Оглавление

Введение 6

1.Предпроектное исследование. 8

1.1. Основные подходы к построению ИМ. 8

1.2. Процесс имитации в РДО. 9

1.3. Основные положения языка РДО. 11

1.4. Постановка задачи. 13

2.Концептуальный этап проектирования. 17

2.1.Анализ альтернативных концепций решения задачи. 17

2.2.Диаграмма компонентов. 18

2.2.Структура логического вывода РДО. 20

2.3.Техническое задание. 20

2.3.1.Общие сведения. 20

2.3.2.Назначение и цели развития системы. 21

2.3.3.Характеристики объекта автоматизации. 21

2.3.4.Требования к системе. 21

3.Технический этап проектирования. 23

3.1.Разработка синтаксиса точки принятия решения. 23

3.2.Разработка архитектуры компонента rdo\_parser. 24

3.3.Разработка архитектуры компонента rdo\_runtime. 25

4.Рабочий этап проектирования. 26

4.1.Синтаксический анализ приоритета логик. 26

4.2.Изменения в пространстве имен rdoParse. 27

4.3.Изменения в пространстве имен rdoRuntime. 29

Заключение 32

Список использованных источников 33

Приложение 1. Полный синтаксический анализ точек принятия решений (rdodpt.y). 34

Приложение 2. Код имитационной модели работы почтового отделения связи на языке РДО. 47

Приложение 3. Функциональная диаграмма компиляции модели на языке РДО. Уровень A-0. 74

# Введение

««Сложные системы», «системность», «бизнес-процессы», «управление сложными системами», «модели» – все эти термины в настоящее время широко используются практически во всех сферах деятельности человека». Причиной этого является обобщение накопленного опыта и результатов в различных сферах человеческой деятельности и естественное желание найти и использовать некоторые общесистемные принципы и методы. Именно системность решаемых задач в перспективе должна стать той базой, которая позволит исследователю работать с любой сложной системой, независимо от ее физической сущности. Именно модели и моделирование систем является тем инструментом, которое обеспечивает эту возможность.

Имитационное моделирование (ИМ) на ЭВМ находит широкое применение при исследовании и управлении сложными дискретными системами (СДС) и процессами в них. К таким системам можно отнести экономические и производственные объекты, морские порты, аэропорты, комплексы перекачки нефти и газа, программное обеспечение сложных систем управления, вычислительные сети и многие другие. Широкое использование ИМ объясняется сложностью (а иногда и невозможностью) применения строгих методов оптимизации, которая обусловлена размерностью решаемых задач и неформализуемостью сложных систем. Так выделяют, например, следующие проблемы в исследовании операций, которые не могут быть решены сейчас и в обозримом будущем без ИМ:

1. Формирование инвестиционной политики при перспективном планировании.
2. Выбор средств обслуживания (или оборудования) при текущем планировании.
3. Разработка планов с обратной информационной связью и операционных предписаний.

Эти классы задач определяются тем, что при их решении необходимо одновременно учитывать факторы неопределенности, динамическую взаимную обусловленность текущих решений и последующих событий, комплексную взаимозависимость между управляемыми переменными исследуемой системы, а часто и строго дискретную и четко определенную последовательность интервалов времени. Указанные особенности свойственны всем сложным системам.

Проведение имитационного эксперимента позволяет:

1. Сделать выводы о поведении СДС и ее особенностях:
   * без ее построения, если это проектируемая система;
   * без вмешательства в ее функционирование, если это действующая система, проведение экспериментов над которой или слишком дорого, или небезопасно;
   * без ее разрушения, если цель эксперимента состоит в определении пределов воздействия на систему.
2. Синтезировать и исследовать стратегии управления.
3. Прогнозировать и планировать функционирование системы в будущем.
4. Обучать и тренировать управленческий персонал и т.д.

ИМ является эффективным, но и не лишенным недостатков, методом. Трудности использования ИМ, связаны с обеспечением адекватности описания системы, интерпретацией результатов, обеспечением стохастической сходимости процесса моделирования, решением проблемы размерности и т.п. К проблемам применения ИМ следует отнести также и большую трудоемкость данного метода.

Интеллектуальное ИМ, характеризующееся возможностью использования методов искусственного интеллекта и, прежде всего, знаний, при принятии решений в процессе имитации, при управлении имитационным экспериментом, при реализации интерфейса пользователя, создании информационных банков ИМ, снимает часть проблем использования ИМ.

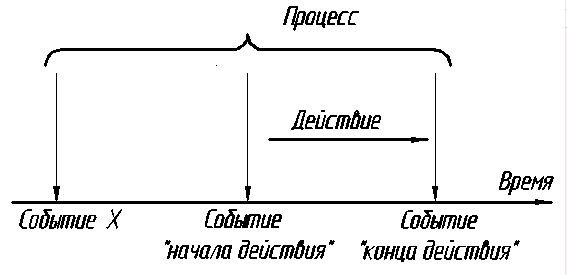
# 1.Предпроектное исследование.

## 1.1. Основные подходы к построению ИМ.

Системы имитационного моделирования СДС в зависимости от способов представления процессов, происходящих в моделируемом объекте, могут быть дискретными и непрерывными, пошаговыми и событийными, детерминированными и статистическими, стационарными и нестационарными.

Рассмотрим основные моменты этапа создания ИМ. Чтобы описать функционирование СДС надо описать интересующие нас события и действия, после чего создать алфавит, то есть дать каждому из них уникальное имя. Этот алфавит определяется как природой рассматриваемой СДС, так и целями ее анализа. Следовательно, выбор алфавита событий СДС приводит к ее упрощению – не рассматриваются многие ее свойства и действия не представляющие интерес для исследователя.

Событие СДС происходит мгновенно, то есть это некоторое действие с нулевой длительностью. Действие, требующее для своей реализации определенного времени, имеет собственное имя и связано с двумя событиями – начала и окончания. Длительность действия зависит от многих причин, среди которых время его начала, используемые ресурсы СДС, характеристики управления, влияние случайных факторов и т.д. В течение времени протекания действия в СДС могут возникнуть события, приводящие к преждевременному завершению действия. Последовательность действий образует процесс в СДС (Рис. 2.).



***Рис. 1. Взаимосвязь между событиями, действием и процессом.***

В соответствии с этим выделяют три альтернативных методологических подхода к построению ИМ: событийный, подход сканирования активностей и процессно-ориентированный.

## 1.2. Процесс имитации в РДО.

Для имитации работы модели в РДО реализованы два подхода: событийный и сканирования активностей.

***Событийный подход.***

При событийном подходе исследователь описывает события, которые могут изменять состояние системы, и определяет логические взаимосвязи между ними. Начальное состояние устанавливается путем задания значений переменным модели и параметров генераторам случайных чисел. Имитация происходит путем выбора из списка будущих событий ближайшего по времени и его выполнения. Выполнение события приводит к изменению состояния системы и генерации будущих событий, логически связанных с выполняемым. Эти события заносятся в список будущих событий и упорядочиваются в нем по времени наступления. Например, событие начала обработки детали на станке приводит к появлению в списке будущих событий события окончания обработки детали, которое должно наступить в момент времени равный текущему времени плюс время, требуемое на обработку детали на станке. В событийных системах модельное время фиксируется только в моменты изменения состояний.

Выбор из списка будущих событий ближайшего

Перевод модельного времени вперед

Выполнение программы для данного события

***Рис. 2. Выполнение событий в ИМ.***

***Подход сканирования активностей.***

При использовании подхода сканирования активностей разработчик описывает все действия, в которых принимают участие элементы системы, и задает условия, определяющие начало и завершение действий. После каждого продвижения имитационного времени условия всех возможных действий проверяются и если условие выполняется, то происходит имитация соответствующего действия. Выполнение действия приводит к изменению состояния системы и возможности выполнения новых действий. Например, для начала действия обработка детали на станке необходимо наличие свободной детали и наличие свободного станка. Если хотя бы одно из этих условий не выполнено, действие не начинается.

Вход

Выбрать из списка очередное действие

Выполнить действие, обновив состояние системы

Условие выполняется?

Все действия просмотрены?

Печать итогового отчета

Завершен ли прогон?

Да

Нет

Нет

Нет

Да

***Рис. 3. Блок-схема реализации***

***подхода сканирования активностей.***

## 1.3. Основные положения языка РДО.

В основе системы РДО – «Ресурсы, Действия, Операции» – лежат следующие положения:

* Все элементы сложной дискретной системы (СДС) представлены как ресурсы, описываемые некоторыми параметрами.
* Состояние ресурса определяется вектором значений всех его параметров; состояние СДС – значением всех параметров всех ресурсов.
* Процесс, протекающий в СДС, описывается как последовательность целенаправленных действий и нерегулярных событий, изменяющих определенным образом состояния ресурсов; действия ограничены во времени двумя событиями: событиями начала и конца.
* Нерегулярные события описывают изменение состояния СДС, непредсказуемые в рамках продукционной модели системы (влияние внешних по отношению к СДС факторов либо факторов, внутренних по отношению к ресурсам СДС). Моменты наступления нерегулярных событий случайны.
* Действия описываются операциями, которые представляют собой модифицированные продукционные правила, учитывающие временные связи. Операция описывает предусловия, которым должно удовлетворять состояние участвующих в операции ресурсов, и правила изменения ресурсов в начале и конце соответствующего действия.

При выполнении работ, связанных с созданием и использованием ИМ в среде РДО, пользователь оперирует следующими основными понятиями:

**Модель** - совокупность объектов РДО-языка, описывающих какой-то реальный объект, собираемые в процессе имитации показатели, кадры анимации и графические элементы, используемые при анимации, результаты трассировки.

**Прогон** - это единая неделимая точка имитационного эксперимента. Он характеризуется совокупностью объектов, представляющих собой исходные данные и результаты, полученные при запуске имитатора с этими исходными данными.

**Проект** - один или более прогонов, объединенных какой-либо общей целью. Например, это может быть совокупность прогонов, которые направлены на исследование одного конкретного объекта или выполнение одного контракта на имитационные исследования по одному или нескольким объектам.

**Объект** - совокупность информации, предназначенной для определенных целей и имеющая смысл для имитационной программы. Состав объектов обусловлен РДО-методом, определяющим парадигму представления СДС на языке РДО.

Объектами исходных данных являются:

* типы ресурсов (с расширением .rtp);
* ресурсы (с расширением .rss);
* образцы операций (с расширением .pat);
* операции (с расширением .opr);
* точки принятия решений (с расширением .dpt);
* константы, функции и последовательности (с расширением .fun);
* кадры анимации (с расширением .frm);
* требуемая статистика (с расширением .pmd);
* прогон (с расширением .smr).

Объекты, создаваемые РДО-имитатором при выполнении прогона:

* результаты (с расширением .pmv);
* трассировка (с расширением .trc).

## 1.4. Постановка задачи.

Основная идея бакалаврской работы – добавление в механизм логического вывода РДО иерархии логик. Под логикой, в данном контексте, понимается такая сущность системы, которая может объединять в себе простые атомарные активности и имеет свой алгоритм работы. Т.е. логиками являются точки принятия решений всех видов (some, prior, free и search).

Другими словами, в РДО должна появиться возможность встраивать точки принятия решений внутрь дригих точек принятия решений.

Сейчас точки принятия решений могут содержать лишь активности.

Это накладывает определенные ограничения на процесс разработки моделей на языке РДО. Для демонстрации этого недостатка рассмотрим модель работы почтового отделения связи. Прием клиентов может вестись в 5-ти окнах. Каждое из них работает по своему расписанию. Каждое из окон предоставляет клиентам весь спектр услуг. Емкость очереди в каждое окно ограничена максимальным значением. Все клиенты, которые не помещаются в очередь, уходят без обслуживания. Все клиенты, которые стоят в очереди на момент окончания рабочего дня также покидают почту не обслужившись. Почта работает с 9 часов утра до 7 вечера.

Модель данной СМО на языке РДО представлена в Приложении 2.

Обратим внимание на описание БЗ модели на закладке DPT:

**$Decision\_point** приход\_клиентов : **some**

**$Condition** Отделение.Состояние = Рабочий\_День

**$Activities**

Еще\_Один\_Пришел : Приход\_Клиента\_На\_Почту

**$End**

**$Decision\_point** размещение\_клиентов : **some**

**$Condition** Отделение.Состояние = Рабочий\_День

**$Activities**

Какое\_Окно : Определить\_Окно

Потерянного\_Вон : Ликвидирование

В\_Очередь\_Его : Приход\_Клиента\_В\_Очередь

**$End**

**$Decision\_point** обслуживание\_клиентов : **some**

**$Condition** Отделение.Состояние = Рабочий\_День

**$Activities**

Наконец\_То\_В\_Окно1 : Обслуживание\_Первого 1

Наконец\_То\_В\_Окно2 : Обслуживание\_Первого 2

Наконец\_То\_В\_Окно3 : Обслуживание\_Первого 3

Наконец\_То\_В\_Окно4 : Обслуживание\_Первого 4

Наконец\_То\_В\_Окно5 : Обслуживание\_Первого 5

**$End**

**$Decision\_point** продвижение\_очередей : **some**

**$Condition** Отделение.Состояние = Рабочий\_День

**$Activities**

Вперед1 : Сдвинуть\_Очередь 1

Вперед2 : Сдвинуть\_Очередь 2

Вперед3 : Сдвинуть\_Очередь 3

Вперед4 : Сдвинуть\_Очередь 4

Вперед5 : Сдвинуть\_Очередь 5

**$End**

**$Decision\_point** уход\_клиентов : **some**

**$Condition** Отделение.Состояние = Рабочий\_День

**$Activities**

От\_Окна1 : Уход\_Клиента 1

От\_Окна2 : Уход\_Клиента 2

От\_Окна3 : Уход\_Клиента 3

От\_Окна4 : Уход\_Клиента 4

От\_Окна5 : Уход\_Клиента 5

**$End**

**$Decision\_point** открытие\_закрытие\_окна : **some**

**$Condition** Отделение.Состояние = Рабочий\_День

**$Activities**

Я\_Работаю : Открыть\_Окно

Я\_Закрылось : Закрыть\_Окно

**$End**

**$Decision\_point** конец\_дня : **some**

**$Condition** Отделение.Состояние = Конец\_Дня

**$Activities**

Поехали\_Снова : Переход\_К\_Новому\_Дню

**$End**

**$Decision\_point** уход\_клиентов\_в\_конце\_дня : **some**

**$Condition** Отделение.Состояние = Конец\_Дня

**$Activities**

Лишних\_Вон1 : Уничтожение 1

Лишних\_Вон2 : Уничтожение 2

Лишних\_Вон3 : Уничтожение 3

Лишних\_Вон4 : Уничтожение 4

Лишних\_Вон5 : Уничтожение 5

**$End**

**$Decision\_point** служебные : **some**

**$Activities**

Часы\_На\_Экран : Соответствие\_Времени

Поехали : Функционирование

**$End**

При разработке данной имитационной модели работа почты была разделена на 2 этапа: рабочий день и окончание рабочего дня. Технически это реализовано с помощью условия запуска точек принятия решений. На первом этапе происходит поступление клиентов в систему, их размещение в очереди либо отказ в обслуживании, ожидание, обслуживание, уход. Также во время рабочего дня некоторые окна закрываются на перерыв и снова открываются. В свою очередь, после окончания рабочего дня на почту перестают приходить новые клиенты, все клиенты из очереди удаляются не обслуженными, клиенты, чьи заявки уже выполняются заканчивают свое обслуживание и уходят. После того, как на почте не остается клиентов, производятся некоторые технические процедуры по завершению текущего рабочего дня и подготовке к следующему. После этого может начинаться следующий день. Кроме этого также можно выделить некоторые служебные знания, присущие имитационным моделям, отображение времени, его синхронизация, переход между описанными выше этапами функционирования почты.

Все это говорит о иерархических знаниях, присущих описанию этой системы массового обслуживания. Но текущая версия РДО не позволяет это записывать в явном виде.

Таким образом, целью бакалаврской работы является разработка логического вывода системы имитационного моделирования РДО, основанного на иерархических логиках.

# 2.Концептуальный этап проектирования.

## 2.1.Анализ альтернативных концепций решения задачи.

Существует несколько альтернативных способов решения поставленной задачи, отличающихся простотой использования, наглядностью, гибкостью, предоставляемыми возможностями и, конечно, сложностью реализации. Поэтому необходимо провести их сравнительный анализ с целью выбора одной из альтернативных концепций решения задачи, которая позволит создать новый инструмент удобным, выразительным и мощным в сроки, отведенные для бакалаврской работы.

Первый вариант решения задачи - помещение полного описания одной (дочерней) точки принятия решений внутрь другой (родительской). Это не вносит никаких изменений в набор ключевых слов системы, и кажется весьма простым, интуитивно понятным решением с точки зрения разработчика моделей. Его основной недостаток заключается в большом объеме плохо структурированного текста, что может сказаться негативно на удобстве написания и сопровождения моделей при недостаточно строгом форматировании их кода.

Второй вариант - использование «ссылок» из дочерних точек на родительские. Т.е. в заголовке точки принятия решений должна появиться новая строка с именем той точки, активностью которой должна стать текущая. Недостатком этого метода является наличие ограничений на порядок следования точек: родительская точка должна быть описана раньше, чем дочерняя. Однако такое решение задачи не вносит каких-либо серьезных изменений в стилистику языка РДО и является весьма лаконичным.

Третьй способ - заведение в БЗ точки в виде активности. Т.е. необходимо добавить в систему новый тип образца активности - образца точки принятия решений. Этот метод является функциональнее других багодаря возможности многократного использования одного образца и передачи ему параметров. Частным случаем этих параметров может быть условие запуска или приоритет активности-точки, которые могут быть переданы от родительской точки дочерней, либо оставлены без изменений (эти значения у дочерней точки фактически являются значениями по умолчанию).

Последний из рассотренных вариантов, без сомнения, является наиболее мощным и гибким инструментом, но его разработка связана с большим числом изменений, которые необходимо внести в РДО, и, соответственно, с наличием временного ресурса большего, чем предоставляется на выполнение бакалаврской работы. Поэтому был сделан выбор в пользу второй концепции решения поставленной задачи.

## 2.2.Диаграмма компонентов.

Система имитационного моделирования РДО безусловно является сложной и статически, и динамически. На это указывает сложная иерархическая структура системы со множеством различных связей между компонентами и ее сложное поведение во времени.

Ярко выраженная иерархическая структура и модульность системы определяют направление изучения системы сверху вниз. Т.е. мне необходимо применять принцип декомпозиции нужных модулей до тех пор, пока не будет достигнут уровень абстракции, представление на котором нужных объектов не нуждается в дальнейшей детализации для решения данной задачи.

Для отображения зависимости между компонентами системы РДО и выделения среди них модернизируемых служит соответствующая диаграмма в нотации UML.

***Рис. 4. Упращенная диаграмма компонентов.***

Базовый функционал представленных на диаграмме компонентов:

rdo\_kernel реализует ядровые функции системы. Не изменяется при разработке системы.

RAO-studio.exe реализует графический интерфейс пользователя. Не изменяется при разработки системы.

rdo\_repository реализует управление потоками данных внутри системы и отвечает за хранение и получение информации о модели. Не изменяется при разработке системы.

rdo\_mbuilder реализует функционал, используемый для программного управления типами ресурсов и ресурсами модели. Не изменяется при разработке системы.

rdo\_simulator управляет процессом моделирования на всех его этапах. Он осуществляет координацию и управление компонентами rdo\_runtime и rdo\_parser. Не изменяется при разработке системы.

rdo\_parser производит лексический и синтаксический разбор исходных текстов модели, написанной на языке РДО. Модернизируется при разработке системы.

rdo\_runtime отвечает за непосредственное выполнение модели, управление базой данных и базой знаний. Модернизируется при разработке системы.

Объекты компонента rdo\_runtime инициализуруются при разборе исходного текста модели компонентом rdo\_parser. Например, конструктор rdoParse::RDODPTSome::RDODPTSome содержит следующее выражение:

m\_rt\_logic = new rdoRuntime::RDODPTSome( parser()->runtime() );

которое выделяет место в свободной памяти и инициализирует объект rdoRuntime::RDODPTSome, учавствующий в дальнейшем процессе имитации.

В дальнейшем компоненты rdo\_parser и rdo\_runtime описываются более детально.

## 2.2.Структура логического вывода РДО.

Логический вывод системы РДО представляет собой алгоритм, который определяет какое событие в моделируемой системе должно произойти следующим в процессе имитации работы системы.

Во время имитации работы модели в системе существует одна МЕТА-логика. Она является контейнером для хранения разных логик. Сами логики являются одновременно и контейнерами, в которых хранятся различные атомарные операции (например, нерегулярные события и правила) и атомарной (базовай) операцией. Т.е. данная архитектура БЗ РДО уже позволяет включать логики внутрь логик и в рамках данного проекта нужно создать инструмент, который будет встраивать логики модели в нужные места существующего дерева БЗ. Это позволит создавать иерархические БЗ.

Поиск активности, которая должна быть запущена следующей, начинается с обращения класса RDOSimulator к своему атрибуту m\_logics, в котором хранится описанная выше МЕТА-логика. Далее от корня дерева к листьям распространяется волна вызовов метода onCheckCondition(). Т.е. onCheckCondition() вызывается у МЕТА-логики, затем циклически у ее логик, и наконец, циклически проверяются все атомарные операции каждой логики. Как только найдена активность, которая может быть выполнена, происходит ее кэширование (запоминание) внутри логики и кэширование самой логики внутри МЕТА-логики. После этого управление снова передается в RDOSimulator и найденная активность выполняется. Т.е. существующий механизм будет правильно обрабатывать многоуровневые иерархические деревья логик.

## 2.3.Техническое задание.

### 2.3.1.Общие сведения.

В системе РДО разрабатывается новый инструмент построения иерархической базы знаний в виде многоуровневых деревьев логик. Основной разработчик РДО – кафедра РК-9, МГТУ им. Н.Э. Баумана.

### 2.3.2.Назначение и цели развития системы.

Основная цель данного курсового проекта – разработать механизм логического вывода в системе имитационного моделирования РДО на основе иерархических логик.

### 2.3.3.Характеристики объекта автоматизации.

РДО – язык имитационного моделирования, включающий все три основные подхода описания дискретных систем: процессный, событийный и сканирования активностей.

### 2.3.4.Требования к системе.

При описании точки принятия решения пользователь может после имени точки указать родительскую точку - слово “$Parent” и имя описанной ранее точки типа Some или Prior.

Т.е. с учетом возможности описания приоритетов точек принятия решений, модель производственного участока должна иметь вид:

**$Decision\_point** main : **some**

**$Activities**

**$End**

**$Decision\_point** рабочий\_день : **some** **trace**

$Parent main

**$Condition** Отделение.Состояние = Рабочий\_День

**$Activities**

**$End**

**$Decision\_point** приход\_клиентов : **some**

$Parent рабочий\_день

**$Activities**

Еще\_Один\_Пришел : Приход\_Клиента\_На\_Почту

**$End**

**$Decision\_point** размещение\_клиентов : **some**

$Parent рабочий\_день

**$Activities**

Какое\_Окно : Определить\_Окно

Потерянного\_Вон : Ликвидирование

В\_Очередь\_Его : Приход\_Клиента\_В\_Очередь

**$End**

**$Decision\_point** обслуживание\_клиентов : **some**

$Parent рабочий\_день

**$Activities**

Наконец\_То\_В\_Окно1 : Обслуживание\_Первого 1

Наконец\_То\_В\_Окно2 : Обслуживание\_Первого 2

Наконец\_То\_В\_Окно3 : Обслуживание\_Первого 3

Наконец\_То\_В\_Окно4 : Обслуживание\_Первого 4

Наконец\_То\_В\_Окно5 : Обслуживание\_Первого 5

**$End**

**$Decision\_point** продвижение\_очередей : **some**

$Parent рабочий\_день

**$Activities**

Вперед1 : Сдвинуть\_Очередь 1

Вперед2 : Сдвинуть\_Очередь 2

Вперед3 : Сдвинуть\_Очередь 3

Вперед4 : Сдвинуть\_Очередь 4

Вперед5 : Сдвинуть\_Очередь 5

**$End**

**$Decision\_point** уход\_клиентов : **some**

$Parent рабочий\_день

**$Activities**

От\_Окна1 : Уход\_Клиента 1

От\_Окна2 : Уход\_Клиента 2

От\_Окна3 : Уход\_Клиента 3

От\_Окна4 : Уход\_Клиента 4

От\_Окна5 : Уход\_Клиента 5

**$End**

**$Decision\_point** открытие\_закрытие\_окна : **some**

$Parent рабочий\_день

**$Activities**

Я\_Работаю : Открыть\_Окно

Я\_Закрылось : Закрыть\_Окно

**$End**

**$Decision\_point** конец\_дня : **some**

$Parent main

**$Condition** Отделение.Состояние = Конец\_Дня

**$Activities**

Поехали\_Снова : Переход\_К\_Новому\_Дню

**$End**

**$Decision\_point** уход\_клиентов\_в\_конце\_дня : **some**

$Parent конец\_дня

**$Activities**

Лишних\_Вон1 : Уничтожение 1

Лишних\_Вон2 : Уничтожение 2

Лишних\_Вон3 : Уничтожение 3

Лишних\_Вон4 : Уничтожение 4

Лишних\_Вон5 : Уничтожение 5

**$End**

**$Decision\_point** служебные : **some**

$Parent main

**$Activities**

Часы\_На\_Экран : Соответствие\_Времени

Поехали : Функционирование

**$End**

# 3.Технический этап проектирования.

## 3.1.Разработка синтаксиса точки принятия решения.

Объект точек принятия решений имеет следующий формат:

<описание\_точки\_принятия\_решений> { <описание\_точки\_принятия\_решений> }

Описание точки принятия решений имеет следующий формат:

[ <заголовок\_точки\_принятия\_решений> ] <блок\_активностей>

У точки принятия решений заголовок может состоять лишь из описания приоритета точки (или в частном случае отсутствовать вовсе). В таком случае она называется свободной точкой принятия решений или блоком свободных активностей.

Заголовок точки принятия решений имеет следующий формат:

$Decision\_point <имя\_точки> : <тип\_точки> [ <признак\_трассировки> ]

$Parent <имя\_родительской\_точки>

$Condition <условие\_активизации\_точки>

$Priority <приоритета\_точки>

[ $Term\_condition <терминальное\_условие>

$Evaluate\_by <оценка\_стоимости\_оставшегося\_пути\_на\_графе>

$Compare\_tops = <признак\_сравнения\_вершин> ]

Имена точек принятия решений должны быть различными для всех точек принятия решений и не должны совпадать с ранее определенными именами.

Тип точки может быть одним из следующих: some, prior или search (подробнее см. в справке РДО).

Имя родительской точки - имя одной из описанных выше точек принятия решений типа Some или Prior.

Признак трассировки может быть одним из следующих: no\_trace, trace\_stat, trace\_tops, trace\_all (подробнее см. в справке РДО).

Условие активизации точки – это логическое выражение. Алгоритм обработки точки принятия решений активизируется только в том случае, если состояние системы удовлетворяет этому выражению.

Приоритет точки - арифметическое выражение целого или вещественного типа данных, ограниченное диапазоном [0; 1].

Терминальное условие поиска, эвристическая оценочная функция стоимости оставшегося пути до целевой вершины и признак трассировки вершин записывается только для точек типа search (подробнее см. в справке РДО).

Блок активностей имеет следующий формат:

$Activities

<описание\_активности> {<описание\_активности>}

$End

Описание каждой активности подробно описано в справке РДО.

## 3.2.Разработка архитектуры компонента rdo\_parser.

Для возможности обработки новой конструкции в коде модели требуют изменений лексический и синтаксический анализаторы РДО.

В пространстве имен rdoParse у конструкторов точек принятия решений должен появиться дополнительный параметр - указатель на родительскую точку. При конструировании точки этот параметр должен передаваться из rdoParse в rdoRuntime.

## 3.3.Разработка архитектуры компонента rdo\_runtime.

В пространстве имен rdoRuntime указатель на родительскую точку должен появиться у базового класса для всех логик - RDOLogic.

Конструктора всех логик должны сохранять передаваемый им из rdoParse указатель на родительскую точку.

После создания точки происходит ее инициализация, в ходе которой точка должна быть «привязана» к нужной ветви существующего дерева БЗ модели.

Если rdoParse не передал в rdoRuntime указатель на родительскую точку, то в качестве нее должна быть использована МЕТА-логика.

# 4.Рабочий этап проектирования.

## 4.1.Синтаксический анализ приоритета логик.

Для реализации в среде имитационного моделирования нового инструмента разработанного на концептуальном и техническом этапах проектирования, в первую очередь необходимо добавить новые термальные символы в лексический анализатор РДО и нетермальные символы в грамматический анализатор.

В лексическом анализаторе (flex) я добавил новый токен RDO\_Parent, который может быть записан двумя разными способами (с заглавной и строчной буквы):

$Parent return(RDO\_Parent);

$parent return(RDO\_Parent);

Этот токен необходимо также добавить в генератор синтаксического анализатора (bison):

%token RDO\_ Parent 377

Далее нужно добавить описание приоритета точки:

dpt\_some\_parent:

/\* empty \*/

{

$$ = 0;

}

| RDO\_Parent RDO\_IDENTIF

{

$$ = $2;

}

| RDO\_Parent error

{

PARSER->error( @1, "Ошибка в имени родительской точки" );

};

dpt\_some\_begin:

RDO\_Decision\_point RDO\_IDENTIF\_COLON RDO\_some dpt\_some\_trace dpt\_some\_parent

{

RDOValue\* name = reinterpret\_cast<RDOValue\*>($2);

RDOValue\* parent\_name = reinterpret\_cast<RDOValue\*>($5);

if ( parent\_name != 0 )

{

const RDODPTPrior\* parentDPTPrior = PARSER->findDPTPrior( parent\_name->value().getIdentificator() );

const RDODPTSearch\* parentDPTSearch = PARSER->findDPTSearch( parent\_name->value().getIdentificator() );

const RDODPTSome\* parentDPTSome = PARSER->findDPTSome( parent\_name->value().getIdentificator() );

if ( parentDPTPrior == NULL && parentDPTSearch == NULL && parentDPTSome == NULL )

{

PARSER->error( @1, rdo::format("Не найдена родительская точка %s", parent\_name->value().getIdentificator().c\_str()) );

}

if ( parentDPTSearch != NULL )

{

PARSER->error( @5, @1, "Точка принятия решений типа search может содержать лишь активности типа rule и не может быть указана в качестве родительской точки" );

}

if ( parentDPTPrior != NULL )

{

LPILogic parent = parentDPTPrior->getLogic();

$$ = (int)new RDODPTSome( PARSER, name->src\_info(), parent );

}

if ( parentDPTSome != NULL )

{

LPILogic parent = parentDPTSome->getLogic();

$$ = (int)new RDODPTSome( PARSER, name->src\_info(), parent );

}

}

if ( parent\_name == 0 )

{

$$ = (int)new RDODPTSome( PARSER, name->src\_info() );

}

};

Из этого кода можно сделать вывод, что точка не обязательно должна иметь родительскую точку. Это сделано для поддержки уже существующих моделей, написанных на РДО. Если пользователь опустит родительскую точку, то для создания новой точки будет вызван специальный конструктор, который сам укажет в качестве родительской логики - МЕТА-логику.

Если родительская точка до сих пор была не объявлена, то пользователь получит сообщение «Не найдена родительская точка». Т.е. это также произойдет, если родительская точка была описана ниже.

Если в качестве родительской указана точка типа Search, то пользователь также получит сообщение об ошибке «Точка типа Search не может быть указана в качестве родительской точки».

Подробное описание всех точек принятия решений приведено в Приложении 1.

## 4.2.Изменения в пространстве имен rdoParse.

*Объвление класса RDODPTPrior*

class RDODPTPrior: public RDOLogicActivity<rdoRuntime::RDODPTPrior, RDODPTPriorActivity>

{

public:

RDODPTPrior( RDOParser\* \_parser, const RDOParserSrcInfo& \_src\_info, LPILogic \_parent = NULL );

LPILogic getLogic() const { return m\_rt\_logic; }

RDOFUNLogic\* getConditon() const { return m\_conditon; }

void setCondition( RDOFUNLogic\* conditon = NULL ) { m\_conditon = conditon; }

void end();

private:

RDOFUNLogic\* m\_conditon;

};

LPILogic getLogic( ) - метод класса RDODPTPrior, возвращает указатель на логику. Этот метод используется для получения указателя на родительскую точку.

*Объвление класса RDODPTSome*

class RDODPTSome: public RDOLogicActivity<rdoRuntime::RDODPTSome, RDODPTSomeActivity>

{

public:

RDODPTSome( RDOParser\* \_parser, const RDOParserSrcInfo& \_src\_info, LPILogic \_parent = NULL );

LPILogic getLogic() const { return m\_rt\_logic; }

RDOFUNLogic\* getConditon() const { return m\_conditon; }

void setCondition( RDOFUNLogic\* conditon = NULL ) { m\_conditon = conditon; }

void end();

private:

RDOFUNLogic\* m\_conditon;

};

LPILogic getLogic( ) - метод класса RDODPTSome, возвращает указатель на логику. Этот метод используется для получения указателя на родительскую точку.

*Определение конструктора класса RDODPTActivityHotKey*

RDODPTActivityHotKey::RDODPTActivityHotKey( LPIBaseOperationContainer dpt, const RDOParserObject\* parent, const RDOParserSrcInfo& \_src\_info, const RDOParserSrcInfo& \_pattern\_src\_info ):

RDODPTActivity( parent, \_src\_info, \_pattern\_src\_info )

{

switch ( pattern()->getType() )

{

case RDOPATPattern::PT\_IE:

{

m\_activity = static\_cast<rdoRuntime::RDOPatternIrregEvent\*>(pattern()->getPatRuntime())->createActivity( dpt, parser()->runtime(), name() );

break;

}

case RDOPATPattern::PT\_Rule:

{

m\_activity = static\_cast<rdoRuntime::RDOPatternRule\*>(pattern()->getPatRuntime())->createActivity( dpt, parser()->runtime(), name() );

break;

}

case RDOPATPattern::PT\_Operation:

{

m\_activity = static\_cast<rdoRuntime::RDOPatternOperation\*>(pattern()->getPatRuntime())->createActivity( dpt, parser()->runtime(), name() );

break;

}

case RDOPATPattern::PT\_Keyboard:

{

m\_activity = static\_cast<rdoRuntime::RDOPatternKeyboard\*>(pattern()->getPatRuntime())->createActivity( dpt, parser()->runtime(), name() );

break;

}

default:

{

parser()->error\_push\_only( src\_info(), "Íåèçâåñòíûé òèï îáðàçöà" );

parser()->error\_push\_only( pattern()->src\_info(), "Ñì. îáðàçåö" );

parser()->error\_push\_done();

}

}

}

Этот конструктор создает активность с помощью метода createActivity(), которому передает в качестве параметра указатель на логику, в которой находится данная активность. С помощью этого указателя активность привязывается к нужной ветви БЗ модели.

## 4.3.Изменения в пространстве имен rdoRuntime.

*Объвление класса RDOLogic*

template <class Order>

class RDOLogic: public IBaseOperation, public IBaseOperationContainer, public ILogic, CAST\_TO\_UNKNOWN

{

QUERY\_INTERFACE\_BEGIN

QUERY\_INTERFACE(IBaseOperation)

QUERY\_INTERFACE(IBaseOperationContainer)

QUERY\_INTERFACE(ILogic)

QUERY\_INTERFACE\_END

public:

typedef BaseOperationList ChildList;

typedef BaseOperationList::iterator Iterator;

typedef BaseOperationList::const\_iterator CIterator;

protected:

RDOLogic(PTR(RDOSimulator) sim = NULL, LPIBaseOperationContainer parent = NULL);

virtual ~RDOLogic();

DECLARE\_IBaseOperationContainer;

PTR(RDOCalc) m\_condition;

rbool m\_lastCondition;

ChildList m\_childList;

LPIBaseOperation m\_first;

LPIBaseOperationContainer m\_parent;

private:

rbool checkSelfCondition(PTR(RDOSimulator) sim);

void start (PTR(RDOSimulator) sim);

void stop (PTR(RDOSimulator) sim);

DECLARE\_IBaseOperation;

DECLARE\_ILogic;

};

В атрибуте m\_parent класса RDOLogic хранится указатель на родительскую логику (точку).

*Определение конструктора класса RDOLogic*

template <class Order>

inline RDOLogic<Order>::RDOLogic(PTR(RDOSimulator) sim, LPIBaseOperationContainer parent)

: m\_condition (NULL )

, m\_lastCondition(false )

, m\_first (NULL )

, m\_parent (parent ? parent : (sim ? sim->m\_metaLogic : NULL))

{}

Если конструктор RDOLogic был вызван с указателем на родительскую точку, то она сохранится атрибуте m\_parent. Если конструктор RDOLogic был вызван без указателя на родительскую точку, но с указателем на симулятор, то в качестве родительской логике сохранится МЕТА-логика. Иначе, если конструктор RDOLogic был вызван без обоих параметров (это нужно для заведения МЕТА-логики), то в атрибут m\_parent запишется нулевой указатель.

*Определение функции-члена init() класса RDOLogic*

template <class Order>

inline void RDOLogic<Order>::init(PTR(RDOSimulator) sim)

{

if (sim)

sim->appendLogic(rdo::UnknownPointer(this).query\_cast<IBaseOperation>(), this->m\_parent);

}

В этом методе происходит «привязывание» логики к дереву БЗ модели.

*Определение функции-члена init() классов RDOPatternRule*

LPIRule RDOPatternRule::createActivity(LPIBaseOperationContainer logic, PTR(RDORuntime) runtime, PTR(RDOCalc) condition, CREF(tstring) \_oprName)

{

LPIRule rule = F(RDORule)::create(runtime, this, traceable(), condition, \_oprName);

runtime->addRuntimeRule(logic, rule);

return rule;

}

LPIOperation RDOPatternOperation::createActivity(LPIBaseOperationContainer parent, PTR(RDORuntime) runtime, CREF(tstring) \_oprName)

{

LPIOperation operation = F(RDOOperation)::create(runtime, this, traceable(), \_oprName);

runtime->addRuntimeOperation(parent, operation);

return operation;

}

LPIIrregEvent RDOPatternIrregEvent::createActivity(LPIBaseOperationContainer parent, PTR(RDORuntime) runtime, CREF(tstring) oprName)

{

LPIIrregEvent ie = F(RDOIrregEvent)::create(runtime, this, traceable(), oprName);

runtime->addRuntimeIE(parent, ie);

return ie;

}

LPIKeyboard RDOPatternKeyboard::createActivity(LPIBaseOperationContainer parent, PTR(RDORuntime) runtime, CREF(tstring) \_oprName)

{

LPIKeyboard keyboard = F(RDOKeyboard)::create(runtime, this, traceable(), \_oprName);

runtime->addRuntimeOperation(parent, keyboard);

return keyboard;

}

В этих методах происходит «привязывание» активностей к дереву БЗ модели.

# Заключение

В рамках данной квалификационной работы бакалавра были получены следующие результаты:

* + 1. Проведено предпроектное исследование системы имитационного моделирования РДО и сформулированы предпосылки создания в системе инструмента для построения иерархической базы знаний модели.
    2. На этапе концептуального проектирования системы проанализированы три варианта решения поставленной задачи и сделан выбор в пользу второго. С помощью диаграммы компонентов нотации UML укрупненно показано внутреннее устройство РДО и выделены те компоненты, которые потребуют внесения изменений в ходе этой работы. Разработана функциональная диаграмма (в нотации IDEF0) компиляции модели, на которой обозначены те функции системы, в которые будут внесены изменения.
    3. На этапе технического проектирования разработан новый синтаксис точек принятия решений, который представлен на синтаксической диаграмме. С помощью диаграммы классов разработана архитектура новой системы. С помощью блок-схемы разработаны алгоритмы, реализующие в системе РДО механизм логического вывода на основе иерархических логик.
    4. На этапе рабочего проектирования написан программный код для реализации спроектированных раннее алгоритмов работы и архитектуры компонентов rdo\_parser и rdo\_runtime системы РДО. Проведены отладка и тестирование новой системы, в ходе которых исправлялись найденные ошибки.
    5. Для демонстрации новых возможностей системы модель, представленная на этапе постановки задачи, была переписана так, чтобы задействовать новый механизм логического вывода РДО. Результаты проведения имитационного исследования позволяют сделать вывод об адекватности нового логического вывода.

Поставленная цель работы достигнута в полном объеме.

# Список использованных источников

* + 1. RAO-Studio – Руководство пользователя, 2007 [[http://rdo.rk9.bmstu.ru/forum/viewtopic.php?t=900](http://rdo.rk9.bmstu.ru/forum/viewforum.php?t=900)].
    2. Справка по языку РДО (в составе программы) [<http://rdo.rk9.bmstu.ru/forum/viewforum.php?f=15>].
    3. Емельянов В.В., Ясиновский С.И. Имитационное моделирование систем: Учеб. пособие. – М.: Изд-во МГТУ им.Н.Э. Баумана, 2009. – 584 с.: ил. (Информатика в техническом университете).
    4. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению. ГОСТ 19.201-78.
    5. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. ГОСТ 19.701-90. Условные обозначения и правила выполнения.
    6. Леоненков. Самоучитель по UML [<http://khpi-iip.mipk.kharkiv.edu/library/case/leon/index.html>].
    7. Бьерн Страуструп. Язык моделирования C++. Специальное издание. Пер. с англ. – М.: ООО «Бином-пресс», 2007 г. – 1104 с.: ил.

# Приложение 1. Полный синтаксический анализ точек принятия решений (rdodpt.y).

// ----------------------------------------------------------------------------

// ---------- DPT

// ----------------------------------------------------------------------------

dpt\_main:

| dpt\_main dpt\_search\_end

| dpt\_main dpt\_some\_end

| dpt\_main dpt\_prior\_end

| dpt\_main dpt\_free\_end

| dpt\_main dpt\_process\_end

| error

{

PARSER->error( @1, "Ожидается описание точки или свободного блока активностей" );

};

// ----------------------------------------------------------------------------

// ---------- DPTSearch

// ----------------------------------------------------------------------------

dpt\_search\_trace: /\* empty \*/

{

$$ = rdoRuntime::RDODPTSearchTrace::DPT\_no\_trace;

}

| RDO\_no\_trace

{

$$ = rdoRuntime::RDODPTSearchTrace::DPT\_no\_trace;

}

| RDO\_trace

{

PARSER->error( @1, "Данный признак трассировки не используется в точке типа search" );

}

| RDO\_trace\_stat

{

$$ = rdoRuntime::RDODPTSearchTrace::DPT\_trace\_stat;

}

| RDO\_trace\_tops

{

$$ = rdoRuntime::RDODPTSearchTrace::DPT\_trace\_tops;

}

| RDO\_trace\_all

{

$$ = rdoRuntime::RDODPTSearchTrace::DPT\_trace\_all;

};

dpt\_search\_parent: /\* empty \*/

{

$$ = 0;

}

| RDO\_Parent RDO\_IDENTIF

{

$$ = $2;

}

| RDO\_Parent error

{

PARSER->error( @1, "Ошибка в имени родительской точки" );

};

dpt\_search\_begin: RDO\_Decision\_point RDO\_IDENTIF\_COLON RDO\_search dpt\_search\_trace dpt\_search\_parent

{

RDOValue\* name = reinterpret\_cast<RDOValue\*>($2);

RDOValue\* parent\_name = reinterpret\_cast<RDOValue\*>($5);

if ( parent\_name != 0 )

{

const RDODPTPrior\* parentDPTPrior = PARSER->findDPTPrior( parent\_name->value().getIdentificator() );

const RDODPTSearch\* parentDPTSearch = PARSER->findDPTSearch( parent\_name->value().getIdentificator() );

const RDODPTSome\* parentDPTSome = PARSER->findDPTSome( parent\_name->value().getIdentificator() );

if ( parentDPTPrior == NULL && parentDPTSearch == NULL && parentDPTSome == NULL )

{

PARSER->error( @1, rdo::format("Не найдена родитеская точка %s", parent\_name->value().getIdentificator().c\_str()) );

}

if ( parentDPTSearch != NULL )

{

PARSER->error( @1, "Точка принятия решений типа search может содержать лишь активности типа rule и не может быть указана в качестве родительской точки" );

}

if ( parentDPTPrior != NULL )

{

LPILogic parent = parentDPTPrior->getLogic();

$$ = (int)new RDODPTSearch( PARSER, name->src\_info(), \*reinterpret\_cast<rdoRuntime::RDODPTSearchTrace::DPT\_TraceFlag\*>(&$4), parent );

}

if ( parentDPTSome != NULL )

{

LPILogic parent = parentDPTSome->getLogic();

$$ = (int)new RDODPTSearch( PARSER, name->src\_info(), \*reinterpret\_cast<rdoRuntime::RDODPTSearchTrace::DPT\_TraceFlag\*>(&$4), parent );

}

}

if ( parent\_name == 0 )

{

$$ = (int)new RDODPTSearch( PARSER, name->src\_info(), \*reinterpret\_cast<rdoRuntime::RDODPTSearchTrace::DPT\_TraceFlag\*>(&$4) );

}

}

| RDO\_Decision\_point RDO\_IDENTIF\_COLON error

{

PARSER->error( @2, @3, "Ожидается тип точки" );

}

| RDO\_Decision\_point RDO\_IDENTIF error

{

PARSER->error( @2, "Ожидается двоеточие" );

}

| RDO\_Decision\_point error

{

PARSER->error( @1, @2, "После ключевого слова $Decision\_point ожидается имя точки" );

};

dpt\_search\_condition: dpt\_search\_begin RDO\_Condition fun\_logic

{

RDODPTSearch\* dpt = reinterpret\_cast<RDODPTSearch\*>($1);

dpt->setCondition((RDOFUNLogic \*)$3);

}

| dpt\_search\_begin RDO\_Condition RDO\_NoCheck

{

RDODPTSearch\* dpt = reinterpret\_cast<RDODPTSearch\*>($1);

dpt->setCondition();

}

| dpt\_search\_begin RDO\_Condition error

{

PARSER->error( @2, @3, "После ключевого слова $Condition ожидается условие начала поиска (начальная вершина)" );

}

| dpt\_search\_begin error

{

PARSER->error( @2, "Ожидается ключевое слово $Condition" );

};

dpt\_search\_prior: dpt\_search\_condition

| dpt\_search\_condition RDO\_Priority fun\_arithm

{

if (!PARSER->getLastDPTSearch()->setPrior( reinterpret\_cast<RDOFUNArithm\*>($3) ))

{

PARSER->error(@3, \_T("Точка принятия решений пока не может иметь приоритет"));

}

}

| dpt\_search\_condition RDO\_Priority error

{

PARSER->error( @1, @2, "Ошибка описания приоритета точки принятия решений" )

}

| dpt\_search\_condition error

{

PARSER->error( @1, @2, "Ожидается ключевое слово $Priority" )

};

dpt\_search\_term: dpt\_search\_prior RDO\_Term\_condition fun\_logic

{

RDODPTSearch\* dpt = reinterpret\_cast<RDODPTSearch\*>($1);

dpt->setTermCondition((RDOFUNLogic \*)$3);

}

| dpt\_search\_prior RDO\_Term\_condition RDO\_NoCheck

{

RDODPTSearch\* dpt = reinterpret\_cast<RDODPTSearch\*>($1);

dpt->setTermCondition();

}

| dpt\_search\_prior RDO\_Term\_condition error

{

PARSER->error( @2, @3, "После ключевого слова $Term\_condition ожидается условие остановки поиска (конечная вершина)" );

}

| dpt\_search\_prior error

{

PARSER->error( @2, "Ожидается ключевое слово $Term\_condition" );

};

dpt\_search\_evaluate: dpt\_search\_term RDO\_Evaluate\_by fun\_arithm

{

RDODPTSearch\* dpt = reinterpret\_cast<RDODPTSearch\*>($1);

dpt->setEvaluateBy((RDOFUNArithm \*)$3);

}

| dpt\_search\_term RDO\_Evaluate\_by error

{

PARSER->error( @2, @3, "После ключевого слова $Evaluate\_by ожидается оценочная функция, например, 0 для поиска в ширину" );

}

| dpt\_search\_term error

{

PARSER->error( @2, "Ожидается ключевое слово $Evaluate\_by" );

};

dp\_searcht\_compare: dpt\_search\_evaluate RDO\_Compare\_tops '=' RDO\_NO

{

RDODPTSearch\* dpt = reinterpret\_cast<RDODPTSearch\*>($1);

dpt->setCompareTops(false);

}

| dpt\_search\_evaluate RDO\_Compare\_tops '=' RDO\_YES

{

RDODPTSearch\* dpt = reinterpret\_cast<RDODPTSearch\*>($1);

dpt->setCompareTops(true);

}

| dpt\_search\_evaluate RDO\_Compare\_tops '=' error

{

PARSER->error( @3, @4, "Ожидается режим запоминания пройденных вершин (YES или NO)" );

}

| dpt\_search\_evaluate RDO\_Compare\_tops error

{

PARSER->error( @2, @3, "Ожидается знак равенства" );

}

| dpt\_search\_evaluate error

{

PARSER->error( @2, "Ожидается ключевое слово $Compare\_tops" );

};

dpt\_search\_descr\_param: /\* empty \*/

| dpt\_search\_descr\_param '\*'

{

PARSER->getLastDPTSearch()->getLastActivity()->addParam( RDOValue(RDOParserSrcInfo(@2, "\*")) )

}

| dpt\_search\_descr\_param RDO\_INT\_CONST

{

PARSER->getLastDPTSearch()->getLastActivity()->addParam( \*reinterpret\_cast<RDOValue\*>($2) )

}

| dpt\_search\_descr\_param RDO\_REAL\_CONST

{

PARSER->getLastDPTSearch()->getLastActivity()->addParam( \*reinterpret\_cast<RDOValue\*>($2) )

}

| dpt\_search\_descr\_param RDO\_BOOL\_CONST

{

PARSER->getLastDPTSearch()->getLastActivity()->addParam( \*reinterpret\_cast<RDOValue\*>($2) )

}

| dpt\_search\_descr\_param RDO\_STRING\_CONST

{

PARSER->getLastDPTSearch()->getLastActivity()->addParam( \*reinterpret\_cast<RDOValue\*>($2) )

}

| dpt\_search\_descr\_param RDO\_IDENTIF

{

PARSER->getLastDPTSearch()->getLastActivity()->addParam( \*reinterpret\_cast<RDOValue\*>($2) )

};

dpt\_search\_descr\_value: RDO\_value\_before fun\_arithm

{

RDODPTSearch\* dpt = PARSER->getLastDPTSearch();

dpt->getLastActivity()->setValue( IDPTSearchActivity::vt\_before, reinterpret\_cast<RDOFUNArithm\*>($2), @1 );

}

| RDO\_value\_after fun\_arithm

{

RDODPTSearch\* dpt = PARSER->getLastDPTSearch();

dpt->getLastActivity()->setValue( IDPTSearchActivity::vt\_after, reinterpret\_cast<RDOFUNArithm\*>($2), @1 );

}

| RDO\_value\_before error

{

PARSER->error( @1, @2, "Ошибка в арифметическом выражении" );

}

| RDO\_value\_after error

{

PARSER->error( @1, @2, "Ошибка в арифметическом выражении" );

};

dpt\_search\_name: RDO\_IDENTIF\_COLON RDO\_IDENTIF

{

RDODPTSearch\* dpt = PARSER->getLastDPTSearch();

RDOValue\* name = reinterpret\_cast<RDOValue\*>($1);

RDOValue\* pattern = reinterpret\_cast<RDOValue\*>($2);

$$ = (int)dpt->addNewActivity( name->src\_info(), pattern->src\_info() );

}

| RDO\_IDENTIF\_COLON error

{

PARSER->error( @1, @2, "Ожидается имя образца" );

}

| RDO\_IDENTIF

{

PARSER->error( @1, "Ожидается ':'" );

}

| error

{

PARSER->error( @1, "Ожидается имя активности" );

};

dpt\_searcht\_activity: /\* empty \*/

| dpt\_searcht\_activity dpt\_search\_name dpt\_search\_descr\_param dpt\_search\_descr\_value

{

RDODPTSearchActivity\* activity = reinterpret\_cast<RDODPTSearchActivity\*>($2);

activity->endParam( @3 );

}

| dpt\_searcht\_activity dpt\_search\_name dpt\_search\_descr\_param error

{

PARSER->error( @3, @4, "Ожидаются ключевые слова value before или value after и стоимость применения правила" );

};

dpt\_search\_header: dp\_searcht\_compare RDO\_Activities dpt\_searcht\_activity

{

}

| dp\_searcht\_compare error

{

PARSER->error( @1, @2, "После режима запоминания пройденных вершин ожидается ключевое слово $Activities" );

};

dpt\_search\_end: dpt\_search\_header RDO\_End

{

RDODPTSearch\* dpt = reinterpret\_cast<RDODPTSearch\*>($1);

dpt->end();

}

| dpt\_search\_header

{

PARSER->error( @1, "Ожидается ключевое слово $End" );

};

// ----------------------------------------------------------------------------

// ---------- DPTSome

// ----------------------------------------------------------------------------

dpt\_some\_trace: /\* empty \*/

{

$$ = 1;

}

| RDO\_no\_trace

{

$$ = 1;

}

| RDO\_trace

{

$$ = 2;

}

| RDO\_trace\_stat

{

PARSER->error( @1, "Данный признак трассировки не используется в точке типа some" );

}

| RDO\_trace\_tops

{

PARSER->error( @1, "Данный признак трассировки не используется в точке типа some" );

}

| RDO\_trace\_all

{

PARSER->error( @1, "Данный признак трассировки не используется в точке типа some" );

};

dpt\_some\_parent: /\* empty \*/

{

$$ = 0;

}

| RDO\_Parent RDO\_IDENTIF

{

$$ = $2;

}

| RDO\_Parent error

{

PARSER->error( @1, "Ошибка в имени родительской точки" );

};

dpt\_some\_begin: RDO\_Decision\_point RDO\_IDENTIF\_COLON RDO\_some dpt\_some\_trace dpt\_some\_parent

{

RDOValue\* name = reinterpret\_cast<RDOValue\*>($2);

RDOValue\* parent\_name = reinterpret\_cast<RDOValue\*>($5);

if ( parent\_name != 0 )

{

const RDODPTPrior\* parentDPTPrior = PARSER->findDPTPrior( parent\_name->value().getIdentificator() );

const RDODPTSearch\* parentDPTSearch = PARSER->findDPTSearch( parent\_name->value().getIdentificator() );

const RDODPTSome\* parentDPTSome = PARSER->findDPTSome( parent\_name->value().getIdentificator() );

if ( parentDPTPrior == NULL && parentDPTSearch == NULL && parentDPTSome == NULL )

{

PARSER->error( @1, rdo::format("Не найдена родитеская точка %s", parent\_name->value().getIdentificator().c\_str()) );

}

if ( parentDPTSearch != NULL )

{

PARSER->error( @5, @1, "Точка принятия решений типа search может содержать лишь активности типа rule и не может быть указана в качестве родительской точки" );

}

if ( parentDPTPrior != NULL )

{

LPILogic parent = parentDPTPrior->getLogic();

$$ = (int)new RDODPTSome( PARSER, name->src\_info(), parent );

}

if ( parentDPTSome != NULL )

{

LPILogic parent = parentDPTSome->getLogic();

$$ = (int)new RDODPTSome( PARSER, name->src\_info(), parent );

}

}

if ( parent\_name == 0 )

{

$$ = (int)new RDODPTSome( PARSER, name->src\_info() );

}

};

dpt\_some\_condition: dpt\_some\_begin RDO\_Condition fun\_logic

{

RDODPTSome\* dpt = reinterpret\_cast<RDODPTSome\*>($1);

dpt->setCondition( reinterpret\_cast<RDOFUNLogic\*>($3) );

}

| dpt\_some\_begin RDO\_Condition RDO\_NoCheck

{

RDODPTSome\* dpt = reinterpret\_cast<RDODPTSome\*>($1);

dpt->setCondition();

}

| dpt\_some\_begin RDO\_Condition error

{

PARSER->error( @2, @3, "После ключевого слова $Condition ожидается условие запуска точки" );

}

| dpt\_some\_begin

{

RDODPTSome\* dpt = reinterpret\_cast<RDODPTSome\*>($1);

dpt->setCondition();

};

dpt\_some\_prior: dpt\_some\_condition

| dpt\_some\_condition RDO\_Priority fun\_arithm

{

if (!PARSER->getLastDPTSome()->setPrior( reinterpret\_cast<RDOFUNArithm\*>($3) ))

{

PARSER->error(@3, \_T("Точка принятия решений пока не может иметь приоритет"));

}

}

| dpt\_some\_condition RDO\_Priority error

{

PARSER->error( @1, @2, "Ошибка описания приоритета точки принятия решений" )

}

| dpt\_some\_condition error

{

PARSER->error( @1, @2, "Ожидается ключевое слово $Priority" )

};

dpt\_some\_name: RDO\_IDENTIF\_COLON RDO\_IDENTIF

{

RDODPTSome\* dpt = PARSER->getLastDPTSome();

RDOValue\* name = reinterpret\_cast<RDOValue\*>($1);

RDOValue\* pattern = reinterpret\_cast<RDOValue\*>($2);

$$ = (int)dpt->addNewActivity( name->src\_info(), pattern->src\_info() );

}

| RDO\_IDENTIF\_COLON error

{

PARSER->error( @1, @2, "Ожидается имя образца" );

};

dpt\_some\_descr\_param: /\* empty \*/

| dpt\_some\_descr\_param '\*'

{

PARSER->getLastDPTSome()->getLastActivity()->addParam( RDOValue(RDOParserSrcInfo(@2, "\*")) )

}

| dpt\_some\_descr\_param RDO\_INT\_CONST

{

PARSER->getLastDPTSome()->getLastActivity()->addParam( \*reinterpret\_cast<RDOValue\*>($2) )

}

| dpt\_some\_descr\_param RDO\_REAL\_CONST

{

PARSER->getLastDPTSome()->getLastActivity()->addParam( \*reinterpret\_cast<RDOValue\*>($2) )

}

| dpt\_some\_descr\_param RDO\_BOOL\_CONST

{

PARSER->getLastDPTSome()->getLastActivity()->addParam( \*reinterpret\_cast<RDOValue\*>($2) )

}

| dpt\_some\_descr\_param RDO\_STRING\_CONST

{

PARSER->getLastDPTSome()->getLastActivity()->addParam( \*reinterpret\_cast<RDOValue\*>($2) )

}

| dpt\_some\_descr\_param RDO\_IDENTIF

{

PARSER->getLastDPTSome()->getLastActivity()->addParam( \*reinterpret\_cast<RDOValue\*>($2) )

}

| dpt\_some\_descr\_param error

{

PARSER->error( @1, @2, "Ошибка описания параметра образца" )

};

dpt\_some\_activity: /\* empty \*/

| dpt\_some\_activity dpt\_some\_name dpt\_some\_descr\_param

{

RDODPTSomeActivity\* activity = reinterpret\_cast<RDODPTSomeActivity\*>($2);

activity->endParam( @3 );

};

dpt\_some\_header: dpt\_some\_prior RDO\_Activities dpt\_some\_activity

{

}

| dpt\_some\_prior error

{

PARSER->error( @1, @2, "Ожидается ключевое слово $Activities" );

};

dpt\_some\_end: dpt\_some\_header RDO\_End

{

RDODPTSome\* dpt = reinterpret\_cast<RDODPTSome\*>($1);

dpt->end();

}

| dpt\_some\_header

{

PARSER->error( @1, "Ожидается ключевое слово $End" );

};

// ----------------------------------------------------------------------------

// ---------- DPT Prior

// ----------------------------------------------------------------------------

dpt\_prior\_trace: /\* empty \*/ {

$$ = 1;

}

| RDO\_no\_trace {

$$ = 1;

}

| RDO\_trace {

$$ = 2;

}

| RDO\_trace\_stat {

PARSER->error( @1, "Данный признак трассировки не используется в точке типа prior" );

}

| RDO\_trace\_tops {

PARSER->error( @1, "Данный признак трассировки не используется в точке типа prior" );

}

| RDO\_trace\_all {

PARSER->error( @1, "Данный признак трассировки не используется в точке типа prior" );

};

dpt\_prior\_parent: /\* empty \*/

{

$$ = 0;

}

| RDO\_Parent RDO\_IDENTIF

{

$$ = $2;

}

| RDO\_Parent error

{

PARSER->error( @1, "Ошибка в имени родительской точки" );

};

dpt\_prior\_begin: RDO\_Decision\_point RDO\_IDENTIF\_COLON RDO\_prior dpt\_prior\_trace dpt\_prior\_parent

{

RDOValue\* name = reinterpret\_cast<RDOValue\*>($2);

RDOValue\* parent\_name = reinterpret\_cast<RDOValue\*>($5);

if ( parent\_name != 0 )

{

const RDODPTPrior\* parentDPTPrior = PARSER->findDPTPrior( parent\_name->value().getIdentificator() );

const RDODPTSearch\* parentDPTSearch = PARSER->findDPTSearch( parent\_name->value().getIdentificator() );

const RDODPTSome\* parentDPTSome = PARSER->findDPTSome( parent\_name->value().getIdentificator() );

if ( parentDPTPrior == NULL && parentDPTSome == NULL && parentDPTSearch == NULL )

{

PARSER->error( @1, rdo::format("Не найдена родитеская точка %s", parent\_name->value().getIdentificator().c\_str()) );

}

if ( parentDPTSearch != NULL )

{

PARSER->error( @5, @1, "Точка принятия решений типа search может содержать лишь активности типа rule и не может быть указана в качестве родительской точки" );

}

if ( parentDPTPrior != NULL )

{

LPILogic parent = parentDPTPrior->getLogic();

$$ = (int)new RDODPTPrior( PARSER, name->src\_info(), parent );

}

if ( parentDPTSome != NULL )

{

LPILogic parent = parentDPTSome->getLogic();

$$ = (int)new RDODPTPrior( PARSER, name->src\_info(), parent );

}

}

if ( parent\_name == 0 )

{

$$ = (int)new RDODPTPrior( PARSER, name->src\_info() );

}

};

dpt\_prior\_condition: dpt\_prior\_begin RDO\_Condition fun\_logic {

RDODPTPrior\* dpt = reinterpret\_cast<RDODPTPrior\*>($1);

dpt->setCondition( reinterpret\_cast<RDOFUNLogic\*>($3) );

}

| dpt\_prior\_begin RDO\_Condition RDO\_NoCheck {

RDODPTPrior\* dpt = reinterpret\_cast<RDODPTPrior\*>($1);

dpt->setCondition();

}

| dpt\_prior\_begin RDO\_Condition error {

PARSER->error( @2, @3, "После ключевого слова $Condition ожидается условие запуска точки" );

}

| dpt\_prior\_begin {

RDODPTPrior\* dpt = reinterpret\_cast<RDODPTPrior\*>($1);

dpt->setCondition();

};

dpt\_prior\_prior: dpt\_prior\_condition

| dpt\_prior\_condition RDO\_Priority fun\_arithm

{

if (!PARSER->getLastDPTPrior()->setPrior( reinterpret\_cast<RDOFUNArithm\*>($3) ))

{

PARSER->error(@3, \_T("Точка принятия решений пока не может иметь приоритет"));

}

}

| dpt\_prior\_condition RDO\_Priority error

{

PARSER->error( @1, @2, "Ошибка описания приоритета точки принятия решений" )

}

| dpt\_some\_condition error

{

PARSER->error( @1, @2, "Ожидается ключевое слово $Priority" )

};

dpt\_prior\_name: RDO\_IDENTIF\_COLON RDO\_IDENTIF {

RDODPTPrior\* dpt = PARSER->getLastDPTPrior();

RDOValue\* name = reinterpret\_cast<RDOValue\*>($1);

RDOValue\* pattern = reinterpret\_cast<RDOValue\*>($2);

$$ = (int)dpt->addNewActivity( name->src\_info(), pattern->src\_info() );

}

| RDO\_IDENTIF\_COLON error {

PARSER->error( @1, @2, "Ожидается имя образца" );

};

dpt\_prior\_descr\_param: /\* empty \*/

| dpt\_prior\_descr\_param '\*' { PARSER->getLastDPTPrior()->getLastActivity()->addParam( RDOValue(RDOParserSrcInfo(@2, "\*")) ) }

| dpt\_prior\_descr\_param RDO\_INT\_CONST { PARSER->getLastDPTPrior()->getLastActivity()->addParam( \*reinterpret\_cast<RDOValue\*>($2) ) }

| dpt\_prior\_descr\_param RDO\_REAL\_CONST { PARSER->getLastDPTPrior()->getLastActivity()->addParam( \*reinterpret\_cast<RDOValue\*>($2) ) }

| dpt\_prior\_descr\_param RDO\_BOOL\_CONST { PARSER->getLastDPTPrior()->getLastActivity()->addParam( \*reinterpret\_cast<RDOValue\*>($2) ) }

| dpt\_prior\_descr\_param RDO\_STRING\_CONST { PARSER->getLastDPTPrior()->getLastActivity()->addParam( \*reinterpret\_cast<RDOValue\*>($2) ) }

| dpt\_prior\_descr\_param RDO\_IDENTIF { PARSER->getLastDPTPrior()->getLastActivity()->addParam( \*reinterpret\_cast<RDOValue\*>($2) ) }

| dpt\_prior\_descr\_param error

{

PARSER->error( @1, @2, "Ошибка описания параметра образца" )

};

dpt\_prior\_activ\_prior: /\* empty \*/

| RDO\_CF '=' fun\_arithm

{

if (!PARSER->getLastDPTPrior()->getLastActivity()->setPrior( reinterpret\_cast<RDOFUNArithm\*>($3) ))

{

PARSER->error(@3, \_T("Активность не может иметь приоритет"));

}

}

| RDO\_CF '=' error

{

PARSER->error( @1, @2, "Ошибка описания приоритета активности" )

}

| RDO\_CF error

{

PARSER->error( @1, @2, "Ошибка: ожидается знак равенства" )

};

dpt\_prior\_activity: /\* empty \*/

| dpt\_prior\_activity dpt\_prior\_name dpt\_prior\_descr\_param dpt\_prior\_activ\_prior{

RDODPTPriorActivity\* activity = reinterpret\_cast<RDODPTPriorActivity\*>($2);

activity->endParam( @3 );

};

dpt\_prior\_header: dpt\_prior\_prior RDO\_Activities dpt\_prior\_activity {

}

| dpt\_prior\_prior error {

PARSER->error( @1, @2, "Ожидается ключевое слово $Activities" );

};

dpt\_prior\_end: dpt\_prior\_header RDO\_End {

RDODPTPrior\* dpt = reinterpret\_cast<RDODPTPrior\*>($1);

dpt->end();

}

| dpt\_prior\_header {

PARSER->error( @1, "Ожидается ключевое слово $End" );

};

// ----------------------------------------------------------------------------

// ---------- DPT Free

// ----------------------------------------------------------------------------

dpt\_free\_prior: dpt\_free\_header

| RDO\_Priority fun\_arithm dpt\_free\_header

{

if (!PARSER->getLastDPTFree()->setPrior( reinterpret\_cast<RDOFUNArithm\*>($2) ))

{

PARSER->error(@3, \_T("Точка принятия решений пока не может иметь приоритет"));

}

}

| RDO\_Priority error dpt\_free\_header

{

PARSER->error( @1, @2, "Ошибка описания приоритета точки принятия решений" )

}

| error dpt\_free\_header

{

PARSER->error( @1, @2, "Ожидается ключевое слово $Priority" )

};

dpt\_free\_header: RDO\_Activities {

$$ = (int)new RDODPTFree( PARSER, @1 );

};

dpt\_free\_activity: /\* empty \*/

| dpt\_free\_activity dpt\_free\_activity\_name dpt\_free\_activity\_param dpt\_free\_activity\_keys {

};

dpt\_free\_activity\_name: RDO\_IDENTIF\_COLON RDO\_IDENTIF {

RDODPTFree\* dpt = PARSER->getLastDPTFree();

RDOValue\* name = reinterpret\_cast<RDOValue\*>($1);

RDOValue\* pattern = reinterpret\_cast<RDOValue\*>($2);

$$ = (int)dpt->addNewActivity( name->src\_info(), pattern->src\_info() );

}

| RDO\_IDENTIF\_COLON error {

PARSER->error( @1, @2, "Ожидается имя образца" );

};

dpt\_free\_activity\_param: /\* empty \*/

| dpt\_free\_activity\_param '\*' { PARSER->getLastDPTFree()->getLastActivity()->addParam( RDOValue(RDOParserSrcInfo(@2, "\*")) ) }

| dpt\_free\_activity\_param RDO\_INT\_CONST { PARSER->getLastDPTFree()->getLastActivity()->addParam( \*reinterpret\_cast<RDOValue\*>($2) ) }

| dpt\_free\_activity\_param RDO\_REAL\_CONST { PARSER->getLastDPTFree()->getLastActivity()->addParam( \*reinterpret\_cast<RDOValue\*>($2) ) }

| dpt\_free\_activity\_param RDO\_BOOL\_CONST { PARSER->getLastDPTFree()->getLastActivity()->addParam( \*reinterpret\_cast<RDOValue\*>($2) ) }

| dpt\_free\_activity\_param RDO\_STRING\_CONST { PARSER->getLastDPTFree()->getLastActivity()->addParam( \*reinterpret\_cast<RDOValue\*>($2) ) }

| dpt\_free\_activity\_param RDO\_IDENTIF { PARSER->getLastDPTFree()->getLastActivity()->addParam( \*reinterpret\_cast<RDOValue\*>($2) ) }

| dpt\_free\_activity\_param error

{

PARSER->error( @1, @2, "Ошибка описания параметра образца" )

};

dpt\_free\_activity\_keys: /\* empty \*/

| dpt\_free\_activity\_keys RDO\_STRING\_CONST {

RDODPTFreeActivity\* activity = PARSER->getLastDPTFree()->getLastActivity();

std::string key = reinterpret\_cast<RDOValue\*>($2)->value().getString();

activity->addHotKey( key, @2 );

}

| dpt\_free\_activity\_keys '+' RDO\_STRING\_CONST {

RDODPTFreeActivity\* activity = PARSER->getLastDPTFree()->getLastActivity();

std::string key = reinterpret\_cast<RDOValue\*>($3)->value().getString();

activity->addHotKey( key, @3 );

};

dpt\_free\_end: dpt\_free\_prior dpt\_free\_activity RDO\_End {

}

| dpt\_free\_header error {

PARSER->error( @1, "Ожидается ключевое слово $End" );

};

# Приложение 2. Код имитационной модели работы почтового отделения связи на языке РДО.

Postoffice.rtp (Типы ресурсов):

**$Resource\_type Отделения : permanent**

**$Parameters**

**Состояние : (Начало\_Дня, Рабочий\_День, Конец\_Дня)**

**$End**

**$Resource\_type Окна : permanent**

**$Parameters**

**Номер : integer[1..5]**

**Работоспособность : (Открыто, Закрыто)**

**Занятость : (Свободен, Занят)**

**Обслужено : integer[0..32530] = 0**

**В\_Очереди : integer[0..32530] = 0**

**Интервал1\_Начало : real**

**Интервал1\_Конец : real**

**Интервал2\_Начало : real**

**Интервал2\_Конец : real**

**Интервал3\_Начало : real**

**Интервал3\_Конец : real**

**$End**

**$Resource\_type Клиенты : temporary**

**$Parameters**

**Заявка : (Rq1,**

**Rq2,**

**Rq3,**

**Rq4,**

**Rq5,**

**Rq6,**

**Rq7,**

**Rq8,**

**Rq9,**

**Rq10,**

**Rq11,**

**Rq12,**

**Rq13,**

**Rq14,**

**Rq15,**

**Rq16,**

**Rq17,**

**Rq18,**

**Rq19)**

**Состояние : (Возник, Пришел, В\_Очереди, Обслуживается, Обслужился, Ушел)**

**Номер\_В\_Очереди : integer[1..32530]**

**Номер\_Окна : such\_as Окна.Номер**

**Приход\_В\_Очередь : real**

**Время\_В\_Очереди : real**

**$End**

**$Resource\_type Изображаемые\_Клиенты : permanent**

**$Parameters**

**Номер : such\_as Окна.Номер**

**Заявка : such\_as Клиенты.Заявка**

**$End**

**$Resource\_type Счетчики : permanent**

**$Parameters**

**Номер : such\_as Окна.Номер**

**Текущий\_Клиент : integer[2..32530]**

**$End**

**$Resource\_type Счетчики\_Заявок : permanent**

**$Parameters**

**Заявка : such\_as Клиенты.Заявка**

**Количество : integer[0..32530]**

**$End**

**$Resource\_type Счетчики\_Времени : permanent**

**$Parameters**

**Номер : such\_as Окна.Номер**

**Время\_На\_Обработку : real**

**$End**

**$Resource\_type Счетчики\_Дней : permanent**

**$Parameters**

**Количество\_Дней : integer**

**$End**

**$Resource\_type Смотрители : permanent**

**$Parameters**

**Номер : such\_as Окна.Номер**

**Разрешить : (Да, Нет)**

**$End**

**$Resource\_type Виды\_Нагрузки : permanent**

**$Parameters**

**Вид : (Постоянный**

**,Линейный\_Увеличение**

**,Линейный\_Уменьшение**

**,Равномерный**

**,Экспоненциальный**

**,Нормальный) = Постоянный**

**Минимум\_Среднее\_Число : real**

**Максимум\_Дисперсия : real**

**Грузить : (Да, Нет)**

**$End**

**$Resource\_type Счетчики\_Часов : permanent**

**$Parameters**

**Часы : real**

**Соответствие : (Да,Нет)**

**Последнее\_Изменение : real**

**$End**

**$Resource\_type Потерянные\_Клиенты : permanent**

**$Parameters**

**Количество : integer**

**$End**

Postoffice.rss (Ресурсы):

**$Resources**

**Окно1 : Окна 1 Закрыто Свободен \* \* 10.0 11.0 12.0 13.0 15.0 18.0**

**Окно2 : Окна 2 Закрыто Свободен \* \* 9.0 18.0 0.0 0.0 0.0 0.0**

**Окно3 : Окна 3 Закрыто Свободен \* \* 9.0 10.0 17.0 18.0 0.0 0.0**

**Окно4 : Окна 4 Закрыто Свободен \* \* 9.0 12.0 0.0 0.0 0.0 0.0**

**Окно5 : Окна 5 Закрыто Свободен \* \* 11.0 16.0 0.0 0.0 0.0 0.0**

**Нагрузка : Виды\_Нагрузки Постоянный 0.016666666667 0.0 Да**

**Из\_Клиент1 : Изображаемые\_Клиенты 1 Rq1**

**Из\_Клиент2 : Изображаемые\_Клиенты 2 Rq1**

**Из\_Клиент3 : Изображаемые\_Клиенты 3 Rq1**

**Из\_Клиент4 : Изображаемые\_Клиенты 4 Rq1**

**Из\_Клиент5 : Изображаемые\_Клиенты 5 Rq1**

**Count1 : Счетчики\_Заявок Rq1 0**

**Count2 : Счетчики\_Заявок Rq2 0**

**Count3 : Счетчики\_Заявок Rq3 0**

**Count4 : Счетчики\_Заявок Rq4 0**

**Count5 : Счетчики\_Заявок Rq5 0**

**Count6 : Счетчики\_Заявок Rq6 0**

**Count7 : Счетчики\_Заявок Rq7 0**

**Count8 : Счетчики\_Заявок Rq8 0**

**Count9 : Счетчики\_Заявок Rq9 0**

**Count10 : Счетчики\_Заявок Rq10 0**

**Count11 : Счетчики\_Заявок Rq11 0**

**Count12 : Счетчики\_Заявок Rq12 0**

**Count13 : Счетчики\_Заявок Rq13 0**

**Count14 : Счетчики\_Заявок Rq14 0**

**Count15 : Счетчики\_Заявок Rq15 0**

**Count16 : Счетчики\_Заявок Rq16 0**

**Count17 : Счетчики\_Заявок Rq17 0**

**Count18 : Счетчики\_Заявок Rq18 0**

**Count19 : Счетчики\_Заявок Rq19 0**

**Счетчик1 : Счетчики 1 2**

**Счетчик2 : Счетчики 2 2**

**Счетчик3 : Счетчики 3 2**

**Счетчик4 : Счетчики 4 2**

**Счетчик5 : Счетчики 5 2**

**Смотритель1 : Смотрители 1 Нет**

**Смотритель2 : Смотрители 2 Нет**

**Смотритель3 : Смотрители 3 Нет**

**Смотритель4 : Смотрители 4 Нет**

**Смотритель5 : Смотрители 5 Нет**

**Отделение : Отделения Начало\_Дня**

**С\_Времени1 : Счетчики\_Времени 1 0.0**

**С\_Времени2 : Счетчики\_Времени 2 0.0**

**С\_Времени3 : Счетчики\_Времени 3 0.0**

**С\_Времени4 : Счетчики\_Времени 4 0.0**

**С\_Времени5 : Счетчики\_Времени 5 0.0**

**Время : Счетчики\_Часов 9.0 Нет 0.0**

**Потерянные : Потерянные\_Клиенты 0**

**С\_Рабочих\_Дней : Счетчики\_Дней 1**

**$End**

Postoffice.fun (Константы, последовательности и функции):

$Constant

Месяц : integer = 25

Квартал : integer = 73

Год : integer = 289

Начало1\_X : integer = 112

Начало\_Y\_R : integer = 55

Начало\_Y\_T : integer = 57

Начало2\_X : integer = 440

Ширина1 : integer = 329

Ширина2 : integer = 185

Высота : integer = 19

Начало\_Текст1 : integer = 119

Начало\_Текст2 : integer = 448

См\_X : integer = 250

См\_Y : integer = 47

См\_Ширина : integer = 329

См\_Высота : integer = 12

Начало\_Рабочего\_Дня : real = 9.0

Время\_Рабочего\_Дня : real = 10.0

Коэф\_Времени : real = 1.0

Переполнение\_Очереди: integer = 10

$End

$Function Заявки\_Окна1 : integer = 0

$Type = list

$Parameters

Заявка : such\_as Клиенты.Заявка

$Body

Rq1 = 1

Rq2 = 1

Rq3 = 1

Rq4 = 1

Rq5 = 1

Rq6 = 1

Rq7 = 1

Rq8 = 1

Rq9 = 1

Rq10 = 1

Rq11 = 1

Rq12 = 1

Rq13 = 1

Rq14 = 1

Rq15 = 1

Rq16 = 1

Rq17 = 1

Rq18 = 1

Rq19 = 1

$End

$Function Заявки\_Окна2 : integer = 0

$Type = list

$Parameters

Заявка : such\_as Клиенты.Заявка

$Body

Rq1 = 2

Rq2 = 2

Rq3 = 2

Rq4 = 2

Rq5 = 2

Rq6 = 2

Rq7 = 2

Rq8 = 2

Rq9 = 2

Rq10 = 2

Rq11 = 2

Rq12 = 2

Rq13 = 2

Rq14 = 2

Rq15 = 2

Rq16 = 2

Rq17 = 2

Rq18 = 2

Rq19 = 2

$End

$Function Заявки\_Окна3 : integer = 0

$Type = list

$Parameters

Заявка : such\_as Клиенты.Заявка

$Body

Rq1 = 3

Rq2 = 3

Rq3 = 3

Rq4 = 3

Rq5 = 3

Rq6 = 3

Rq7 = 3

Rq8 = 3

Rq9 = 3

Rq10 = 3

Rq11 = 3

Rq12 = 3

Rq13 = 3

Rq14 = 3

Rq15 = 3

Rq16 = 3

Rq17 = 3

Rq18 = 3

Rq19 = 3

$End

$Function Заявки\_Окна4 : integer = 0

$Type = list

$Parameters

Заявка : such\_as Клиенты.Заявка

$Body

Rq1 = 4

Rq2 = 4

Rq3 = 4

Rq4 = 4

Rq5 = 4

Rq6 = 4

Rq7 = 4

Rq8 = 4

Rq9 = 4

Rq10 = 4

Rq11 = 4

Rq12 = 4

Rq13 = 4

Rq14 = 4

Rq15 = 4

Rq16 = 4

Rq17 = 4

Rq18 = 4

Rq19 = 4

$End

$Function Заявки\_Окна5 : integer = 0

$Type = list

$Parameters

Заявка : such\_as Клиенты.Заявка

$Body

Rq1 = 5

Rq2 = 5

Rq3 = 5

Rq4 = 5

Rq5 = 5

Rq6 = 5

Rq7 = 5

Rq8 = 5

Rq9 = 5

Rq10 = 5

Rq11 = 5

Rq12 = 5

Rq13 = 5

Rq14 = 5

Rq15 = 5

Rq16 = 5

Rq17 = 5

Rq18 = 5

Rq19 = 5

$End

$Sequence Заявка\_Клиента : such\_as Клиенты.Заявка

$Type = by\_hist

$Body

Rq1 4.0

Rq2 44.0

Rq3 2.0

Rq4 7.0

Rq5 9.0

Rq6 5.0

Rq7 2.0

Rq8 30.0

Rq9 2.0

Rq10 2.0

Rq11 2.0

Rq12 2.0

Rq13 3.0

Rq14 74.0

Rq15 28.0

Rq16 6.0

Rq17 30.0

Rq18 25.0

Rq19 11.0

$End

$Sequence Нормальный\_Закон : real

$Type = normal

$End

$Sequence Равномерный\_Закон : real

$Type = uniform

$End

$Sequence Экспоненциальный\_Закон : real

$Type = exponential

$End

$Function Показать\_Спрятать : such\_as Смотрители.Разрешить

$Type = algorithmic

$Parameters

Сейчас : such\_as Смотрители.Разрешить

Номер : such\_as Окна.Номер

$Body

Calculate\_if Сейчас = Да Показать\_Спрятать = Нет

Calculate\_if Сейчас = Нет and

Номер = 1 and

[Смотритель2.Разрешить = Да or

Смотритель3.Разрешить = Да or

Смотритель4.Разрешить = Да or

Смотритель5.Разрешить = Да ]

Показать\_Спрятать = Нет

Calculate\_if Сейчас = Нет and

Номер = 1 and

[Смотритель2.Разрешить = Нет and

Смотритель3.Разрешить = Нет and

Смотритель4.Разрешить = Нет and

Смотритель5.Разрешить = Нет ]

Показать\_Спрятать = Да

Calculate\_if Сейчас = Нет and

Номер = 2 and

[Смотритель1.Разрешить = Да or

Смотритель3.Разрешить = Да or

Смотритель4.Разрешить = Да or

Смотритель5.Разрешить = Да ]

Показать\_Спрятать = Нет

Calculate\_if Сейчас = Нет and

Номер = 2 and

[Смотритель1.Разрешить = Нет and

Смотритель3.Разрешить = Нет and

Смотритель4.Разрешить = Нет and

Смотритель5.Разрешить = Нет ]

Показать\_Спрятать = Да

Calculate\_if Сейчас = Нет and

Номер = 3 and

[Смотритель1.Разрешить = Да or

Смотритель2.Разрешить = Да or

Смотритель4.Разрешить = Да or

Смотритель5.Разрешить = Да ]

Показать\_Спрятать = Нет

Calculate\_if Сейчас = Нет and

Номер = 3 and

[Смотритель1.Разрешить = Нет and

Смотритель2.Разрешить = Нет and

Смотритель4.Разрешить = Нет and

Смотритель5.Разрешить = Нет ]

Показать\_Спрятать = Да

Calculate\_if Сейчас = Нет and

Номер = 4 and

[Смотритель1.Разрешить = Да or

Смотритель2.Разрешить = Да or

Смотритель3.Разрешить = Да or

Смотритель5.Разрешить = Да ]

Показать\_Спрятать = Нет

Calculate\_if Сейчас = Нет and

Номер = 4 and

[Смотритель1.Разрешить = Нет and

Смотритель2.Разрешить = Нет and

Смотритель3.Разрешить = Нет and

Смотритель5.Разрешить = Нет ]

Показать\_Спрятать = Да

Calculate\_if Сейчас = Нет and

Номер = 5 and

[Смотритель1.Разрешить = Да or

Смотритель2.Разрешить = Да or

Смотритель3.Разрешить = Да or

Смотритель4.Разрешить = Да ]

Показать\_Спрятать = Нет

Calculate\_if Сейчас = Нет and

Номер = 5 and

[Смотритель1.Разрешить = Нет and

Смотритель2.Разрешить = Нет and

Смотритель3.Разрешить = Нет and

Смотритель4.Разрешить = Нет ]

Показать\_Спрятать = Да

$End

$Function Нормальный\_Закон\_Abs : real

$Type = algorithmic

$Parameters

Результат : real

$Body

Calculate\_if Результат > 0 Нормальный\_Закон\_Abs = Результат

Calculate\_if Результат < 0 Нормальный\_Закон\_Abs = -1 \* Результат

Calculate\_if Результат = 0 Нормальный\_Закон\_Abs = 1/60

$End

$Function Время\_Прихода\_Клиента : real

$Type = algorithmic

$Parameters

Вид : such\_as Виды\_Нагрузки.Вид

Число1 : real

Число2 : real

$Body

Calculate\_if Вид = Постоянный

Время\_Прихода\_Клиента = Число1

Calculate\_if Вид = Линейный\_Увеличение and

Время.Часы = 9.0

Время\_Прихода\_Клиента = Число1

Calculate\_if Вид = Линейный\_Увеличение and

Время.Часы = 10.0

Время\_Прихода\_Клиента = Число1 + 1 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Увеличение and

Время.Часы = 11.0

Время\_Прихода\_Клиента = Число1 + 2 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Увеличение and

Время.Часы = 12.0

Время\_Прихода\_Клиента = Число1 + 3 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Увеличение and

Время.Часы = 13.0

Время\_Прихода\_Клиента = Число1 + 4 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Увеличение and

Время.Часы = 14.0

Время\_Прихода\_Клиента = Число1 + 5 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Увеличение and

Время.Часы = 15.0

Время\_Прихода\_Клиента = Число1 + 6 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Увеличение and

Время.Часы = 16.0

Время\_Прихода\_Клиента = Число1 + 7 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Увеличение and

Время.Часы = 17.0

Время\_Прихода\_Клиента = Число1 + 8 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Увеличение and

Время.Часы = 18.0

Время\_Прихода\_Клиента = Число1 + 9 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Увеличение and

Время.Часы = 19.0

Время\_Прихода\_Клиента = Число1 + 10 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Увеличение and

Время.Часы = 20.0

Время\_Прихода\_Клиента = Число1 + 11 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Уменьшение and

Время.Часы = Время\_Рабочего\_Дня + 9.0

Время\_Прихода\_Клиента = Число1

Calculate\_if Вид = Линейный\_Уменьшение and

Время.Часы = 9.0

Время\_Прихода\_Клиента = Число2

Calculate\_if Вид = Линейный\_Уменьшение and

Время.Часы = 10.0

Время\_Прихода\_Клиента = Число2 - 1 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Уменьшение and

Время.Часы = 11.0

Время\_Прихода\_Клиента = Число2 - 2 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Уменьшение and

Время.Часы = 12.0

Время\_Прихода\_Клиента = Число2 - 3 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Уменьшение and

Время.Часы = 13.0

Время\_Прихода\_Клиента = Число2 - 4 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Уменьшение and

Время.Часы = 14.0

Время\_Прихода\_Клиента = Число2 - 5 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Уменьшение and

Время.Часы = 15.0

Время\_Прихода\_Клиента = Число2 - 6 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Уменьшение and

Время.Часы = 16.0

Время\_Прихода\_Клиента = Число2 - 7 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Уменьшение and

Время.Часы = 17.0

Время\_Прихода\_Клиента = Число2 - 8 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Уменьшение and

Время.Часы = 18.0

Время\_Прихода\_Клиента = Число2 - 9 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Уменьшение and

Время.Часы = 19.0

Время\_Прихода\_Клиента = Число2 - 10 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Линейный\_Уменьшение and

Время.Часы = 20.0

Время\_Прихода\_Клиента = Число2 - 11 \* (Число2 - Число1) / (Время\_Рабочего\_Дня - 1)

Calculate\_if Вид = Равномерный

Время\_Прихода\_Клиента = Равномерный\_Закон(Число1,Число2)

Calculate\_if Вид = Нормальный

Время\_Прихода\_Клиента = Нормальный\_Закон\_Abs(Нормальный\_Закон(Число1,Число2))

Calculate\_if Вид = Экспоненциальный

Время\_Прихода\_Клиента = Экспоненциальный\_Закон(Число1)

$End

$Function Время\_Обслуживания : real

$Type = algorithmic

$Parameters

Заявка : such\_as Клиенты.Заявка

$Body

Calculate\_if Заявка = Rq1 Время\_Обслуживания = Равномерный\_Закон(0.0375, 0.045833333333)

Calculate\_if Заявка = Rq2 Время\_Обслуживания = Равномерный\_Закон(0.023333333333, 0.028666666667)

Calculate\_if Заявка = Rq3 Время\_Обслуживания = Равномерный\_Закон(0.06, 0.073333333333)

Calculate\_if Заявка = Rq4 Время\_Обслуживания = Равномерный\_Закон(0.024, 0.029333333333)

Calculate\_if Заявка = Rq5 Время\_Обслуживания = Равномерный\_Закон(0.102, 0.12466666667)

Calculate\_if Заявка = Rq6 Время\_Обслуживания = Равномерный\_Закон(0.039, 0.047666666667)

Calculate\_if Заявка = Rq7 Время\_Обслуживания = Равномерный\_Закон(0.1725, 0.21083333333)

Calculate\_if Заявка = Rq8 Время\_Обслуживания = Равномерный\_Закон(0.0515, 0.062833333333)

Calculate\_if Заявка = Rq9 Время\_Обслуживания = Равномерный\_Закон(0.0225, 0.0275)

Calculate\_if Заявка = Rq10 Время\_Обслуживания = Равномерный\_Закон(0.0225, 0.0275)

Calculate\_if Заявка = Rq11 Время\_Обслуживания = Равномерный\_Закон(0.0975, 0.11916666667)

Calculate\_if Заявка = Rq12 Время\_Обслуживания = Равномерный\_Закон(0.0375, 0.045833333333)

Calculate\_if Заявка = Rq13 Время\_Обслуживания = Равномерный\_Закон(0.015, 0.018333333333)

Calculate\_if Заявка = Rq14 Время\_Обслуживания = Равномерный\_Закон(0.019666666667, 0.024)

Calculate\_if Заявка = Rq15 Время\_Обслуживания = Равномерный\_Закон(0.0165, 0.020166666667)

Calculate\_if Заявка = Rq16 Время\_Обслуживания = Равномерный\_Закон(0.035, 0.042666666667)

Calculate\_if Заявка = Rq17 Время\_Обслуживания = Равномерный\_Закон(0.020333333333, 0.024833333333)

Calculate\_if Заявка = Rq18 Время\_Обслуживания = Равномерный\_Закон(0.0275, 0.0335)

Calculate\_if Заявка = Rq19 Время\_Обслуживания = Равномерный\_Закон(0.036, 0.044)

$End

$Function ВремечкО : real = 0.0

$Type = list

$Parameters

Заявка : such\_as Клиенты.Заявка

$Body

Rq1 = 0.033333333333

Rq2 = 0.033333333333

Rq3 = 0.016666666667

Rq4 = 0.033333333333

Rq5 = 0.021666666667

Rq6 = 0.145

Rq7 = 0.03

Rq8 = 0.0016666666667

Rq9 = 0.145

Rq10 = 0.033333333333

Rq11 = 0.033333333333

Rq12 = 0.03

Rq13 = 0.033333333333

Rq14 = 0.033333333333

Rq15 = 0.033333333333

Rq16 = 0.033333333333

Rq17 = 0.033333333333

Rq18 = 0.033333333333

Rq19 = 0.033333333333

$End

Postoffice.pat (Образцы):

**$Pattern** Приход\_Клиента\_На\_Почту : **operation**

**$Relevant\_resources**

Рел\_Отделение : Отделения **NoChange** **NoChange**

Рел\_Нагрузка : Виды\_Нагрузки **Keep** **Keep**

Рел\_Клиент : Клиенты **Create** **NoChange**

**$Time** = Время\_Прихода\_Клиента(Рел\_Нагрузка.Вид,

Рел\_Нагрузка.Минимум\_Среднее\_Число,

Рел\_Нагрузка.Максимум\_Дисперсия)

**$Body**

Рел\_Отделение

**Choice** **from** Рел\_Отделение.Состояние = Рабочий\_День

**first**

Рел\_Нагрузка

**Choice** **from** Рел\_Нагрузка.Грузить = Да

**first**

**Convert\_begin**

Грузить **set** Нет

**Convert\_end**

Грузить **set** Да

Рел\_Клиент

**Convert\_begin**

Заявка **set** Заявка\_Клиента

Состояние **set** Возник

Номер\_В\_Очереди **set** 1

Номер\_Окна **set** 1

Приход\_В\_Очередь **set** **Time\_now**

Время\_В\_Очереди **set** 0.0

**$End**

**$Pattern** Определить\_Окно : **rule**

**$Relevant\_resources**

Рел\_Клиент : Клиенты **Keep**

Рел\_Окно : Окна **NoChange**

**$Body**

Рел\_Клиент

**Choice** **from** Рел\_Клиент.Состояние = Возник

**first**

**Convert\_rule**

Состояние **set** Пришел

Номер\_Окна **set** Рел\_Окно.Номер

Рел\_Окно

**Choice** **from** [Рел\_Окно.Номер = Заявки\_Окна1(Рел\_Клиент.Заявка) **and** Окно1.Работоспособность = Открыто] **or**

[Рел\_Окно.Номер = Заявки\_Окна2(Рел\_Клиент.Заявка) **and** Окно2.Работоспособность = Открыто] **or**

[Рел\_Окно.Номер = Заявки\_Окна3(Рел\_Клиент.Заявка) **and** Окно3.Работоспособность = Открыто] **or**

[Рел\_Окно.Номер = Заявки\_Окна4(Рел\_Клиент.Заявка) **and** Окно4.Работоспособность = Открыто] **or**

[Рел\_Окно.Номер = Заявки\_Окна5(Рел\_Клиент.Заявка) **and** Окно5.Работоспособность = Открыто]

**with\_min**(Рел\_Окно.В\_Очереди)

**$End**

**$Pattern** Ликвидирование : **rule**

**$Relevant\_resources**

Рел\_Клиент : Клиенты **Erase**

Рел\_Потерянный\_Клиент : Потерянные\_Клиенты **Keep**

**$Body**

Рел\_Клиент

**Choice** **from** Рел\_Клиент.Состояние = Возник

**first**

Рел\_Потерянный\_Клиент

**Choice** **NoCheck**

**first**

**Convert\_rule**

Количество **set** Рел\_Потерянный\_Клиент.Количество + 1

**$End**

**$Pattern** Открыть\_Окно : **rule**

**$Relevant\_resources**

Рел\_Время : Время **NoChange**

Рел\_Окно : Окна **Keep**

**$Body**

Рел\_Время

**Choice** **NoCheck**

**first**

Рел\_Окно

**Choice** **from** Рел\_Окно.Работоспособность = Закрыто **and**

(Рел\_Окно.Интервал1\_Начало = Рел\_Время.Часы **or**

Рел\_Окно.Интервал2\_Начало = Рел\_Время.Часы **or**

Рел\_Окно.Интервал3\_Начало = Рел\_Время.Часы)

**first**

**Convert\_rule**

Работоспособность **set** Открыто

**$End**

**$Pattern** Закрыть\_Окно : **rule**

**$Relevant\_resources**

Рел\_Время : Время **NoChange**

Рел\_Окно : Окна **Keep**

**$Body**

Рел\_Время

**Choice** **NoCheck**

**first**

Рел\_Окно

**Choice** **from** Рел\_Окно.Работоспособность = Открыто **and**

(Рел\_Окно.Интервал1\_Конец = Рел\_Время.Часы **or**

Рел\_Окно.Интервал2\_Конец = Рел\_Время.Часы **or**

Рел\_Окно.Интервал3\_Конец = Рел\_Время.Часы)

**first**

**Convert\_rule**

Работоспособность **set** Закрыто

**$End**

**$Pattern** Приход\_Клиента\_В\_Очередь : **rule**

**$Relevant\_resources**

Рел\_Клиент : Клиенты **Keep**

Рел\_Окно : Окна **Keep**

**$Body**

Рел\_Клиент

**Choice** **from** Рел\_Клиент.Состояние = Пришел

**first**

**Convert\_rule**

Состояние **set** В\_Очереди

Номер\_В\_Очереди **set** Рел\_Окно.В\_Очереди + 1

Приход\_В\_Очередь **set** **Time\_now**

Рел\_Окно

**Choice** **from** Рел\_Окно.Номер = Рел\_Клиент.Номер\_Окна

**first**

**Convert\_rule**

В\_Очереди **set** Рел\_Окно.В\_Очереди + 1

**$End**

**$Pattern** Сдвинуть\_Очередь : **rule**

**$Parameters**

Номер : **such\_as** Окна.Номер

**$Relevant\_resources**

Рел\_Счетчик : Счетчики **Keep**

Рел\_Клиент : Клиенты **Keep**

**$Body**

Рел\_Счетчик

**Choice** **from** Рел\_Счетчик.Номер = Номер

**first**

**Convert\_rule**

Текущий\_Клиент **set** Рел\_Счетчик.Текущий\_Клиент + 1

Рел\_Клиент

**Choice** **from** Рел\_Клиент.Номер\_В\_Очереди = Рел\_Счетчик.Текущий\_Клиент **and**

Рел\_Клиент.Номер\_Окна = Номер

**first**

**Convert\_rule**

Номер\_В\_Очереди **set** Рел\_Клиент.Номер\_В\_Очереди - 1

**$End**

**$Pattern** Функционирование : **operation**

**$Relevant\_resources**

Рел\_Отделение : Отделения **Keep** **Keep**

**$Time** = Время\_Рабочего\_Дня

**$Body**

Рел\_Отделение

**Choice** **from** Рел\_Отделение.Состояние = Начало\_Дня

**first**

**Convert\_begin**

Состояние **set** Рабочий\_День

**Convert\_end**

Состояние **set** Конец\_Дня

**$End**

**$Pattern** Обслуживание\_Первого : **operation**

**$Parameters**

Номер : **such\_as** Окна.Номер

**$Relevant\_resources**

Рел\_Отделение : Отделения **NoChange** **NoChange**

Рел\_Окно : Окна **Keep** **Keep**

Рел\_Клиент : Клиенты **Keep** **Keep**

Рел\_Счетчик : Счетчики **Keep** **NoChange**

Рел\_Счетчик\_Заявок : Счетчики\_Заявок **NoChange** **Keep**

Рел\_Счетчик\_Времени : Счетчики\_Времени **NoChange** **Keep**

Рел\_Из\_Клиент : Изображаемые\_Клиенты **Keep** **NoChange**

**$Time** = Коэф\_Времени \* Время\_Обслуживания(Рел\_Клиент.Заявка)

**$Body**

Рел\_Отделение

**Choice** **from** Рел\_Отделение.Состояние = Рабочий\_День

**first**

Рел\_Окно

**Choice** **from** Рел\_Окно.Номер = Номер **and**

Рел\_Окно.Занятость = Свободен

**first**

**Convert\_begin**

Занятость **set** Занят

**Convert\_end**

Занятость **set** Свободен

Обслужено **set** Рел\_Окно.Обслужено + 1

В\_Очереди **set** Рел\_Окно.В\_Очереди - 1

Рел\_Клиент

**Choice** **from** Рел\_Клиент.Состояние = В\_Очереди **and**

Рел\_Клиент.Номер\_Окна = Номер **and**

Рел\_Окно.Занятость = Свободен **and**

Рел\_Клиент.Номер\_В\_Очереди = 1

**first**

**Convert\_begin**

Состояние **set** Обслуживается

Время\_В\_Очереди **set** **Time\_now** - Рел\_Клиент.Приход\_В\_Очередь

**Convert\_end**

Состояние **set** Ушел

Рел\_Счетчик

**Choice** **from** Рел\_Счетчик.Номер = Номер **and**

Рел\_Отделение.Состояние = Рабочий\_День

**first**

**Convert\_begin**

Текущий\_Клиент **set** 2

Рел\_Счетчик\_Заявок

**Choice** **from** Рел\_Счетчик\_Заявок.Заявка = Рел\_Клиент.Заявка

**first**

**Convert\_end**

Количество **set** Рел\_Счетчик\_Заявок.Количество + 1

Рел\_Счетчик\_Времени

**Choice** **from** Рел\_Счетчик\_Времени.Номер = Номер

**first**

**Convert\_end**

Время\_На\_Обработку **set** Рел\_Счетчик\_Времени.Время\_На\_Обработку +

ВремечкО(Рел\_Клиент.Заявка)

Рел\_Из\_Клиент

**Choice** **from** Рел\_Из\_Клиент.Номер = Номер

**first**

**Convert\_begin**

Заявка **set** Рел\_Клиент.Заявка

**$End**

**$Pattern** Уничтожение : **rule**

**$Parameters**

Номер : **such\_as** Окна.Номер

**$Relevant\_resources**

Рел\_Отделение : Отделения **NoChange**

Рел\_Окно : Окна **Keep**

Рел\_Клиент : Клиенты **Erase**

**$Body**

Рел\_Отделение

**Choice** **from** Рел\_Отделение.Состояние = Конец\_Дня

**first**

Рел\_Окно

**Choice** **from** Рел\_Окно.Номер = Номер **and**

Рел\_Окно.Занятость = Свободен **and**

Рел\_Окно.В\_Очереди > 0

**first**

**Convert\_rule**

В\_Очереди **set** Рел\_Окно.В\_Очереди - 1

Рел\_Клиент

**Choice** **from** Рел\_Клиент.Состояние = В\_Очереди **and**

Рел\_Клиент.Номер\_Окна = Номер

**first**

**$End**

**$Pattern** Переход\_К\_Новому\_Дню : **rule**

**$Relevant\_resources**

Рел\_Отделение : Отделения **Keep**

Рел\_Счетчик\_Дней : Счетчики\_Дней **Keep**

Рел\_Счетчик\_Времени1 : Счетчики\_Времени **Keep**

Рел\_Счетчик\_Времени2 : Счетчики\_Времени **Keep**

Рел\_Счетчик\_Времени3 : Счетчики\_Времени **Keep**

Рел\_Счетчик\_Времени4 : Счетчики\_Времени **Keep**

Рел\_Счетчик\_Времени5 : Счетчики\_Времени **Keep**

Рел\_Время : Время **Keep**

**$Body**

Рел\_Отделение

**Choice** **from** Окно1.В\_Очереди + Окно2.В\_Очереди + Окно3.В\_Очереди +

Окно4.В\_Очереди + Окно5.В\_Очереди = 0 **and**

Рел\_Отделение.Состояние = Конец\_Дня

**first**

**Convert\_rule**

Состояние **set** Начало\_Дня

Рел\_Счетчик\_Дней

**Choice** **NoCheck**

**first**

**Convert\_rule**

Количество\_Дней **set** Рел\_Счетчик\_Дней.Количество\_Дней + 1

Рел\_Счетчик\_Времени1

**Choice** **from** Рел\_Счетчик\_Времени1.Номер = 1

**first**

**Convert\_rule**

Время\_На\_Обработку **set** 0.0

Рел\_Счетчик\_Времени2

**Choice** **from** Рел\_Счетчик\_Времени2.Номер = 2

**first**

**Convert\_rule**

Время\_На\_Обработку **set** 0.0

Рел\_Счетчик\_Времени3

**Choice** **from** Рел\_Счетчик\_Времени3.Номер = 3

**first**

**Convert\_rule**

Время\_На\_Обработку **set** 0.0

Рел\_Счетчик\_Времени4

**Choice** **from** Рел\_Счетчик\_Времени4.Номер = 4

**first**

**Convert\_rule**

Время\_На\_Обработку **set** 0.0

Рел\_Счетчик\_Времени5

**Choice** **from** Рел\_Счетчик\_Времени5.Номер = 5

**first**

**Convert\_rule**

Время\_На\_Обработку **set** 0.0

Рел\_Время

**Choice** **NoCheck**

**first**

**Convert\_rule**

Часы **set** 9.0

Последнее\_Изменение **set** **Time\_now**

**$End**

**$Pattern** Уход\_Клиента : **rule**

**$Parameters**

Номер : **such\_as** Окна.Номер

**$Relevant\_resources**

Рел\_Окно : Окна **NoChange**

Рел\_Клиент : Клиенты **Erase**

**$Body**

Рел\_Окно

**Choice** **from** Рел\_Окно.Номер = Номер

**first**

Рел\_Клиент

**Choice** **from** Рел\_Клиент.Состояние = Ушел

**first**

**$End**

**$Pattern** Обслуживание : **keyboard**

**$Parameters**

Номер : **such\_as** Окна.Номер

**$Relevant\_resources**

Рел\_Смотритель : Смотрители **Keep** **NoChange**

**$Time** = 0.0

**$Body**

Рел\_Смотритель

**Choice** **from** Рел\_Смотритель.Номер = Номер

**first**

**Convert\_begin**

Разрешить **set** Показать\_Спрятать(Рел\_Смотритель.Разрешить,Номер)

**$End**

**$Pattern** Соответствие\_Времени : **operation**

**$Relevant\_resources**

Рел\_Время : Время **Keep** **Keep**

**$Time** = 1.0

**$Body**

Рел\_Время

**Choice** **from** Рел\_Время.Соответствие = Нет **and**

(**Time\_now** - Рел\_Время.Последнее\_Изменение) >= 1.0

**first**

**Convert\_begin**

Часы **set** Рел\_Время.Часы + 1

Соответствие **set** Да

Последнее\_Изменение **set** **Time\_now**

**Convert\_end**

Соответствие **set** Нет

**$End**

Postoffice.dpt (Точки принятия решений):

**$Decision\_point** main : **some**

**$Activities**

**$End**

**$Decision\_point** рабочий\_день : **some** **trace**

$Parent main

**$Condition** Отделение.Состояние = Рабочий\_День

**$Activities**

**$End**

**$Decision\_point** приход\_клиентов : **some**

$Parent рабочий\_день

**$Activities**

Еще\_Один\_Пришел : Приход\_Клиента\_На\_Почту

**$End**

**$Decision\_point** размещение\_клиентов : **some**

$Parent рабочий\_день

**$Activities**

Какое\_Окно : Определить\_Окно

Потерянного\_Вон : Ликвидирование

В\_Очередь\_Его : Приход\_Клиента\_В\_Очередь

**$End**

**$Decision\_point** обслуживание\_клиентов : **some**

$Parent рабочий\_день

**$Activities**

Наконец\_То\_В\_Окно1 : Обслуживание\_Первого 1

Наконец\_То\_В\_Окно2 : Обслуживание\_Первого 2

Наконец\_То\_В\_Окно3 : Обслуживание\_Первого 3

Наконец\_То\_В\_Окно4 : Обслуживание\_Первого 4

Наконец\_То\_В\_Окно5 : Обслуживание\_Первого 5

**$End**

**$Decision\_point** продвижение\_очередей : **some**

$Parent рабочий\_день

**$Activities**

Вперед1 : Сдвинуть\_Очередь 1

Вперед2 : Сдвинуть\_Очередь 2

Вперед3 : Сдвинуть\_Очередь 3

Вперед4 : Сдвинуть\_Очередь 4

Вперед5 : Сдвинуть\_Очередь 5

**$End**

**$Decision\_point** уход\_клиентов : **some**

$Parent рабочий\_день

**$Activities**

От\_Окна1 : Уход\_Клиента 1

От\_Окна2 : Уход\_Клиента 2

От\_Окна3 : Уход\_Клиента 3

От\_Окна4 : Уход\_Клиента 4

От\_Окна5 : Уход\_Клиента 5

**$End**

**$Decision\_point** открытие\_закрытие\_окна : **some**

$Parent рабочий\_день

**$Activities**

Я\_Работаю : Открыть\_Окно

Я\_Закрылось : Закрыть\_Окно

**$End**

**$Decision\_point** конец\_дня : **some**

$Parent main

**$Condition** Отделение.Состояние = Конец\_Дня

**$Activities**

Поехали\_Снова : Переход\_К\_Новому\_Дню

**$End**

**$Decision\_point** уход\_клиентов\_в\_конце\_дня : **some**

$Parent конец\_дня

**$Activities**

Лишних\_Вон1 : Уничтожение 1

Лишних\_Вон2 : Уничтожение 2

Лишних\_Вон3 : Уничтожение 3

Лишних\_Вон4 : Уничтожение 4

Лишних\_Вон5 : Уничтожение 5

**$End**

**$Decision\_point** служебные : **some**

$Parent main

**$Activities**

Часы\_На\_Экран : Соответствие\_Времени

Поехали : Функционирование

**$End**

Postoffice.frm (Анимация):

**$Frame** Зал

**$Back\_picture** = <255 255 255> post

**Show**

**bitmap**[318,398,w\_day,w\_day\_m]

**text**[555,405,150,12,**transparent**,<0 255 0>,С\_Рабочих\_Дней.Количество\_Дней]

**Show\_if** Окно1.Работоспособность = Открыто

**bitmap**[138,120,k1,k\_m]

**bitmap**[140,160,oper,oper\_m]

**Show\_if** Окно1.Работоспособность = Закрыто **and**

Окно1.В\_Очереди >= 1

**bitmap**[138,120,nk1,k\_m]

**bitmap**[140,160,oper,oper\_m]

**Show\_if** Окно1.Работоспособность = Закрыто **and**

Окно1.В\_Очереди = 0

**bitmap**[138,120,nk1,k\_m]

**Show\_if** Окно2.Работоспособность = Открыто

**bitmap**[243,120,k2,k\_m]

**bitmap**[245,160,oper,oper\_m]

**Show\_if** Окно2.Работоспособность = Закрыто **and**

Окно2.В\_Очереди >= 1

**bitmap**[243,120,nk2,k\_m]

**bitmap**[245,160,oper,oper\_m]

**Show\_if** Окно2.Работоспособность = Закрыто **and**

Окно2.В\_Очереди = 0

**bitmap**[243,120,nk2,k\_m]

**Show\_if** Окно3.Работоспособность = Открыто

**bitmap**[348,120,k3,k\_m]

**bitmap**[350,160,oper,oper\_m]

**Show\_if** Окно3.Работоспособность = Закрыто **and**

Окно3.В\_Очереди >= 1

**bitmap**[348,120,nk3,k\_m]

**bitmap**[350,160,oper,oper\_m]

**Show\_if** Окно3.Работоспособность = Закрыто **and**

Окно3.В\_Очереди = 0

**bitmap**[348,120,nk3,k\_m]

**Show\_if** Окно4.Работоспособность = Открыто

**bitmap**[453,120,k4,k\_m]

**bitmap**[455,160,oper,oper\_m]

**Show\_if** Окно4.Работоспособность = Закрыто **and**

Окно4.В\_Очереди >= 1

**bitmap**[453,120,nk4,k\_m]

**bitmap**[455,160,oper,oper\_m]

**Show\_if** Окно4.Работоспособность = Закрыто **and**

Окно4.В\_Очереди = 0

**bitmap**[453,120,nk4,k\_m]

**Show\_if** Окно5.Работоспособность = Открыто

**bitmap**[559,120,k5,k\_m]

**bitmap**[561,160,oper,oper\_m]

**Show\_if** Окно5.Работоспособность = Закрыто **and**

Окно5.В\_Очереди >= 1

**bitmap**[559,120,nk5,k\_m]

**bitmap**[561,160,oper,oper\_m]

**Show\_if** Окно5.Работоспособность = Закрыто **and**

Окно5.В\_Очереди = 0

**bitmap**[559,120,nk5,k\_m]

**Show\_if** Окно1.В\_Очереди >= 1

**bitmap**[148,157,others,others\_m]

**Show\_if** Окно1.В\_Очереди >= 2

**bitmap**[171,157,second,second\_m]

**Show\_if** Окно1.В\_Очереди >= 3

**bitmap**[195,160,others,others\_m]

**Show\_if** Окно1.В\_Очереди >= 4

**bitmap**[195,168,others,others\_m]

**Show\_if** Окно1.В\_Очереди >= 5

**bitmap**[195,176,others,others\_m]

**Show\_if** Окно1.В\_Очереди >= 6

**bitmap**[195,184,others,others\_m]

**Show\_if** Окно1.В\_Очереди >= 7

**bitmap**[195,192,others,others\_m]

**Show\_if** Окно1.В\_Очереди >= 8

**bitmap**[195,200,others,others\_m]

**Show\_if** Окно1.В\_Очереди >= 9

**bitmap**[195,208,others,others\_m]

**Show\_if** Окно1.В\_Очереди >= 10

**bitmap**[195,216,others,others\_m]

**Show\_if** Окно2.В\_Очереди >= 1

**bitmap**[255,157,others,others\_m]

**Show\_if** Окно2.В\_Очереди >= 2

**bitmap**[278,157,second,second\_m]

**Show\_if** Окно2.В\_Очереди >= 3

**bitmap**[302,160,others,others\_m]

**Show\_if** Окно2.В\_Очереди >= 4

**bitmap**[302,168,others,others\_m]

**Show\_if** Окно2.В\_Очереди >= 5

**bitmap**[302,176,others,others\_m]

**Show\_if** Окно2.В\_Очереди >= 6

**bitmap**[302,184,others,others\_m]

**Show\_if** Окно2.В\_Очереди >= 7

**bitmap**[302,192,others,others\_m]

**Show\_if** Окно2.В\_Очереди >= 8

**bitmap**[302,200,others,others\_m]

**Show\_if** Окно2.В\_Очереди >= 9

**bitmap**[302,208,others,others\_m]

**Show\_if** Окно2.В\_Очереди >= 10

**bitmap**[302,216,others,others\_m]

**Show\_if** Окно3.В\_Очереди >= 1

**bitmap**[365,157,others,others\_m]

**Show\_if** Окно3.В\_Очереди >= 2

**bitmap**[388,157,second,second\_m]

**Show\_if** Окно3.В\_Очереди >= 3

**bitmap**[412,160,others,others\_m]

**Show\_if** Окно3.В\_Очереди >= 4

**bitmap**[412,168,others,others\_m]

**Show\_if** Окно3.В\_Очереди >= 5

**bitmap**[412,176,others,others\_m]

**Show\_if** Окно3.В\_Очереди >= 6

**bitmap**[412,184,others,others\_m]

**Show\_if** Окно3.В\_Очереди >= 7

**bitmap**[412,192,others,others\_m]

**Show\_if** Окно3.В\_Очереди >= 8

**bitmap**[412,200,others,others\_m]

**Show\_if** Окно3.В\_Очереди >= 9

**bitmap**[412,208,others,others\_m]

**Show\_if** Окно3.В\_Очереди >= 10

**bitmap**[412,216,others,others\_m]

**Show\_if** Окно4.В\_Очереди >= 1

**bitmap**[465,157,others,others\_m]

**Show\_if** Окно4.В\_Очереди >= 2

**bitmap**[488,157,second,second\_m]

**Show\_if** Окно4.В\_Очереди >= 3

**bitmap**[512,160,others,others\_m]

**Show\_if** Окно4.В\_Очереди >= 4

**bitmap**[512,168,others,others\_m]

**Show\_if** Окно4.В\_Очереди >= 5

**bitmap**[512,176,others,others\_m]

**Show\_if** Окно4.В\_Очереди >= 6

**bitmap**[512,184,others,others\_m]

**Show\_if** Окно4.В\_Очереди >= 7

**bitmap**[512,192,others,others\_m]

**Show\_if** Окно4.В\_Очереди >= 8

**bitmap**[512,200,others,others\_m]

**Show\_if** Окно4.В\_Очереди >= 9

**bitmap**[512,208,others,others\_m]

**Show\_if** Окно4.В\_Очереди >= 10

**bitmap**[512,216,others,others\_m]

**Show\_if** Окно5.В\_Очереди >= 1

**bitmap**[565,157,others,others\_m]

**Show\_if** Окно5.В\_Очереди >= 2

**bitmap**[588,157,second,second\_m]

**Show\_if** Окно5.В\_Очереди >= 3

**bitmap**[612,160,others,others\_m]

**Show\_if** Окно5.В\_Очереди >= 4

**bitmap**[612,168,others,others\_m]

**Show\_if** Окно5.В\_Очереди >= 5

**bitmap**[612,176,others,others\_m]

**Show\_if** Окно5.В\_Очереди >= 6

**bitmap**[612,184,others,others\_m]

**Show\_if** Окно5.В\_Очереди >= 7

**bitmap**[612,192,others,others\_m]

**Show\_if** Окно5.В\_Очереди >= 8

**bitmap**[612,200,others,others\_m]

**Show\_if** Окно5.В\_Очереди >= 9

**bitmap**[612,208,others,others\_m]

**Show\_if** Окно5.В\_Очереди >= 10

**bitmap**[612,216,others,others\_m]

**Show\_if** [Окно1.В\_Очереди >= Переполнение\_Очереди **or**

Окно2.В\_Очереди >= Переполнение\_Очереди **or**

Окно3.В\_Очереди >= Переполнение\_Очереди **or**

Окно4.В\_Очереди >= Переполнение\_Очереди **or**

Окно5.В\_Очереди >= Переполнение\_Очереди] **and**

**Frac**(**Floor**(**Seconds**) / 2) = 0

**bitmap**[318,426,ocher,ocher\_m]

**Show\_if** [Окно1.В\_Очереди >= Переполнение\_Очереди **or**

Окно2.В\_Очереди >= Переполнение\_Очереди **or**

Окно3.В\_Очереди >= Переполнение\_Очереди **or**

Окно4.В\_Очереди >= Переполнение\_Очереди **or**

Окно5.В\_Очереди >= Переполнение\_Очереди] **and**

**Frac**(**Floor**(**Seconds**) / 2) <> 0

**bitmap**[318,426,ocher\_1,ocher\_m]

**Show\_if** Время.Часы = 9.0

**bitmap**[601,398,c9,c0\_m]

**Show\_if** Время.Часы = 10.0

**bitmap**[601,398,c10,c0\_m]

**Show\_if** Время.Часы = 11.0

**bitmap**[601,398,c11,c0\_m]

**Show\_if** Время.Часы = 12.0

**bitmap**[601,398,c12,c0\_m]

**Show\_if** Время.Часы = 13.0

**bitmap**[601,398,c13,c0\_m]

**Show\_if** Время.Часы = 14.0

**bitmap**[601,398,c14,c0\_m]

**Show\_if** Время.Часы = 15.0

**bitmap**[601,398,c15,c0\_m]

**Show\_if** Время.Часы = 16.0

**bitmap**[601,398,c16,c0\_m]

**Show\_if** Время.Часы = 17.0

**bitmap**[601,398,c17,c0\_m]

**Show\_if** Время.Часы = 18.0

**bitmap**[601,398,c18,c0\_m]

**Show\_if** Время.Часы = 19.0

**bitmap**[601,398,c19,c0\_m]

**Show\_if** Время.Часы = 20.0

**bitmap**[601,398,c20,c0\_m]

**$End**

**$Frame** Окошки

**$Back\_picture** = <255 255 255> post\_1

**Show\_if** Окно1.Работоспособность = Открыто

**bitmap**[7,79,f1,f1\_m]

**bitmap**[109,75,oper,oper\_m]

**Show\_if** Окно1.Работоспособность = Закрыто **and**

Окно1.В\_Очереди >= 1

**bitmap**[7,79,nf1,nf1\_m]

**bitmap**[109,75,oper,oper\_m]

**Show\_if** Окно1.Работоспособность = Закрыто **and**

Окно1.В\_Очереди = 0

**bitmap**[7,79,nf1,nf1\_m]

**Show\_if** Окно2.Работоспособность = Открыто

**bitmap**[7,164,f2,f2\_m]

**bitmap**[109,160,oper,oper\_m]

**Show\_if** Окно2.Работоспособность = Закрыто **and**

Окно2.В\_Очереди >= 1

**bitmap**[7,164,nf2,nf2\_m]

**bitmap**[109,160,oper,oper\_m]

**Show\_if** Окно2.Работоспособность = Закрыто **and**

Окно2.В\_Очереди = 0

**bitmap**[7,164,nf2,nf2\_m]

**Show\_if** Окно3.Работоспособность = Открыто

**bitmap**[7,249,f3,f3\_m]

**bitmap**[109,246,oper,oper\_m]

**Show\_if** Окно3.Работоспособность = Закрыто **and**

Окно3.В\_Очереди >= 1

**bitmap**[7,249,nf3,nf3\_m]

**bitmap**[109,246,oper,oper\_m]

**Show\_if** Окно3.Работоспособность = Закрыто **and**

Окно3.В\_Очереди = 0

**bitmap**[7,249,nf3,nf3\_m]

**Show\_if** Окно4.Работоспособность = Открыто

**bitmap**[7,334,f4,f4\_m]

**bitmap**[109,332,oper,oper\_m]

**Show\_if** Окно4.Работоспособность = Закрыто **and**

Окно4.В\_Очереди >= 1

**bitmap**[7,334,nf4,nf4\_m]

**bitmap**[109,332,oper,oper\_m]

**Show\_if** Окно4.Работоспособность = Закрыто **and**

Окно4.В\_Очереди = 0

**bitmap**[7,334,nf4,nf4\_m]

**Show\_if** Окно5.Работоспособность = Открыто

**bitmap**[7,419,f5,f5\_m]

**bitmap**[109,418,oper,oper\_m]

**Show\_if** Окно5.Работоспособность = Закрыто **and**

Окно5.В\_Очереди >= 1

**bitmap**[7,419,nf5,nf5\_m]

**bitmap**[109,418,oper,oper\_m]

**Show\_if** Окно5.Работоспособность = Закрыто **and**

Окно5.В\_Очереди = 0

**bitmap**[7,419,nf5,nf5\_m]

**Show**

**text**[155, 32, 150, 15, **transparent**, <0 128 0>, < С\_Рабочих\_Дней.Количество\_Дней]

**text**[542, 99, 57, 12, **transparent**, <0 0 0>, = Окно1.В\_Очереди]

**text**[542, 184, 57, 12, **transparent**, <0 0 0>, = Окно2.В\_Очереди]

**text**[542, 270, 57, 12, **transparent**, <0 0 0>, = Окно3.В\_Очереди]

**text**[542, 356, 57, 12, **transparent**, <0 0 0>, = Окно4.В\_Очереди]

**text**[542, 442, 57, 12, **transparent**, <0 0 0>, = Окно5.В\_Очереди]

**text**[632, 99, 150, 12, **transparent**, <0 0 0>, < Окно1.Обслужено]

**text**[632, 184, 150, 12, **transparent**, <0 0 0>, < Окно2.Обслужено]

**text**[632, 270, 150, 12, **transparent**, <0 0 0>, < Окно3.Обслужено]

**text**[632, 356, 150, 12, **transparent**, <0 0 0>, < Окно4.Обслужено]

**text**[632, 442, 150, 12, **transparent**, <0 0 0>, < Окно5.Обслужено]

**text**[729, 99, 65, 12, **transparent**, <0 0 0>, < С\_Времени1.Время\_На\_Обработку]

**text**[729, 184, 65, 12, **transparent**, <0 0 0>, < С\_Времени2.Время\_На\_Обработку]

**text**[729, 270, 65, 12, **transparent**, <0 0 0>, < С\_Времени3.Время\_На\_Обработку]

**text**[729, 356, 65, 12, **transparent**, <0 0 0>, < С\_Времени4.Время\_На\_Обработку]

**text**[729, 442, 65, 12, **transparent**, <0 0 0>, < С\_Времени5.Время\_На\_Обработку]

**Show\_if** [Окно1.В\_Очереди >= 1] **and** [Из\_Клиент1.Заявка = Rq1]

**text**[200, 99, 320, 17, **transparent**, <0 0 0>, < 'Выдача до востребования']

**Show\_if** [Окно2.В\_Очереди >= 1] **and** [Из\_Клиент2.Заявка = Rq1]

**text**[200, 184, 320, 17, **transparent**, <0 0 0>, < 'Выдача до востребования']

**Show\_if** [Окно3.В\_Очереди >= 1] **and** [Из\_Клиент3.Заявка = Rq1]

**text**[200, 270, 320, 17, **transparent**, <0 0 0>, < 'Выдача до востребования']

**Show\_if** [Окно4.В\_Очереди >= 1] **and** [Из\_Клиент4.Заявка = Rq1]

**text**[200, 356, 320, 17, **transparent**, <0 0 0>, < 'Выдача до востребования']

**Show\_if** [Окно5.В\_Очереди >= 1] **and** [Из\_Клиент5.Заявка = Rq1]

**text**[200, 442, 320, 17, **transparent**, <0 0 0>, < 'Выдача до востребования']

**Show\_if** [Окно1.В\_Очереди >= 1] **and** [Из\_Клиент1.Заявка = Rq2]

**text**[200, 99, 320, 17, **transparent**, <0 0 0>, < 'Выплата пенсий']

**Show\_if** [Окно2.В\_Очереди >= 1] **and** [Из\_Клиент2.Заявка = Rq2]

**text**[200, 184, 320, 17, **transparent**, <0 0 0>, < 'Выплата пенсий']

**Show\_if** [Окно3.В\_Очереди >= 1] **and** [Из\_Клиент3.Заявка = Rq2]

**text**[200, 270, 320, 17, **transparent**, <0 0 0>, < 'Выплата пенсий']

**Show\_if** [Окно4.В\_Очереди >= 1] **and** [Из\_Клиент4.Заявка = Rq2]

**text**[200, 356, 320, 17, **transparent**, <0 0 0>, < 'Выплата пенсий']

**Show\_if** [Окно5.В\_Очереди >= 1] **and** [Из\_Клиент5.Заявка = Rq2]

**text**[200, 442, 320, 17, **transparent**, <0 0 0>, < 'Выплата пенсий']

**Show\_if** [Окно1.В\_Очереди >= 1] **and** [Из\_Клиент1.Заявка = Rq3]

**text**[200, 99, 320, 17, **transparent**, <0 0 0>, < 'Ксерокопия']

**Show\_if** [Окно2.В\_Очереди >= 1] **and** [Из\_Клиент2.Заявка = Rq3]

**text**[200, 184, 320, 17, **transparent**, <0 0 0>, < 'Ксерокопия']

**Show\_if** [Окно3.В\_Очереди >= 1] **and** [Из\_Клиент3.Заявка = Rq3]

**text**[200, 270, 320, 17, **transparent**, <0 0 0>, < 'Ксерокопия']

**Show\_if** [Окно4.В\_Очереди >= 1] **and** [Из\_Клиент4.Заявка = Rq3]

**text**[200, 356, 320, 17, **transparent**, <0 0 0>, < 'Ксерокопия']

**Show\_if** [Окно5.В\_Очереди >= 1] **and** [Из\_Клиент5.Заявка = Rq3]

**text**[200, 442, 320, 17, **transparent**, <0 0 0>, < 'Ксерокопия']

**Show\_if** [Окно1.В\_Очереди >= 1] **and** [Из\_Клиент1.Заявка = Rq4]

**text**[200, 99, 320, 17, **transparent**, <0 0 0>, < 'Оплата п/перевода']

**Show\_if** [Окно2.В\_Очереди >= 1] **and** [Из\_Клиент2.Заявка = Rq4]

**text**[200, 184, 320, 17, **transparent**, <0 0 0>, < 'Оплата п/перевода']

**Show\_if** [Окно3.В\_Очереди >= 1] **and** [Из\_Клиент3.Заявка = Rq4]

**text**[200, 270, 320, 17, **transparent**, <0 0 0>, < 'Оплата п/перевода']

**Show\_if** [Окно4.В\_Очереди >= 1] **and** [Из\_Клиент4.Заявка = Rq4]

**text**[200, 356, 320, 17, **transparent**, <0 0 0>, < 'Оплата п/перевода']

**Show\_if** [Окно5.В\_Очереди >= 1] **and** [Из\_Клиент5.Заявка = Rq4]

**text**[200, 442, 320, 17, **transparent**, <0 0 0>, < 'Оплата п/перевода']

**Show\_if** [Окно1.В\_Очереди >= 1] **and** [Из\_Клиент1.Заявка = Rq5]

**text**[200, 99, 320, 17, **transparent**, <0 0 0>, < 'Оплата т/перевода']

**Show\_if** [Окно2.В\_Очереди >= 1] **and** [Из\_Клиент2.Заявка = Rq5]

**text**[200, 184, 320, 17, **transparent**, <0 0 0>, < 'Оплата т/перевода']

**Show\_if** [Окно3.В\_Очереди >= 1] **and** [Из\_Клиент3.Заявка = Rq5]

**text**[200, 270, 320, 17, **transparent**, <0 0 0>, < 'Оплата т/перевода']

**Show\_if** [Окно4.В\_Очереди >= 1] **and** [Из\_Клиент4.Заявка = Rq5]

**text**[200, 356, 320, 17, **transparent**, <0 0 0>, < 'Оплата т/перевода']

**Show\_if** [Окно5.В\_Очереди >= 1] **and** [Из\_Клиент5.Заявка = Rq5]

**text**[200, 442, 320, 17, **transparent**, <0 0 0>, < 'Оплата т/перевода']

**Show\_if** [Окно1.В\_Очереди >= 1] **and** [Из\_Клиент1.Заявка = Rq6]

**text**[200, 99, 320, 17, **transparent**, <0 0 0>, < 'Отправление з/письма']

**Show\_if** [Окно2.В\_Очереди >= 1] **and** [Из\_Клиент2.Заявка = Rq6]

**text**[200, 184, 320, 17, **transparent**, <0 0 0>, < 'Отправление з/письма']

**Show\_if** [Окно3.В\_Очереди >= 1] **and** [Из\_Клиент3.Заявка = Rq6]

**text**[200, 270, 320, 17, **transparent**, <0 0 0>, < 'Отправление з/письма']

**Show\_if** [Окно4.В\_Очереди >= 1] **and** [Из\_Клиент4.Заявка = Rq6]

**text**[200, 356, 320, 17, **transparent**, <0 0 0>, < 'Отправление з/письма']

**Show\_if** [Окно5.В\_Очереди >= 1] **and** [Из\_Клиент5.Заявка = Rq6]

**text**[200, 442, 320, 17, **transparent**, <0 0 0>, < 'Отправление з/письма']

**Show\_if** [Окно1.В\_Очереди >= 1] **and** [Из\_Клиент1.Заявка = Rq7]

**text**[200, 99, 320, 17, **transparent**, <0 0 0>, < 'Отправление ц/б']

**Show\_if** [Окно2.В\_Очереди >= 1] **and** [Из\_Клиент2.Заявка = Rq7]

**text**[200, 184, 320, 17, **transparent**, <0 0 0>, < 'Отправление ц/б']

**Show\_if** [Окно3.В\_Очереди >= 1] **and** [Из\_Клиент3.Заявка = Rq7]

**text**[200, 270, 320, 17, **transparent**, <0 0 0>, < 'Отправление ц/б']

**Show\_if** [Окно4.В\_Очереди >= 1] **and** [Из\_Клиент4.Заявка = Rq7]

**text**[200, 356, 320, 17, **transparent**, <0 0 0>, < 'Отправление ц/б']

**Show\_if** [Окно5.В\_Очереди >= 1] **and** [Из\_Клиент5.Заявка = Rq7]

**text**[200, 442, 320, 17, **transparent**, <0 0 0>, < 'Отправление ц/б']

**Show\_if** [Окно1.В\_Очереди >= 1] **and** [Из\_Клиент1.Заявка = Rq8]

**text**[200, 99, 320, 17, **transparent**, <0 0 0>, < 'Перерыв']

**Show\_if** [Окно2.В\_Очереди >= 1] **and** [Из\_Клиент2.Заявка = Rq8]

**text**[200, 184, 320, 17, **transparent**, <0 0 0>, < 'Перерыв']

**Show\_if** [Окно3.В\_Очереди >= 1] **and** [Из\_Клиент3.Заявка = Rq8]

**text**[200, 270, 320, 17, **transparent**, <0 0 0>, < 'Перерыв']

**Show\_if** [Окно4.В\_Очереди >= 1] **and** [Из\_Клиент4.Заявка = Rq8]

**text**[200, 356, 320, 17, **transparent**, <0 0 0>, < 'Перерыв']

**Show\_if** [Окно5.В\_Очереди >= 1] **and** [Из\_Клиент5.Заявка = Rq8]

**text**[200, 442, 320, 17, **transparent**, <0 0 0>, < 'Перерыв']

**Show\_if** [Окно1.В\_Очереди >= 1] **and** [Из\_Клиент1.Заявка = Rq9]

**text**[200, 99, 320, 17, **transparent**, <0 0 0>, < 'Получение з/письма']

**Show\_if** [Окно2.В\_Очереди >= 1] **and** [Из\_Клиент2.Заявка = Rq9]

**text**[200, 184, 320, 17, **transparent**, <0 0 0>, < 'Получение з/письма']

**Show\_if** [Окно3.В\_Очереди >= 1] **and** [Из\_Клиент3.Заявка = Rq9]

**text**[200, 270, 320, 17, **transparent**, <0 0 0>, < 'Получение з/письма']

**Show\_if** [Окно4.В\_Очереди >= 1] **and** [Из\_Клиент4.Заявка = Rq9]

**text**[200, 356, 320, 17, **transparent**, <0 0 0>, < 'Получение з/письма']

**Show\_if** [Окно5.В\_Очереди >= 1] **and** [Из\_Клиент5.Заявка = Rq9]

**text**[200, 442, 320, 17, **transparent**, <0 0 0>, < 'Получение з/письма']

**Show\_if** [Окно1.В\_Очереди >= 1] **and** [Из\_Клиент1.Заявка = Rq10]

**text**[200, 99, 320, 17, **transparent**, <0 0 0>, < 'Получение п/б']

**Show\_if** [Окно2.В\_Очереди >= 1] **and** [Из\_Клиент2.Заявка = Rq10]

**text**[200, 184, 320, 17, **transparent**, <0 0 0>, < 'Получение п/б']

**Show\_if** [Окно3.В\_Очереди >= 1] **and** [Из\_Клиент3.Заявка = Rq10]

**text**[200, 270, 320, 17, **transparent**, <0 0 0>, < 'Получение п/б']

**Show\_if** [Окно4.В\_Очереди >= 1] **and** [Из\_Клиент4.Заявка = Rq10]

**text**[200, 356, 320, 17, **transparent**, <0 0 0>, < 'Получение п/б']

**Show\_if** [Окно5.В\_Очереди >= 1] **and** [Из\_Клиент5.Заявка = Rq10]

**text**[200, 442, 320, 17, **transparent**, <0 0 0>, < 'Получение п/б']

**Show\_if** [Окно1.В\_Очереди >= 1] **and** [Из\_Клиент1.Заявка = Rq11]

**text**[200, 99, 320, 17, **transparent**, <0 0 0>, < 'Получение посылки']

**Show\_if** [Окно2.В\_Очереди >= 1] **and** [Из\_Клиент2.Заявка = Rq11]

**text**[200, 184, 320, 17, **transparent**, <0 0 0>, < 'Получение посылки']

**Show\_if** [Окно3.В\_Очереди >= 1] **and** [Из\_Клиент3.Заявка = Rq11]

**text**[200, 270, 320, 17, **transparent**, <0 0 0>, < 'Получение посылки']

**Show\_if** [Окно4.В\_Очереди >= 1] **and** [Из\_Клиент4.Заявка = Rq11]

**text**[200, 356, 320, 17, **transparent**, <0 0 0>, < 'Получение посылки']

**Show\_if** [Окно5.В\_Очереди >= 1] **and** [Из\_Клиент5.Заявка = Rq11]

**text**[200, 442, 320, 17, **transparent**, <0 0 0>, < 'Получение посылки']

**Show\_if** [Окно1.В\_Очереди >= 1] **and** [Из\_Клиент1.Заявка = Rq12]

**text**[200, 99, 320, 17, **transparent**, <0 0 0>, < 'Получение ц/б']

**Show\_if** [Окно2.В\_Очереди >= 1] **and** [Из\_Клиент2.Заявка = Rq12]

**text**[200, 184, 320, 17, **transparent**, <0 0 0>, < 'Получение ц/б']

**Show\_if** [Окно3.В\_Очереди >= 1] **and** [Из\_Клиент3.Заявка = Rq12]

**text**[200, 270, 320, 17, **transparent**, <0 0 0>, < 'Получение ц/б']

**Show\_if** [Окно4.В\_Очереди >= 1] **and** [Из\_Клиент4.Заявка = Rq12]

**text**[200, 356, 320, 17, **transparent**, <0 0 0>, < 'Получение ц/б']

**Show\_if** [Окно5.В\_Очереди >= 1] **and** [Из\_Клиент5.Заявка = Rq12]

**text**[200, 442, 320, 17, **transparent**, <0 0 0>, < 'Получение ц/б']

**Show\_if** [Окно1.В\_Очереди >= 1] **and** [Из\_Клиент1.Заявка = Rq13]

**text**[200, 99, 320, 17, **transparent**, <0 0 0>, < 'Получение ц/письма']

**Show\_if** [Окно2.В\_Очереди >= 1] **and** [Из\_Клиент2.Заявка = Rq13]

**text**[200, 184, 320, 17, **transparent**, <0 0 0>, < 'Получение ц/письма']

**Show\_if** [Окно3.В\_Очереди >= 1] **and** [Из\_Клиент3.Заявка = Rq13]

**text**[200, 270, 320, 17, **transparent**, <0 0 0>, < 'Получение ц/письма']

**Show\_if** [Окно4.В\_Очереди >= 1] **and** [Из\_Клиент4.Заявка = Rq13]

**text**[200, 356, 320, 17, **transparent**, <0 0 0>, < 'Получение ц/письма']

**Show\_if** [Окно5.В\_Очереди >= 1] **and** [Из\_Клиент5.Заявка = Rq13]

**text**[200, 442, 320, 17, **transparent**, <0 0 0>, < 'Получение ц/письма']

**Show\_if** [Окно1.В\_Очереди >= 1] **and** [Из\_Клиент1.Заявка = Rq14]

**text**[200, 99, 320, 17, **transparent**, <0 0 0>, < 'Прием коммун. платежей']

**Show\_if** [Окно2.В\_Очереди >= 1] **and** [Из\_Клиент2.Заявка = Rq14]

**text**[200, 184, 320, 17, **transparent**, <0 0 0>, < 'Прием коммун. платежей']

**Show\_if** [Окно3.В\_Очереди >= 1] **and** [Из\_Клиент3.Заявка = Rq14]

**text**[200, 270, 320, 17, **transparent**, <0 0 0>, < 'Прием коммун. платежей']

**Show\_if** [Окно4.В\_Очереди >= 1] **and** [Из\_Клиент4.Заявка = Rq14]

**text**[200, 356, 320, 17, **transparent**, <0 0 0>, < 'Прием коммун. платежей']

**Show\_if** [Окно5.В\_Очереди >= 1] **and** [Из\_Клиент5.Заявка = Rq14]

**text**[200, 442, 320, 17, **transparent**, <0 0 0>, < 'Прием коммун. платежей']

**Show\_if** [Окно1.В\_Очереди >= 1] **and** [Из\_Клиент1.Заявка = Rq15]

**text**[200, 99, 320, 17, **transparent**, <0 0 0>, < 'Прием платы за ГТС']

**Show\_if** [Окно2.В\_Очереди >= 1] **and** [Из\_Клиент2.Заявка = Rq15]

**text**[200, 184, 320, 17, **transparent**, <0 0 0>, < 'Прием платы за ГТС']

**Show\_if** [Окно3.В\_Очереди >= 1] **and** [Из\_Клиент3.Заявка = Rq15]

**text**[200, 270, 320, 17, **transparent**, <0 0 0>, < 'Прием платы за ГТС']

**Show\_if** [Окно4.В\_Очереди >= 1] **and** [Из\_Клиент4.Заявка = Rq15]

**text**[200, 356, 320, 17, **transparent**, <0 0 0>, < 'Прием платы за ГТС']

**Show\_if** [Окно5.В\_Очереди >= 1] **and** [Из\_Клиент5.Заявка = Rq15]

**text**[200, 442, 320, 17, **transparent**, <0 0 0>, < 'Прием платы за ГТС']

**Show\_if** [Окно1.В\_Очереди >= 1] **and** [Из\_Клиент1.Заявка = Rq16]

**text**[200, 99, 320, 17, **transparent**, <0 0 0>, < 'Прием подписки']

**Show\_if** [Окно2.В\_Очереди >= 1] **and** [Из\_Клиент2.Заявка = Rq16]

**text**[200, 184, 320, 17, **transparent**, <0 0 0>, < 'Прием подписки']

**Show\_if** [Окно3.В\_Очереди >= 1] **and** [Из\_Клиент3.Заявка = Rq16]

**text**[200, 270, 320, 17, **transparent**, <0 0 0>, < 'Прием подписки']

**Show\_if** [Окно4.В\_Очереди >= 1] **and** [Из\_Клиент4.Заявка = Rq16]

**text**[200, 356, 320, 17, **transparent**, <0 0 0>, < 'Прием подписки']

**Show\_if** [Окно5.В\_Очереди >= 1] **and** [Из\_Клиент5.Заявка = Rq16]

**text**[200, 442, 320, 17, **transparent**, <0 0 0>, < 'Прием подписки']

**Show\_if** [Окно1.В\_Очереди >= 1] **and** [Из\_Клиент1.Заявка = Rq17]

**text**[200, 99, 320, 17, **transparent**, <0 0 0>, < 'Продажа ЗПО']

**Show\_if** [Окно2.В\_Очереди >= 1] **and** [Из\_Клиент2.Заявка = Rq17]

**text**[200, 184, 320, 17, **transparent**, <0 0 0>, < 'Продажа ЗПО']

**Show\_if** [Окно3.В\_Очереди >= 1] **and** [Из\_Клиент3.Заявка = Rq17]

**text**[200, 270, 320, 17, **transparent**, <0 0 0>, < 'Продажа ЗПО']

**Show\_if** [Окно4.В\_Очереди >= 1] **and** [Из\_Клиент4.Заявка = Rq17]

**text**[200, 356, 320, 17, **transparent**, <0 0 0>, < 'Продажа ЗПО']

**Show\_if** [Окно5.В\_Очереди >= 1] **and** [Из\_Клиент5.Заявка = Rq17]

**text**[200, 442, 320, 17, **transparent**, <0 0 0>, < 'Продажа ЗПО']

**Show\_if** [Окно1.В\_Очереди >= 1] **and** [Из\_Клиент1.Заявка = Rq18]

**text**[200, 99, 320, 17, **transparent**, <0 0 0>, < 'Справка о пенсии']

**Show\_if** [Окно2.В\_Очереди >= 1] **and** [Из\_Клиент2.Заявка = Rq18]

**text**[200, 184, 320, 17, **transparent**, <0 0 0>, < 'Справка о пенсии']

**Show\_if** [Окно3.В\_Очереди >= 1] **and** [Из\_Клиент3.Заявка = Rq18]

**text**[200, 270, 320, 17, **transparent**, <0 0 0>, < 'Справка о пенсии']

**Show\_if** [Окно4.В\_Очереди >= 1] **and** [Из\_Клиент4.Заявка = Rq18]

**text**[200, 356, 320, 17, **transparent**, <0 0 0>, < 'Справка о пенсии']

**Show\_if** [Окно5.В\_Очереди >= 1] **and** [Из\_Клиент5.Заявка = Rq18]

**text**[200, 442, 320, 17, **transparent**, <0 0 0>, < 'Справка о пенсии']

**Show\_if** [Окно1.В\_Очереди >= 1] **and** [Из\_Клиент1.Заявка = Rq19]

**text**[200, 99, 320, 17, **transparent**, <0 0 0>, < 'Справка о подписке']

**Show\_if** [Окно2.В\_Очереди >= 1] **and** [Из\_Клиент2.Заявка = Rq19]

**text**[200, 184, 320, 17, **transparent**, <0 0 0>, < 'Справка о подписке']

**Show\_if** [Окно3.В\_Очереди >= 1] **and** [Из\_Клиент3.Заявка = Rq19]

**text**[200, 270, 320, 17, **transparent**, <0 0 0>, < 'Справка о подписке']

**Show\_if** [Окно4.В\_Очереди >= 1] **and** [Из\_Клиент4.Заявка = Rq19]

**text**[200, 356, 320, 17, **transparent**, <0 0 0>, < 'Справка о подписке']

**Show\_if** [Окно5.В\_Очереди >= 1] **and** [Из\_Клиент5.Заявка = Rq19]

**text**[200, 442, 320, 17, **transparent**, <0 0 0>, < 'Справка о подписке']

**Show\_if** Окно1.Занятость = Занят

**bitmap**[125,73,others,others\_m]

**Show\_if** Окно2.Занятость = Занят

**bitmap**[125,158,others,others\_m]

**Show\_if** Окно3.Занятость = Занят

**bitmap**[125,244,others,others\_m]

**Show\_if** Окно4.Занятость = Занят

**bitmap**[125,330,others,others\_m]

**Show\_if** Окно5.Занятость = Занят

**bitmap**[125,416,others,others\_m]

**Show**

// active Кликнул1 [65,66,114,80]

// active Кликнул2 [65,151,114,80]

// active Кликнул3 [65,237,114,80]

// active Кликнул4 [65,323,114,80]

// active Кликнул5 [65,409,114,80]

**Show\_if** Смотритель1.Разрешить = Да

**bitmap**[См\_X - 10, См\_Y - 40, order\_1, order\_m]

**rect** [См\_X - 10, См\_Y - 5, См\_Ширина, 20 \* См\_Высота - 2, <0 255 255>, <0 0 0>]

**text** [См\_X, См\_Y + 0 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, < 'Выдача до востребования']

**text** [См\_X, См\_Y + 1 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, < 'Выплата пенсий']

**text** [См\_X, См\_Y + 2 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, < 'Ксерокопия']

**text** [См\_X, См\_Y + 3 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, < 'Оплата п/перевода']

**text** [См\_X, См\_Y + 4 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, < 'Оплата т/перевода']

**text** [См\_X, См\_Y + 5 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, < 'Отправление з/письма']

**text** [См\_X, См\_Y + 6 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, < 'Отправление ц/б']

**text** [См\_X, См\_Y + 7 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, < 'Перерыв']

**text** [См\_X, См\_Y + 8 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, < 'Получение з/письма']

**text** [См\_X, См\_Y + 9 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, < 'Получение п/б']

**text** [См\_X, См\_Y + 10 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, < 'Получение посылки']

**text** [См\_X, См\_Y + 11 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, < 'Получение ц/б']

**text** [См\_X, См\_Y + 12 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, < 'Получение ц/письма']

**text** [См\_X, См\_Y + 13 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, < 'Прием коммун. платежей']

**text** [См\_X, См\_Y + 14 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, < 'Прием платы за ГТС']

**text** [См\_X, См\_Y + 15 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, < 'Прием подписки']

**text** [См\_X, См\_Y + 16 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, < 'Продажа ЗПО']

**text** [См\_X, См\_Y + 17 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, < 'Справка о пенсии']

**text** [См\_X, См\_Y + 18 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, < 'Справка о подписке']

**Show\_if** Смотритель2.Разрешить = Да

**bitmap**[См\_X - 10, См\_Y - 40, order\_2, order\_m]

**rect** [См\_X - 10, См\_Y - 5, См\_Ширина, 20 \* См\_Высота - 2, <0 255 255>, <0 0 0>]

**text** [См\_X, См\_Y + 0 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Выдача до востребования']

**text** [См\_X, См\_Y + 1 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Выплата пенсий']

**text** [См\_X, См\_Y + 2 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Ксерокопия']

**text** [См\_X, См\_Y + 3 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Оплата п/перевода']

**text** [См\_X, См\_Y + 4 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Оплата т/перевода']

**text** [См\_X, См\_Y + 5 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Отправление з/письма']

**text** [См\_X, См\_Y + 6 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Отправление ц/б']

**text** [См\_X, См\_Y + 7 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Перерыв']

**text** [См\_X, См\_Y + 8 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Получение з/письма']

**text** [См\_X, См\_Y + 9 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Получение п/б']

**text** [См\_X, См\_Y + 10 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Получение посылки']

**text** [См\_X, См\_Y + 11 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Получение ц/б']

**text** [См\_X, См\_Y + 12 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Получение ц/письма']

**text** [См\_X, См\_Y + 13 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Прием коммун. платежей']

**text** [См\_X, См\_Y + 14 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Прием платы за ГТС']

**text** [См\_X, См\_Y + 15 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Прием подписки']

**text** [См\_X, См\_Y + 16 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Продажа ЗПО']

**text** [См\_X, См\_Y + 17 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Справка о пенсии']

**text** [См\_X, См\_Y + 18 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Справка о подписке']

**Show\_if** Смотритель3.Разрешить = Да

**bitmap**[См\_X - 10, См\_Y - 40, order\_3, order\_m]

**rect** [См\_X - 10, См\_Y - 5, См\_Ширина, 20 \* См\_Высота - 2, <0 255 255>, <0 0 0>]

**text** [См\_X, См\_Y + 0 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Выдача до востребования']

**text** [См\_X, См\_Y + 1 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Выплата пенсий']

**text** [См\_X, См\_Y + 2 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Ксерокопия']

**text** [См\_X, См\_Y + 3 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Оплата п/перевода']

**text** [См\_X, См\_Y + 4 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Оплата т/перевода']

**text** [См\_X, См\_Y + 5 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Отправление з/письма']

**text** [См\_X, См\_Y + 6 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Отправление ц/б']

**text** [См\_X, См\_Y + 7 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Перерыв']

**text** [См\_X, См\_Y + 8 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Получение з/письма']

**text** [См\_X, См\_Y + 9 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Получение п/б']

**text** [См\_X, См\_Y + 10 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Получение посылки']

**text** [См\_X, См\_Y + 11 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Получение ц/б']

**text** [См\_X, См\_Y + 12 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Получение ц/письма']

**text** [См\_X, См\_Y + 13 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Прием коммун. платежей']

**text** [См\_X, См\_Y + 14 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Прием платы за ГТС']

**text** [См\_X, См\_Y + 15 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Прием подписки']

**text** [См\_X, См\_Y + 16 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Продажа ЗПО']

**text** [См\_X, См\_Y + 17 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Справка о пенсии']

**text** [См\_X, См\_Y + 18 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Справка о подписке']

**Show\_if** Смотритель4.Разрешить = Да

**bitmap**[См\_X - 10, См\_Y - 40, order\_4, order\_m]

**rect** [См\_X - 10, См\_Y - 5, См\_Ширина, 20 \* См\_Высота - 2, <0 255 255>, <0 0 0>]

**text** [См\_X, См\_Y + 0 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Выдача до востребования']

**text** [См\_X, См\_Y + 1 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Выплата пенсий']

**text** [См\_X, См\_Y + 2 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Ксерокопия']

**text** [См\_X, См\_Y + 3 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Оплата п/перевода']

**text** [См\_X, См\_Y + 4 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Оплата т/перевода']

**text** [См\_X, См\_Y + 5 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Отправление з/письма']

**text** [См\_X, См\_Y + 6 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Отправление ц/б']

**text** [См\_X, См\_Y + 7 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Перерыв']

**text** [См\_X, См\_Y + 8 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Получение з/письма']

**text** [См\_X, См\_Y + 9 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Получение п/б']

**text** [См\_X, См\_Y + 10 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Получение посылки']

**text** [См\_X, См\_Y + 11 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Получение ц/б']

**text** [См\_X, См\_Y + 12 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Получение ц/письма']

**text** [См\_X, См\_Y + 13 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Прием коммун. платежей']

**text** [См\_X, См\_Y + 14 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Прием платы за ГТС']

**text** [См\_X, См\_Y + 15 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Прием подписки']

**text** [См\_X, См\_Y + 16 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Продажа ЗПО']

**text** [См\_X, См\_Y + 17 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Справка о пенсии']

**text** [См\_X, См\_Y + 18 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Справка о подписке']

**Show\_if** Смотритель5.Разрешить = Да

**bitmap**[См\_X - 10, См\_Y - 40, order\_5, order\_m]

**rect** [См\_X - 10, См\_Y - 5, См\_Ширина, 20 \* См\_Высота - 2, <0 255 255>, <0 0 0>]

**text** [См\_X, См\_Y + 0 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Выдача до востребования']

**text** [См\_X, См\_Y + 1 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Выплата пенсий']

**text** [См\_X, См\_Y + 2 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Ксерокопия']

**text** [См\_X, См\_Y + 3 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Оплата п/перевода']

**text** [См\_X, См\_Y + 4 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Оплата т/перевода']

**text** [См\_X, См\_Y + 5 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Отправление з/письма']

**text** [См\_X, См\_Y + 6 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Отправление ц/б']

**text** [См\_X, См\_Y + 7 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Перерыв']

**text** [См\_X, См\_Y + 8 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Получение з/письма']

**text** [См\_X, См\_Y + 9 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Получение п/б']

**text** [См\_X, См\_Y + 10 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Получение посылки']

**text** [См\_X, См\_Y + 11 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Получение ц/б']

**text** [См\_X, См\_Y + 12 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Получение ц/письма']

**text** [См\_X, См\_Y + 13 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Прием коммун. платежей']

**text** [См\_X, См\_Y + 14 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Прием платы за ГТС']

**text** [См\_X, См\_Y + 15 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Прием подписки']

**text** [См\_X, См\_Y + 16 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Продажа ЗПО']

**text** [См\_X, См\_Y + 17 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Справка о пенсии']

**text** [См\_X, См\_Y + 18 \* См\_Высота, См\_Ширина, См\_Высота, **transparent**, <0 0 0>, 'Справка о подписке']

**$End**

**$Frame** Статистика

**$Back\_picture** = <0 255 0> post\_2

**Show**

**rect** [Начало1\_X , Начало\_Y\_R + Высота \* 0, Ширина1, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст1, Начало\_Y\_T + Высота \* 0, 320, 17, **transparent**, <0 0 0>, < 'Выдача до востребования']

**rect** [Начало2\_X , Начало\_Y\_R + Высота \* 0, Ширина2, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст2, Начало\_Y\_T + Высота \* 0, 167, 17, **transparent**, <0 0 0>, = Count1.Количество]

**rect** [Начало1\_X , Начало\_Y\_R + Высота \* 1, Ширина1, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст1, Начало\_Y\_T + Высота \* 1, 320, 17, **transparent**, <0 0 0>, < 'Выплата пенсий']

**rect** [Начало2\_X , Начало\_Y\_R + Высота \* 1, Ширина2, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст2, Начало\_Y\_T + Высота \* 1, 167, 17, **transparent**, <0 0 0>, = Count2.Количество]

**rect** [Начало1\_X , Начало\_Y\_R + Высота \* 2, Ширина1, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст1, Начало\_Y\_T + Высота \* 2, 320, 17, **transparent**, <0 0 0>, < 'Ксерокопия']

**rect** [Начало2\_X , Начало\_Y\_R + Высота \* 2, Ширина2, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст2, Начало\_Y\_T + Высота \* 2, 167, 17, **transparent**, <0 0 0>, = Count3.Количество]

**rect** [Начало1\_X , Начало\_Y\_R + Высота \* 3, Ширина1, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст1, Начало\_Y\_T + Высота \* 3, 320, 17, **transparent**, <0 0 0>, < 'Оплата п/перевода']

**rect** [Начало2\_X , Начало\_Y\_R + Высота \* 3, Ширина2, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст2, Начало\_Y\_T + Высота \* 3, 167, 17, **transparent**, <0 0 0>, = Count4.Количество]

**rect** [Начало1\_X , Начало\_Y\_R + Высота \* 4, Ширина1, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст1, Начало\_Y\_T + Высота \* 4, 320, 17, **transparent**, <0 0 0>, < 'Оплата т/перевода']

**rect** [Начало2\_X , Начало\_Y\_R + Высота \* 4, Ширина2, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст2, Начало\_Y\_T + Высота \* 4, 167, 17, **transparent**, <0 0 0>, = Count5.Количество]

**rect** [Начало1\_X , Начало\_Y\_R + Высота \* 5, Ширина1, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст1, Начало\_Y\_T + Высота \* 5, 320, 17, **transparent**, <0 0 0>, < 'Отправление з/письма']

**rect** [Начало2\_X , Начало\_Y\_R + Высота \* 5, Ширина2, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст2, Начало\_Y\_T + Высота \* 5, 167, 17, **transparent**, <0 0 0>, = Count6.Количество]

**rect** [Начало1\_X , Начало\_Y\_R + Высота \* 6, Ширина1, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст1, Начало\_Y\_T + Высота \* 6, 320, 17, **transparent**, <0 0 0>, < 'Отправление ц/б']

**rect** [Начало2\_X , Начало\_Y\_R + Высота \* 6, Ширина2, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст2, Начало\_Y\_T + Высота \* 6, 167, 17, **transparent**, <0 0 0>, = Count7.Количество]

**rect** [Начало1\_X , Начало\_Y\_R + Высота \* 7, Ширина1, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст1, Начало\_Y\_T + Высота \* 7, 320, 17, **transparent**, <0 0 0>, < 'Перерыв']

**rect** [Начало2\_X , Начало\_Y\_R + Высота \* 7, Ширина2, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст2, Начало\_Y\_T + Высота \* 7, 167, 17, **transparent**, <0 0 0>, = Count8.Количество]

**rect** [Начало1\_X , Начало\_Y\_R + Высота \* 8, Ширина1, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст1, Начало\_Y\_T + Высота \* 8, 320, 17, **transparent**, <0 0 0>, < 'Получение з/письма']

**rect** [Начало2\_X , Начало\_Y\_R + Высота \* 8, Ширина2, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст2, Начало\_Y\_T + Высота \* 8, 167, 17, **transparent**, <0 0 0>, = Count9.Количество]

**rect** [Начало1\_X , Начало\_Y\_R + Высота \* 9, Ширина1, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст1, Начало\_Y\_T + Высота \* 9, 320, 17, **transparent**, <0 0 0>, < 'Получение п/б']

**rect** [Начало2\_X , Начало\_Y\_R + Высота \* 9, Ширина2, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст2, Начало\_Y\_T + Высота \* 9, 167, 17, **transparent**, <0 0 0>, = Count10.Количество]

**rect** [Начало1\_X , Начало\_Y\_R + Высота \* 10, Ширина1, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст1, Начало\_Y\_T + Высота \* 10, 320, 17, **transparent**, <0 0 0>, < 'Получение посылки']

**rect** [Начало2\_X , Начало\_Y\_R + Высота \* 10, Ширина2, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст2, Начало\_Y\_T + Высота \* 10, 167, 17, **transparent**, <0 0 0>, = Count11.Количество]

**rect** [Начало1\_X , Начало\_Y\_R + Высота \* 11, Ширина1, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст1, Начало\_Y\_T + Высота \* 11, 320, 17, **transparent**, <0 0 0>, < 'Получение ц/б']

**rect** [Начало2\_X , Начало\_Y\_R + Высота \* 11, Ширина2, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст2, Начало\_Y\_T + Высота \* 11, 167, 17, **transparent**, <0 0 0>, = Count12.Количество]

**rect** [Начало1\_X , Начало\_Y\_R + Высота \* 12, Ширина1, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст1, Начало\_Y\_T + Высота \* 12, 320, 17, **transparent**, <0 0 0>, < 'Получение ц/письма']

**rect** [Начало2\_X , Начало\_Y\_R + Высота \* 12, Ширина2, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст2, Начало\_Y\_T + Высота \* 12, 167, 17, **transparent**, <0 0 0>, = Count13.Количество]

**rect** [Начало1\_X , Начало\_Y\_R + Высота \* 13, Ширина1, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст1, Начало\_Y\_T + Высота \* 13, 320, 17, **transparent**, <0 0 0>, < 'Прием коммун. платежей']

**rect** [Начало2\_X , Начало\_Y\_R + Высота \* 13, Ширина2, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст2, Начало\_Y\_T + Высота \* 13, 167, 17, **transparent**, <0 0 0>, = Count14.Количество]

**rect** [Начало1\_X , Начало\_Y\_R + Высота \* 14, Ширина1, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст1, Начало\_Y\_T + Высота \* 14, 320, 17, **transparent**, <0 0 0>, < 'Прием платы за ГТС']

**rect** [Начало2\_X , Начало\_Y\_R + Высота \* 14, Ширина2, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст2, Начало\_Y\_T + Высота \* 14, 167, 17, **transparent**, <0 0 0>, = Count15.Количество]

**rect** [Начало1\_X , Начало\_Y\_R + Высота \* 15, Ширина1, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст1, Начало\_Y\_T + Высота \* 15, 320, 17, **transparent**, <0 0 0>, < 'Прием подписки']

**rect** [Начало2\_X , Начало\_Y\_R + Высота \* 15, Ширина2, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст2, Начало\_Y\_T + Высота \* 15, 167, 17, **transparent**, <0 0 0>, = Count16.Количество]

**rect** [Начало1\_X , Начало\_Y\_R + Высота \* 16, Ширина1, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст1, Начало\_Y\_T + Высота \* 16, 320, 17, **transparent**, <0 0 0>, < 'Продажа ЗПО']

**rect** [Начало2\_X , Начало\_Y\_R + Высота \* 16, Ширина2, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст2, Начало\_Y\_T + Высота \* 16, 167, 17, **transparent**, <0 0 0>, = Count17.Количество]

**rect** [Начало1\_X , Начало\_Y\_R + Высота \* 17, Ширина1, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст1, Начало\_Y\_T + Высота \* 17, 320, 17, **transparent**, <0 0 0>, < 'Справка о пенсии']

**rect** [Начало2\_X , Начало\_Y\_R + Высота \* 17, Ширина2, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст2, Начало\_Y\_T + Высота \* 17, 167, 17, **transparent**, <0 0 0>, = Count18.Количество]

**rect** [Начало1\_X , Начало\_Y\_R + Высота \* 18, Ширина1, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст1, Начало\_Y\_T + Высота \* 18, 320, 17, **transparent**, <0 0 0>, < 'Справка о подписке']

**rect** [Начало2\_X , Начало\_Y\_R + Высота \* 18, Ширина2, Высота + 1, **transparent**, <0 0 0>]

**text** [Начало\_Текст2, Начало\_Y\_T + Высота \* 18, 167, 17, **transparent**, <0 0 0>, = Count19.Количество]

**$End**

Postoffice.smr (Прогон):

**Model\_name** = Postoffice

**Resource\_file** = Postoffice

**Frame\_file** = Postoffice

**Statistic\_file** = Postoffice

**Results\_file** = Postoffice

**Trace\_file** = Postoffice

**Show\_mode** = **NoShow**

**Frame\_number** = 1

**Show\_rate** = 50.0

**Terminate\_if** С\_Рабочих\_Дней.Количество\_Дней = 2

**Break\_point** Завершение\_моделирования С\_Рабочих\_Дней.Количество\_Дней = Год

Postoffice.pmd (Показатели):

**$Results**

Время\_Пребывания\_В\_Очереди : **watch\_value** Клиенты **NoCheck** Клиенты.Время\_В\_Очереди

Количество\_В\_Очереди\_1 : **watch\_par** Окно1.В\_Очереди

Количество\_В\_Очереди\_2 : **watch\_par** Окно2.В\_Очереди

Количество\_В\_Очереди\_3 : **watch\_par** Окно3.В\_Очереди

Количество\_В\_Очереди\_4 : **watch\_par** Окно4.В\_Очереди

Количество\_В\_Очереди\_5 : **watch\_par** Окно5.В\_Очереди

Количество\_Обслуженных1 : **get\_value** Окно1.Обслужено

Количество\_Обслуженных2 : **get\_value** Окно2.Обслужено

Количество\_Обслуженных3 : **get\_value** Окно3.Обслужено

Количество\_Обслуженных4 : **get\_value** Окно4.Обслужено

Количество\_Обслуженных5 : **get\_value** Окно5.Обслужено

Количество\_Обслуженных : **get\_value** Окно5.Обслужено + Окно4.Обслужено + Окно3.Обслужено + Окно2.Обслужено + Окно1.Обслужено

Число\_Клиентов\_Без\_Очереди : **watch\_quant** Клиенты Клиенты.Время\_В\_Очереди = 0 **and** Клиенты.Состояние = Ушел

Время\_моделирования : **get\_value** **Time\_now**

Занятость\_Оператора\_з\_1 : **watch\_state** Окно1.Занятость = Занят

Занятость\_Оператора\_з\_2 : **watch\_state** Окно2.Занятость = Занят

Занятость\_Оператора\_з\_3 : **watch\_state** Окно3.Занятость = Занят

Занятость\_Оператора\_з\_4 : **watch\_state** Окно4.Занятость = Занят

Занятость\_Оператора\_з\_5 : **watch\_state** Окно5.Занятость = Занят

Занятость\_Оператора\_с\_1 : **watch\_state** Окно1.Занятость = Свободен **and** Окно1.Работоспособность = Открыто

Занятость\_Оператора\_с\_2 : **watch\_state** Окно2.Занятость = Свободен **and** Окно2.Работоспособность = Открыто

Занятость\_Оператора\_с\_3 : **watch\_state** Окно3.Занятость = Свободен **and** Окно3.Работоспособность = Открыто

Занятость\_Оператора\_с\_4 : **watch\_state** Окно4.Занятость = Свободен **and** Окно4.Работоспособность = Открыто

Занятость\_Оператора\_с\_5 : **watch\_state** Окно5.Занятость = Свободен **and** Окно5.Работоспособность = Открыто

Время\_Обработки\_В\_Окне1 : **watch\_par** С\_Времени1.Время\_На\_Обработку

Время\_Обработки\_В\_Окне2 : **watch\_par** С\_Времени2.Время\_На\_Обработку

Время\_Обработки\_В\_Окне3 : **watch\_par** С\_Времени3.Время\_На\_Обработку

Время\_Обработки\_В\_Окне4 : **watch\_par** С\_Времени4.Время\_На\_Обработку

Время\_Обработки\_В\_Окне5 : **watch\_par** С\_Времени5.Время\_На\_Обработку

Потерянных : **get\_value** Потерянные.Количество

Rq1 : **get\_value** Count1.Количество

Rq2 : **get\_value** Count2.Количество

Rq3 : **get\_value** Count3.Количество

Rq4 : **get\_value** Count4.Количество

Rq5 : **get\_value** Count5.Количество

Rq6 : **get\_value** Count6.Количество

Rq7 : **get\_value** Count7.Количество

Rq8 : **get\_value** Count8.Количество

Rq9 : **get\_value** Count9.Количество

Rq10 : **get\_value** Count10.Количество

Rq11 : **get\_value** Count11.Количество

Rq12 : **get\_value** Count12.Количество

Rq13 : **get\_value** Count13.Количество

Rq14 : **get\_value** Count14.Количество

Rq15 : **get\_value** Count15.Количество

Rq16 : **get\_value** Count16.Количество

Rq17 : **get\_value** Count17.Количество

Rq18 : **get\_value** Count18.Количество

Rq19 : **get\_value** Count19.Количество

**$End**

# Приложение 3. Функциональная диаграмма компиляции модели на языке РДО. Уровень A-0.