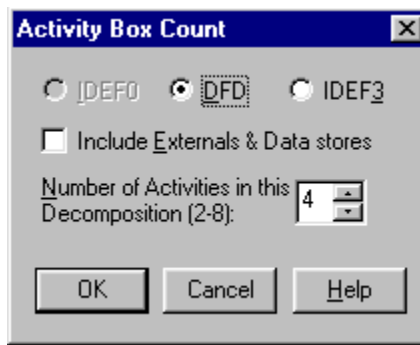


Рисунок 2.6 – Окно диалога *Activity Box Count*

Стрелки, которые были внесены на контекстной диаграмме, показываются и на диаграмме декомпозиции (миграция стрелок), но при этом не касаются работ. Такие стрелки называются несвязанными и воспринимаются, как синтаксическая ошибка. Для связывания стрелки необходимо перейти в режим редактирования стрелок, выбрать стрелку и соответствующий сегмент работы.

Для связи работ между собой используются внутренние стрелки, т.е. стрелки, которые не касаются границы диаграммы, начинаются у одной и кончаются у другой работы.

Вновь внесенные граничные стрелки на диаграмме декомпозиции нижнего уровня изображаются в квадратных скобках и автоматически не появляются на диаграмме верхнего уровня. Чтобы переместить стрелку на уровень выше необходимо сначала выбрать кнопку на палитре инструментов и нажать на квадратные скобки граничной стрелки. В появившемся окне *Border Arrow Editor* (рисунок 2.7.) выбрать либо кнопку *Resolve Border Arrow* (стрелка мигрирует на диаграмму верхнего уровня), либо кнопку *Change To Tunnel* (стрелка будет затуннелирована и не попадет на другую диаграмму).

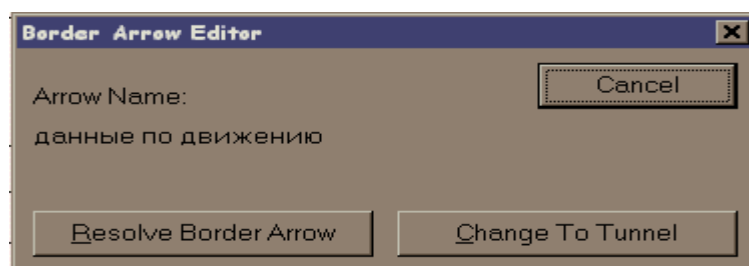


Рисунок 2.7 – Диалог *Border Arrow Editor*

Одна и та же информация может обрабатываться в нескольких работах, в то же время из нескольких работ могут выходить одинаковые данные, то есть стрелки могут разветвляться и сливаться. Для разветвления стрелки нужно в режиме редактирования стрелки нажать на фрагмент стрелки и на соответствующий сегмент работы.

Case – средство VPwin для формирования проектной документации предлагает редактор отчетов, который можно вызвать с помощью пункта меню Report VPwin, позволяет выбрать несколько категорий отчетов: Model Report (отчет о модели), Diagram Report (отчет по диаграмме), Diagram Object Report (отчет по элементам диаграмм), Activity Cost Report (позволяет создать отчет о стоимости и длительности процессов и прежде всего предназначен для реинжиниринга системы), Arrow Report (позволяет создать словарь данных), Data Usage Report (отчет по использованию данных, описанных в словаре сущностей и атрибутов), Model Consistency Report (позволяет проверить систему на соответствие правилам нотации и выдает список ошибок в модели).

Таким образом, VPwin является мощным средством для создания и анализа модели информационных потоков системы, которое включает в себя средства построения графической документации (иерархия диаграмм) в различных методологиях, а также средства создания проектной документации.

Отчет по лабораторной работе оформляется в электронной форме и содержит следующие разделы:

- 1 Тема лабораторной работы.
- 2 Цель работы.
- 3 Индивидуальное задание (предметная область).
- 4 Описание модели (алгоритм).
- 5 Выводы по работе.

### **Вопросы коллоквиума**

- 1) Что описывает диаграмма DFD?

- 2) Какая нотация используется в BPWin для построения диаграммы DFD?
- 3) Какие операции над данными могут совершать работы в модели процессов?
- 4) Как выполнить переход от одной модели к другой?
- 5) Что означает проверить модель на полноту и согласованность?
- 6) Какие нотации включает DFD диаграмма?
- 7) Что такое «внешняя сущность»?
- 8) Что определяет «хранилище данных»?
- 9) Описание миниспецификаций.
- 10) Какие правила используются при интерпретации DFD-диаграммы?

### **3 Лабораторная работа № 2. Разработка функциональной модели предметной области принятия решений с использованием Case – средств**

Учебные вопросы:

- 3.1 Инструментальные средства BPwin для разработки функциональной модели предметной области принятия решений.
- 3.2 Методика выполнения задания.

#### **3.1 Инструментальные средства BPwin для разработки функциональной модели предметной области принятия решений**

Основу методологии IDEF0 составляет графический язык описания бизнес - процессов. Модель в нотации IDEF0 представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм. Вершина этой древовидной структуры, представляющая собой самое общее описание системы и ее взаимодействия с внешней средой, называется контекстной диаграммой.



После описания системы в целом проводится разбиение ее на крупные фрагменты. Этот процесс называется функциональной декомпозицией, а диаграммы, которые описывают каждый фрагмент и взаимодействие фрагментов, называются диаграммами декомпозиции. После декомпозиции контекстной диаграммы проводится декомпозиция каждого большого фрагмента системы на более мелкие и так далее до достижения нужного уровня подробности описания. Синтаксис описания системы в целом и каждого ее фрагмента одинаков во всей модели. Работы (Activity), которые означают некие поименованные процессы, функции или задачи, изображаются в виде прямоугольников. Именем работы должен быть глагол или глагольная форма. Взаимодействие работ с внешним миром и между собой описывается в виде стрелок. Стрелки представляют собой некую информацию и именуется существительными.

В IDEF0 различают пять типов стрелок:

**Вход (Input)** - материал или информация, которая используется или преобразовывается работой.

**Управление (Control)** - правила, стратегии, процедуры или стандарты, которыми руководствуется работа. Каждая работа должна иметь хотя бы одну стрелку управления.

**Выход (Output)** - материал или информация, которая производится работой. Каждая работа должна иметь хотя бы одну стрелку выхода.

**Механизм (Mechanism)** - ресурсы, которые выполняют работу, например, персонал, ЭВМ, механизмы и т.д.

**Вызов** - специальная стрелка, указывающая на другую модель работы.

Каждый тип стрелок подходит или выходит к определенной стороне прямоугольника, изображающего работу. К левой стороне подходят стрелки входов, к верхней - стрелки управления, к нижней - механизмов реализации выполняемой функции, а из правой - выходят стрелки выходов (рисунок 3.1)

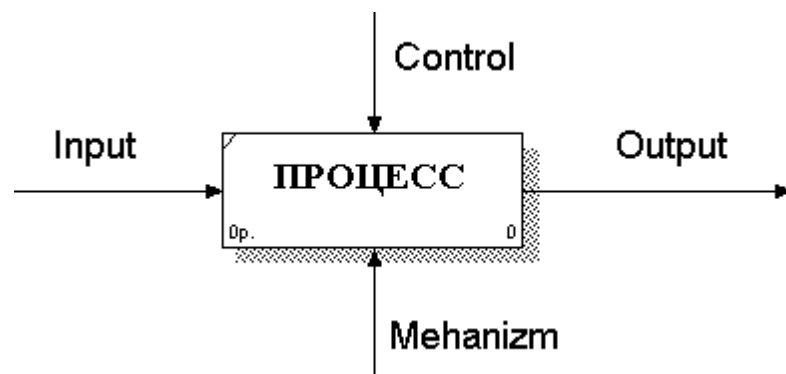


Рисунок 3.1 - Пример диаграммы в нотации IDEF0

Такое соглашение предполагает, что используя управляющую информацию и реализующий ее механизм, функция преобразует свои входы в соответствующие выходы.

### 3.2 Методика выполнения задания

**Цель работы:** Построение функциональной модели предметной области по методологии SADT Case-средством BPwin.

#### Последовательность выполнения:

Для создания контекстной диаграммы (функциональной модели) по методологии IDEF0 необходимо в диалоговом окне «*I would like to*» ввести имя модели «СППР Клиническая больница» и выбрать методологию IDEF0.

После того как откроется диалоговое окно *Properties for New Models* (свойства новой модели) необходимо ввести имя автора модели (поле *Autor*) и его инициалы (поле *Autor initials*). После чего автоматически создастся контекстная диаграмма, представленная на рисунке 3.2

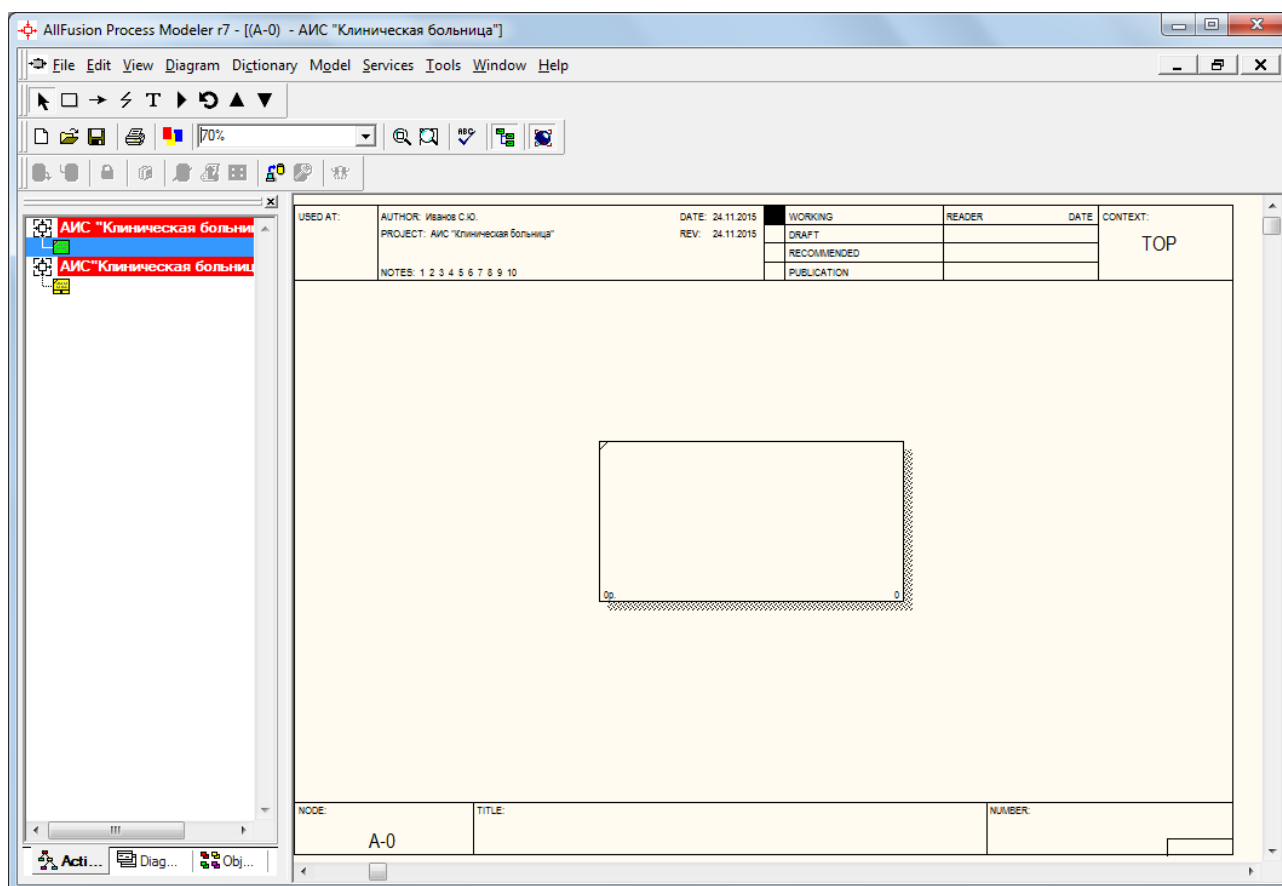


Рисунок 3.2 – Незаполненная контекстная диаграмма

Для документирования модели необходимо воспользоваться диалоговым окном *Model Properties*. На вкладке *General* (рисунок 3.3) в поле *Model name* (имя модели) следует ввести имя модели «Автоматизация информационного процесса определения степени состояния здоровья пациентов клинической больницы». В поле *Project* (имя проекта) ввести имя проекта «АИС «Клиническая больница»» и установить тип диаграммы в поле *Time Frame* (временной охват) AS – IS или TO – BE. В данном случае временной охват устанавливаем TO – BE.

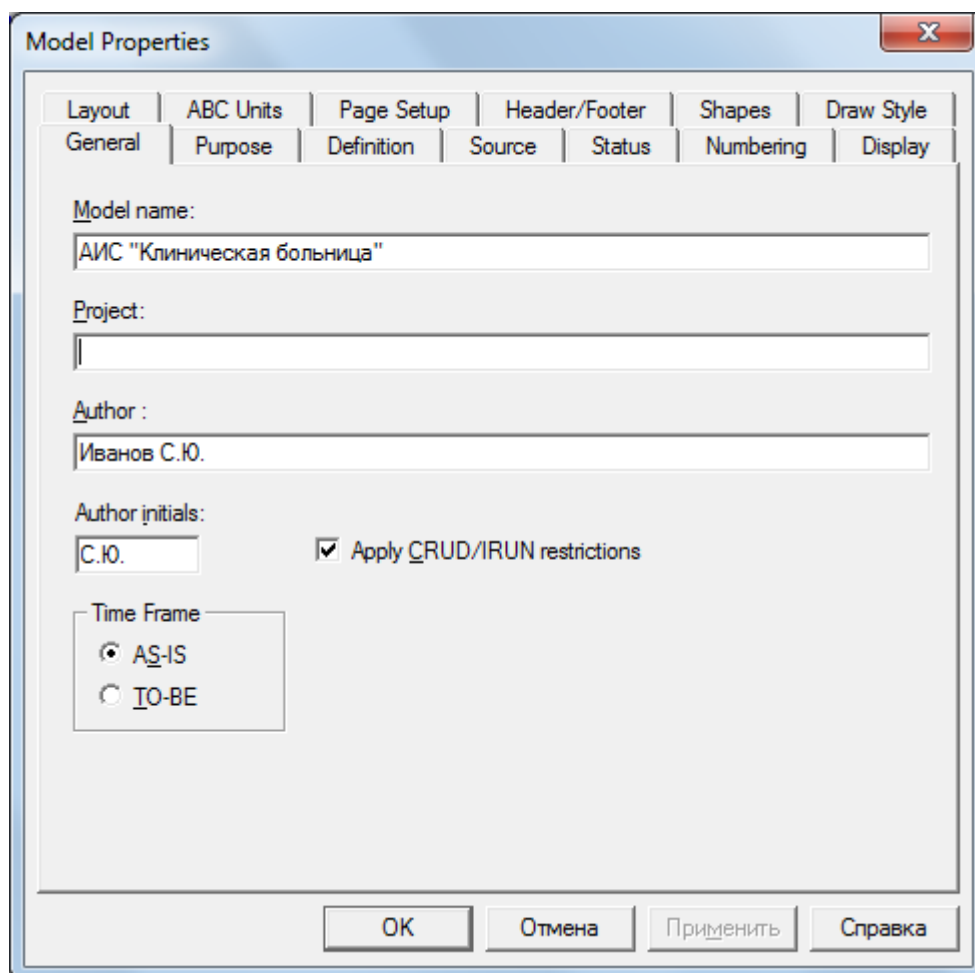


Рисунок 3.3 – Окно задания свойств модели

На вкладке **Purpose** (рисунок 3.4) в поле *Purpose* (цель) необходимо ввести данные о цели разработки модели – «Моделировать процесс автоматизации определения степени состояния здоровья пациентов клинической больницы» и в поле *Viewpoint* (точка зрения) указать с точки зрения кого представлена данная модель. Разрабатываемая модель, для данного примера, представлена с точки зрения главного врача.

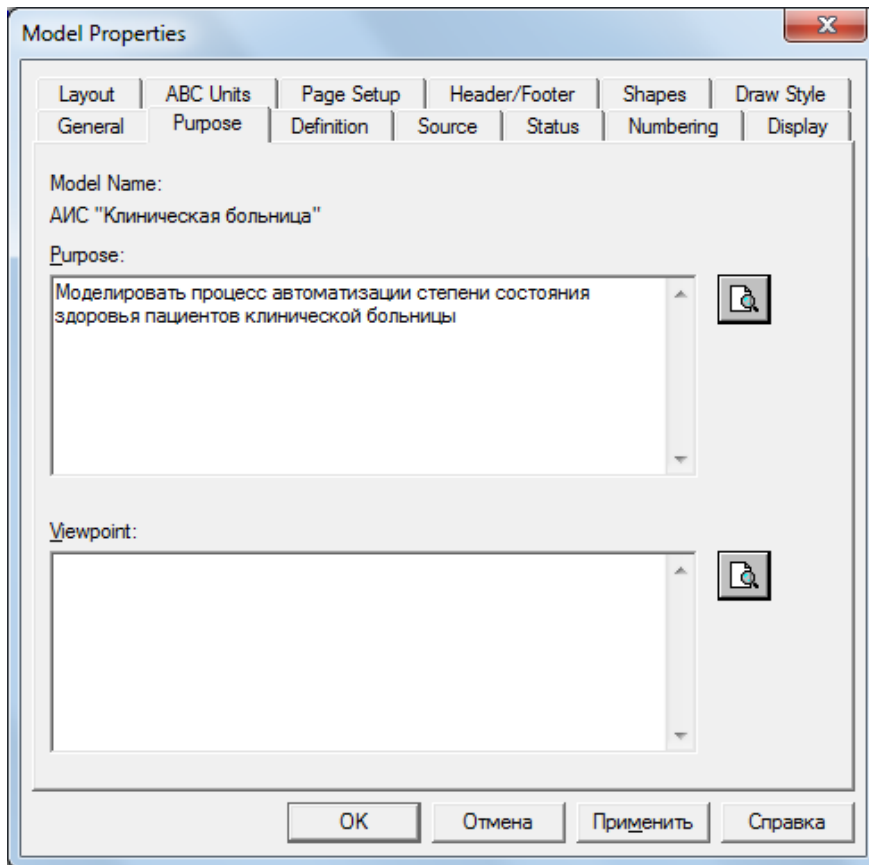


Рисунок 3.4 – Диалоговое окно для внесения данных о модели

Функциональная модель процесса автоматизации, построенная по методологии IDEF0, представлена на рисунке 3.5.

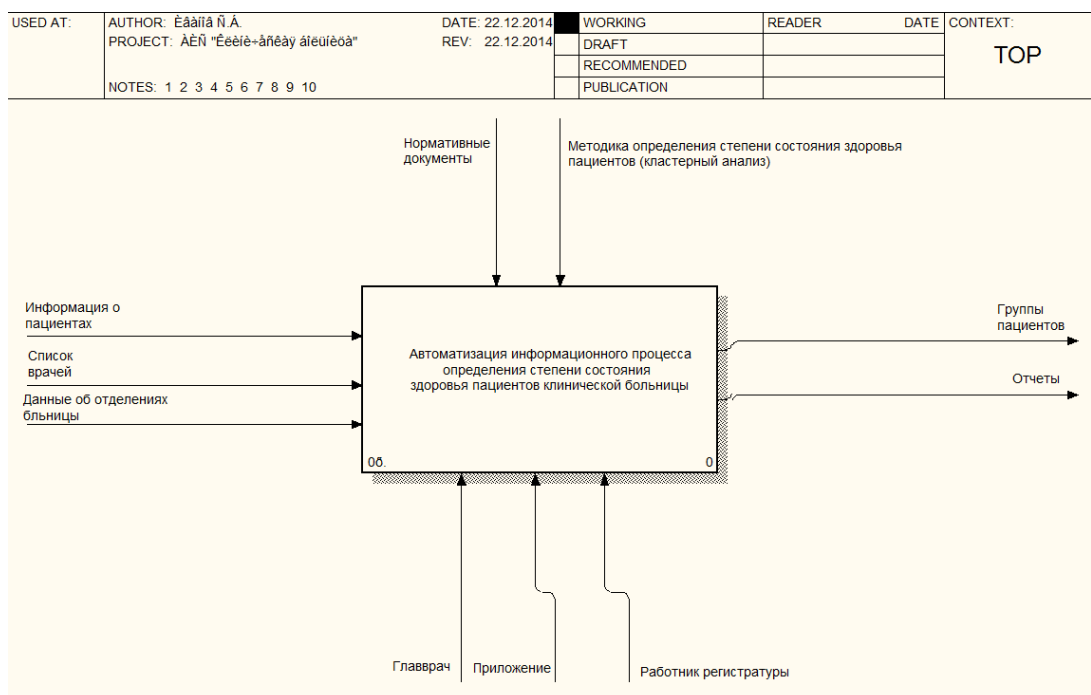


Рисунок 3.5 – Контекстная диаграмма функциональной модели

Процесс автоматизации можно представить как совокупность подпроцессов на основе диаграммы декомпозиции (рисунок 3.6).

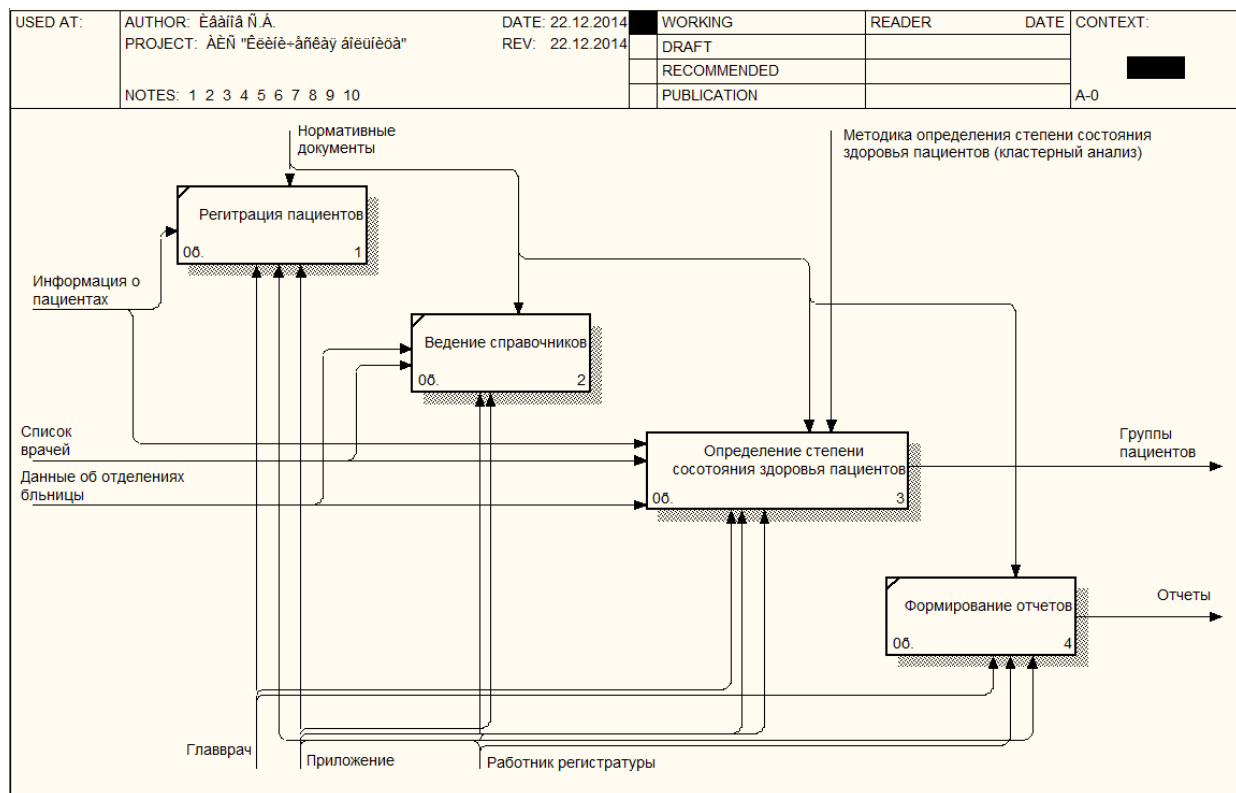


Рисунок 3.6 – Диаграмма декомпозиции функциональной модели

Декомпозиция процесса «Определение степени состояния здоровья пациентов» методом кластерного анализа представлена на рисунке 3.7.

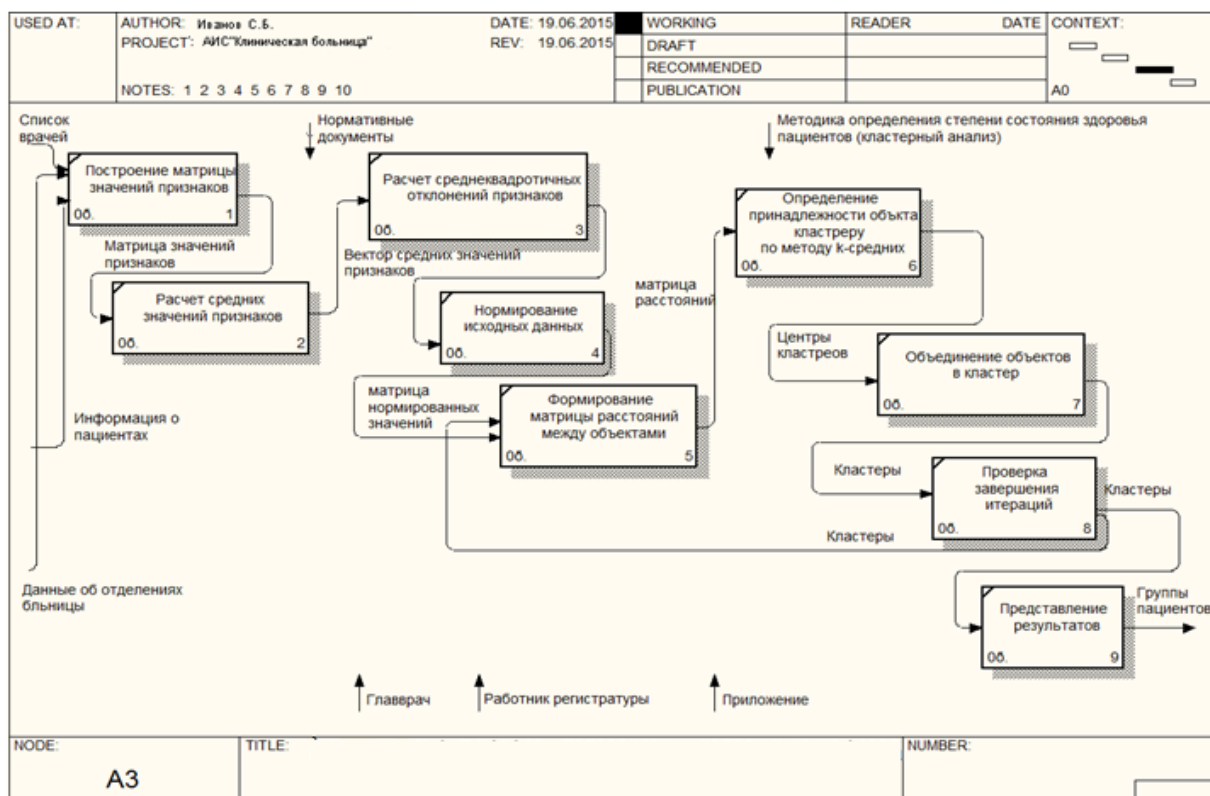


Рисунок 3.7 – Декомпозиция процесса метода кластерного анализа

Отчет по лабораторной работе оформляется в электронной форме и состоит:

- 1 Тема лабораторной работы.
- 2 Цель работы.
- 3 Индивидуальное задание (предметная область).
- 4 Описание модели (алгоритм).
- 5 Выводы по работе.

### Вопросы коллоквиума

- 1) Какие методологии поддерживает Case-средство BPWin?
- 2) Что представляет собой модель в нотации IDEF0?
- 3) Для чего предназначена контекстная диаграмма?
- 4) Перечислите последовательность построения диаграмм в нотации IDEF0?
- 5) Какие существуют правила именования работ?

- 6) В чем заключается процесс декомпозиции работ?
- 7) Каким образом можно создать новую модель в BPWin?
- 8) Какое назначение сторон прямоугольников работ на диаграммах?
- 9) Какие операции над данными могут совершать работы в модели процессов?
- 10) Какая нотация используется в BPWin для построения диаграммы DFD?

#### **4 Лабораторная работа № 3. Разработка средства поддержки принятия решений в условиях определенности**

Учебные вопросы:

- 4.1 Основы теории оценивания систем в условиях определенности
- 4.2 Методика выполнения задания

##### **4.1 Основы теории оценивания систем в условиях определенности**

Оценивание систем в условиях определенности проводится на основе матрицы частных показателей (таблица 4.1) с использованием методов векторной оптимизации.

Таблица 4.1 – Матрица частных показателей

$a_i$	$n_j$				$K(a_i)$
	$n_1$	$n_2$	...	$n_k$	
$a_1$	$k_{11}$	$k_{21}$		$k_{1k}$	
$a_2$	$k_{21}$	$k_{22}$		$k_{2k}$	
...	...	...		...	
$a_m$	$k_{m1}$	$k_{m2}$		$k_{mk}$	



Пусть  $K = \{k_1, k_2, \dots, k_l\}$  – векторный критерий, представляющий собой отображение

$$K : A \rightarrow R^i$$

где  $K(a)$  – векторная оценка альтернативы  $a \in A$ ,

$R^i$  – шкала действительных чисел.

Отсюда общая задача векторной оптимизации формулируется следующим образом

$$\begin{aligned} K(a) \rightarrow \text{opt } K(a_i) \\ a \in A \end{aligned} \quad (4.1)$$

где  $\text{opt}$  – оператор, определяющий семантику оптимальности.

Решение задачи (4.1) осуществляется в 3 этапа:

1 На основе системного анализа предметной области определяются частные показатели состояния объекта управления и формируется критерий оценивания.

2 Формулируется задача многокритериальной оптимизации в форме (4.1).

3 Путем скаляризации критерия устраняется многокритериальность.

Устранение многокритериальности в условиях определенности осуществляется на основе следующих методов:

1 Выделения главного критерия.

2 Лексикографическая оптимизация.

3 Последовательных уступок.

4 Свертки векторного критерия в скалярный.

5 Агрегирование.

В методе выделения главного критерия лицо, принимающее решение (ЛПР) назначает один критерий главным, а остальные выводятся в состав

ограничений (указываются границы, в которых эти критерии могут изменяться).

Выражение (4.1) примет вид:

$$\begin{aligned} K(a_i) &\rightarrow \text{opt } K(a_i), \\ a_i &\in A \end{aligned} \tag{4.2}$$

$$k_j \leq k_j^{\text{don}}, j \neq i, j = \overline{1, n}.$$

Если условия ограничения для альтернативы  $a_i$  не выполняются, решение принято быть не может.

В методе лексикографической оптимизации предполагается, что частные показатели  $k_j$ , составляющие векторный критерий  $\mathbf{K}$ , могут быть упорядочены на основе отношения абсолютной предпочтительности. При этом частные показатели  $k_j$  перенумеруются так, что наиболее важному из них соответствует номер 1. Тогда на первом шаге выбирается множество альтернатив  $a_i \subseteq A$ , имеющих наилучшие оценки по первому частному показателю  $k_1$ . Если  $a_i$  единично, то решение принято. Если  $a_i > 1$ , то на втором шаге выбирается множество  $a_i \subseteq A_i$ , имеющее наилучшие оценки по  $k_2$  и так далее, пока не будет выявлена лучшая альтернатива. При поиске решения задачи (4.1) в описанной процедуре, как правило, будут использоваться не все, а лишь наиболее важные критерии, что не всегда может быть оправдано. Кроме того, остальные частные показатели переводятся в ограничения. Если условия ограничения не выполняются, решение принято быть не может.

Поэтому в методе последовательных уступок для каждого из проранжированных по важности частного показателя назначается допустимое отклонение значения  $k_i$  от наилучшего. Затем на первом шаге производится построение подмножества альтернатив  $A_1 \subseteq A$ , для которых отклонение оценки по первому критерию от экстремального значения не превышает допустимого отклонения – «уступки». Далее строится подмножество  $A_2 \subseteq A_1$  на основе

второго критерия и его уступки и т.д. Однако, даже при допущении некоторого процентного интервала величин частных показателей возможна ситуация, при которой не одна из альтернатив решением признана быть не может.

**Методы свертывания векторного критерия в скалярный.** В этих методах задача (3.1) заменяется задачей

$$\begin{aligned} K(a) \rightarrow \text{extr}, \\ a \in A \end{aligned} \quad (4.3)$$

где  $K(a)$  – скалярный критерий, представляющий собой некоторую функцию от значений компонентов векторного критерия  $K(a) = f(k_1(a), k_2(a), \dots, k_i(a))$ .

Функция  $f$  называется сверткой.

Методика получения функции свертки распадается на 4 задачи:

- 1 Обоснование допустимости свертки.
- 2 Нормализация критериев для их сопоставления.
- 3 Учет приоритетов (важности) критериев.
- 4 Построение функции свертки, позволяющей решить задачу выбора.

**Обоснование допустимости свертки.** Требуется подтверждения, что рассматриваемые частные показатели являются однородными. Известно, что показатели эффективности разделяются на 3 группы: показатели результативности, ресурсоемкости и оперативности. Разрешается свертка показателей, входящих в обобщенный показатель для каждой группы отдельно. Нарушение этого принципа ведет к потере физического смысла критерия.

**Нормализация критериев.** Проводится для исключения влияния различной размерности частных показателей на результат решения.

**Учет приоритетов.** Осуществляется путем задания вектора коэффициентов важности критериев  $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_I)$ , с учетом  $\sum_{i=1}^I \lambda_i = 1$ .

В результате нормализации и учета приоритетов частных показателей вместо исходной векторной оценки  $K(a)$  альтернативы  $a$  образуется новая векторная оценка  $K(a) = (\lambda_1 \cdot k_1(a), \lambda_2 \cdot k_2(a), \dots, \lambda_j \cdot k_j(a))$ . Именно эта полученная оценка подлежит преобразованию с использованием функции свертки.

**Построение функции свертки.** Известны несколько способов свертки нескольких частных показателей в критерий, использование которых зависит от характера частных показателей и целей оценивания. Наиболее часто используются аддитивная и мультипликативная свертки.

Аддитивная свертка компонентов векторного критерия состоит в представлении обобщенного скалярного критерия в виде суммы взвешенных нормированных частных показателей

$$K(a) = \sum_{i=1}^I \lambda_i \frac{k_i(a)}{k_i^0} \quad (4.4)$$

Свертка (3.4) основана на использовании принципа справедливой компенсации абсолютных значений нормированных частных показателей. Суть принципа: справедливым следует считать компромисс, при котором суммарный уровень абсолютного снижения значения одного или нескольких показателей не превышает суммарного уровня абсолютного увеличения значений других показателей.

Недостатком данной методики является то, что низкие оценки по одним критериям могут компенсироваться высокими оценками по другим критериям.

Мультипликативная свертка компонентов векторного критерия состоит в представлении обобщенного скалярного критерия в виде произведения

$$K(a) = \prod_{i=1}^I k_i(a)^{\lambda_i} \quad (4.5)$$

В этом критерии схема компромисса предполагает оперирование не

абсолютными, а относительными изменениями частных критериев.

Правомочность мультипликативного критерия основывается на принципе справедливой относительной компенсации: справедливым следует считать такой компромисс, при котором суммарный уровень относительного снижения значения одного или нескольких критериев не превышает суммарного уровня относительного увеличения значений других критериев. В математической форме такое условие имеет вид:

$$\sum_{i=1}^I \frac{\Delta k_i(a)}{k_i(a)} = 0$$

где  $\Delta k_i(a)$  – приращение величины  $i$ -го частного показателя;

$k_i(a)$  – первоначальная величина  $i$ -го частного показателя.

Достоинством мультипликативного критерия является то, что при его использовании не требуется нормировка частных показателей.

Недостатки критерия: критерий компенсирует недостающую величину одного частного критерия избыточной величиной другого и имеет тенденцию сглаживать уровни частных показателей за счет их неравнозначных первоначальных значений.

Выбор между аддитивной и мультипликативной свертками частных показателей определяется степенью важности абсолютных или относительных изменений значений частных показателей соответственно.

Кроме свертки векторного критерия в теории векторной оптимизации особое место занимает **агрегирование**. Если из сущности задачи следует полная недопустимость компенсации значений одних показателей другими, т.е. требуется обеспечить равномерное подтягивание всех показателей к наилучшему уровню, то используют функцию агрегирования вида:

$$K(a) = \frac{\prod_{i=1}^m k_i(a)}{\prod_{i=m+1}^I k_i(a)} \quad (4.6)$$

где  $i = \overline{1, m}$  – номер показателя, значения которого желательно увеличить,

$i = \overline{m+1, I}$  – номер показателя, значения которого желательно уменьшить.

Часто первая группа показателей отождествляется с целевым эффектом, а вторая – с затратами на его достижение.

Рассмотренные группы методов представляют широкие возможности для анализа многокритериальных оценок в целях выявления наилучших альтернатив. Однако условия применимости тех или иных методов вследствие эвристического характера последних не могут быть сформулированы строго.

## 4.2 Методика выполнения задания

**Цель работы:** разработать программное средство принятия решений в условиях определенности.

### Индивидуальное задание

Требуется оценить результат разработки автоматизированной системы с учетом известных аналогов (3 альтернативы  $a_i$ ) на основе 4-х частных показателей  $k_j$ , представленных в таблице 4.2.

Таблица 4.2 - Матрица частных показателей автоматизированной системы

A	$k_j$			
	$k_1$	$k_2$	$k_3$	$k_4$
$a_1$	$k_{11}$	$k_{12}$	$k_{13}$	$k_{14}$
$a_2$	$k_{21}$	$k_{22}$	$k_{23}$	$k_{24}$
$a_3$	$k_{31}$	$k_{32}$	$k_{33}$	$k_{34}$

Частные показатели:

$k_1$  – время решения задачи (процессорное время);

$k_2$  – время разработки средства (написание кода);

$k_3$  – цена проекта (инструмента программирования, аналога);

$k_4$  – объем требуемой памяти (оперативной, дисковой).

Альтернативы:

$a_1$  – разработанная система;

$a_1$  – аналог 1;

$a_1$  – аналог 2.

Метод решения задачи выбирается исходя из варианта (см. Приложение А)

Порядок выполнения задания:

1. Разработать алгоритм решения задачи.
2. Написать программу в высокоуровневой среде.
3. Протестировать программу.

В качестве примера выполнения задания рассматривается следующая постановка задачи: научной лаборатории необходимо оценить воздействие трех медицинских препаратов при одном из 4-х заболеваний. Матрица эффективности представлена в таблице, где  $a_i$  - медицинские препараты, а  $n_j$  - заболевания.

Таблица 4.3 – Матрица эффективности

$a_i$	$n_j$				$K(a_i)$
	$n_1$	$n_2$	...	$n_k$	
$a_1$	$k_{11}$	$k_{21}$		$k_{1k}$	
$a_2$	$k_{21}$	$k_{22}$		$k_{2k}$	
...	...	...		...	
$a_m$	$k_{m1}$	$k_{m2}$		$k_{mn}$	

## **Вопросы коллоквиума**

- 1) На основе чего производится оценивание систем в условиях определенности?
  - 2) В чем заключается сущность методов векторной оптимизации?
  - 3) Какие методы решения задач векторной оптимизации Вы знаете?
  - 4) На основе каких методов осуществляется устранение многокритериальности в условиях неопределенности?
  - 5) В какой аксиоме предпочтение исхода  $a_i$  исходу  $a_j$  не зависит от порядка, в котором они представлены.
  - 6) В каком случае для метода выделения главного критерия решение принято быть не может?
  - 7) В чем заключается метод лексикографической оптимизации?
  - 8) Метод последовательных уступок.
  - 9) На какие задачи распадается методика получения функции свертки?
- Поясните каждую из задач.
- 10) Мультипликативная свертка.

## **5 Лабораторная работа № 4. Разработка средств СПР в условиях стохастической неопределенности**

Учебные вопросы:

- 5.1 Основы теории оценивания систем в условиях стохастической неопределенности.
- 5.2 Методика выполнения задания.

### **5.1 Основы теории оценивания систем в условиях стохастической неопределенности**

**Оценивание систем в условиях риска.** Принятие решений в условиях



риска предполагает, что каждой альтернативе  $a_i$ ,  $i = \overline{1, m}$  соответствует свое распределение вероятностей на множестве исходов  $y_j$ ,  $j = \overline{1, n}$ . Если множества альтернатив и исходов конечны, то считаются известными вероятности всех исходов, возможных при выборе данной альтернативы.

Ситуации выбора альтернатив можно представить в виде матрицы (таблицы 4.1), называемой матрицей решений.

Матрица решений интерпретируется следующим образом – решение  $a_i$  может реализовать различные исходы из соответствующей строки матрицы:  $y_{i1}, y_{i2}, \dots, y_{in}$ . Какой именно исход реализуется, зависит от значения параметра неопределенности  $z$ , который может иметь различный содержательный смысл.

Тогда задачу принятия решений в условиях риска можно представить в форме функции реализации вида

$$y = F(a, z). \quad (5.1)$$

Таблица 5.1 – Матрица решений

A	Z				
	$z_1$		$z_j$		$z_n$
$a_1$	$y_{11}$		$y_{1j}$		$y_{1n}$
$a_i$	$y_{i1}$		$y_{ij}$		$y_{in}$
$a_m$	$y_{mj}$		$y_{mj}$		$y_{mn}$

Представление задачи принятия решений в условиях риска в форме функции реализации означает, что статистическую неопределенность, проявляющуюся в неоднозначной (вероятностной) связи между средством и результатом, можно интерпретировать как существование некоторой внешней среды, оказывающей влияние на результат.

Каждому состоянию среды  $z_j$  соответствует вероятность его наступления  $p(z_j)$

$$p(z_j) = \prod_{i=1}^n p_i(y_j(a_i)), \quad (5.2)$$

где  $p_i(y_j(a_i))$  - вероятность наступления исхода  $y_j$  при выборе альтернативы  $a_i$ .

Тогда существует функционал  $K: Y \times Z \rightarrow R$  и задача принятия решений сводится задаче оптимизации

$$K(a, z) \rightarrow \underset{a \in A}{opt}. \quad (5.3)$$

Для задачи принятия решений при многократно реализуемым исходам используется математическое ожидание критерия (4.3) вида

$$K(a) = \overline{K(a, z)} = \underset{a \in A}{opt}. \quad (5.4)$$

**Оценивание систем на основе функции полезности.** В теории полезности исходят из того, что критерий предназначен для выявления предпочтений на альтернативах (исходах операций), что позволяет обеспечить обоснованный выбор решения. При этом полезность исхода операции – это действительное число, приписываемое исходу операции, которое характеризует его предпочтительность по сравнению с другими альтернативами относительно цели.

Отличие данного подхода состоит в том, что свертывание векторного критерия в скалярный производится на основе аксиоматизации предпочтений ЛПР. Естественные отношения порядка на шкальных значениях критериев здесь не используются, т.к. все компоненты векторного критерия на основе предпочтений ЛПР преобразуются в функции полезности компонентов и лишь затем осуществляется свертывание.

Зная возможные альтернативы с их показателями полезности, можно построить функцию полезности, которая дает основу для сравнения и выбора

решений. Функция полезности представляет собой числовую функцию  $F(a)$ , определенную на множестве альтернатив  $A = \{a_k\}$ ,  $k = \overline{1, l}$ , так, что  $F(a_i) = F(a_j)$ , когда альтернативы  $a_i$  и  $a_j$  неразличимы ( $a_i \sim a_j$ );  $F(a_i) > F(a_j)$ , когда альтернатива  $a_i$  предпочтительнее  $a_j$  ( $a_i \succ a_j$ ).

Примером построения  $F(a)$  является функция, представленная на рисунке:

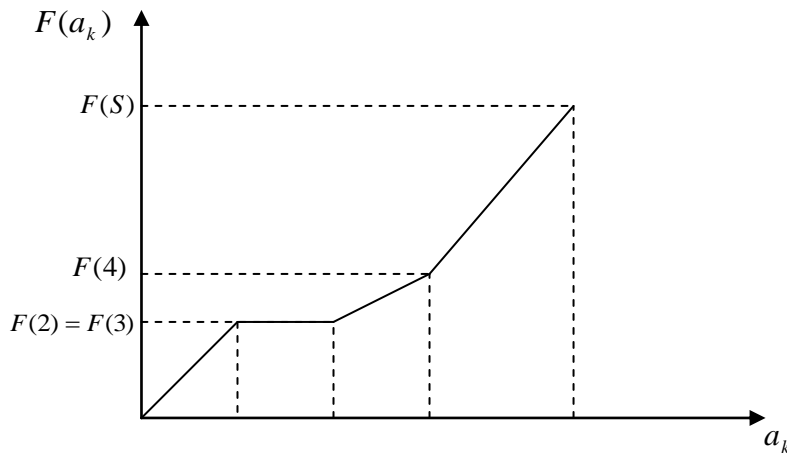


Рисунок 5.1 – Пример построения числовой функции  $F(a)$

Задача принятия решений в вероятностных операциях определяется через математическое ожидание  $M_a$  функции полезности на множестве исходов

$$K(a) = M_a[F(y)], \quad (5.5)$$

где  $F(y)$  – функция полезности на множестве исходов  $y$  по альтернативам  $a$ .

При исходах  $y_k$  ( $k = \overline{1, m}$ ) с дискретными значениями показателей, каждый из которых появляется с условной вероятностью  $p(y_k / a_i)$  и имеет полезность  $F(y_k)$ , выражение для определения математического ожидания функции полезности записывается в виде

$$K(a_i) = 1/m \sum_{k=1}^m p(y_k / a_i) \cdot F(y_k), \quad i = \overline{1, n}. \quad (5.6)$$

Условия оценки альтернатив в случае, когда показатели исхода операций являются дискретными величинами, задаются таблично.

Таблица 5.2 – Условия оценки альтернатив

$a_i$	$y_k$	$p(y_k / a_i)$	$F(y_k)$	$K(a_i)$
$a_1$	$y_1$	$p(y_1 / a_1)$	$F(y_1)$	$K(a_1)$
	$y_2$	$p(y_2 / a_1)$	$F(y_2)$	
	$y_m$	$p(y_m / a_1)$	$F(y_m)$	
$a_2$	$y_1$	$p(y_1 / a_2)$	$F(y_1)$	$K(a_2)$
	$y_2$	$p(y_2 / a_2)$	$F(y_2)$	
	$y_m$	$p(y_m / a_2)$	$F(y_m)$	

Критерий оптимальности для вероятностных операций имеет вид

$$K(a_i) = \max_{a_i} M_{a_i} [F(y)] = \max K(a_i), i = \overline{1, m}. \quad (5.4)$$

В соответствии с этим критерием оптимальной системой в условиях стохастической неопределенности считается система с максимальным значением математического ожидания функции полезности на множестве исходов операции.

Сведение задачи оценки систем в вероятностной постановке применимо для операций, имеющих массовый характер, для которых имеется вероятность определить объективные показатели исходов, вероятностные характеристики по параметрам обстановки и законы распределения вероятностей на множестве исходов операции.

В теории полезности доказывается существование функции полезности, для которой предпочтения ЛПР формулируются в виде аксиом.

Основными аксиомами теории полезности является:

Аксиома 1 – измеримость (каждому альтернативному исходу  $y_k$  ( $k = \overline{1, m}$ ) может быть поставлено неотрицательное действительное число  $p_i$  ( $0 \leq p_i \leq 1$ ) рассматриваемое как мера относительной полезности исхода.

Аксиома 2 – сравнимость (любые два исхода (альтернативы) сравнимы: либо один исход предпочтительней другого, либо они эквивалентны).

Аксиома 3 – транзитивность (соотношения предпочтения исходов транзитивны, если исход  $a_i$  предпочтительнее  $a_j$ , а исход  $a_j$  предпочтительнее  $a_k$ , то исход  $a_i$  также предпочтительнее  $a_k$ . По эквивалентности аналогично)

Аксиома 4 – коммутативность (предпочтение исхода  $a_i$  исходу  $a_j$  не зависит от порядка, в котором они представлены).

Аксиома 5 – независимость (если исход  $a_i$  предпочтительнее исхода  $a_j$  и, кроме того существует исход  $a_k$ , который не оценивается относительно исходов  $a_i$  и  $a_j$ , то смесь исходов  $a_i$  и  $a_k$  предпочтительнее смеси исходов  $a_j$  и  $a_k$ ).

Согласно теории полезности при выполнении в реальной задаче оценки систем всех пяти аксиом существует функция полезности, однозначно определенная на множестве всех альтернатив с точностью до монотонного строгого возрастающего линейного преобразования, т.е. полезность измеряется в шкале интервалов.

Процедура определения функции полезности включает в себя 3 этапа:

- выявление показателей исходов операции;
- определение множества допустимых исходов;
- определение показателей полезности исходов операции.

Определение полезности как меры оценки того или иного исхода операции представляет сложную задачу, точные методы решения которой пока не найдены. Все известные способы определения функции полезности носят приближенный характер. Такими способами являются экспертное оценивание и методы аппроксимации.

Определение функции полезности на основе аппроксимации заключается в

следующем. При рассмотрении исходов конкретной операции отыскиваются характерные точки, соответствующие, например, экстремумам функции полезности, а неизвестные значения между ними определяются некоторой известной зависимостью. Вид аппроксимации выбирается на основе имеющихся сведений или качественных соображений о показателе полезности исходов. На практике применяются многоступенчатые и другие сложные функции полезности. Наиболее простыми аппроксимациями являются одноступенчатое (1), косинусоидальное (2) и треугольное (3) представления функции полезности (см. рисунок 4.2).

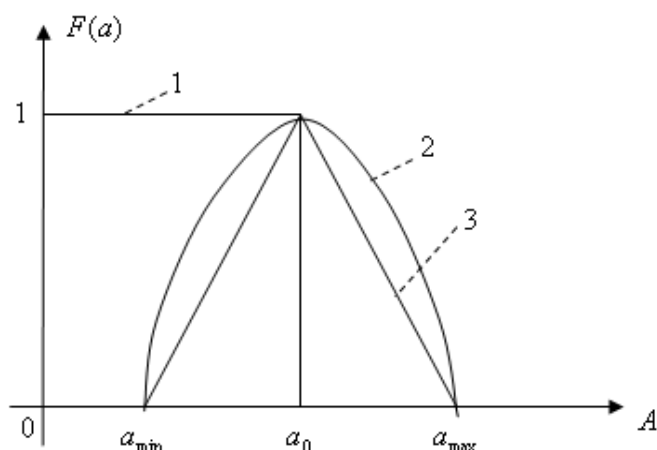


Рисунок 5.2 – Представление аппроксимации полезности (1 – одноступенчатое, 2 – косинусоидальное, 3 – треугольное)

Одноступенчатое представление функции полезности (1) может быть приемлемым для операций, в которых показателем исхода является срок выполнения работ. Например, подготовка презентации в ситуационном центре. В этом случае под исходами  $A$  понимается фактическое время готовности компьютерной презентации к работе. Очевидно, что полезность системы при  $\alpha_i < \alpha_0$  равна 1, а при  $\alpha_i > \alpha_0$  равна 0.

Косинусоидальное и треугольное представление функции полезности могут быть приемлемыми для операций, в которых показателями исхода является интервал времени, при этом функция полезности может быть представлена либо отрезком косинусоиды, либо треугольником.

Кроме оптимизации «в среднем» в вероятностных операциях используются и другие критерии оценки систем:

- максимум вероятности случайного события;
- максимум степени вероятностной гарантии достижения результата не ниже требуемого уровня;
- максимум среднего квадрата отклонения результата от требуемого;
- минимум дисперсии результата;
- минимум среднего (байесовского) риска (минимум средних потерь).

## **5.2 Методика выполнения задания**

**Цель работы:** Разработать программное средство СППР в условиях стохастической неопределенности

Отчет по лабораторной работе оформляется в электронной форме и состоит из разделов:

1. Тема лабораторной работы.
2. Цель работы.
3. Индивидуальное задание.
4. Описание модели (алгоритм) и текст программы.
5. Тестирование средства.
6. Выводы по работе.

### **Вопросы коллоквиума**

- 1) Оценивание систем в условиях риска.
- 2) Оценивание систем на основе функции полезности.
- 3) Перечислите основные аксиомы теории полезности.
- 4) Каким образом определяется функция полезности?
- 5) В чем заключается определение функции полезности на основе аппроксимации? Виды аппроксимации для представления функции полезности.

- 6) Что представляет собой функция реализации?
- 7) Отличие функции полезности от функции реализации?
- 8) Каким образом можно представить выбор альтернатив?
- 9) В каком виде можно представить задачу принятия решений в условиях риска?
- 10) Полезность исхода операции это....?

## **6 Лабораторная работа №5. Разработка средств СППР в стохастически неопределенных ситуациях**

Учебные вопросы:

- 6.1 Основы принятия решений в стохастически неопределенных условиях.
- 6.2 Методика выполнения задания.

### **6.1 Основы принятия решений в стохастически неопределенных условиях**

**Критерий среднего выигрыша.** Критерий среднего выигрыша предполагает задание вероятностей состояния обстановки  $P_i$ . Эффективность системы оценивается как среднее ожидаемое значение (МОЖ) оценок эффективности по всем состояниям обстановки:

$$K(a_i) = \sum_{j=1}^l P_j k_{ij}$$

Оптимальной системе будет соответствовать эффективность

$$K^{onm} = \max_i \sum_{j=1}^l P_j k_{ij}$$



## Критерий Лапласа

В основе данного критерия лежит предположение, поскольку о состоянии обстановки ничего не известно, то их можно считать равновероятностными. Следовательно,

$$K(a_i) = \frac{1}{n} \sum_{j=1}^l k_{ij}$$

Оптимальной системе будет соответствовать эффективность

$$K^{opt}(a_i) = \max_i \left( \frac{1}{n} \sum_{j=1}^l k_{ij} \right)$$

Таким образом, Критерий Лапласа – частный случай критерия среднего выигрыша.

## Критерий осторожного наблюдателя (Вальда)

Критерий Вальда гарантирует определенный выигрыш при наилучших условиях. Данный критерий основывается на том, что если состояние обстановки неизвестно, нужно поступать самым осторожным образом.

В каждом столбце матрицы эффективности находится минимальная из оценок системы по различным состояниям обстановки:

$$K(a_i) = \min_j k_{ij}$$

Оптимальной считается система из строки с максимальным значением коэффициента:

$$K^{opt}(a_i) = \max_i \left( \min_j k_{ij} \right)$$

Данный критерий ориентирован на решение, не содержащее элементов риска. Такая осторожность в ряде случаев является недостатком.

## Критерий максимакса

Данным критерием предписывается оценивать системы по максимальному значению эффективности и выбирать в качестве оптимального решения обследующую эффективность с наибольшим из максимумов:

$$K(a_i) = \max_j k_{ij}$$

$$K^{onm}(a_i) = \max_i \left( \max_j k_{ij} \right)$$

Данный критерий ориентирован на самое оптимистическое решение, при котором риск max.

## Критерий пессимизма-оптимизма (Гурвица)

Критерий Гурвица – это критерий обобщенного максимина, согласно которому при принятии решений не разумно проявлять как осторожность, так и азарт, а занимать промежуточную позицию.

Для этого вводится коэффициент оптимизма  $\alpha$  ( $\alpha = \overline{0,1}$ ). Эффективность решений находится как взвешенная с помощью коэффициента  $\alpha$  сумма максимальной и минимальной оценок.

$$K(a_i) = \alpha \max_j k_{ij} + (1 - \alpha) \min_j k_{ij}$$

Условия оптимальности записываются в следующем виде:

$$K^{onm}(a_i) = \max_i \left( \alpha \max_j k_{ij} + (1 - \alpha) \min_j k_{ij} \right)$$

## Критерий минимального риска (Сэвиджа)

Критерий Сэвиджа минимизирует потери эффективности при наихудших условиях.

Для оценки решений матрица эффективности преобразуется в матрицу потерь. Каждый элемент матрицы потерь определяется как разность между максимальным и текущим значениями оценок эффективности в столбце:

$$\Delta K_{ij} = \max_i k_{ij} - k_{ij}$$

После преобразования матрицы используется критерий минимакса:

$$K(a_i) = \max_j \Delta k_{ij}$$

$$K^{opt} = \min_i \left( \max_j \Delta k_{ij} \right)$$

## 5.2 Методика выполнения задания

Цель работы: разработка средств СППР в статистически неопределенных ситуациях

### Последовательность выполнения

Для выбранной предметной области согласно номеру по списку реализовать программное средство реализующие метод решения.

Отчет по лабораторной работе оформляется в электронной форме и содержит разделы:

- 1 Постановка задачи.
- 2 Описание метода решения.
- 3 Схема алгоритма и программный код.
- 4 Тестирование разработанного средства.
- 5 Вывод.
- 6 Список использованной литературы.

Для защиты отчета представить разработанное программное средство.

### **Вопросы коллоквиума**

- 1) Что лежит в основе метода принятия решения в условиях неопределенности?
- 2) Какие критерии используются в неопределенных операциях?
- 3) Что влияет на выбор того или иного критерия?
- 4) Критерии среднего выигрыша
- 5) Критерий Лапласа
- 6) Критерий осторожного наблюдателя (Вальда).
- 7) Критерий максимакса.
- 8) Критерий пессимизма-оптимизма (Гурвица).
- 9) Критерий минимального риска (Сэвиджа)
- 10) Какой из критериев минимизирует потери эффективности при наихудших условиях?

## Заключение

Лабораторный практикум включает в себя перечень лабораторных работ, имеющих целью выработки у студентов умений и навыков проектирования и разработки СППР с использованием современных инструментальных средств.

Выполнение лабораторных работ состоит из следующих основных этапов:

- краткое описание теории, раскрывающей понятия и важнейшие особенности принятия решений в конкретной предметной области при заданных условиях, на основе которых определяются методы решения задач;
- реализация программного средства на базе современной методологии программирования с учетом специфики методов (методик, алгоритмов) поддержки принятия решений применительно к автоматизированной системе.

Для выполнения всех лабораторных работ предлагается единый порядок, предусматривающий следующие этапы:

- уяснение цели работы и формулировка инженерных задач для её достижения;
- ознакомление с инструментарием, используемым для описания модели (диаграммы), и разработка алгоритма программного средства;
- разработка и реализация программного средства (модели);
- тестирование разработанного средства (компонента автоматизированной системы – аналитического приложения);
- сохранение результата;
- отчет по лабораторной работе.

Отчет по расчетно-графическому заданию (РГЗ) необходимо оформить исходя из типового отчета по РГЗ. Для защиты РГЗ продемонстрировать преподавателю работу разработанного программного средства и выполнить презентацию в соответствии с типовым отчетом по РГЗ.

## Список использованных источников

1 Государственный образовательный стандарт высшего профессионального образования по направлению 09.03.01 – Информатика и вычислительная техника. – М.: МинОбрНауки, 2013. – 44 с.

2 Анфилатов, В.С. Системный анализ в управлении /Под ред. А.А. Емельянов– М.: Финансы и статистика, 2006. – С. 9-15.

3 Вентцель, Е. С. Исследование операций: задачи, принципы, методология: учеб. пособие для вузов / Е. С. Вентцель.- 3-е изд., стер. – М.: Дрофа, 2004. - 208 с. : ил.

4 Соловьев, Н.А. Основы теории принятия решений для программистов: учебное пособие /Н.А. Соловьев, Е.Н. Чернопрудова, Д.А. Лесовой – Оренбург: ООО ИПК «Университет», 2012. – 168 с. – ISBN 978-5-4417-0092-4.

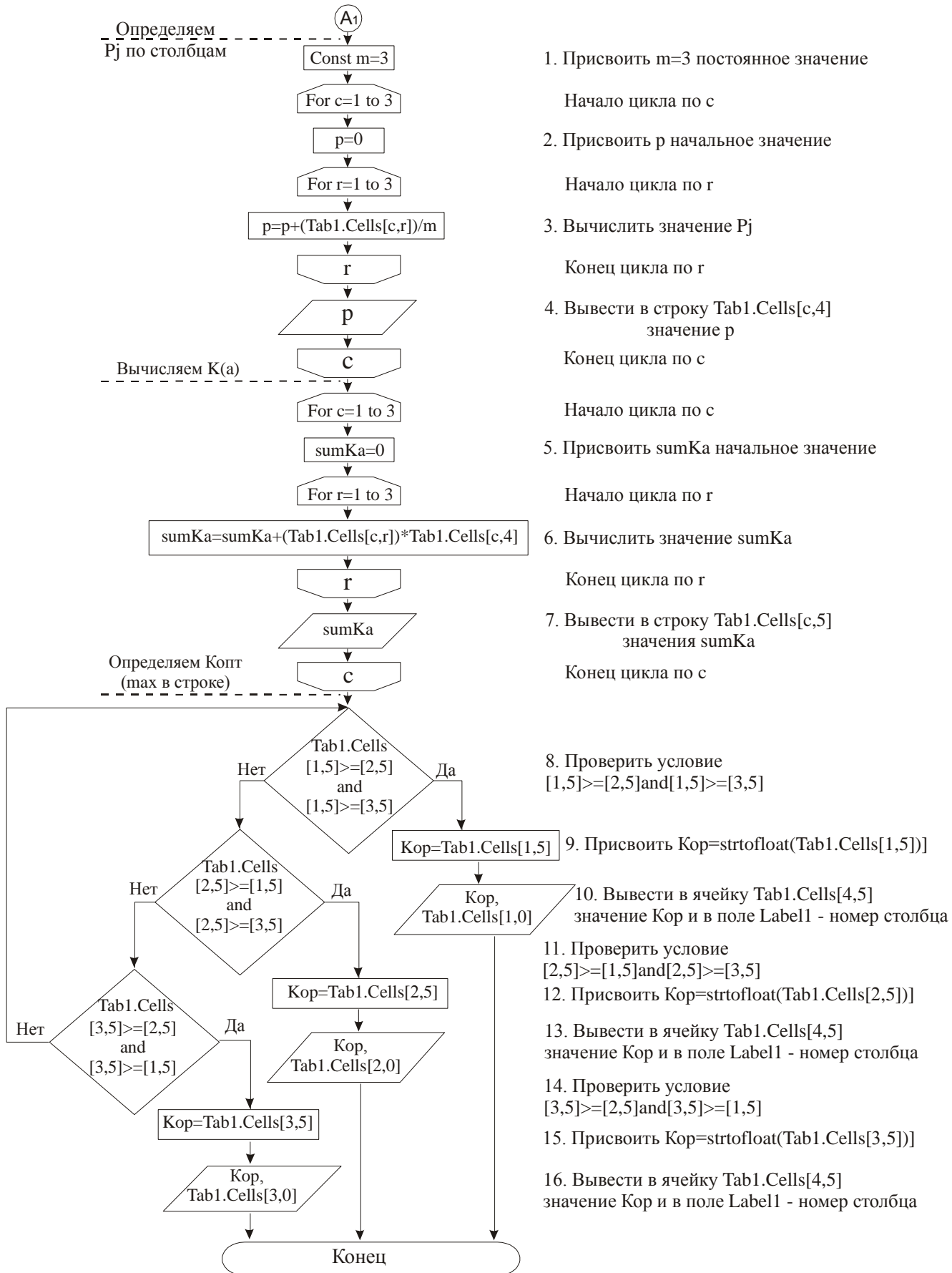
## Приложение А

### Варианты заданий на лабораторные работы № 3-5

№	В условиях определенности						В условиях неопределенности					
	ГК	ЛГ	ПУ	АС	МС	АГ	Св	Л	В	ММ	Г	С
1	+											+
2		+									+	
3			+							+		
4				+					+			
5					+			+				
6						+	+					
7	+											+
8		+									+	
9			+							+		
10				+					+			
11					+			+				
12						+	+					
13				+							+	
14					+		+					

# Приложение Б

## Пример алгоритма приложения





## Приложение В

### Пример составления списка использованных источников для системного анализа предметной области разработки

1 1С: Медицина. Больница .– Режим доступа: <http://1c.ru/news/info.jsp>.

2 ДОКА+ .– Режим доступа: <http://www.docaplus.com/russian/pages/pages.php>.

3 Инфоклиника. – Режим доступа: <http://www.sdsys.ru/page/ProdDir2>

4 Соловьев, Н.А. Системы автоматизации разработки программного обеспечения: учебное пособие / Н.А. Соловьев, Е.Н. Чернопрудова. – Оренбург: ООО ИПК «Университет», 2012. – 191 с. ISBN 978-5-4417-0086-3

5 Microsoft ADO.NET/Пер. с англ. – М.: Изд-во Торговый дом «Русская редакция», 2009. – 640 стр.: ил.

6 Вендров, А.М. Проектирование программного обеспечения экономических информационных систем: учебник/ А.М. Вендров – М.: Финансы и статистика, 2008. – 352 с.:ил.

7 Зубкова, Т.М. Технология разработки программного обеспечения: Учебное пособие/ Т.М Зубкова – Оренбург: ГОУ ОГУ, 2008. – 101 с.

8 Леоненков, А.В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose БИНОМ/ А.В. Леоненков Лаборатория знаний, Интернет-университет информационных технологий. - ИНТУИТ.ру, 2008 – 453 с.

9 Уэнди Боггс, Майкл Боггс. UML и Rational Rose 2002. – СПб.:Изд-во “Лори”, 2008. - 509 с.

10 Гмурман, В.Е. Руководство к решению задач по теории вероятностей и математической статистики. учебник/ В.Е. Гмурман – М.: Высшая школа, 2008.– 395 с.