

### 1.1.3.1 Построение бизнес-модели организации средствами описания бизнес-процессов

#### Основные термины и определения

Описание бизнес-процессов – это документирование в свободной форме описания последовательности действий сотрудников, направленных на создание какого-либо продукта или оказания услуг, например, простое текстовое описание.

Моделирование бизнес-процессов – это создание некоторой формальной модели реальной или предполагаемой деятельности предприятия, т.е. формализованная процедура, описанная на каком-либо формализованном языке.

Методология моделирования - это системные основы исследования, проектирования, принципы построения моделей бизнес-процессов, а также совокупность применяемых при этом методов. Таким образом, методом будет являться процедура, систематически применяемая для формирования описания системы с применением определенных нотаций.

Под нотацией принято понимать систему условных знаков и правил их использования для описания различных категорий моделируемой системы. Категории могут включать в себя, например, объекты, процессы, взаимосвязи между ними и т.п.

В методологии моделирования выделяют несколько видов моделирования, которые различаются подходом к построению и отображению моделей бизнес-процессов, например, структурный, объектно-ориентированный или интегрированный подход.

При структурном подходе изучается структура системы с точки зрения состава элементов, подсистем и отношений между ними (структура), а также с точки зрения свойств системы, которые позволяют достигать заданной цели (функции). Другими словами, главным элементом является функция (операция), а бизнес-процесс представляется в виде последовательности функций, преобразующих входы процесса в выходы с использованием определенных ресурсов. Методы и нотации вида моделирования, использующего структурный подход, представлены в таблице 1.

Таблица 1 - Методы и нотации структурного моделирования

Вид моделирования	Применяемый метод	Используемая нотация
Структурный подход	SADT	IDEF0
	DFD	Гайна - Сарсона Gane / Sarson Йордана - Де Марко Yourdon / De Marko
	WFD	IDEF3 (PFDD)
	ARIS	eEPC
		BPMN

**Цель практической работы:** изучение функционала программного средства Vpwin, построение контекстной диаграммы деятельности медицинской клиники.

**Ограничения:** Бизнес-процессы организации должны быть наглядно представлены руководству бизнеса и исполнителям процессов. Графическое изображение модели бизнес-процессов не должно иметь большой объем деталей, оно должно быть визуально простым и интуитивно понятным.

В результате была выбрана методология структурного анализа и проектирования SADT (Structured analysis and design technique), как наиболее подходящий метод для моделирования бизнес-процессов медицинской клиники. Данный метод представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели объекта (производимые им действия и связи между этими действиями) какой-либо предметной области.

Главная идея методологии SADT - построение иерархической (древовидной) модели бизнес-процессов предприятия. Основным рабочим элементом методологии является диаграмма. Элементы методологии SADT (синтаксис):

- **Функциональный блок (Activities)** – совокупность действий;
- **Управление (Control)** – определенные правила, стратегии, стандарты для выполнения действий;
- **Механизм (Mechanism)** – потребляемые информационные, человеческие и производственные ресурсы, которые выполняют процесс;
- **Вход (Input)** – четко определенный вход - материал или информация, которые используются или преобразуются для получения результата;
- **Выход (Output)** – четко определенный выход - материал или информация, получаемые в результате выполнения процесса.

Самая распространенная нотация методологии SADT: IDEF0– это графическая схема, применяемая для функционального моделирования бизнес-процессов.

Построение модели состоит из следующих этапов:

1. Моделирование текущих бизнес-процессов.
2. Определение нерациональных действий.
3. Обнаружение слабых мест.
4. Моделирование будущих бизнес-процессов.

### **Описание деятельности медицинской клиники.**

Деятельность медицинской клиники представляет собой непрерывный процесс, начинающийся с обращения нового пациента в клинику до удовлетворения его потребности в получении медицинской помощи. Клиника, предоставляет услуги медицинского профиля, в частности: прием врачей-клиницистов широкого спектра специализаций, прием врачей функциональной диагностики, услуги лабораторной службы. Общая схема оказания медицинской помощи в амбулаторных условиях:

1. Обращение пациента.
2. Выяснение его потребностей в оказании медицинских услуг.
3. Оформление документов (амбулаторная карта, договор).
4. Взаимодействие по оплате медицинских услуг.
5. Проведение приема пациента специалистом.
6. Оформление истории болезни.
7. Выписка направлений на обследование и анализы (при необходимости).
8. Талон на последующее посещение (при необходимости).

В клинике также работает административный персонал, руководство, средний медицинский персонал и врачи.

Итак, бизнес-процессы медицинской клиники можно классифицировать следующим образом:

1. Процессы, связанные с оказанием амбулаторной медицинской помощи.
2. Процессы, связанные с консультативной деятельностью врачей-специалистов.
3. Административно-управленческая деятельность.
4. Финансово-экономическая деятельность.
5. Материальное обеспечение лечебно-диагностического процесса.
6. Медицинская статистика и архив медицинских документов.

Работа медицинских учреждений, как государственных, так и частных, регламентируется соответствующей законодательной базой – нормативными документами. Основным законом для всех медучреждений является Федеральный закон от 21.11.2011 N 323-ФЗ (ред. от 22.12.2020) «Об основах охраны здоровья граждан в Российской Федерации». Данный закон регламентирует отношения в области здравоохранения граждан РФ, права и обязанности медработников и организацию системы документооборота. Это лишь один пример нормативного документа. Помимо этого закона, в регулировании медицинской практики участвуют иные документы, регламентирующие отдельные аспекты меддеятельности и отражающие правовые отношения между учреждением и пациентом, страховой компанией и клиникой. Всего их в настоящее время не менее полутора десятков: законы, приказы, письма, методические рекомендации.

Для упрощения процесса построения модели бизнес-процессов в IDEF0, можно использовать CASE-средство BPwin, которое позволяет наглядно представить функциональную декомпозицию системы.

Но предварительно необходимо проанализировать деятельность медицинской клиники и представить ее в синтаксисе методологии SADT.

Итак, на первом этапе элемент «Функциональный блок» - это деятельность медицинской клиники.

«Управление» представляет собой нормативные документы общие для деятельности всех медицинских учреждений. Кроме того, в зависимости от специализации каждой конкретной клиники существуют определенные

методики лечения различных заболеваний, поэтому их тоже стоит включить в элемент «Управление». К правилам можно отнести любой документ, регламентирующий порядок приема пациентов и оказания им медицинской помощи, например, утвержденные часы работы медучреждения.

Далее, определим, что будет входить в элемент «Механизмы». Как описывалось выше, в деятельности медицинской клиник участвует медицинский персонал, обеспечивающий диагностические и лечебные процессы, и административный персонал, выполняющий управленческие функции. В настоящее время трудно представить себе деятельность любой клиники, больницы, медицинского центра и, даже, поликлиники без применения специального оборудования. Сложное медицинское оборудование используется в диагностических целях, проведении операций или лечении заболеваний, например, лазерное, ультразвуковое, электромагнитное и т.п. Подводя итог, определим элемент «Механизмы»: «Персонал» и «Медицинское оборудование».

Для того чтобы любое медицинское учреждение начало функционировать, необходимо обеспечить его оборудованием и персоналом, т.е. произвести закупку и установку оборудования, и набор персонала. Оборудование закупается в соответствии с профилем клиники, например, стоматологическое, офтальмологическое и т.д. Закупка сопровождается соответствующими документами, которые используются в дальнейшей деятельности: приходные накладные, технические паспорта, инструкции по эксплуатации и другие. Расширение направлений деятельности медицинского учреждения или появление новых технологий также способствует приобретению новых медицинских приборов и аппаратов.

При наборе персонала в любое медицинское учреждение учитывается наличие дипломов и действующих сертификатов, ученых степеней и высших медицинских категорий, опыт работы, подтверждаемый документами кадрового учета.

Пациенты, которые обращаются за медицинской помощью, предъявляют паспорт, в определенных случаях и страховой полис. Эти данные являются основанием для заполнения документов, законодательно утвержденных в документообороте медучреждений.

Подводя итог вышесказанному, определяет составляющие элемента «Вход»: «Данные о сотрудниках», «Данные о закупках», «Данные пациентов».

И, последнее, что осталось, определить элемент «Выход». Каждый пациент после проведения диагностических и лечебных мероприятий получает следующие документы: выписка, рецепт, назначение лечебных или профилактических мер. Помимо этого составляется достаточно большое количество статистических отчетов. Для всех органов и учреждений здравоохранения существуют единые формы статистической отчетности, утверждаемые Федеральной службой государственной статистики (Росстат), а также единые формы учета и инструкции по их заполнению. Медицинская статистика крайне важна для обобщения и отражения в государственных

масштабах состояния системы здравоохранения, состояния здоровья граждан РФ и планирования профилактических мероприятий.

Таким образом, составляющими элемента «Выход» будут являться две группы документов: «Выписки, рецепты, назначения», «Статистические и регламентные отчеты».

Представим результаты анализа деятельности медклиники в табличном виде (Таблица 2).

Таблица 2 – Значение элементов методологии SADT в соответствии с деятельностью медклиники

Элемент	Значение
Функциональный блок	Деятельность медклиники
Управление	«Нормативные документы», «Методики, правила»
Механизмы	«Персонал» и «Медицинское оборудование»
Вход	«Данные о сотрудниках», «Данные о закупках», «Данные пациентов»
Выход	«Выписки, рецепты, назначения», «Статистические и регламентные отчеты»

Графическое представление текстовой информации предназначено для ее дополнения или представления в более наглядном виде. Выполним графическое представление деятельности медклиники, в общем. Для этого предназначена контекстная диаграмма.

#### **Задание 1. Запуск программы и создание новой модели.**

1. Запустить **BPwin**.
2. Выбрать режим работы: создать новую модель (**Create model**).
3. В поле **Name** ввести имя модели: **Медклиника**.
4. Из группы **Type** выбрать тип диаграммы **Business Process (IDEF0)** (рис. 1).

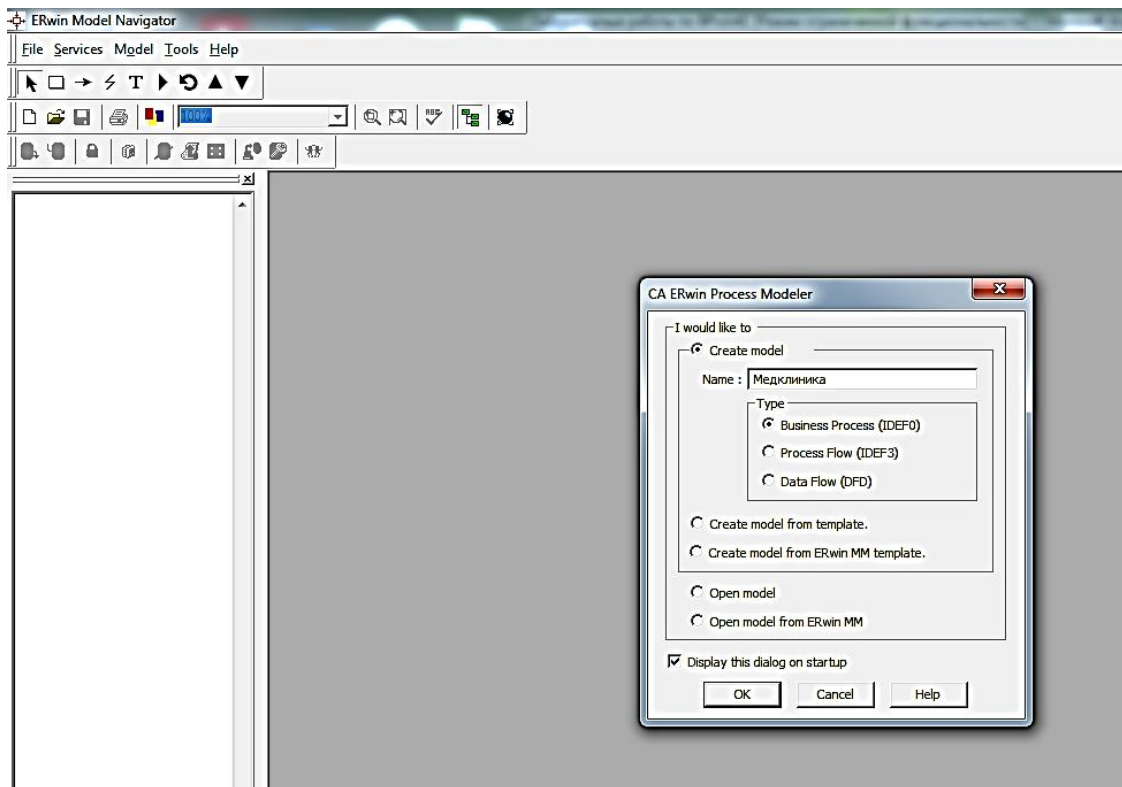


Рис.1 - Диалоговое окно создания модели

5. Нажмите ОК.

После щелчка на кнопке ОК появляется диалог **Properties for New Models** (рис. 2).

6. Укажите свою фамилию и инициалы.

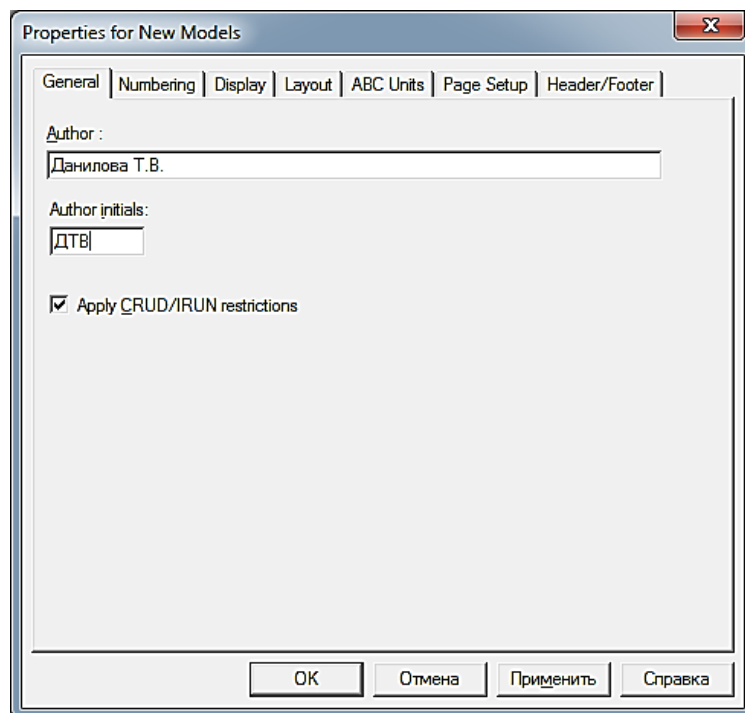


Рис.2 – Диалоговое окно Properties for New Models  
После этого вы увидите стандартное окно VPwin (рис. 3):

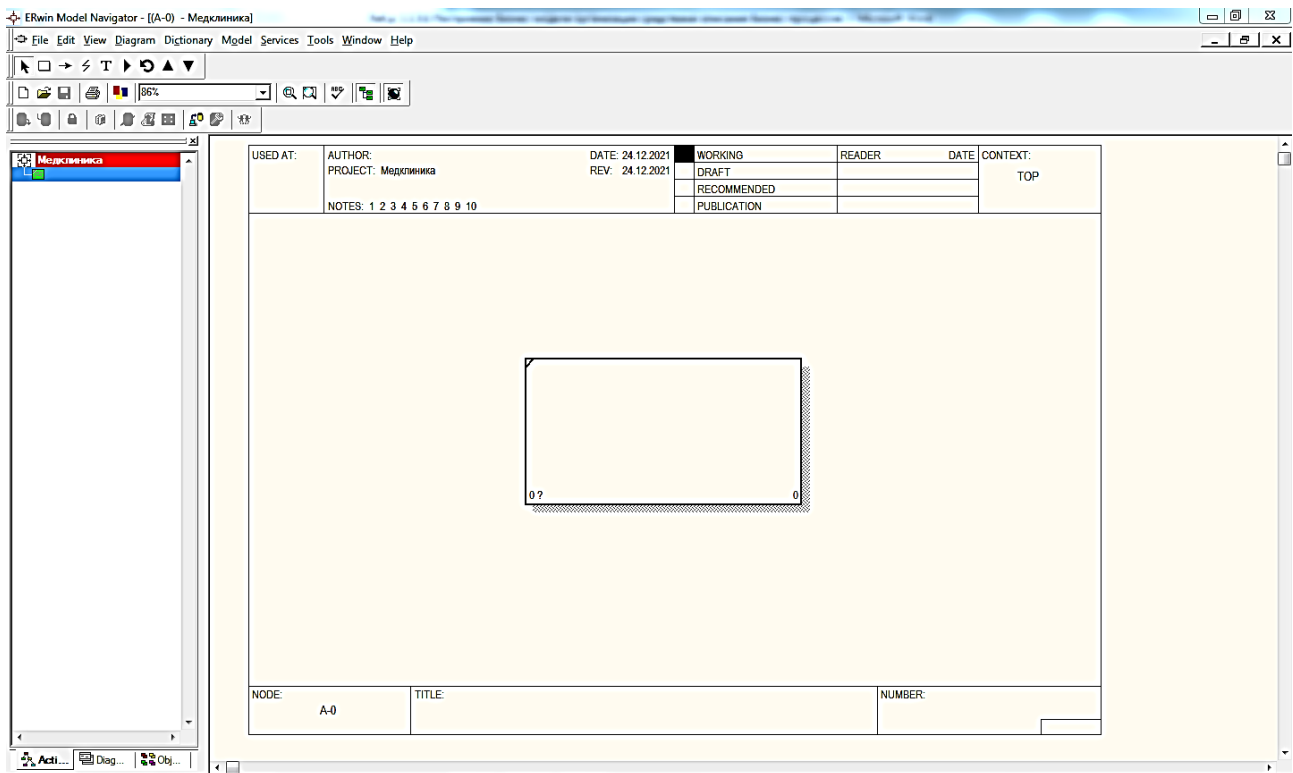


Рис. 3 - Стандартное окно VPrwin

Рассмотри интерфейс программы и основные инструменты VPrwin. Все основные действия с диаграммами, такие как создание, редактирование и т.д., можно выполнить с помощью главного меню, либо контекстно-зависимого меню (меню, появляющееся при нажатии правой кнопке мыши). Принципы работы с меню являются стандартными для среды Windows: объект сначала делается активным, затем над ним осуществляются необходимые действия.

На основной панели инструментов расположены элементы управления, в основном знакомые по другим Windows-интерфейсам (рис. 4):






Рис. 4 - Элементы управления VPrwin

Функциональность панели инструментов доступна из основного меню VPrwin (таблица 3).










Таблица 3 - Элементы управления основной панели инструментов VPrwin

Элемент управления	Описание	Пункт меню
	Создать новую модель	File/New
	Открыть модель	File/Open
	Сохранить модель	File/Save
	Печатать модель	File/Print
	Вызвать генератор отчетов	Tools/Report Builder
	Выбор масштаба	View/Zoom
	Масштабирование	View/Zoom

	Проверка правописания	Tools/Spelling
	Включение и выключение навигатора модели	View/Model Explorer
	Включение и выключение дополнительной панели инструментов	Model Mart

На основной панели инструментов (либо в любом желаемом месте экрана) расположены инструменты редактора VPwin для IDEF0-диаграмм (таблица 4).

Таблица 4 - Инструменты редактора VPwin

Элемент редактирования	Описание	Название элемента
	Выбор и определение позиции объектов, добавленных в диаграмму	Pointer Tool
	Установка блоков в диаграмме	Activity Box Tool
	Установка дуги в диаграмме	Arrow Tool
	Создание тильды, которая соединяет дугу с ее названием	Squiggle Tool
	Создание текстовых блоков	Text Block Tool
	Отображение следующей диаграммы того же уровня	Go to Sibling Diagram
	Отображение диалогового окна Diagram Dictionary Editor, где можно перейти на какую-либо диаграмму или создать новую диаграмму	Diagram Dictionary Editor
	Переход на родительскую диаграмму	Go to Parent Diagram
	Отображение диаграммы потомка или разложение выделенного блока на диаграмму потомка	Go to Child Diagram

## Задание 2. Создание контекстной диаграммы.

Сначала функциональность предприятия описывается в целом, без подробностей. Такое описание называется контекстной диаграммой. Для построения контекстной диаграммы нам необходимо определить:

- входную информацию (данные или материальные ресурсы), которая преобразуется в процессе для получения результата;
- выходную информацию - готовый результат;
- управление, которое влияет на процесс, но не преобразуется процессом.;



– механизмы, которые выполняют процесс.

Для ввода имени блока необходимо:

1. Щелкнуть правой клавишей мыши по блоку.
2. Выбрать команду **Name**.

В диалоговом окне ввести название «Деятельность медклиники» (рис.5.).

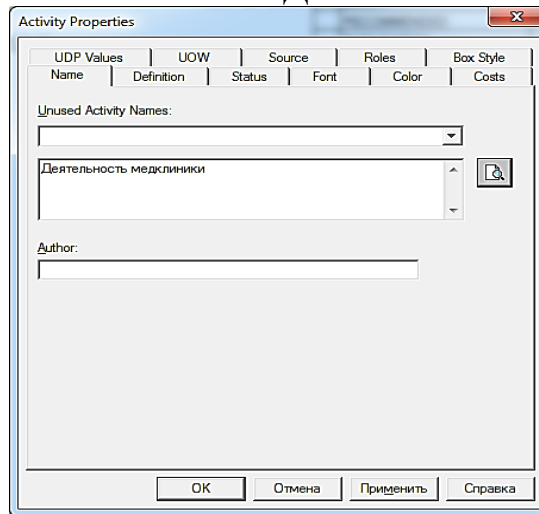



Рис. 5 – Присвоение блоку имени

### Технология построения дуг.

Для построения дуги необходимо выполнить следующие действия:

1. Выбрать на панели инструментов кнопку .
2. Подвести курсор к краю окна построения диаграммы до появления черной полосы и произвести щелчок левой кнопкой мыши по этой полосе.
3. Подвести курсор мыши к соответствующей стороне блока до образования темного треугольника и щелкнуть левой кнопкой мыши.

Пример построения дуги управления приведен на рис. 6.

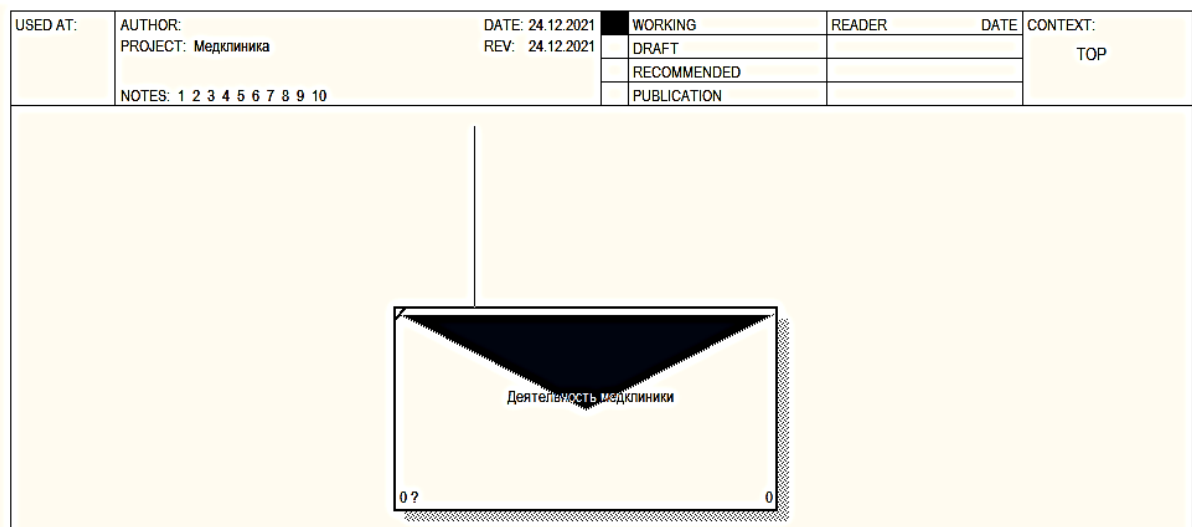



Рис. 6 – Построение дуги управления

Для идентификации дуги нужно выбрать на панели инструментов кнопку  и дважды щелкнуть по дуге левой кнопкой мыши. После этого

откроется окно, показанное на рис.7. В поле Arrow Name нужно ввести название дуги и нажать кнопку ОК.

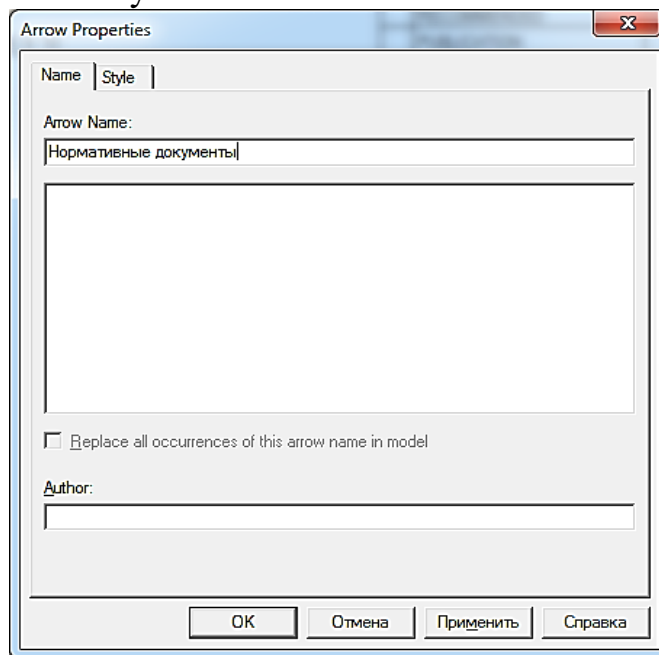


Рис. 7 – Идентификация дуги

После добавления и идентификации всех дуг контекстная диаграмма деятельности медицинской клиники будет выглядеть, как показано на рис.8. Для контекстного процесса «Деятельность медклиники» была определена информация:

- ВХОД – «Данные пациентов», «Данные сотрудников», «Данные о закупках»;
- УПРАВЛЕНИЕ – «Нормативные документы», «Методики, правила»;
- МЕХАНИЗМЫ – «Персонал», «Медицинское оборудование»;
- ВЫХОД – «Выписки, рецепты, назначения», «Статистические и регламентные отчеты».

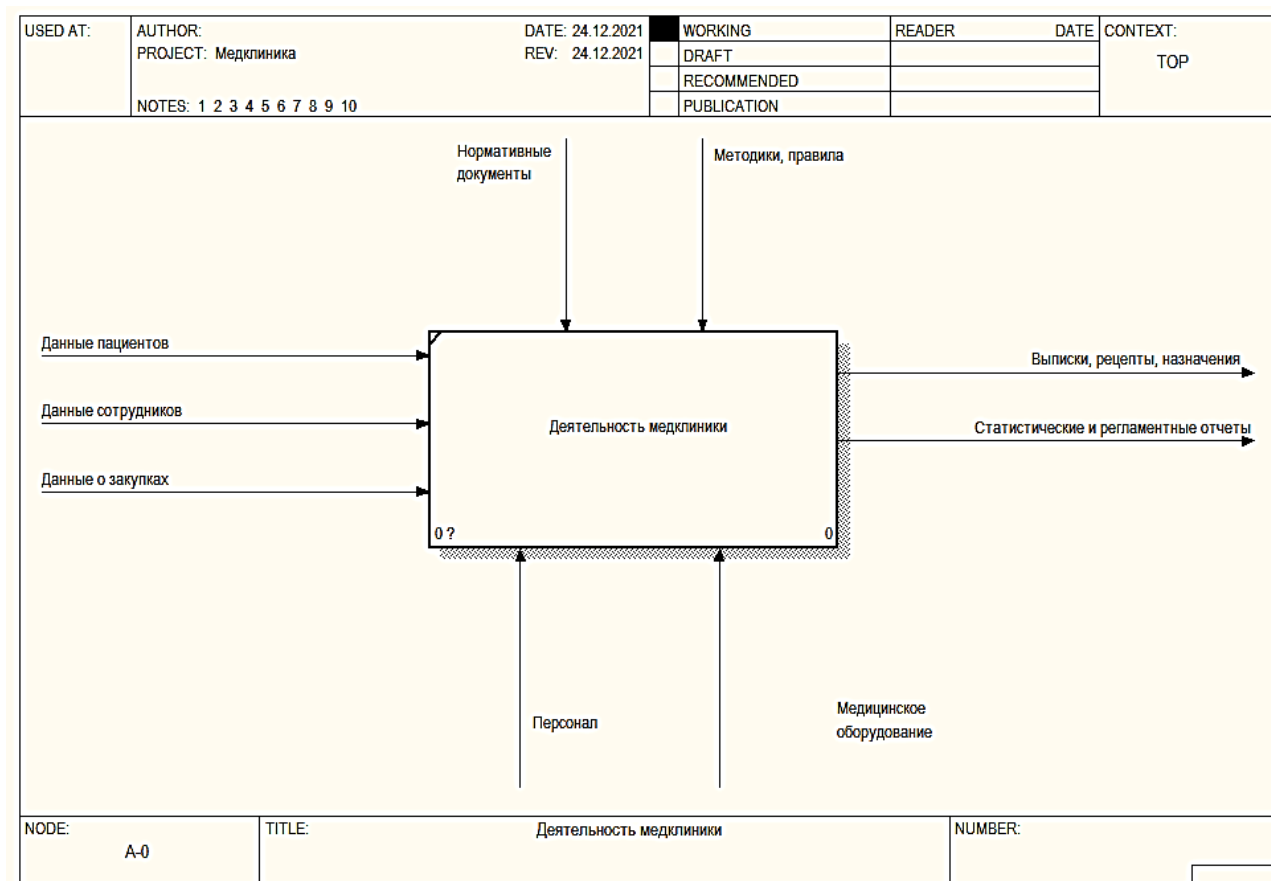


Рис. 8 – Контекстная диаграмма деятельности медицинской клиники

### Вопросы для самопроверки.

1. Моделирование бизнес-процессов. Дайте определение.
2. Нотация – что это?
3. Виды моделирования бизнес-процессов.
4. Что представляет собой структурное моделирование.
5. Методология структурного анализа и проектирования SADT.
6. Главная идея методологии SADT.
7. Элементы методологии SADT (синтаксис).
8. Что представляет собой модель IDEF0.
9. Этапы построения модели IDEF0.
10. В чем заключается смысл контекстной диаграммы.

#### Выполните тест:

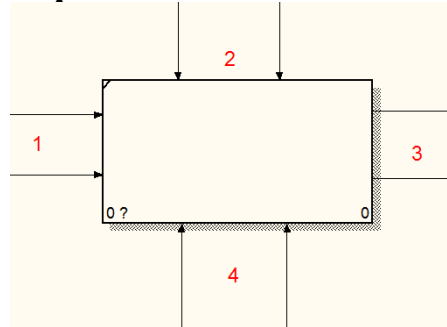
#### 1. Моделирование бизнес-процессов – это ...

- a) системные основы исследования или проектирования.
- b) формализованная процедура, подразумевающая создание некоторой формальной модели процесса, описанной на математическом или любом другом формализованном языке.
- c) систематическая процедура, применяемая для генерации описания системы с использованием соответствующих нотаций.

2. Систематическая процедура, применяемая для генерации описания системы с использованием соответствующих нотаций – это ...





- a) диаграмма.
- b) методология моделирования.
- c) метод.

3. Укажите, для чего предназначена левая сторона блока (1):

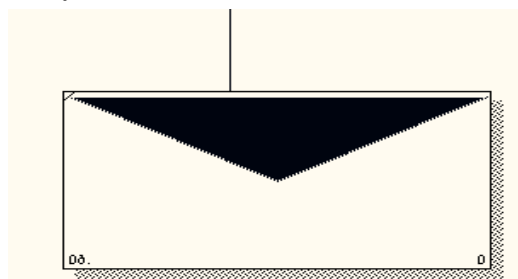


- a) это «Управление» - правила, стратегии, стандарты.
- b) это «Механизмы»- ресурсы, которые выполняют процесс.
- c) это «Вход» - материал или информация, которые используются или преобразуются для получения результата.
- d) это «Выход» - материал или информация, получаемые в результате выполнения процесса.

4. Каким инструментом на диаграмме изображается дуга?




- a) 
- b) 
- c) 
- d) 

5. Какая дуга будет построена в результате выполнения следующих действий?



- a) дуга управления
- b) дуга механизмов
- c) дуга входа
- d) дуга выхода





6. Включение и выключение дополнительной панели инструментов производится по кнопке:

- a) 
- b) 
- c) 

7. Для чего предназначен инструмент **T**:

- a) используется для создания тильды (squiggle), которая соединяет дугу с ее названием.
- b) используется для выбора и определения позиции объектов, добавленных в диаграмму.
- c) используется для создания текстовых блоков.
- d) используется для выбора цвета текста и фона.

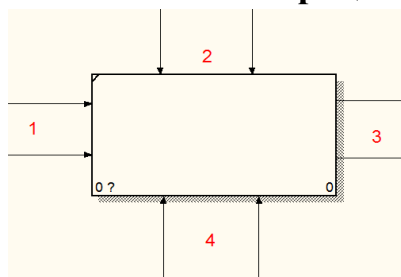
8. Отображение диалогового окна **Diagram Dictionary Editor**, где можно перейти на какую-либо диаграмму или создать новую диаграмму, выполняется по нажатию на кнопку:

- a) 
- b) 
- c) 
- d) 

9. Укажите этап, в котором происходит построение диаграммы процесса верхнего уровня:

- a) построение контекстной диаграммы.
- b) рисование дуги управления.
- c) определение основного бизнес-процесса.

10. С какой стороны блока указываются материалы или информация, получаемые в результате выполнения процесса:



- a) 1
- b) 2
- c) 3
- d) 4

Проверьте правильность ваших ответов по таблице 5.

Таблица 5 – Правильные ответы по тесту

Номер вопроса	Номер правильного ответа
1	b
2	c
3	c
4	b
5	a
6	b
7	c
8	b
9	a
10	c

### 1.1.3.2 Построение бизнес-модели в стандартах IDEF, UML.


#### Построение бизнес-модели в стандартах IDEF.

**Цель практической работы:** выполнение функциональной декомпозиции контекстной диаграммы, моделирование и представление в графическом виде бизнес-процессов медицинской клиники.

**Функциональная декомпозиция** – это следующий шаг после создания контекстной диаграммы, которая представляет собой описание контекста моделируемой системы. На этом шаге проводится разбиение системы более мелкие функциональные подсистемы, с описанием каждой в том же синтаксисе, что и система в целом. В результате детализации контекстного процесса создаются диаграммы верхнего уровня. Затем полученные подсистемы могут быть вновь разбиты на еще более, и так до достижения нужного уровня подробности. В результате этих действий, каждый фрагмент системы изображается на отдельной диаграмме декомпозиции.

В рассматриваемом примере деятельность медицинской клиники разбивается на такие бизнес-процессы, как:

1. Оказание амбулаторной медицинской помощи.
2. Консультативная деятельность врачей-специалистов.
3. Административно-управленческая деятельность.
4. Финансово-экономическая деятельность.
5. Материальное обеспечение лечебно-диагностического процесса.
6. Сбор медицинской статистики.
7. Формирование архива медицинских документов.

В программе Vrwip для выполнения декомпозиции нужно щелкнуть по блоку «Деятельность медклиники» и выбрать инструмент  или в меню, появившемся после нажатия на правую клавишу мыши, выбрать команду Decompose (рис. 9).

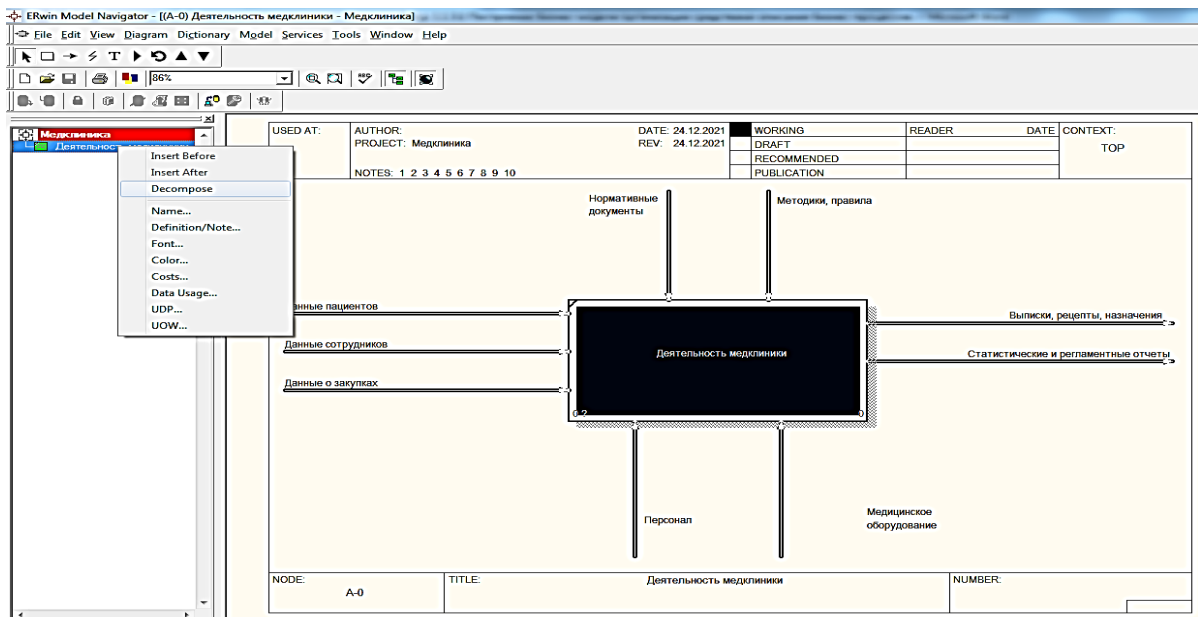


Рис. 9 – Выбор команды Decompose

Высветится в диалоговом окне, в котором требуется указать число блоков, на которое будет произведена декомпозиция (в данном примере – 7) и выбрать тип диаграммы IDEF0 из трех предложенных (рис. 10). Нажать ОК.

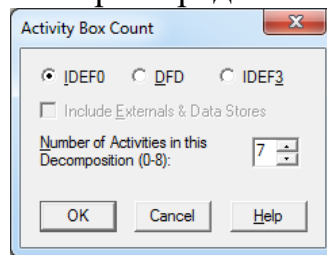



Рис. 10 – Диалоговое окно настройки декомпозиции

При декомпозиции функции происходит миграция дуг, т.е. дуги, входящие в функцию и исходящие из нее автоматически появляются на диаграмме декомпозиции. Но при этом дуги изображаются так, что не касаются блоков. Такие дуги называются несвязанными и считаются синтаксической ошибкой в ERwin (рис. 11). Для исправления этой ошибки или связывания дуг с соответствующими блоками нужно выполнить следующие действия:

1. Выбрать на панели инструмент  - Arrow Tool (Установка дуги в диаграмме).
2. Подвести указатель мыши к стрелочке на конце дуги и кликнуть по ней.
3. Затем, кликнуть по соответствующей стороне требуемого блока.
4. Для соединения с дугами Выход, наоборот, вначале нужно кликнуть по блоку, а затем по стрелке.

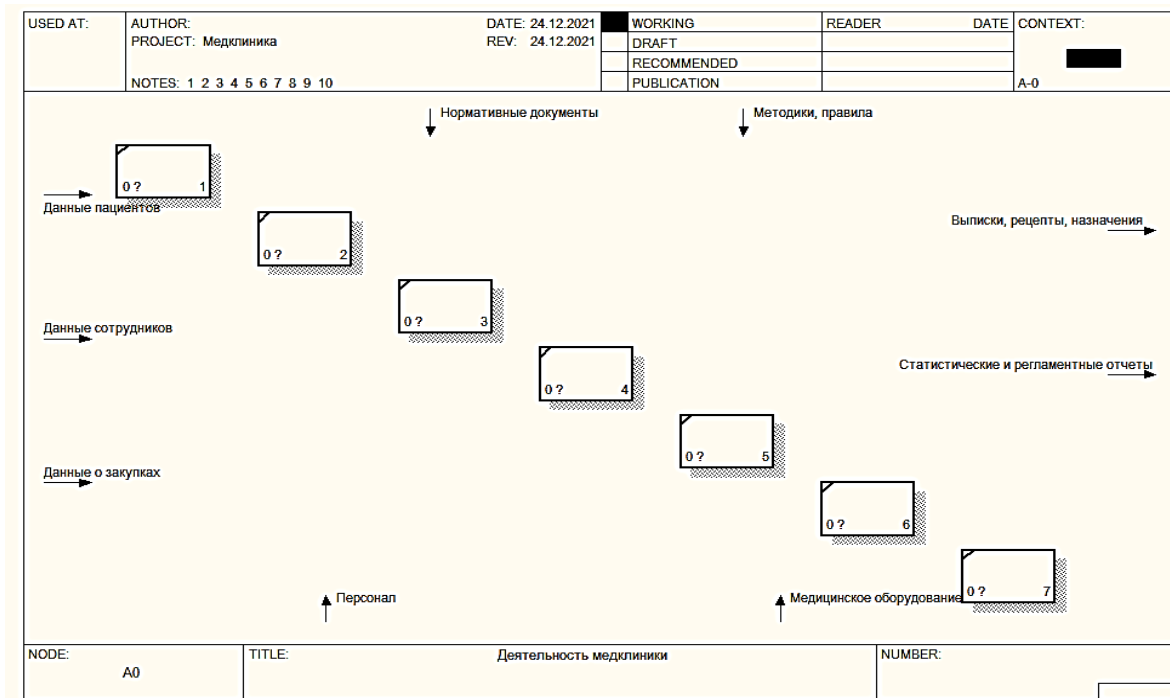


Рис. 11 – Блоки и дуги после декомпозиции контекстной диаграммы

Далее указываются названия новых блоков.

### Задание 3. Декомпозировать контекстную диаграмму.

Произвести декомпозицию на 7 блоков. Внести название каждого блока. Подвести соответствующие дуги, как показано на рисунке 12.

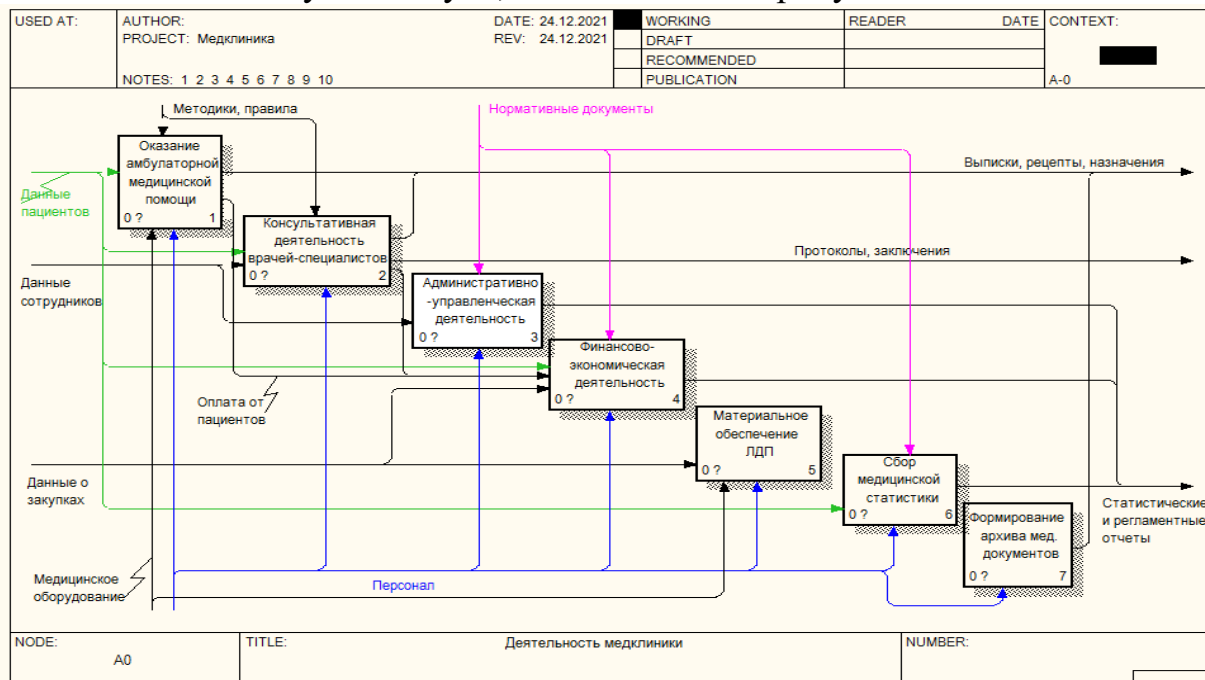



Рис. 12 – Бизнес-процессы медклиники

Название дуги является независимым объектом, который можно перемещать относительно дуги. Текст может располагаться по отношению к дуге в свободной форме, либо соединяться с дугой символом тильды.



Чтобы установить тильду нужно на панели инструментов нажать кнопку  и кликнуть левой кнопкой мыши по тексту, а затем по дуге (рис. 13);

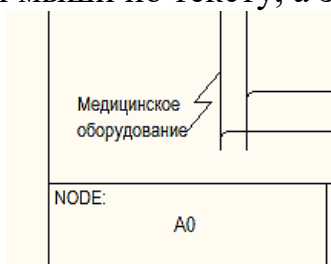


Рис. 13. Установление тильды

Для изменения цвета предназначена команда контекстно-зависимого меню Color (рис. 14):

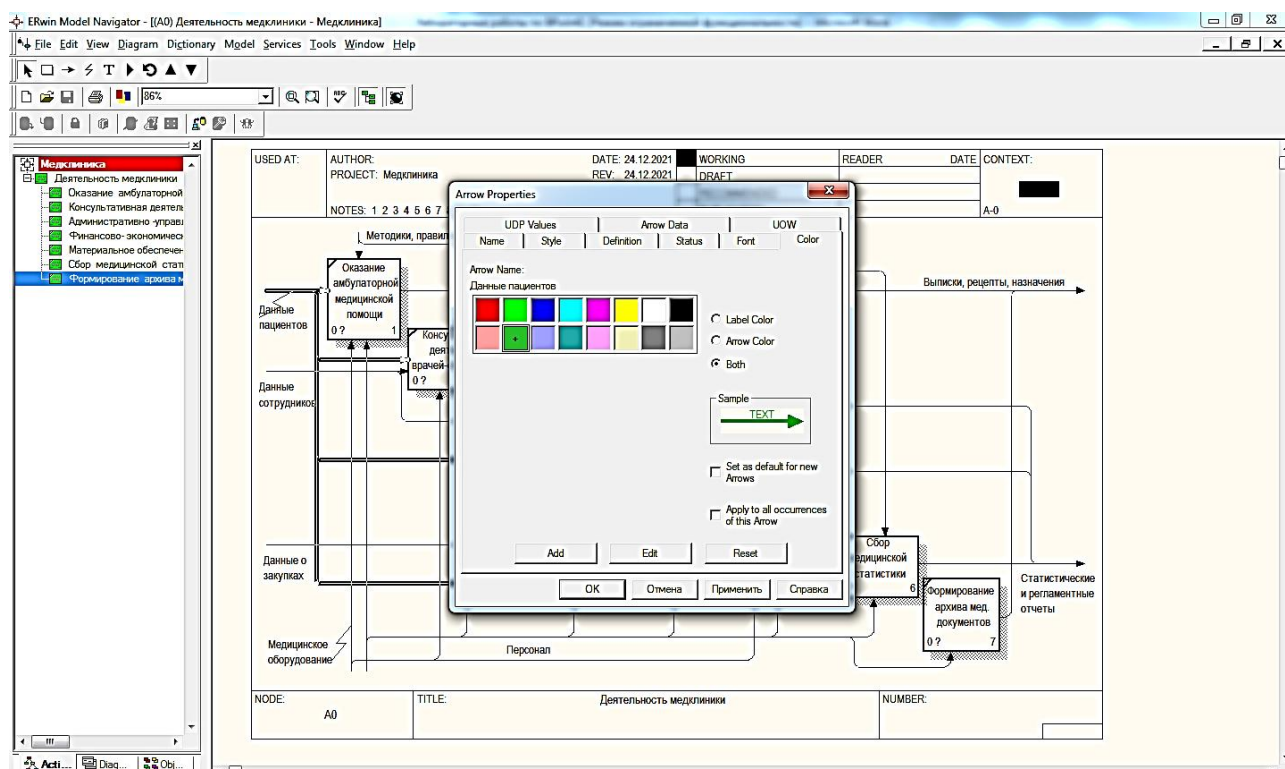


Рис. 14 –Изменения цвета дуг на диаграмме

Аналогично, воспользовавшись командой Color для предварительно выбранного блока, можно выделить его цветом (рис.15).

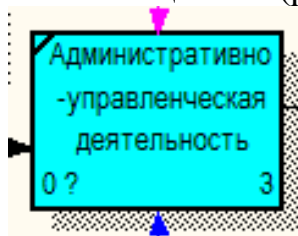


Рис. 15 – Изменение цвета блока

Для удаления блока и дуги или текста необходимо их активизировать щелчком левой кнопки мыши и нажать клавишу Delete, а затем подтвердить намерения по поводу удаления.

Способы редактирования объектов диаграмм представлены в таблице 6.

Таблица 6 - Способы редактирования объектов

Объект	Требуемое действие	Способ редактирования
Блок	Изменение размера	Для изменения высоты установить указатель мыши на верхнюю или нижнюю границу блока и при нажатой левой клавише потянуть вверх или вниз. Аналогично меняется размер по горизонтали
Текст	Стандартные способы форматирования	После клика правой клавишей мышки по тексту появится контекстно-зависимое меню, в котором нужно выбрать пункт <b>Font Editor</b> .
Дуга	Изменение размера	Кликнуть правой клавишей мышки по дуге и в контекстно-зависимом меню выбрать соответствующий пункт: <b>Trim</b>
Дуга	Изменение стиля	Кликнуть правой клавишей мышки по дуге и в контекстно-зависимом меню выбрать соответствующий пункт: <b>Style Editor</b>
Дуга	Изменение цвета	Кликнуть правой клавишей мышки по дуге и в контекстно-зависимом меню выбрать соответствующий пункт: <b>Color Editor</b>
Блоков, дуга, текст	Удаление	Активизировать нажатием левой кнопки мышки необходимый объект и нажать <b>Delete</b> , и в появившемся окне подтвердить запрос на удаление.

#### Задание 4. САМОСТОЯТЕЛЬНО.

Предложенные задания предназначены для самостоятельной разработки модели информационной системы. Номер варианта задания для каждого студента преподаватель назначает сам. В ходе выполнения задания студенты должны разработать модель информационной системы для автоматизации описанных бизнес-процессов:

- 1) построить контекстную диаграмму;
- 2) произвести функциональную декомпозицию;
- 3) построить диаграмму верхнего уровня.

В вариантах заданий приведены примерные перечни бизнес-процессов, которые студент может изменять и дополнять по своему усмотрению.

#### Вариант 1.

Разработать модель информационной системы для автоматизации библиотеки.

Описание бизнес-процессов:

1. Закупка произведений печати.

2. Хранение книг, журналов, газет и других изданий для общественного пользования.
3. Ведение картотеки читателей (регистрация, ввод и изменение данных).
4. Предоставление доступа к литературным, учебным и другим печатным произведениям (выбор, заказ, выдача, оформление возврата).
5. Учет недобросовестного обращения с библиотечной собственностью (потеря, порча, начисление возмещения ущерба и штрафа).
6. Списание печатной продукции.
7. Проведение культурно-массовых мероприятий (встреча с авторами, тематические чтения, олимпиады и конкурсы и др.).
8. Работы по переводу книжных изданий в цифровой формат.
9. Административная и финансовая деятельность библиотеки.

Деятельность библиотек регламентируется государственными нормативно-правовыми актами, Указами и распоряжениями Президента РФ в сфере государственной культурной политики.

### **Вариант 2.**

Разработать модель информационной системы для автоматизации отеля.

Описание бизнес-процессов:

1. Предоставление услуг размещения и проживания постояльцам отеля.
2. Информационное и бытовое обслуживание.
3. Организация экскурсий и активного отдыха.
4. Предоставление трансфера.
5. Организация питания.
6. Организационно-административная деятельность.
7. Ведение бухгалтерского и финансового учета.
8. Материально-техническое обеспечение.

Деятельность отеля регламентируется законодательными и нормативными документами, внутренними документами (кадровыми, распорядительными), документами, подтверждающими соответствие объекта санитарным и противопожарным нормативам и т.д.

### **Вариант 3.**

Разработать модель информационной системы для автоматизации деятельности ресторана.

Описание бизнес-процессов:

1. Закупка оборудования и продуктов.
2. Технологические процессы приготовления пищи.
3. Обслуживание гостей официантами.
4. Оформление и обслуживание мероприятий: свадеб, корпоративов.
5. Расчеты с посетителями.
6. Административная и кадровая работа.
7. Ведение бухгалтерского и финансового учета.
8. Проведение рекламных и маркетинговых мероприятий.

Деятельность отеля регламентируется законодательными и нормативными документами, внутренними документами (кадровыми, распорядительными), документами, подтверждающими соответствие объекта санитарным и противопожарным нормативам и т.д.

#### **Вариант 4.**

Разработать модель информационной системы для автоматизации контроля успеваемости студентов ВУЗа.

Описание бизнес-процессов:

1. Ведение и учет контингента обучающихся в учебном заведении.
2. Проведение занятий в соответствии с образовательными программами, учебными планами и расписанием.
3. Проведение текущей аттестации (проставление текущих оценок на занятиях, оценка тестов, курсовых и научных проектов, производственных практик).
4. Проведение промежуточной аттестации (контрольные точки, сессии).
5. Формирование, контроль заполнения и анализ аттестационных ведомостей.
6. Формирование приказов о движении студентов (перевод на следующий курс, отчисление).
7. Заполнение портфолио студентов.
8. Формирование итогового рейтинга студентов, статистической отчетности.

К документам, которыми руководствуется учебное заведение при контроле успеваемости, следует отнести Закон РФ «Об образовании», Положение самого вуза о проведении текущей и промежуточной аттестации и др.

#### **Вариант 5.**

Разработать модель информационной системы для автоматизации деятельности агентства недвижимости.

Описание бизнес-процессов:

1. Формирование и поддержание в актуальном состоянии базы данных объектов недвижимости.
2. Работа с клиентами (прием заявок на аренду, продажу и приобретение недвижимости).
3. Оказание услуг по показу объектов недвижимости клиентам (подбор вариантов по параметрам и пожеланиям).
4. Информационное и юридическое сопровождение сделок.
5. Сбор и проверка документов, необходимых при аренде, купле-продаже недвижимости.
6. Административно-управленческая деятельность.
7. Деятельность, обеспечивающая ведение бухгалтерского и финансового учета, предоставление отчетности в контролирующие органы.

Регулируется деятельность агентства нормативно-правовыми актами в сфере недвижимости.

**!** Выполнение самостоятельного задания способствует изучению программной среды VRwin и отработке навыков работы в ней.

Приступая к выполнению контрольного задания:

1. Определите нормативные документы, которыми руководствуется организация, применяемые стандарты.
2. Определите входную информацию (данные или материальные ресурсы), подлежащую обработке и преобразованию в процессе деятельности организации для получения результата.
3. Определите выходную информацию - готовый результат.
4. Укажите механизмы, которые будут задействованы в бизнес-процессах.
5. Запустите программу VRwin и создайте новый файл.
6. С помощью инструментов программы создайте контекстную диаграмму и, затем, выполните ее функциональную декомпозицию.
7. После выполненных действий сохраните файл в своей папке с именем **ФИО студента.br1**.

**!** Покажите преподавателю результаты, полученные в ходе выполнения самостоятельного задания.

#### **Вопросы для самопроверки:**




1. Как создается диаграмма верхнего уровня?
2. Как на диаграмме отображается декомпозиция?
3. Каким образом указывается количество блоков для декомпозиции?
4. В каком случае VRwin воспринимает стрелки, как синтаксическую ошибку?
5. Какую функцию выполняет на диаграмме символ тильда?



#### **Выполните тест:**

**1. В каком порядке надо располагать блоки на диаграмме детализации:**

- a) в произвольном;
- b) обязательно справа налево;
- c) в соответствии с уровнем сложности последующей детализации;
- d) в соответствии с их доминированием.

**2. Какой инструмент необходимо выбрать, чтобы с его помощью произвести декомпозицию блока?**

- a) ;
- b) ;
- c) .

- d) ;
- e) .

**3. Выберите правильные выражения:**

- a) Название дуги является независимым объектом, который можно произвольно перемещать относительно дуги.
- b) Текст может располагаться по отношению к дуге в свободной форме, либо соединяться с дугой символом «стрелка».
- c) Название дуги является зависимым объектом, который можно перемещать только вместе с дугой.
- d) Текст может располагаться по отношению к дуге в свободной форме, либо соединяться с дугой символом «тильда».

**4. Для чего предназначена диаграмма верхнего уровня:**

- a) для представления миграции дуг;
- b) для описания контекста моделируемой системы;
- c) для детализации работы.

**5. Для изменения размера дуги потребуется:**

- a) Щелкнуть правой клавишей мышки по дуге и выбрать соответствующий пункт меню: **Font Editor**
- b) Щелкнуть правой клавишей мышки по дуге и выбрать соответствующий пункт меню: **Trim**
- c) Щелкнуть правой клавишей мышки по дуге и выбрать соответствующий пункт меню: **Style**

Таблица 7 – Правильные ответы по тесту

Номер вопроса	Номер правильного ответа
1	b
2	d
3	a, d
4	c
5	b

**Построение бизнес-модели в стандартах UML.**

**Цель практической работы:** изучение языка моделирования UML, получение навыков построения диаграмм UML в программе RSA.

UML – унифицированный язык моделирования (Unified Modeling Language) – это система обозначений, которую можно применять для объектно-ориентированного анализа и проектирования. UML полезен для визуализации, спецификации, конструирования и документирования программных систем. Его также используют для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

UML является языком широкого профиля, это — открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью.

UML-модель представляет собой набор диаграмм двух типов (структурных и поведенческих), позволяющих кратко, но наглядно представить систему с различных точек зрения.

Существует достаточно много CASE-средств, поддерживающих язык UML. Одно из них - Rational Software Architect (RSA), который создан на расширяемой платформе Eclipse и предоставляет широкий диапазон инструментов проектирования и разработки, с помощью которых можно быстро создавать, оценивать и связывать архитектуры программного обеспечения и эскизы.

Используя RSA, построим диаграмму вариантов использования (диаграмма прецедентов). Достоинства этой диаграммы заключаются в том, что она:

- определяет пользователей и границы системы;
- определяет системный интерфейс;
- наглядно представляет участников и их действия в бизнес-процессах предприятия;
- хорошо вписывается в любые методы проектирования (как объектно-ориентированные, так и структурные).

Диаграмма вариантов использования - это графическая диаграмма, которая описывает функциональность рассматриваемой системы в виде графа, в вершинах которого расположены актёры/прецеденты, а связи между вершинами – разного вида отношения.

Основные элементы диаграммы вариантов использования:

- актёры - это субъект (система), который взаимодействует с моделируемой системой извне. Как выявляются актеры? Для этого необходимо задаться вопросом: кто участвует в бизнес-процессах, выполняет какие-либо значимые действия, предоставляет или получает информацию? В рассматриваемом примере деятельности медклиники можно выделить пациента, врача (медперсонал, выполняющий лечебные, диагностические и профилактические функции) и административный персонал (функции регистрации пациентов, управления медклиникой, формирования отчетности, материального обеспечения и финансовой деятельности). У каждого актера должно быть короткое, осмысленное с прикладной точки зрения имя.
- прецеденты (варианты использования) — множества последовательности действий (включая варианты), которые выполняются системой для того, чтобы актер мог получить определенный результат. Прецеденты всегда инициируются актером и описываются с точки зрения актера. Для выявления прецедента следует задаться вопросом: что делает актер? Какие действия

выполняет. У каждого прецедента должно быть короткое описательное имя, представляющее собой глагольную группу. Может случиться такое, что в процессе идентификации прецедентов могут выявиться новые актеры.

### Задание 5.

1. Для представления модели бизнес-процессов медклиники с помощью UML диаграммы вариантов использования запустить программу RSA.
2. Создайте новый UML проект и модель, выполнив следующие шаги:
  - Убедитесь, что вы находитесь в режиме Modeling perspective. Если нет, нажмите click Window > Open Perspective > Modeling
  - Нажмите File > New > Project (рис.16).

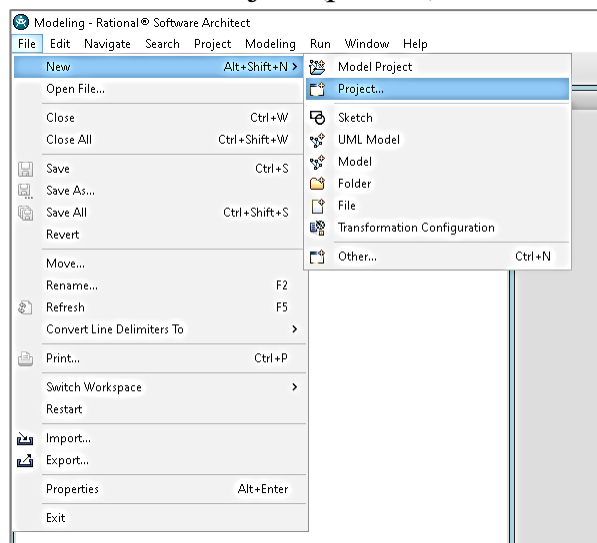


Рис. 16 – Создание нового проекта в RSA

- Находясь в Мастере, раскройте папку Modeling, выберите UML Project. Затем нажмите Next (рис. 17).

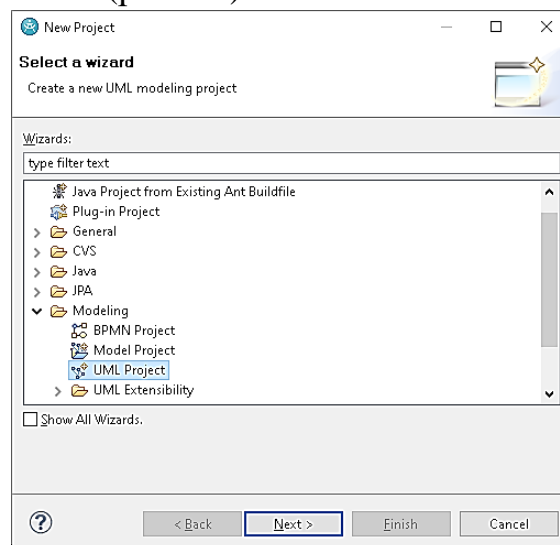


Рис. 17 – Выбор типа проекта

- Назовите проект "Medklinika", кликнете Next (рис. 18).



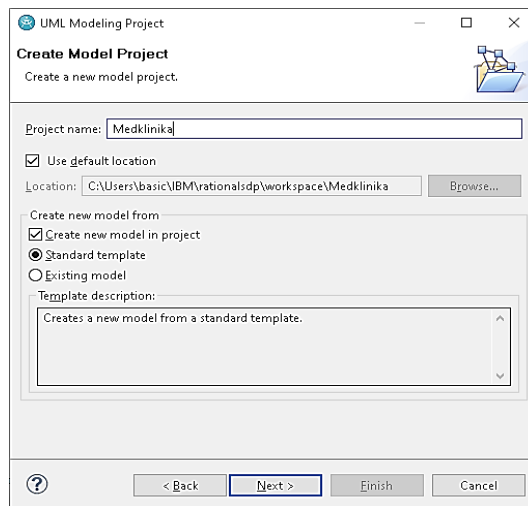


Рис. 18 – Задание имени проекта

- Выберите категорию General , шаблон Blank Package. Укажите имя файла модели "Analysis Model". Кликните Next.
- Выберите тип корневого пакета – Model, кликните Finish.
- Содержимое Project Explorer будет выглядеть следующим образом (рис. 19).

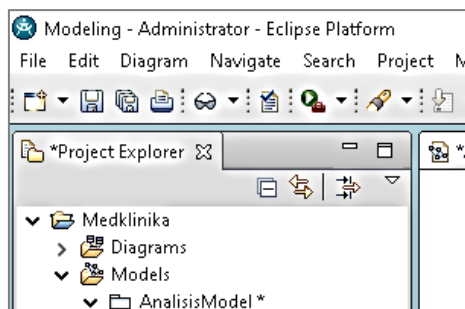


Рис. 19 - Содержимое Project Explorer

3. Создайте новый UML пакет в Project Explorer:
  - Находясь в Project Explorer, сделайте щелчок правой кнопкой мыши на Analysis Model и выберите Add UML > Package.
  - Заметьте, что новый пакет по умолчанию будет иметь имя "Package 1".
  - Введите 'Business Layer' в качестве имени нового пакета.
4. Создание пустой use case диаграммы
  - В Project Explorer, кликните правой кнопкой My Interaction Diagram Model (под узлом Models), и кликните Add Diagram > Use Case Diagram.
  - Новая use case диаграмма будет открыта. Если нет, дважды кликните на Usecase Diagram1 в Project Explorer , чтобы открыть ее.
5. Добавьте элементы на диаграмму. Выберите их в панели инструментов (например, актера) и при нажатой левой клавише мыши перетащите на область диаграммы (рис. 20).

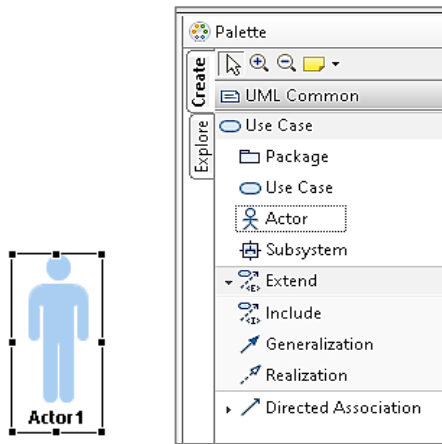


Рис. 20 – Добавление элементов на диаграмму  
 6. Добавляя на диаграмму актеров и варианты использования, разработайте диаграмму, представленную на рисунке 21.

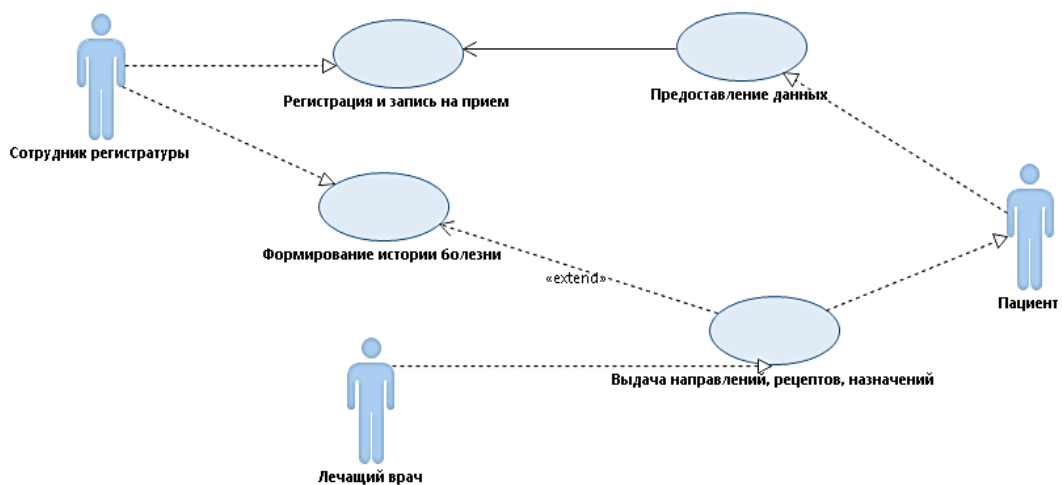
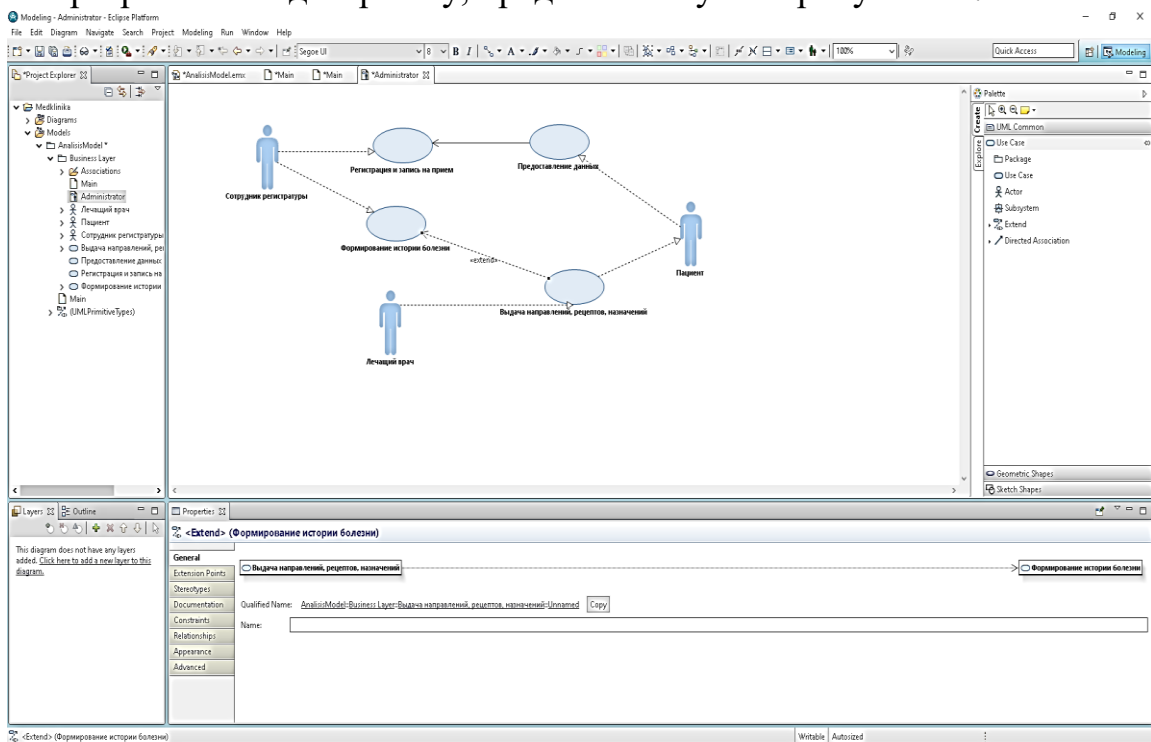


Рис. 21 – Диаграмма вариантов использования

! Сохраните проект с именем **ФИОстудентаUML**.

### Вопросы для самопроверки.

1. Что представляет собой UML-модель?
2. Что такое прецедент?
3. На какие типы делятся UML диаграммы?
4. Перечислите, какими, на ваш взгляд, достоинствами обладает диаграмма прецедентов?
5. Какие типы отношений применяются в диаграмме прецедентов?

### Выполните тест.

#### 1. UML – это унифицированный язык ...

- a) проектирования;
- b) программирования;
- c) моделирования.

#### 2. Графическое представление совокупности элементов модели в форме связного графа, вершинам и ребрам (дугам) которого приписывается определенная семантика – это...

- a) модель;
- b) диаграмма;
- c) нотация.

#### 3. Между компонентами диаграммы вариантов использования могут существовать различные отношения, которые описывают взаимодействие экземпляров одних актеров и вариантов использования с экземплярами других актеров и вариантов. На рисунке показано отношение:



- a) обобщения;
- b) ассоциации;
- c) включения;
- d) расширения.

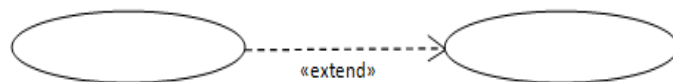
#### 4. На рисунке, представленном ниже, между актером и вариантом использования показано отношение:



- a) обобщения;

- b) ассоциации;
- c) включения;
- d) расширения.


5. **Фрагмент диаграммы вариантов использования на рисунке ниже иллюстрирует отношение:**



- a) обобщения;
- b) ассоциации;
- c) включения;
- d) расширения.

**Таблица 8 – Правильные ответы по тесту**

Номер вопроса	Номер правильного ответа
1	с
2	б
3	б
4	а
5	д

 Сохранить проект.

### 1.1.3.3 Построение бизнес-модели с помощью CASE-средств BPWin, UML

**Цель практической работы:** разбиение сложных бизнес-процессов на более мелкие и конкретные и визуализация их с помощью диаграмм декомпозиции второго уровня.

Последним шагом построения модели является функциональная декомпозиция, т.е. разбиение сложных процессов на более простые. Этот процесс декомпозиции продолжается до достижения нужного уровня подробности.

1. Оказание амбулаторной медицинской помощи.
  - Регистрация и запись на прием
  - Формирование документов
  - Ведение амбулаторных карт пациентов
  - Проведение диагностики
2. Консультативная деятельность врачей-специалистов.
  - Проведение консультаций
  - Проведение консилиумов
  - Оформление протоколов
  - Экспертиза качества медицинской помощи

3. Административно-управленческая деятельность.
  - Управление медицинским учреждением
  - Подбор персонала
  - Кадровый учет
  - Повышение квалификации врачей-специалистов
4. Финансово-экономическая деятельность.
  - Бухгалтерский учет
  - Планово-экономическая деятельность
  - Ведение договоров с пациентами
  - Взаиморасчеты за оказанные услуги
  - Оплата труда сотрудников
5. Материальное обеспечение лечебно-диагностического процесса.
  - Закупка и складской учет
  - Учет и обслуживание медицинского оборудования
  - Обеспечение стерилизационных процессов
  - Обеспечение лабораторных исследований
6. Сбор медицинской статистики.
  - Составление планов и программ исследований
  - Сбор статистических данных
  - Разработка и сводка материалов
  - Анализ полученных данных, выводы, предложения
7. Формирование архива медицинских документов.
  - Формирование историй болезни
  - Ведение архива медицинских документов

**Задание 6. Провести функциональную декомпозицию бизнес-процессов, в соответствии со структурой модели, описанной выше.**

Предварительно сохраните файл (в меню File выберите команду Save as) и сделайте его копию. Затем, откройте файл и произведите декомпозицию.

Пример декомпозиции процесса «Оказание амбулаторной медицинской помощи» показан на рис.15.

Управляющие стрелки и стрелки механизмов, указанные на диаграмме верхнего уровня должны быть и в диаграмме детализации.

Блоки на диаграмме также могут соединяться между собой дугами. Например, выходные данные бизнес-процесса «Регистрация и запись на прием» являются входными для процессов «Оформление документов» и «Ведение амбулаторных карт пациентов». А результаты, полученные на выходе процесса «Проведение диагностики» должны быть отражены в амбулаторной карте, поэтому дуга проведена на вход бизнес-процесса «Ведение амбулаторных карт пациентов».

Таким образом, на рисунке 22 в графическом виде представлена достаточно большая область деятельности медклиники, одна из самых важных и востребованных – это оказание амбулаторной медицинской помощи

пациентам. Первая инстанция, в которую обращается посетитель медицинской клиники – регистратура. После предъявления паспорта и действующего страхового полиса (стрелка входа – Данные пациентов) производится регистрация пациента. В зависимости от того, к какому специалисту пришел пациент, выбирается врач.

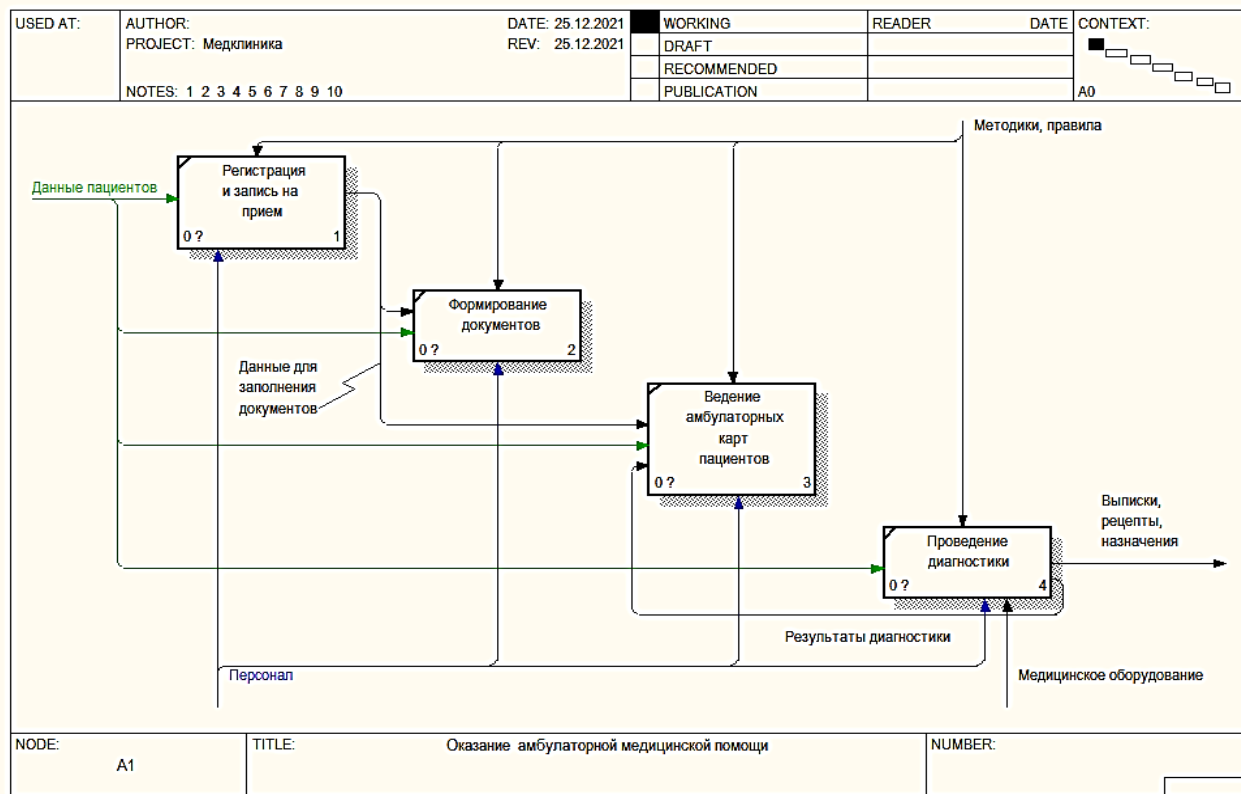


Рис. 22 – Диаграмма декомпозиции бизнес-процесса «Оказание амбулаторной медицинской помощи»

С посетителем согласовывается время посещения врача с учетом его часов приема и пожеланий пациента. Производится запись на прием. При этом выдается талон на посещение на будущее время. Если прием будет произведен сразу, то пациенту распечатывается талон амбулаторного приема. При проведении приема врач-специалист заполняет этот талон, занося в него жалобы пациента на самочувствие, данные первичного осмотра, предварительный диагноз, назначение процедур и лекарственных препаратов. При необходимости проведения дополнительного обследования с помощью анализов или медицинского оборудования (например, ультразвукового исследования или компьютерной томографии) пациенту выписываются соответствующие направления. Информация об этих направлениях отражается в талоне амбулаторного приема. В дальнейшем вся информация о приеме, назначениях врача, ходе лечения, результатах обследования и диагностики хранится в медицинской карте амбулаторного пациента.

Далее рассмотрен бизнес-процесс «Консультативная деятельность врачей-специалистов», который представлен в виде диаграммы декомпозиции 2-го уровня на рисунке 23.

Перед построением диаграммы предварительно опишем этот бизнес-процесс нормальным текстовым языком.

Иногда пациенту со сложным и редким заболеванием требуется консультация врача-специалиста, чья профессиональная компетенция выше, чем у его лечащего врача. Эта проблема решается проведением консультаций, которые зачастую оказываются платно. Также при определении точного диагноза и плана проведения лечебных мероприятий в сложных случаях собираются консилиумы врачей-специалистов. После проведения консилиума оформляется протокол.

При поступлении жалоб от пациентов на оказание медицинских услуг ненадлежащего качества, неисполнение врачом-специалистом своих должностных обязанностей, нанесение вреда жизни и здоровью пациентов проводится экспертиза качества медицинских услуг. Результатом работы специалистов, привлекаемых к проведению такой экспертизы, также являются протоколы или заключения.

Теперь переведем вышеописанное в графическую форму (рис. 23).

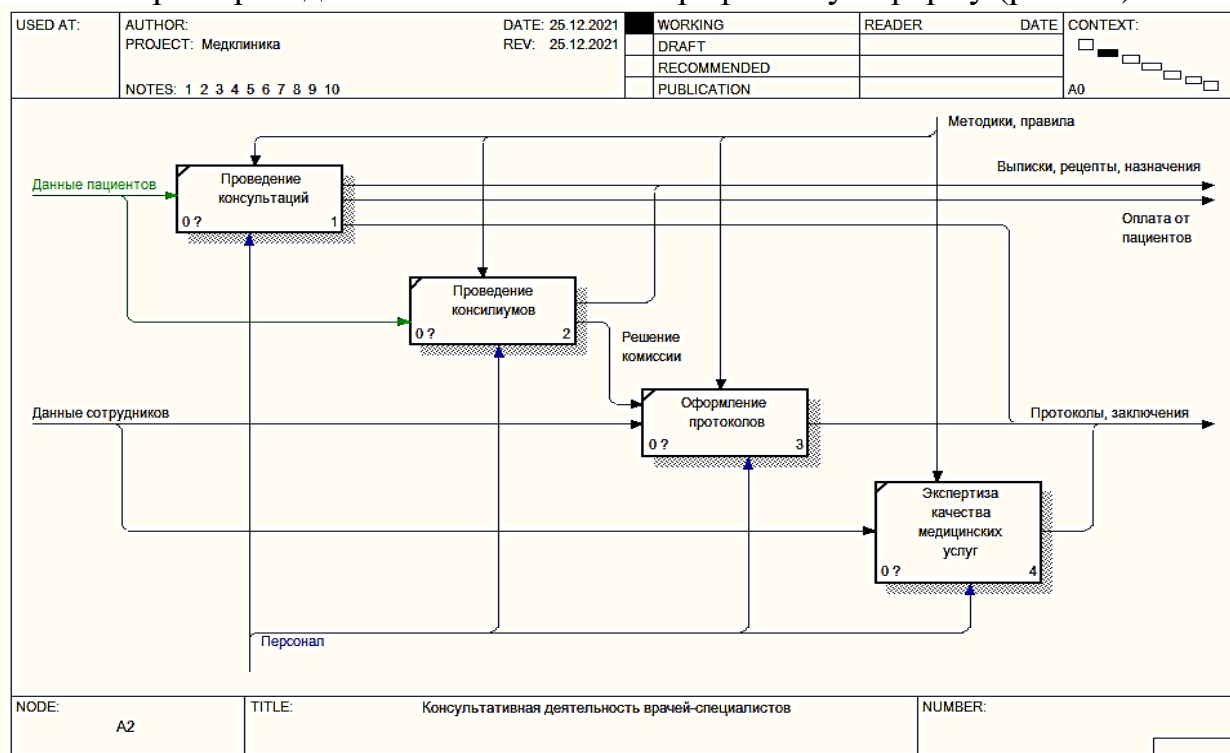


Рис. 23 – Диаграмма декомпозиции бизнес-процесса «Консультативная деятельность врачей-специалистов»

В процессе «Оформление протоколов» понадобилось построить новую граничную дугу «Протоколы, заключения». Она отобразилась с квадратными скобками (рис.24).

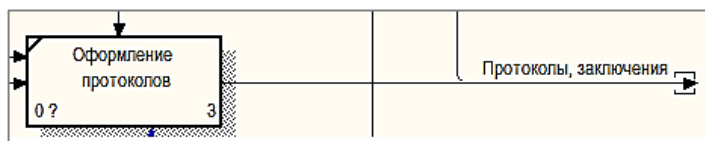



Рис. 24 – Новая дуга «Протоколы, заключения»

Вновь внесенные граничные дуги на диаграмме декомпозиции нижнего уровня изображаются в квадратных скобках и автоматически не появляются на диаграмме верхнего уровня.

Для их отображения наверх нужно:

1. Выбрать инструмент редактирования .
2. Кликнуть правой кнопкой мыши по квадратным скобкам.
3. Выбрать в контекстном меню пункт **Arrow Tunnel** (рис. 25).

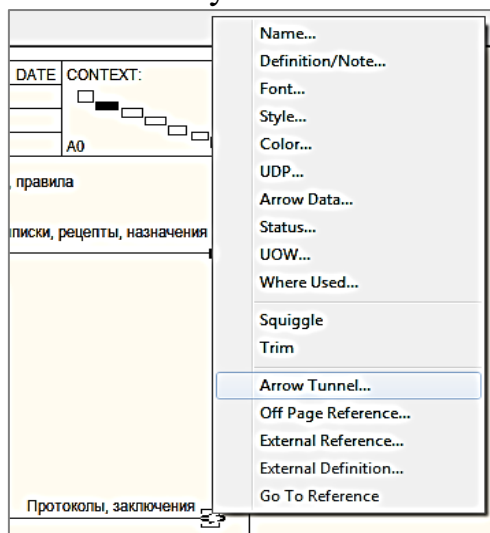


Рис. 25 – Выбор команды **Arrow Tunnel**

4. В появившемся диалоге **Border Arrow Editor** щелкнуть по кнопке **Resolve it to border arrow** для миграции стрелки на диаграмму верхнего уровня или по кнопке **Change it to resolved rounded tunnel** для «туннелирования» дуги.

Далее показаны диаграммы декомпозиции бизнес-процессов:

- «Административно-управленческая деятельность» (рис.26);
- «Финансово-экономическая деятельность» (рис.27);
- «Материальное обеспечение лечебно-диагностического процесса» (рис.28);
- «Сбор медицинской статистики» (рис.29);
- «Формирование архива медицинских документов» (рис.30).

Управление медицинским учреждением может включать в себя, как глобальные аспекты, например, формирование имиджа медицинского учреждения, поддержание деловой репутации и др., так и обычные текущие моменты, например, составление расписания работы врачей-специалистов, распределение кабинетов, организация профосмотров, юридическое урегулирование различных ситуаций с пациентами, обеспечение функционирования медучреждения в целом.

Особое место в этом процессе занимает подбор персонала. Без обеспечения кадрами соответствующего уровня компетенции оказание медицинских услуг, в принципе, невозможно.



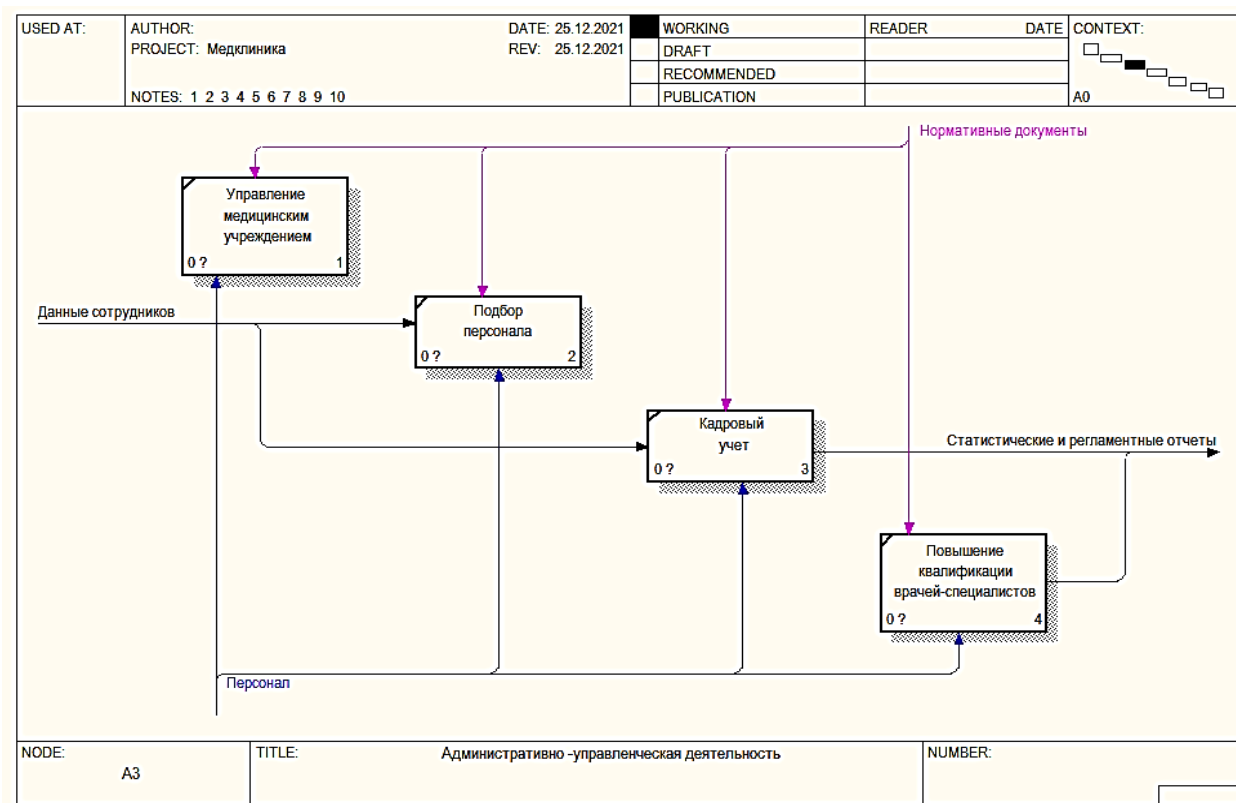


Рис. 26 – Диаграмма декомпозиции бизнес-процесса «Административно-управленческая деятельность»

Кадровый учет сотрудников клиники ведется в соответствии с российским трудовым законодательством. И, поскольку, медицинские работники на протяжении своей деятельности должны не реже 1-го раза в 5 лет проходить повышение квалификации, это отмечено на диаграмме отдельным функциональным блоком.

Продолжая описание бизнес-процессов, следует отметить, что в финансово-экономическую деятельность любого учреждения, в первую очередь, входит организация и ведение бухгалтерского учета. И медклиника не является исключением из этого правила. Ведется учет хозяйственных операций в соответствии с планом счетов, формируются и уплачиваются налоги, представляются отчеты в контролирующие органы.

Услуги клиники могут предоставляться платно и бесплатно, в соответствии со страховым полисом. Планируемые объемы предоставления платных услуг, калькуляция и обоснование их стоимости, квоты на услуги по медицинскому полису, планирование затрат на расширение направлений деятельности, внедрение новых технологий и другие процессы относятся к плано-экономической деятельности.

На оказание медицинских услуг, оплачиваемых пациентами, с ними заключаются договоры, на основании которых принимается оплата, и производятся взаиморасчеты.

В рамках плано-экономической деятельности формируется система оплаты труда, разрабатываются схемы мотивации персонала, производятся соответствующие начисления и выплаты.

Описанные процессы и представлены на диаграмме ниже (рис. 27).

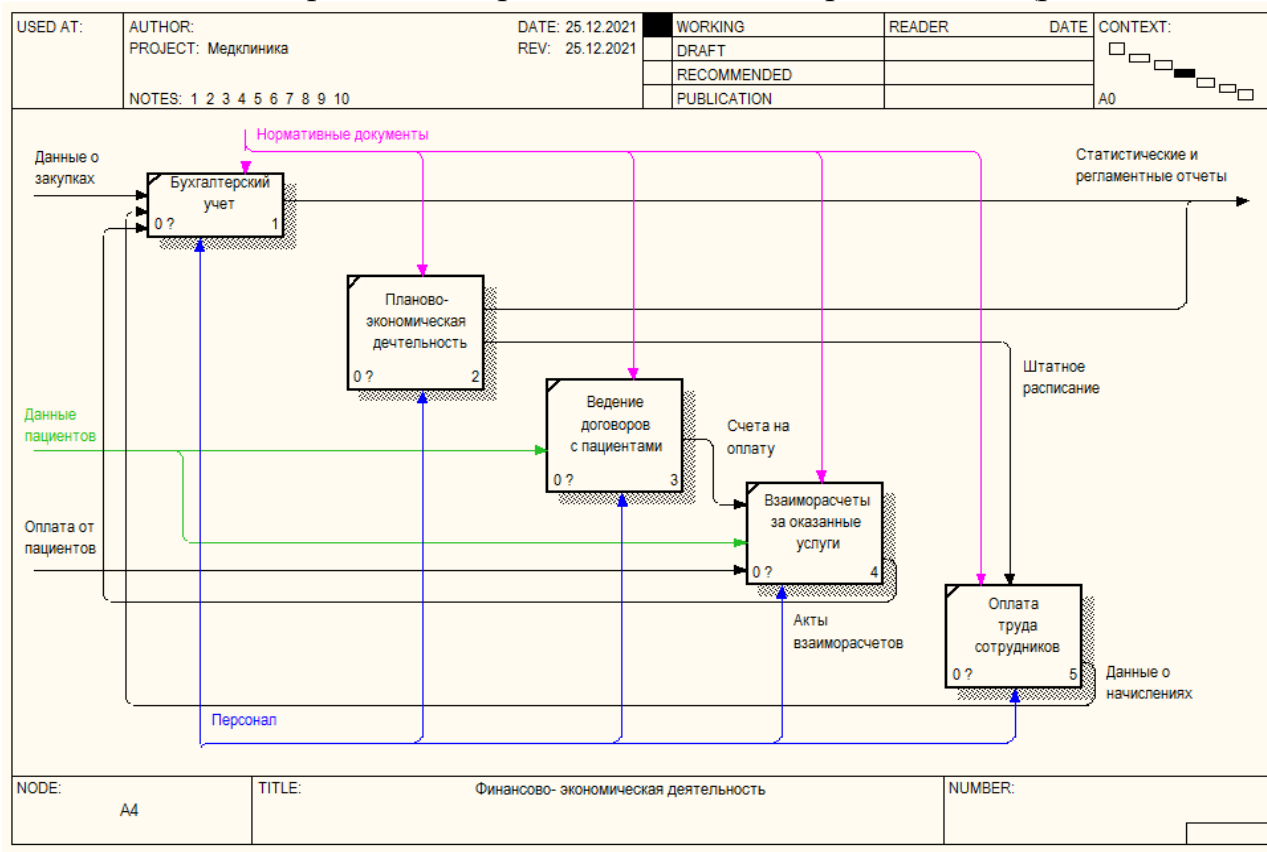


Рис. 27 - Диаграмма декомпозиции бизнес-процесса «Финансово-экономическая деятельность»

Бизнес-процесс «Материальное обеспечение лечебно-диагностического процесса (ЛДП)» включает в себя процессы, обеспечивающие механизмы деятельности медклиники. Это закупка, учет и обслуживание медицинского оборудования. Также немаловажным является обеспечение расходными материалами и проведение стерилизационных процессов используемых инструментов.

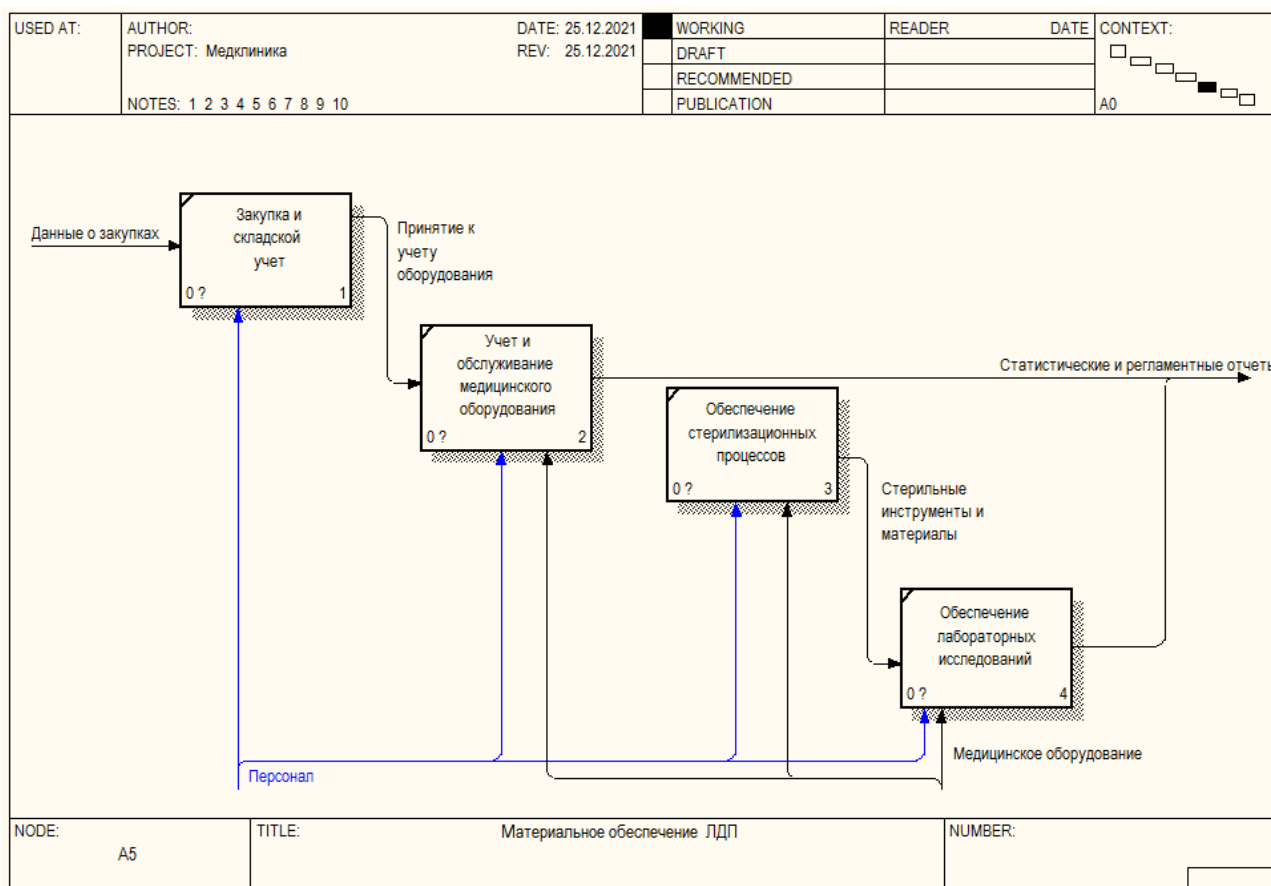


Рис.28 - Диаграмма декомпозиции бизнес-процесса «Материальное обеспечение лечебно-диагностического процесса (ЛДП)»

Краткое и лаконичное представление бизнес-процесса «Сбор медицинской статистики» скрывает за собой кропотливую работу по сбору и обработке огромных массивов данных о состоянии здоровья населения, росте или динамике заболеваний различными болезнями.

Статистические данные могут свидетельствовать о научных достижениях в различных областях медицины, эффективности лечебных, профилактических, оздоровительных мер, предпринятых для борьбы с тем или иным заболеванием.

Также на основе статистических данных строятся прогнозы, разрабатываются программы научных исследований, планы предпринимаемых мер для предотвращения или снижения профессиональных заболеваний, сезонных заболеваний, уровня заболеваемости в период эпидемий и пандемий.

Сбор и обработка статистических данных является обязательным процессом для всех медицинских учреждений.

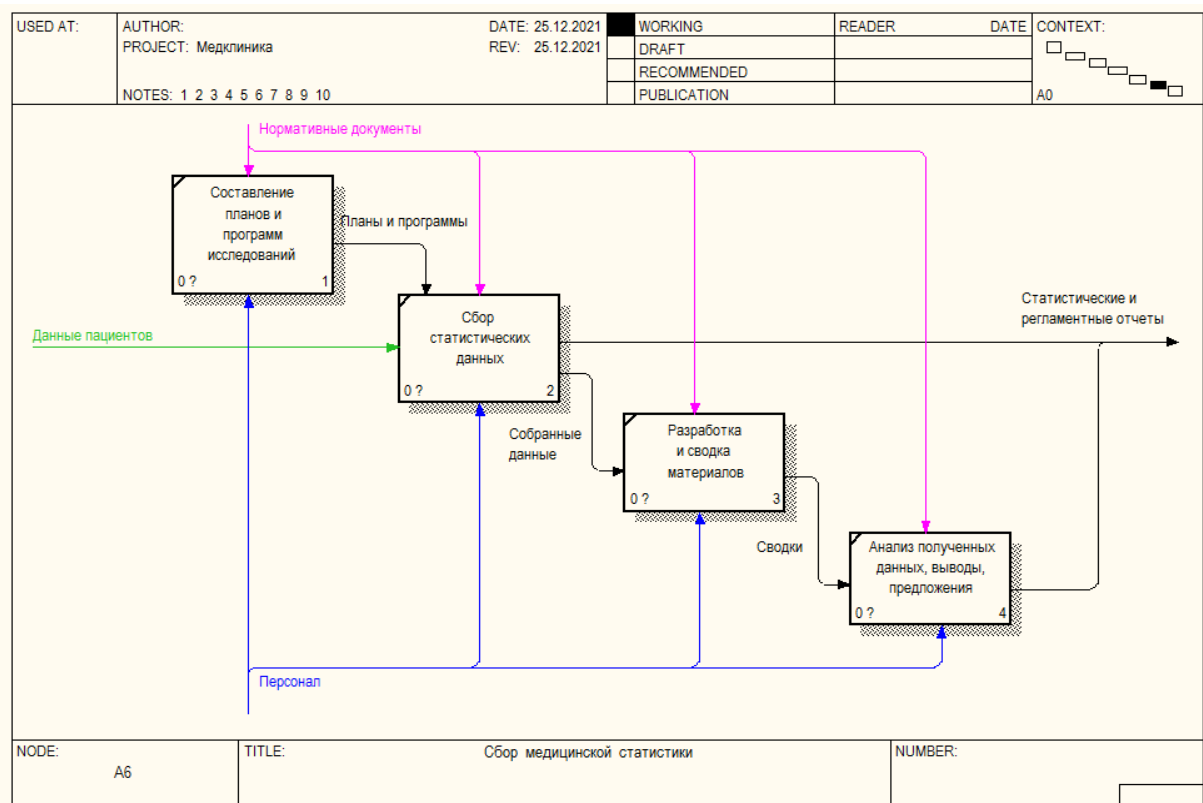


Рис.29 - Диаграмма декомпозиции бизнес-процесса «Сбор медицинской статистики»

Медицинская карта амбулаторного пациента в соответствии с законодательством РФ должна храниться 25 лет. Поэтому формирование архива медицинских документов является неотъемлемым процессом деятельности медицинского учреждения любой формы собственности (государственной, частной).

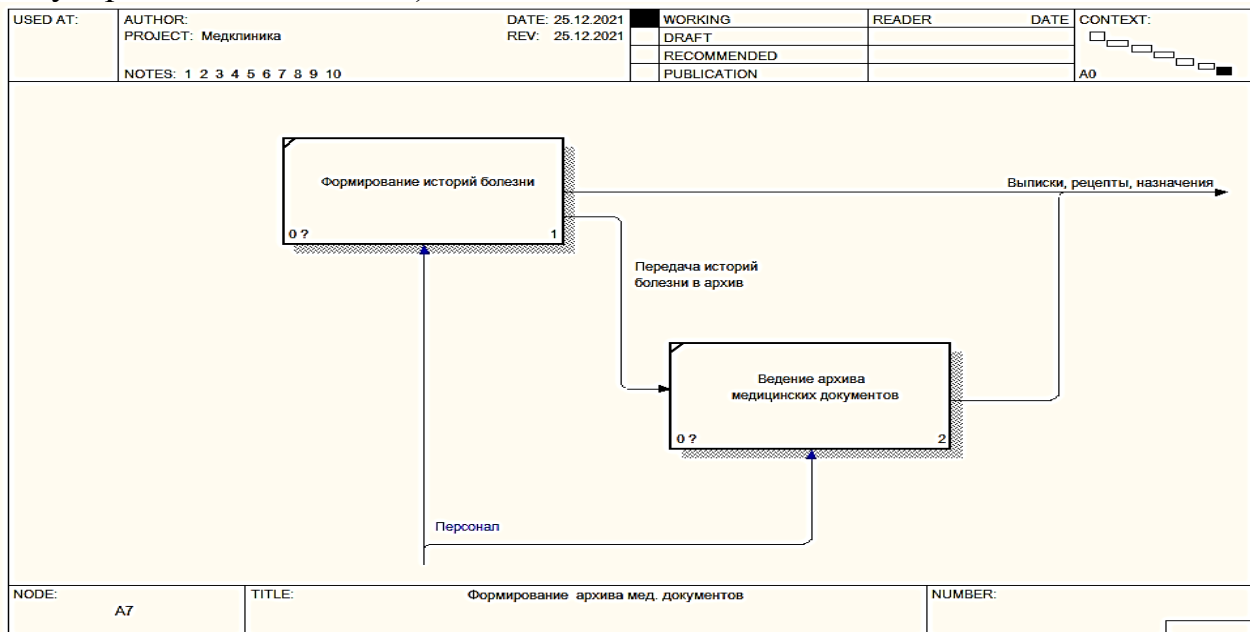


Рис.30 - Диаграмма декомпозиции бизнес-процесса «Формирование архива медицинских документов»

Подробное описание бизнес-процессов способствует более точной разработке диаграмм в нотации IDEF3 в следующем задании.

### **Задание 7. Построить диаграмму IDEF3.**

Методология моделирования IDEF3, называемая также **workflow diagramming** – это методология, использующая графическое описание информационных потоков, взаимоотношений между процессами обработки информации и объектов, являющихся частью этих процессов.

Диаграммы **Workflow** могут быть использованы в моделировании бизнес-процессов для анализа завершенности процедур обработки информации. С их помощью можно описывать сценарии действий сотрудников организации, например последовательность обработки заказа или события, которые необходимо обработать за конечное время. Каждый сценарий сопровождается описанием процесса и может быть использован для документирования каждой функции.

IDEF3 - это метод, имеющий основной целью дать возможность аналитикам описать ситуацию, когда процессы выполняются в определенной последовательности, а также описать объекты, участвующие совместно в одном процессе.

Каждая работа в IDEF3 описывает какой-либо сценарий бизнес-процесса и может являться составляющей другой работы. Поскольку сценарий описывает цель и рамки модели, важно, чтобы работы именовались отглагольным существительным, обозначающим процесс действия, или именным словосочетанием, содержащим такое существительное.

Точка зрения на модель должна быть задокументирована. Обычно это точка зрения человека, ответственного за работу в целом. Также необходимо задокументировать цель модели - те вопросы, на которые призвана ответить модель.

Диаграмма является основной единицей описания в IDEF3. Важно правильно построить диаграммы, поскольку они предназначены для чтения другими людьми (а не только автором).

Единицы работы – **Unit of Work (UOW)**, также называемые работами (activity), являются центральными компонентами модели. В IDEF3 работы изображаются прямоугольниками с прямыми углами (рис. 31) и имеют имя, выраженное отглагольным существительным, обозначающим процесс действия, одиночным или в составе словосочетания, и номер (идентификатор); другое имя существительное в составе того же словосочетания, зависимое от отглагольного существительного, обычно отображает основной выход (результат) работы (например, "Регистрация обращения").

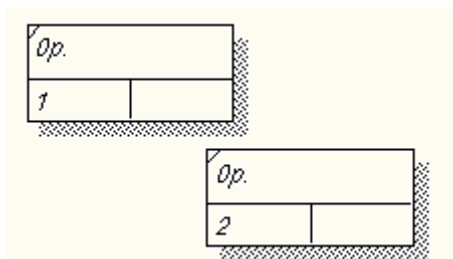


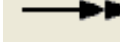


Рис. 31 - Обозначение работы в диаграмме IDEF3

**Связи** показывают взаимоотношения работ. Все связи в **IDEF3** однонаправлены и могут быть направлены куда угодно, но обычно диаграммы IDEF3 стараются построить так, чтобы связи были направлены слева направо. В IDEF3 различают три типа стрелок, изображающих связи, стиль которых устанавливается во вкладке **Style** диалога **Arrow Properties** (пункт контекстного меню **Style**).

**Старшая (Precedence) стрелка**  **Precedence** - сплошная линия, связывающая единицы работ (UOW). Рисуется слева направо или сверху вниз. Показывает, что работа-источник должна закончиться прежде, чем работа-цель начнется.

**Стрелка отношения (Relational)**  **Relational** - пунктирная линия, используемая для изображения связей между единицами работ (UOW), а также между единицами работ и объектами ссылок.

**Потоки объектов (Object Flow)**  **Object flow** - стрелка с двумя наконечниками, применяется для описания того факта, что объект используется в двух или более единицах работы, например, когда объект порождается в одной работе и используется в другой.

**Старшая связь** показывает, что работа-источник заканчивается ранее, чем начинается работа-цель. Часто результатом работы-источника становится объект, необходимый для запуска работы-цели. В этом случае стрелку, обозначающую объект, изображают с двойным наконечником. Имя стрелки должно ясно идентифицировать отображаемый объект. Поток объектов имеет ту же семантику, что и старшая стрелка.

**Отношение** показывает, что стрелка является альтернативой старшей стрелке или потоку объектов в смысле задания последовательности выполнения работ - работа-источник не обязательно должна закончиться прежде, чем работа-цель начнется. Более того, работа-цель может закончиться прежде, чем закончится работа-источник. Временные варианты работ представлены в таблице 9

Таблица 9 - Временные варианты работ

Начало работы источника	Окончания работы источника	Начало работы цели	Окончание работы цели
Начало работы источника	Начало работы цели	Окончания работы источника	Окончание работы цели

Начало работы источника	Начало работы цели	Окончание работы цели	Окончания работы источника
-------------------------	--------------------	-----------------------	----------------------------

**Перекрестки (Junction).** Окончание одной работы может служить сигналом к началу нескольких работ, или же одна работа для своего запуска может ожидать окончания нескольких работ. Перекрестки используются для отображения логики взаимодействия стрелок при слиянии и разветвлении или для отображения множества событий, которые могут или должны быть завершены перед началом следующей работы.

Различают перекрестки для слияния (**Fan-in Junction**) и разветвления (**Fan-out Junction**) стрелок. Перекресток не может использоваться одновременно для слияния и разветвления.


Для внесения перекрестка служит кнопка  в палитре инструментов. Значения типов перекрестков приведены в таблице 10.

Таблица 10 - Значения типов перекрестков

Графическое обозначение	Название перекрестка	Действия при слиянии стрелок Fan-in Junction	Действия при разветвлении стрелок Fan-out Junction
Asynchronous:  AND	Асинхронное «И» (Asynchronous AND)	Все предшествующие процессы должны быть завершены	Все следующие процессы должны быть запущены
Synchronous:  AND	Синхронное «И» (Synchronous AND)	Все предшествующие процессы завершены одновременно	Все следующие процессы запускаются одновременно
Asynchronous:  OR	Асинхронное «ИЛИ» (Asynchronous OR)	Один или несколько предшествующих процессов должны быть завершены	Один или несколько следующих процессов должны быть запущены
Synchronous:  OR	Синхронное «ИЛИ» (Synchronous OR)	Один или несколько предшествующих процессов завершены одновременно	Один или несколько следующих процессов запускаются одновременно
Exclusive OR:  XOR	Исключающее «ИЛИ» XOR (Exclusive OR)	Только один предшествующий процесс завершен	Только один следующий процесс запускается

Все перекрестки на диаграмме нумеруются, каждый номер имеет префикс **J** (рис. 32).

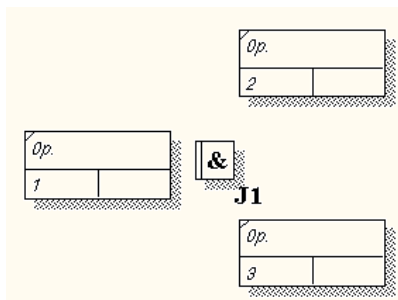


Рис. 32 - Обозначение нумерации перекрестка

В диаграммах IDEF3 стрелки могут сливаться и разветвляться только через перекрестки.

Существуют определенные правила создания перекрестков. На одной диаграмме IDEF3 может быть создано несколько перекрестков различных типов. Определенные сочетания перекрестков для слияния и разветвления могут приводить к логическим несоответствиям. Чтобы избежать конфликтов, необходимо соблюдать следующее:

1. Каждому перекрестку для слияния должен предшествовать перекресток для разветвления.
2. Перекресток для слияния «И» не может следовать за перекрестком для разветвления типа *синхронного* или *асинхронного* «ИЛИ».
3. Перекресток для слияния «И» не может следовать за перекрестком для разветвления типа *исключающего* «ИЛИ».
4. Перекресток для слияния типа *исключающего* «ИЛИ» не может следовать за перекрестком для разветвления типа «И».

Для создания диаграммы IDEF3 бизнес-процесса «Регистрация и запись на прием» необходимо выполнить следующие действия:

1. Перейти на диаграмму «Оказание амбулаторной медицинской помощи» и декомпозировать работу «Регистрация и запись на прием».
2. В диалоге **Activity Box Count** установите число работ 6 и нотацию **IDEF3**. Появится диаграмма **IDEF3**, содержащая работы (**UOW**).
3. Дважды щелкнув левой клавишей мыши по работе, внести в окне **Activity Properties** имя работы (рис. 33).

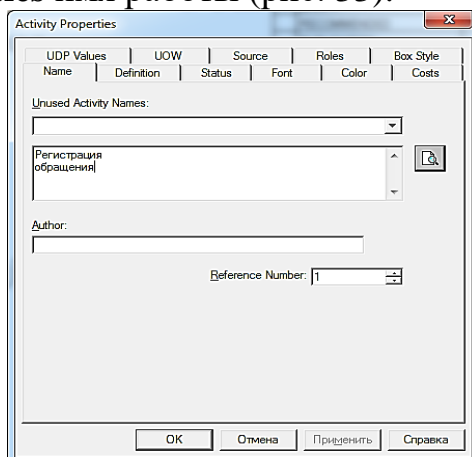


Рис. 33 – Ввод имени работы



4. Во вкладку **UOW**, вносятся свойства работы (рис. 34).

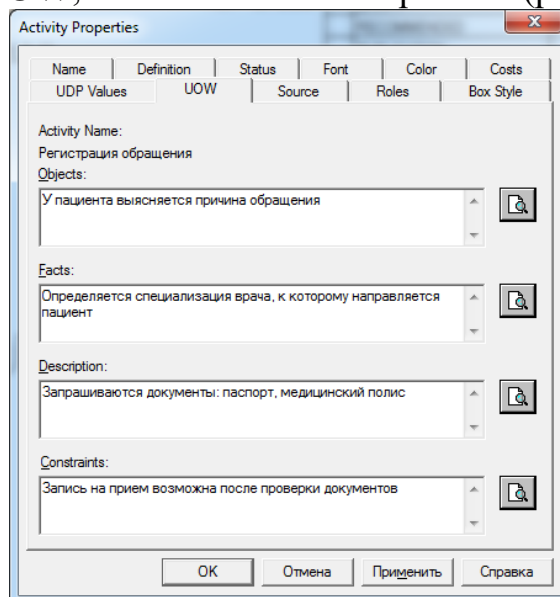





Рис. 34 – Внесение свойств работы

5. Так как работы «Регистрация обращения» и «Проверка документов» должны следовать только друг за другом, они соединяются стрелкой **Precedence**   Precedence, которая показывает, что работа-источник должна закончиться прежде, чем работа-цель начнется

6. С помощью кнопки  на палитре инструментов вносятся перекрестки, которые выбираются в окне **Select Junction Type** (рис. 35).

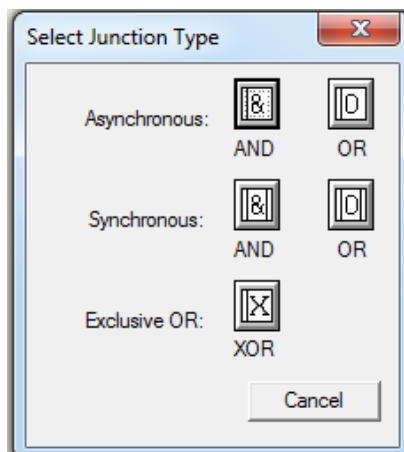


Рис. 35 – Выбор типа перекрестка

На рисунке 36 представлена IDEF3 диаграмма бизнес-процесса «Регистрация и запись на прием».

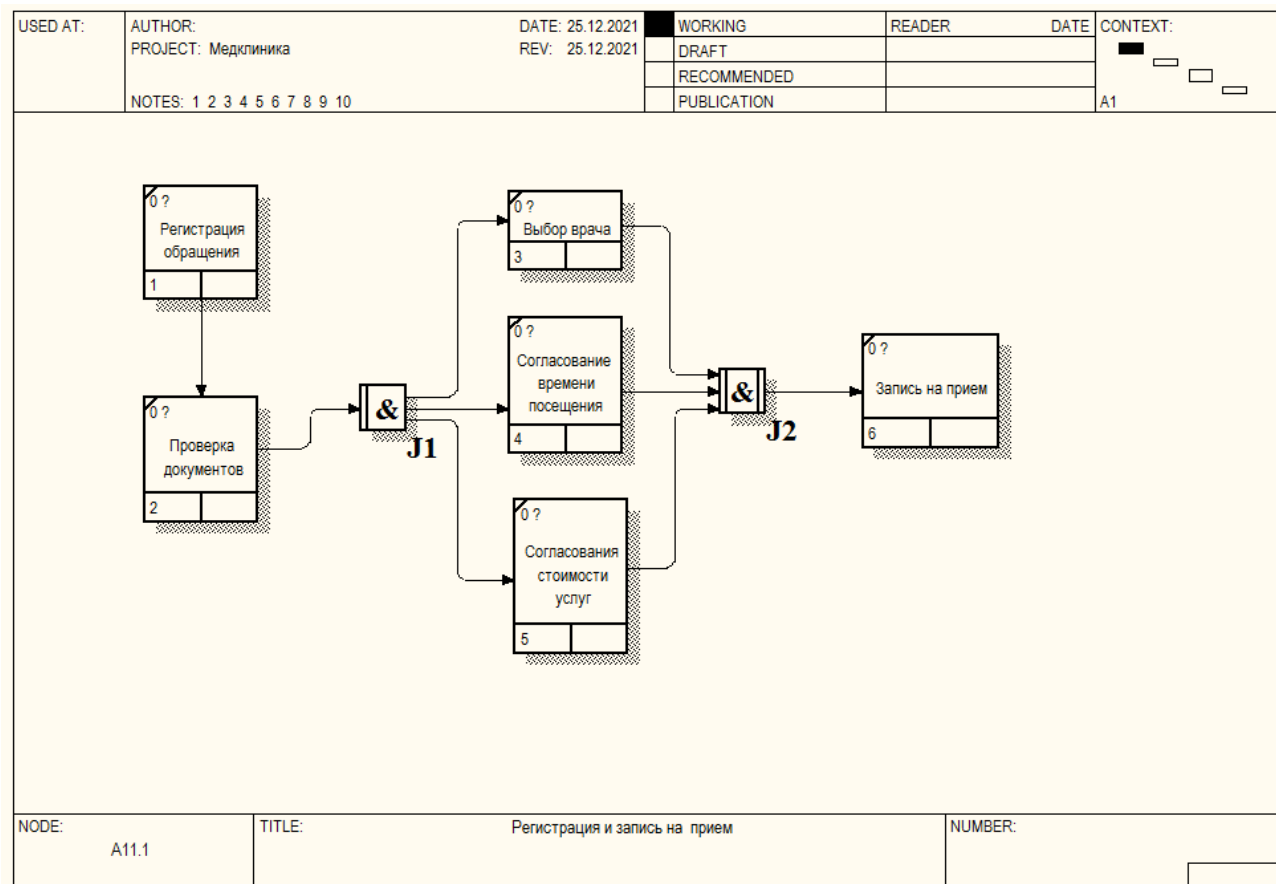


Рис.36 – IDEF3 диаграмма «Регистрация и запись на прием»

### САМОСТОЯТЕЛЬНО.

Построить диаграмму IDEF3 любого бизнес-процесса из выбранного варианта самостоятельной работы.

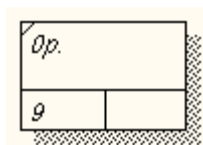
⚠ Показать результаты работы преподавателю.

#### Вопросы для самопроверки:

1. Для чего строится диаграмма IDEF3?
2. Чем диаграмма IDEF3 отличается от диаграммы IDEF0?
3. Как графически обозначается работа в диаграмме IDEF3?
4. С какой целью между работами устанавливают перекресток?
5. Какие типы перекрестков вы знаете?

#### Выполните тест:






1. В какой нотации используется следующее графическое изображение работы:







- a) DFD;
- b) IDEF3;

- c) IDEF0;
- d) правильные ответы b), c);
- e) правильные ответы a), c).

2. С помощью какого инструмента можно установить между работами перекресток?

- a) ;
- b) ;
- c) ;
- d) ;
- e) .

3. Укажите тип перекрестка, если при разветвлении стрелок Fan-in Junction выполняется действие: все следующие процессы должны быть запущены.

- a) Synchronous:  AND
- b) Asynchronous:  AND
- c) Synchronous:  OR
- d) Asynchronous:  OR

4. Перекресток, обозначаемый графическим значком, представленным на рисунке, имеет название:



- a) Асинхронное «И» (Asynchronous AND)
- b) Синхронное «И» (Synchronous AND)
- c) Асинхронное «ИЛИ» (Asynchronous OR)
- d) Синхронное «ИЛИ» (Synchronous OR)
- e) Исключающее «ИЛИ» XOR (Exclusive OR)

5. Различают перекрестки для:

- a) Слияния
- b) Исключения
- c) Разветвления
- d) Отношения

Таблица 11 – Правильные ответы по тесту

Номер вопроса	Номер правильного ответа
---------------	--------------------------

1	b
2	b
3	b
4	a
5	a, c

#### 1.1.3.4 Проектирование интерфейса пользователя. Проектирование экранных форм и отчетов приложения.

К элементам и компонентам программы для обеспечения интерфейса пользователя относятся следующие:

1. Набор задач пользователя, которые он решает при помощи системы.
2. Используемая системой метафора (например, рабочий стол в MS Windows).
3. Элементы управления системой.
4. Навигация между блоками системы.
5. Дизайн экранов программы.
6. Средства отображения информации, отображаемая информация и форматы.
7. Диалоги, взаимодействие и транзакции между пользователем и компьютером.

Этапы разработки пользовательского интерфейса можно подразделить на:

- I. Сбор и анализ требований, поступающих от пользователей
- II. Разработка пользовательского интерфейса
- III. Построение пользовательского интерфейса
- IV. Тестирование и подтверждение качества пользовательского интерфейса

I. Сбор и анализ требований, поступающих от пользователей проводится по следующим направлениям:

1. Определение профилей (моделей) пользователей.
2. Анализ задач, стоящих перед пользователями.
3. Сбор требований.
4. Анализ рабочей среды пользователей.
5. Анализ соответствия требований пользователей стоящим перед ним задачам.

При определении профиля пользователя особенно важно учитывать опыт работы с компьютерами и прикладными программами, так как это в значительной степени влияет на:

- содержание и время обучения;
- объем и структуру справочной системы;
- организацию протокола взаимодействия (привычность).

Ниже приведен примерный перечень вопросов, на которые нужно дать ответы в ходе анализа задач, стоящих перед пользователями:

- Какие задачи решают пользователи?
- Какие задачи являются наиболее важными?
- Какие шаги предпринимаются для решения задач?
- Какие цели преследуют пользователи при решении тех или иных задач?
- Какой информацией необходимо располагать для выполнения задач?
- Какой инструментарий используется для решения задач?
- Каков ожидаемый итог от решения каждой задачи?
- Каким образом пользователи выполняют свою работу (вручную, на компьютере, по телефону и т.д.)?
- Каким образом они взаимодействуют с другими лицами при решении задач?
- Каким образом задачи учитываются в общем бизнес-процессе?
- Как часто пользователям приходится решать их задачи?
- Каким образом компьютер или другая компьютерная техника помогает пользователям в решении задач?

Сбор требований можно провести методом анкетирования.

Анкетирование заключается в составлении списка вопросов и предоставлении их всем заинтересованным лицам (потенциальным пользователям программы), которых необходимо выявить заранее. Большую роль при анкетировании играют правильность составления вопросов, исключая их разное трактование, и заинтересованность респондентов. Спектр вопросов может быть достаточно широк: от перечисления списка решаемых задач в проектируемой системе до расположения различных меню и кнопок, и предпочитаемой цветовой гаммы. Следует иметь в виду, что в силу различных факторов (например, низкая мотивация или компетенция респондентов) анкетирование не дает 100 процентный эффект, поэтому обязательно нужно провести анализ соответствия требований пользователей стоящим перед ним задачам.

II. Разработка пользовательского интерфейса должна решать следующие задачи:

1. Определение цели с точки зрения удобства применения продукта.
2. Разработка задач и сценария действий пользователей.
3. Определение объектов и операций интерфейса.
4. Определение визуального представления.
5. Разработка меню объектов и окон.
6. Оптимизация визуальной разработки.

Рассмотрим несколько областей целей и задач с точки зрения удобства применения продукта:

#### 1. Пригодность:

цель: пользователи получают возможность использовать программу для выполнения своих задач;

задачи: достаточно высокий процент пользователей смогут использовать систему для решения своих задач с первой попытки/после прохождения обучения/...;

## 2. Эффективность:

цель: деятельность пользователей станет более эффективной, чем была до использования программы;

задачи: сравнительно высокий процент пользователей выполнят свои задачи в течение заданного промежутка времени/ с заданными показателями качества/...;

## 3. Легкость в освоении:

цель: пользователям потребуется минимальный тренинг;

задачи: пользователи будут в состоянии успешно работать с продуктом после прохождения соответствующего обучения (длительность, форма обучения, средства обучения) или самостоятельного изучения программы по руководству пользователя/инструкциям по работе с программой/видеокурсу... .

## 4. Оценка пользователями качества продукта:

цель: пользователи будут удовлетворены качеством продукта;

задачи: пользователи будут высоко оценивать степень своего удовлетворения качеством продукта (шкала оценки);

Разработка задач и сценария действий пользователей.

Сценарий – последовательность задач, стоящих перед пользователями, или событий, направленных на достижение единой цели, описание действий, выполняемых пользователем.

Для определения объектов и операций нужно выполнить ряд действий:

### 1. Выделить из сценариев и задач объекты, данные и действия:

- выделение в тексте существительных, глаголов, отглагольных форм и пр.
- классификация объектов (данные, контейнер, устройство):
- соотнесение объектов и действий.

2. Просмотреть и уточнить список объектов и действий совместно с пользователем.

3. Построить диаграммы пользовательских сценариев и диаграммы взаимодействия между объектами.

При проектировании пользовательского интерфейса (User Interface - UI) учитывается большое количество разнообразных факторов, для того, чтобы взаимодействие с программной системой было наиболее продуктивно и эффективно. Рассматриваются не только технические или экономические факторы, но и, например, психологические. Так, на основании проведенных исследований движения глазного яблока были сделаны выводы, что человек сначала направляет взгляд в верхний левый угол экрана, далее перемещает его по экрану по часовой стрелке. Следовательно, при определении местоположения кнопок на экране, те, что используются чаще, стоит располагать в верхнем левом углу, уже потом размещать остальные элементы управления (кнопки, меню и т. д.).

А результатом изучения психологии восприятия человеком цвета, в частности, как смещается внимание, стали утверждения:

- от тёмных областей к светлым;

- от насыщенных цветов к ненасыщенным.

В исследованиях восприятия отдельных объектов, их параметров, взаиморасположения, а также представление о том, что встречаются другие виды объектов, выявлены такие закономерности: восприятие размера начинается с больших объектов и переходит к маленьким или от необычных форм к обычным.

В РФ существуют государственные стандарты, регламентирующие разработку пользовательских интерфейсов - ГОСТ-стандарт ISO 9241-110 (Принципы организации диалогов) является частью ISO 9241 (Эргономика взаимодействия человек—система) и описывает общие принципы и рекомендации по проектированию интерактивных систем — сайтов, интерфейсов, программ, сервисов и т.п.

К принципам, описанным в этом стандарте относятся следующие:

- пригодность для выполнения задачи;
- информативность (самоописываемость);
- соответствие ожиданиям пользователя;
- пригодность для обучения.

Первый принцип подразумевает, что интерфейс, в первую очередь, должен предоставлять все возможности для успешного решения пользователем стоящих перед ним задач.

По принципу информативности рабочее окно системы, экранные пользовательские формы должны содержать столько информации, что пользователю должно быть понятно, где он находится в определенный момент, в результате каких действий здесь оказался, какую последовательность действий необходимо выполнить для достижения результата.

Расположение и обозначение кнопок, команд меню должны демонстрировать функциональность программы и соответствовать ожиданиям пользователя, учитывать его навыки и опыт работы с программными приложениями.

Степень пригодности системы для обучения тем выше, чем более простым и понятным будет пользовательский интерфейс, чем меньше понадобится пользователю на изучение функциональности и начала полноценной работы.

Анкетирование можно проводить, как ручным заполнением анкет, так и в электронном варианте с автоматизированной обработкой результатов. Предварительно определяется круг лиц, для которых будет составлен опрос.

Ниже приведен примерный перечень вопросов для сбора требований пользователей к интерфейсу программы (таблица 12).

Таблица 12 – Пример заполнения анкеты для определения требований пользователей к интерфейсу

Вопросы	Варианты ответов
1. Укажите ФИО	<i>ФИО респондента</i>
2. Укажите должность	Должность респондента

3. Ваши функциональные обязанности	Перечень функциональных обязанностей
4. Какие данные являются исходными	Перечень данных
5. Результат ваших действий	Документ, отчет и т.п.
6. Предпочтительная цветовая гамма интерфейса	Указание цветовой гаммы интерфейса
7. Использование цветowych индикаторов	Да <input type="checkbox"/> V Нет <input type="checkbox"/>
8. Использование графических элементов дизайна (обозначение кнопок, элементы идентификации бренда и т.п.)	Да <input type="checkbox"/> V Нет <input type="checkbox"/>
9. Вспомогательные кнопки: вызов справочной системы, календарь, калькулятор и т.п.)	Да <input type="checkbox"/> V Нет <input type="checkbox"/>
10.Использование всплывающих подсказок	Да <input type="checkbox"/> V Нет <input type="checkbox"/>
11.Наличие помощника	Да <input type="checkbox"/> V Нет <input type="checkbox"/>
12.Оцените количество свободного пространства на экране по отношению к элементам интерфейса	50% <input type="checkbox"/> 70% <input type="checkbox"/> V 80% <input type="checkbox"/>
13. Размер шрифта	12 пт
14.Возможность увеличения размера шрифта для текста на экране	Да <input type="checkbox"/> V Нет <input type="checkbox"/>
15.Возможность индивидуальных настроек интерфейса	Да <input type="checkbox"/> V Нет <input type="checkbox"/>
16.Сообщение об ошибке	Всплывающее окно сообщения <input type="checkbox"/> V Звуковой сигнал <input type="checkbox"/> V
17.Возможность исправления ошибочных действий (отмена)	Да <input type="checkbox"/> V Нет <input type="checkbox"/>
18.Максимальное количество переключения экранов (пунктов меню) для достижения цели	Количество
19.Индикаторы выполнения задания	Всплывающее окно сообщения <input type="checkbox"/> V Звуковой сигнал <input type="checkbox"/> V
20.Возможности возврата к предыдущему экрану /к главному экрану?	Стрелки <input type="checkbox"/> V Текст <input type="checkbox"/> Другие значки <input type="checkbox"/>
21.Быстрый поиск информации	Да <input type="checkbox"/> Нет <input type="checkbox"/>



**Задание 8.**

Добавить в анкету не менее 3-х вопросов.

Следует отметить, что анкетирование не приводит к стопроцентному результату. Рекомендуется использовать анкеты вместе с другими способами сбора требований к интерфейсу, например, использование CASE-средств.

VRwin имеет мощный инструмент навигации по модели - Model Explorer (рис. 37), который позволяет представить иерархию работ и диаграмм в удобном и компактном виде, однако этот инструмент не является составляющей стандарта IDEF0.

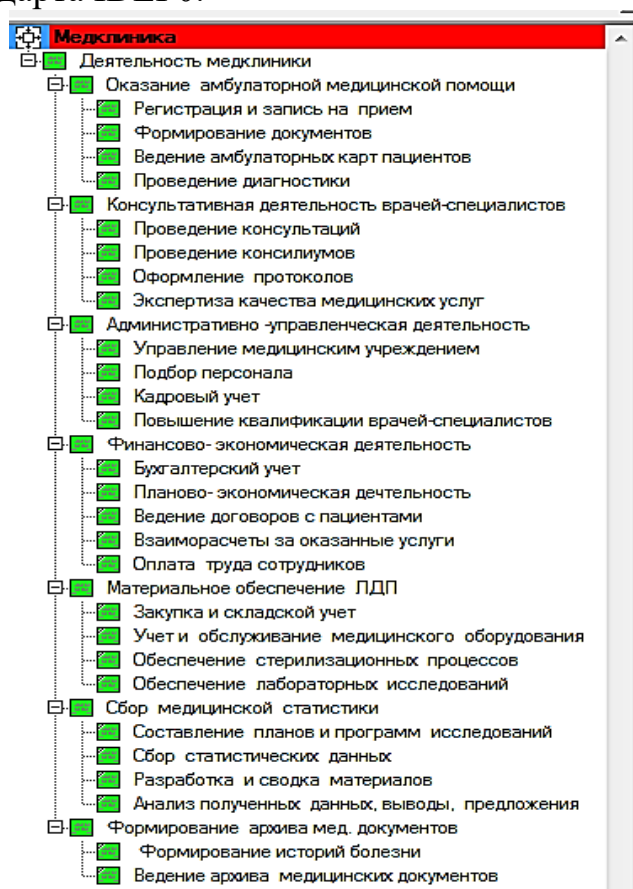


Рис. 37 - Model Explorer

Процесс создания модели работ является итерационным (повторяющимся, многократно меняющимся), следовательно, работы могут менять свое расположение в дереве узлов многократно. Чтобы не запутаться и проверить способ декомпозиции, следует после каждого изменения создавать диаграмму дерева узлов.

**Задание 9. Создание диаграммы дерева узлов модели «Деятельность медклиники».**

Для создания диаграммы дерева узлов следует: выбрать в меню пункт Diagram - Add Node Tree. Появится диалог создания диаграммы дерева узлов Node Tree Wizard (рис. 38).

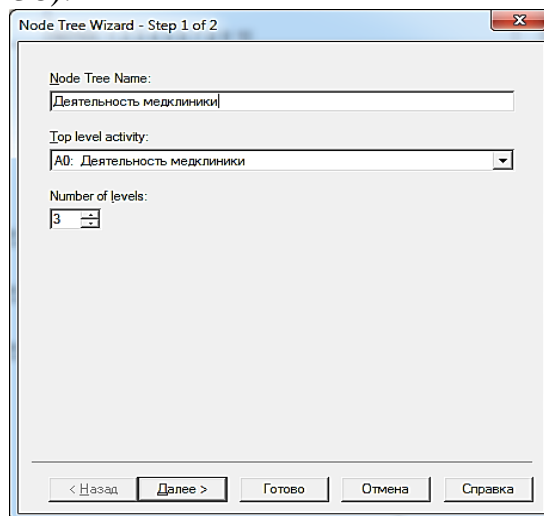


Рис. 38 - Диалог создания диаграммы дерева узлов Node Tree Wizard

В первом окне эксперта можно ввести имя диаграммы дерева узлов, узел верхнего уровня и глубину дерева – Number of Levels (по умолчанию глубина имеет значение 3 уровня).

Дерево узлов не обязательно в качестве верхнего уровня должно иметь контекстную работу и может иметь произвольную глубину. В одной модели можно создавать множество диаграмм деревьев узлов.

Имя дерева узлов по умолчанию совпадает с именем работы верхнего уровня, а номер диаграммы автоматически генерируется как номер узла верхнего уровня плюс литера "N", например, A0N.

Второе окно эксперта Node Tree Wizard, которое высвечивается после нажатия на кнопку «Далее» (рис. 39) позволяет задать свойства диаграммы дерева узлов.

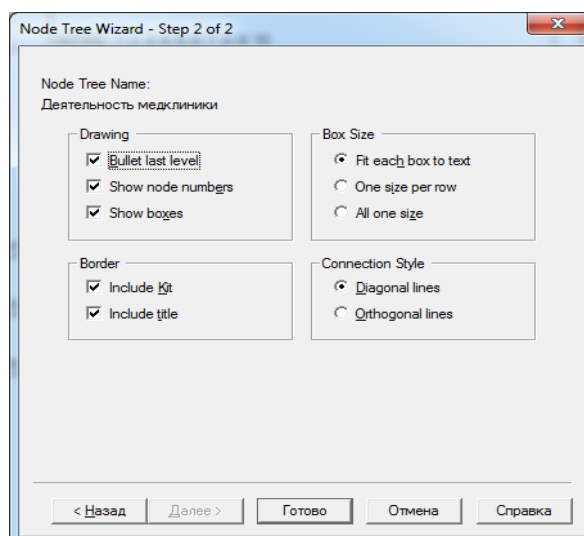


Рис. 39 – Продолжение диалога эксперта Node Tree Wizard

В нотации IDEF0 диаграмма дерева узлов показывает иерархию работ в модели и позволяет рассмотреть всю модель целиком, но не показывает

взаимосвязи между работами. Эта диаграмма может служить прототипом главного меню проектируемой информационной системы.

По умолчанию нижний уровень декомпозиции показывается в виде списка, остальные работы - в виде прямоугольников (рис. 40).

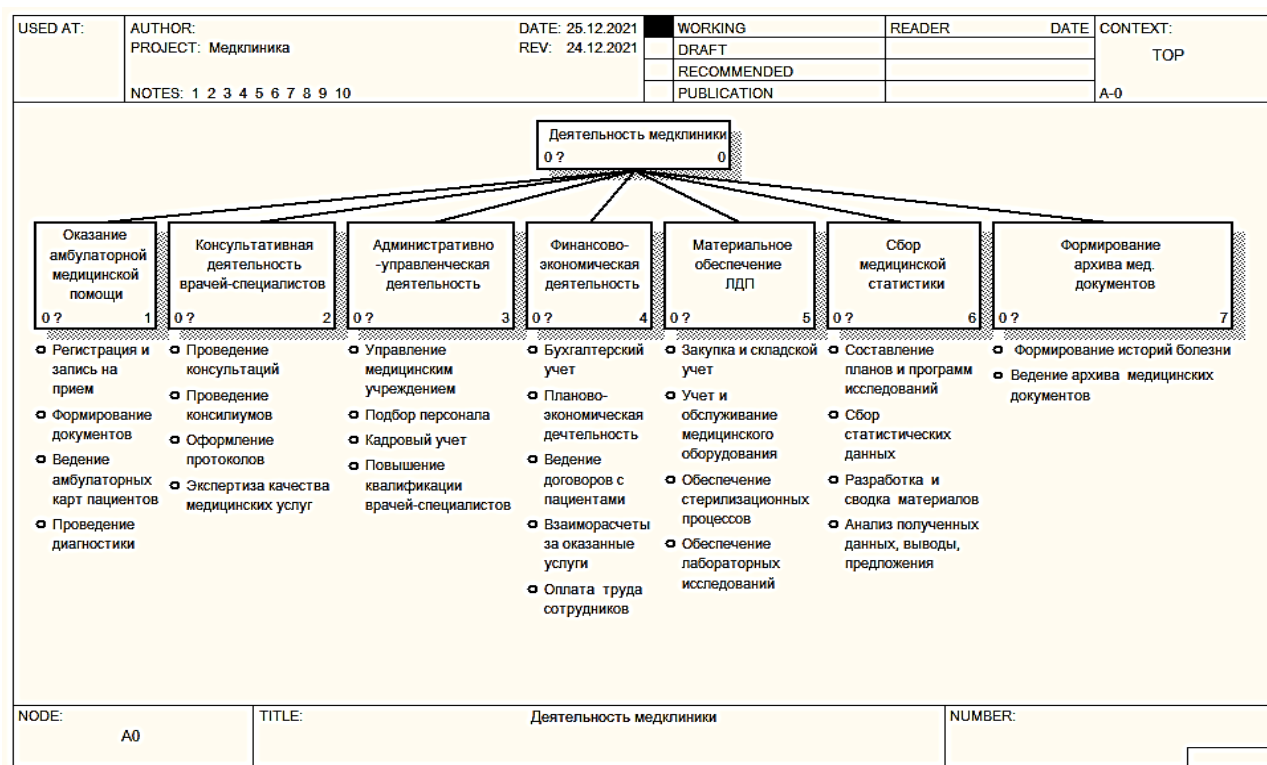


Рис. 40 - Диаграмма дерева узлов модели «Деятельность медклиники»

Для отображения всего дерева в виде прямоугольников следует убрать опцию **Bullet Last Level**. Группа **Connection Style** позволяет выбрать стиль соединительных линий - диагональные (по умолчанию) или ортогональные. Нужно щелкнуть правой кнопкой мыши по свободному месту, не занятому объектами, и выбрать меню **Node tree Diagram Properties** (рис. 41).

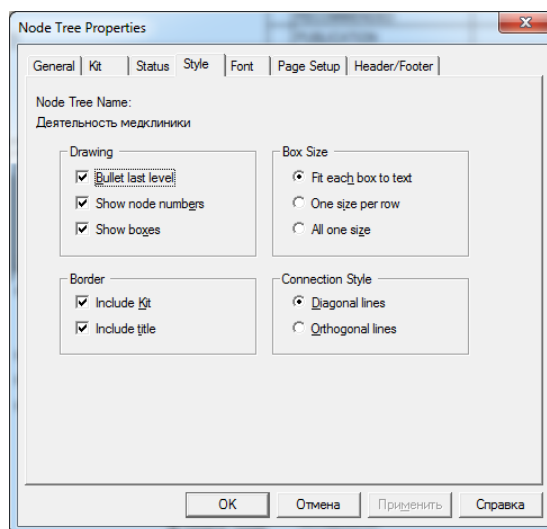


Рис. 41 - Выбор меню **Node tree Diagram Properties**

После отключения во вкладке **Style** диалога **Node Tree Properties** опции **Bullet Last Level** диаграмма дерева узлов примет вид, показанный на рис. 42.

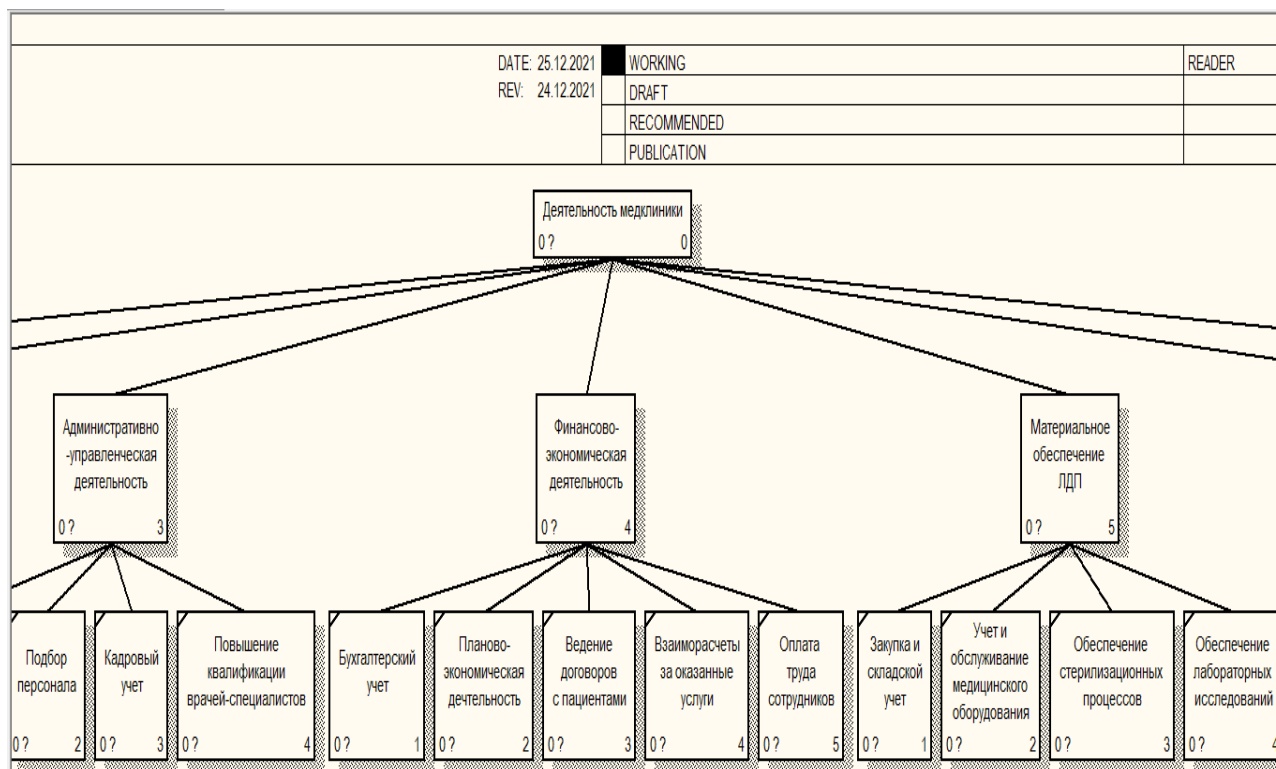


Рис. 42 – Фрагмент диаграммы дерева узлов

Построение диаграмм пользовательских сценариев и диаграмм взаимодействия между объектами удобно выполнить с применением языка моделирования UML.

Исходя из описания бизнес-процесса «Оказание амбулаторной медицинской помощи» определим 4-х действующих лиц (актеров):

1. Сотрудник регистратуры, принимающий и проверяющий документы пациентов, осуществляющий ввод данных пациентов, выдающий талон на амбулаторный прием.
2. Врач-специалист, ведущий прием пациентов, выписывающий назначения, рецепты, направления на анализы или проведение обследование (например, рентген, флюорограмма и т.п.),
3. Лаборант по приему анализов, результаты которых вносятся в медкарту пациента.
4. Специалист по проведению обследования. Результаты обследований также отражаются в медкарте амбулаторной больного.

Разработаем диаграмму сценариев (вариантов использования) для актера – сотрудника регистратуры (рис. 43). После обращения пациента в регистратуру, сотрудник проверяет его документы и осуществляет поиск в базе данных. Если пациент найден, то при необходимости можно отредактировать его данные (например, изменился адрес, фамилия и т.п.). Если пациент не найден, то информация о нем заносится в базу. Далее, вбирается врач-специалист, к которому пациент хочет попасть на прием, и время работы врача.

Следующие действия сотрудника регистратуры: заполнение и печать талона амбулаторного приема.

Откройте проект в программе RSA проект **ФИОстудентUML**. Добавьте диаграммы сценариев, как описано в п.1.1.3.2.

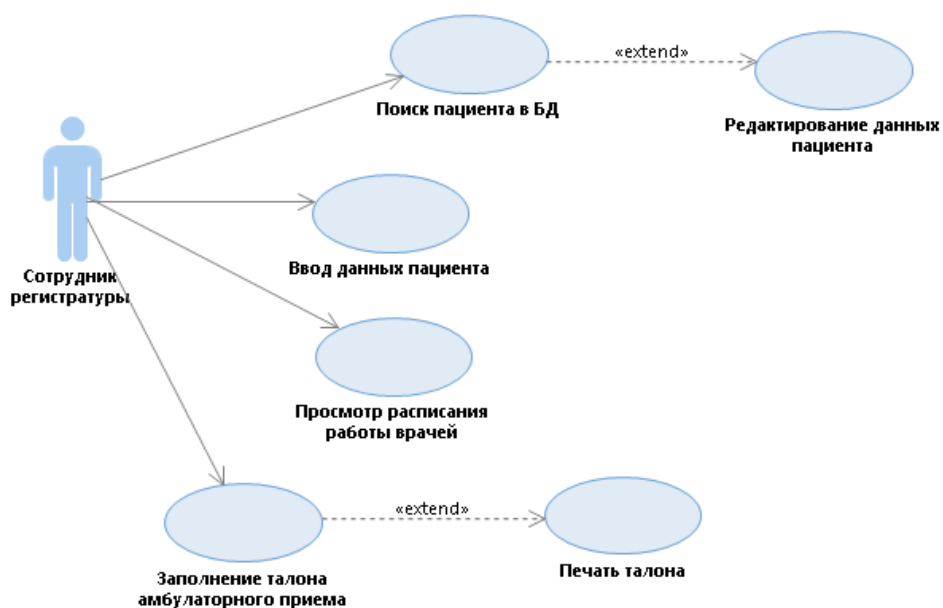


Рис. 43 – Диаграмма сценариев сотрудника регистратуры

С выписанным талоном, пациент, получающий медицинскую помощь в амбулаторных условиях, обращается к врачу-специалисту, который производит заполнение талона в соответствии с принятыми нормативами. В талоне указываются жалобы пациента, диагноз (предварительный, а. затем, и заключительный), направления на анализы, консультацию, обследование, рецепты на лекарственные препараты и т.д. Диаграмма сценариев с актером – врач-специалист приведена на рисунке 44.

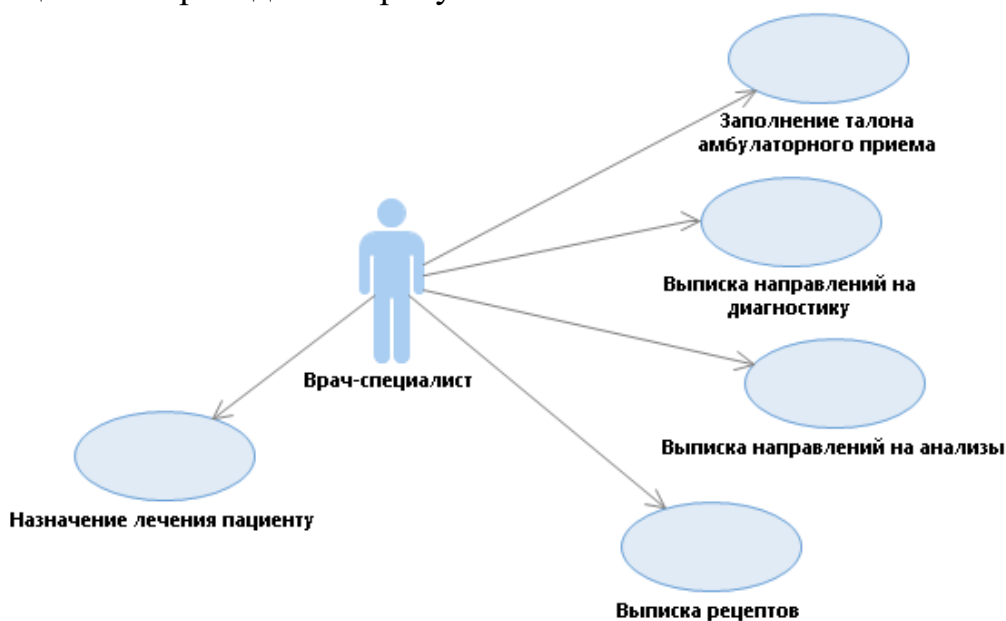


Рис. 44 – Диаграмма сценариев врача-специалиста

Для создания диаграммы взаимодействия в окне Project Explorer, кликните правой кнопкой на My Interaction Diagram Model (под узлом Models), и кликните Add Diagram > Sequence Diagram. Откроется ваша новая пустая диаграмма последовательности. Если нет, дважды щелкните на SequenceDiagram1 в Project Explorer, чтобы открыть ее. На рис. 45 представлена панель инструментов для создания диаграммы взаимодействия (последовательности).

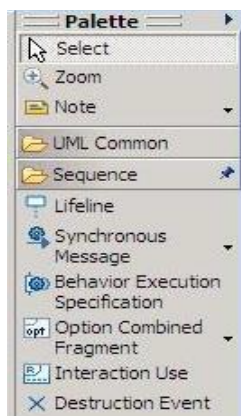


Рис. 45 – Панель инструментов для создания диаграммы взаимодействия

Далее, на рисунке 46 приведена диаграмма взаимодействия (последовательностей) объектов бизнес-процесса «Оказание амбулаторной медицинской помощи».

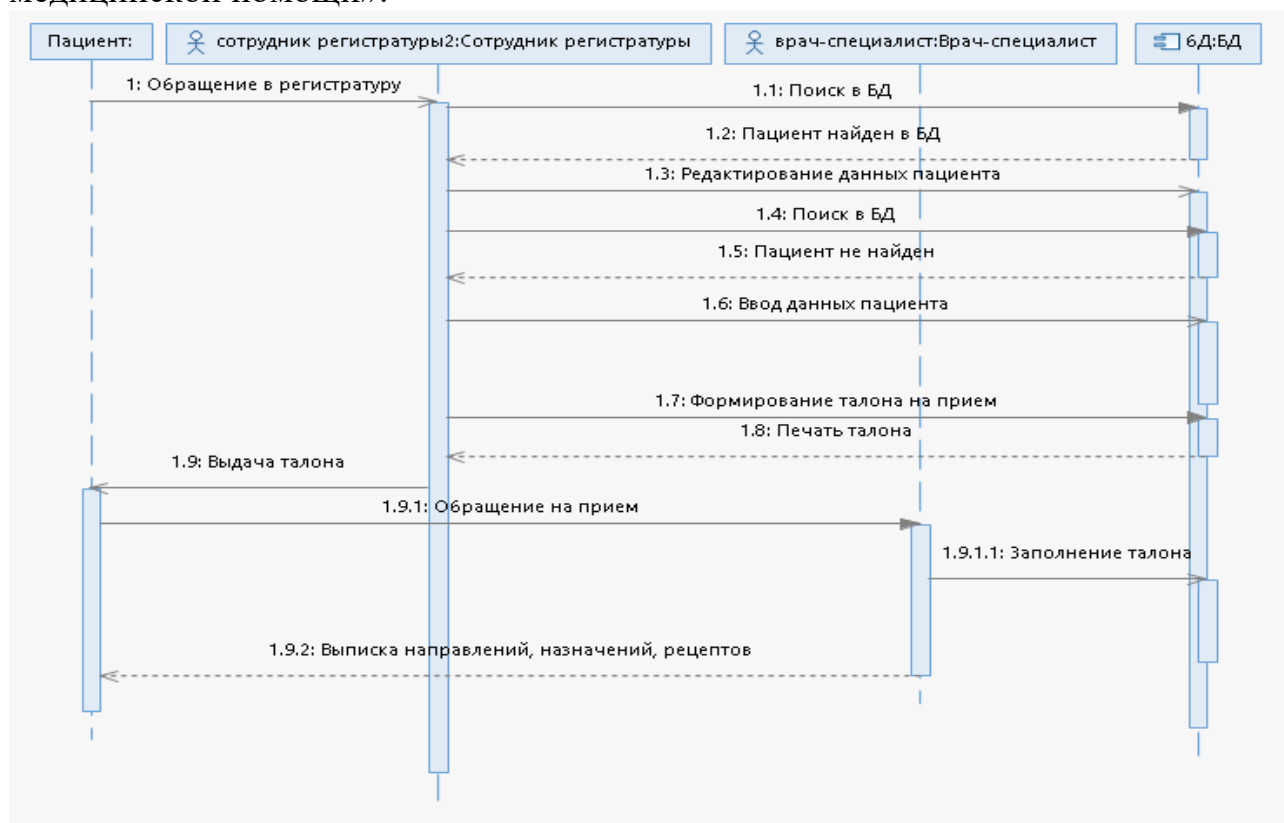


Рис. 46 – Диаграмма взаимодействия объектов бизнес-процесса «Оказание амбулаторной медицинской помощи»

На диаграммах этого вида показываются связи, включающие множество объектов и отношений между ними, в том числе сообщения, которыми объекты обмениваются.

При этом в диаграмме взаимодействия уделяет большое внимание временной упорядоченности сообщений.

! Показать результаты работы преподавателю.

! Сохранить проект **ФИОстудентаUML**.

Для визуального представления пользовательского интерфейса воспользуемся методом прототипирования - это один из этапов разработки, который заключается в предварительном наброске содержания и расположения основных элементов интерфейса. Для реализации этого метода можно воспользоваться любым редактором: MS Word, MS Excel, графическими редакторами и т.д.

На рисунке 47 представлен фрагмент пользовательского интерфейса программы, позволяющий решить задачи оказания пациенту медицинской помощи амбулаторно. На рисунке представлены объекты интерфейса:

1. Справочники (для хранения списочной информации).
2. Документы (для ввода в систему входных данных).
3. Отчеты (для анализа и обработки информации по заданным алгоритмам в целом по системе).

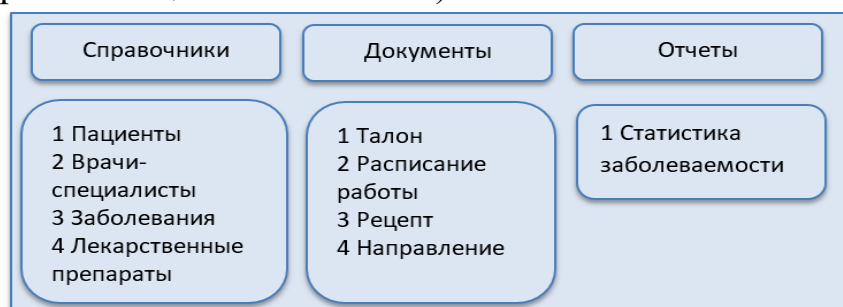


Рис. 47 – Фрагмент прототипа пользовательского интерфейса

На рисунке 48 представлен эскиз экранной формы талона пациента, получающего медицинскую помощь в амбулаторных условиях.

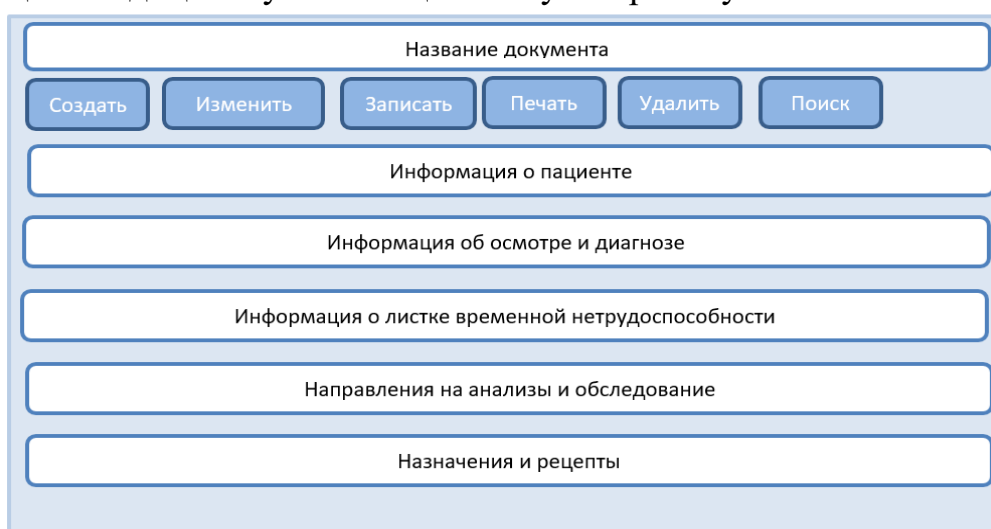


Рис. 48 – Эскиз экранной формы документа

Отчет – это форматированное представление данных, которое выводится на экран, в печать или файл. Они позволяют извлечь из базы нужные сведения и представить их в виде, удобном для восприятия, а также предоставляют широкие возможности для обобщения и анализа данных. В экранной форме отчета, как правило, присутствуют поля для ввода даты или периода, за который требуется сформировать данные. Для конкретизации анализа и вывода данных по определенному параметру в форму отчета добавляются поля настройки. Эскиз экранного интерфейса отчета представлен на рисунке 49.

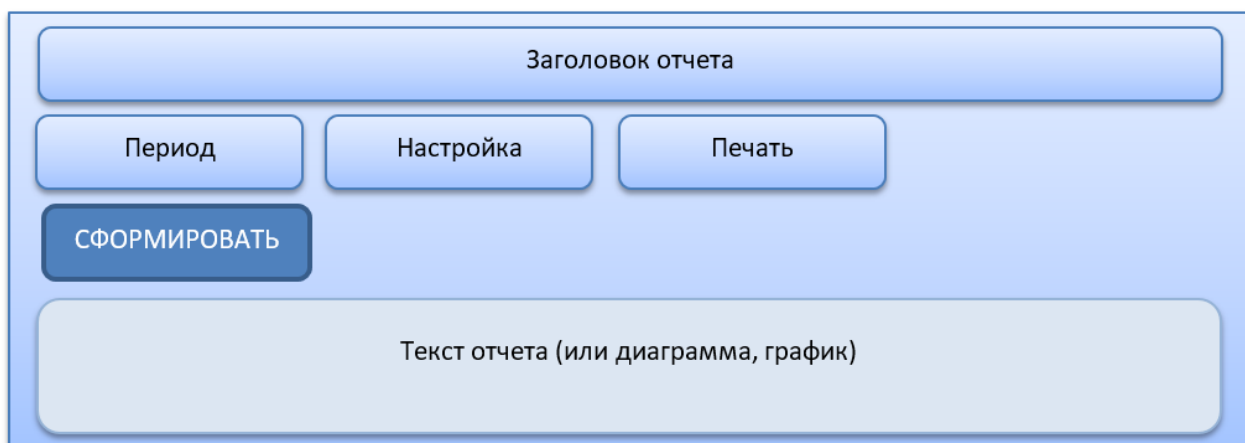


Рис. 49 - Эскиз экранного интерфейса отчета

### **САМОСТОЯТЕЛЬНО.**

Разработать диаграммы сценариев для актеров: лаборант по приему анализов, специалист по проведению обследования.

Разработать диаграмму взаимодействия лаборанта и специалиста в системе.

ⓘ Показать результаты работы преподавателю.

### **САМОСТОЯТЕЛЬНО.**

Разработать диаграммы сценариев, диаграммы взаимодействия, прототип пользовательского интерфейса, эскизы экранных форм документа и отчета по ранее выбранному варианту самостоятельной работы.

ⓘ Показать результаты работы преподавателю.

### **Вопросы для самопроверки:**

1. Какие элементы и компоненты программы обеспечивают интерфейс пользователя?
2. Этапы разработки пользовательского интерфейса.
3. Какие задачи решает разработка пользовательского интерфейса?
4. Что представляет собой сценарий действий пользователей?
5. Приведите пример психологических факторов, учитываемых при разработке интерфейса пользователя.



6. Приведите пример существующего государственного стандарта, регламентирующего разработку пользовательских интерфейсов.
7. Что представляет собой метод анкетирования, применяемый для сбора требований к проектированию пользовательского интерфейса?
8. Что характеризует степень пригодности программной системы для обучения?
9. Что представляет собой диаграмма Node Tree Wizard?
10. Для чего предназначена UML диаграмма взаимодействия?

### 1.1.3.5 Создание диаграммы состояний

При современном подходе к проектированию сложной системы принято делить ее на части, каждую из которых затем рассматривать отдельно. Таким образом, Система разбивается на объекты или компоненты, которые взаимодействуют друг с другом, обмениваясь сообщениями. Сообщения описывают или представляют собой некоторые события. Получение объектом сообщения активизирует его и побуждает выполнять предписанные его программным кодом действия.

При данном подходе Система становится событийно управляемой, поэтому разработчикам зачастую важно знать, как должен реагировать тот или иной объект на определенные события. Инициаторами событий могут быть как объекты самой Системы, так и её внешнее окружение.

Описать поведение отдельно взятого объекта помогает **диаграмма состояний**.

Данный вид диаграмм используется аналитиками для описания последовательности переходов объекта из одного состояния в другое.

Диаграмма состояний показывает все возможные состояния, в которых может находиться объект, а также процесс смены состояний в результате внешнего влияния.

Основными элементами диаграммы состояний являются «Состояние» и «Переход». Диаграмма состояний имеет схожую семантику с диаграммой деятельности, только деятельность здесь заменена состоянием, переходы символизируют действия. Таким образом, если для диаграммы деятельности отличие между понятиями «Деятельность» и «Действие» заключается в возможности дальнейшей декомпозиции, то на диаграмме состояний деятельность символизирует состояние, в котором объект находится продолжительное количество времени, в то время как действие моментально.

### **Задание 10. Создание диаграммы состояний (statemachinediagram) в RSA.**

1. В окне Project Explorer, кликните правой кнопкой на My State Machine Diagram Model.
2. Кликните Add Diagram > State Machine Diagram.

3. Откроется ваша новая пустая диаграмма состояний. Если нет, дважды щелкните на StateMachineDiagram1 в Project Explorer, чтобы открыть ее.

4. Усовершенствуйте вид диаграммы, выбрав Diagram>Arrange>Allmenuoption.

5. Добавьте элементы на диаграмму (Состояния, составные состояния, переходы), используя панель инструментов в соответствии с примером на рис. 50.

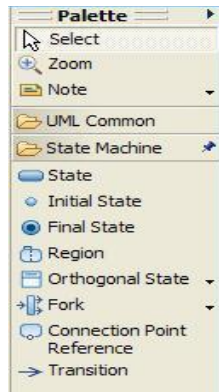


Рис. 50 – Панель инструментов UML диаграммы состояний

На рисунке 51 приведен пример диаграммы состояний документа «Талон на прием».

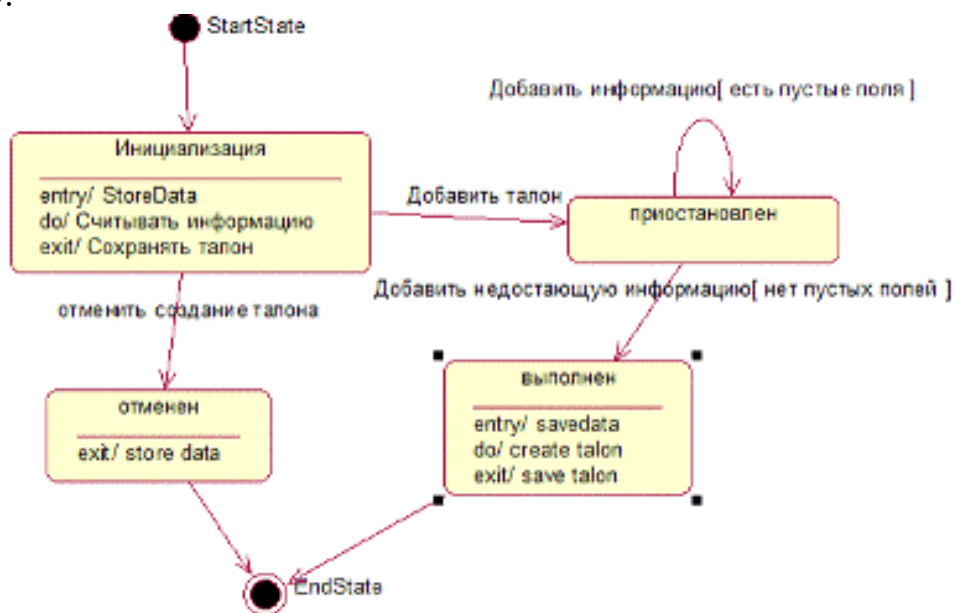


Рис. 51 – Диаграмма состояний документа «Талон на прием»

Целью построения диаграмм состояний является исследования сложного поведения объекта или системы. С помощью нее можно досконально проработать динамику и предупредить возможные проблемы в программировании.

Рассмотрим пример светофора.

Даже такая, казалось бы, простая система потребует детального изучения, в чем поможет диаграмма состояний.

Первым состоянием будет красный. Обратим внимание, мы моделируем светофор в целом, как единый объект, а не отдельные его фонари. То есть состояние – это именно состояние светофора.

Переходы совершаются по прошествии времени.

После состояния Красный светофор переходит в состояние Красный+Желтый. Это состояние говорит о том, что транспорт не может начинать движение, но информирован, что скоро движение будет разрешено после очередного перехода состояний.

Следующее состояние – Зеленый.

Далее по прошествии времени включается Мигающий зеленый.

Далее Желтый.

Затем Красный.

Соответствующая диаграмма показана на рисунке 52.

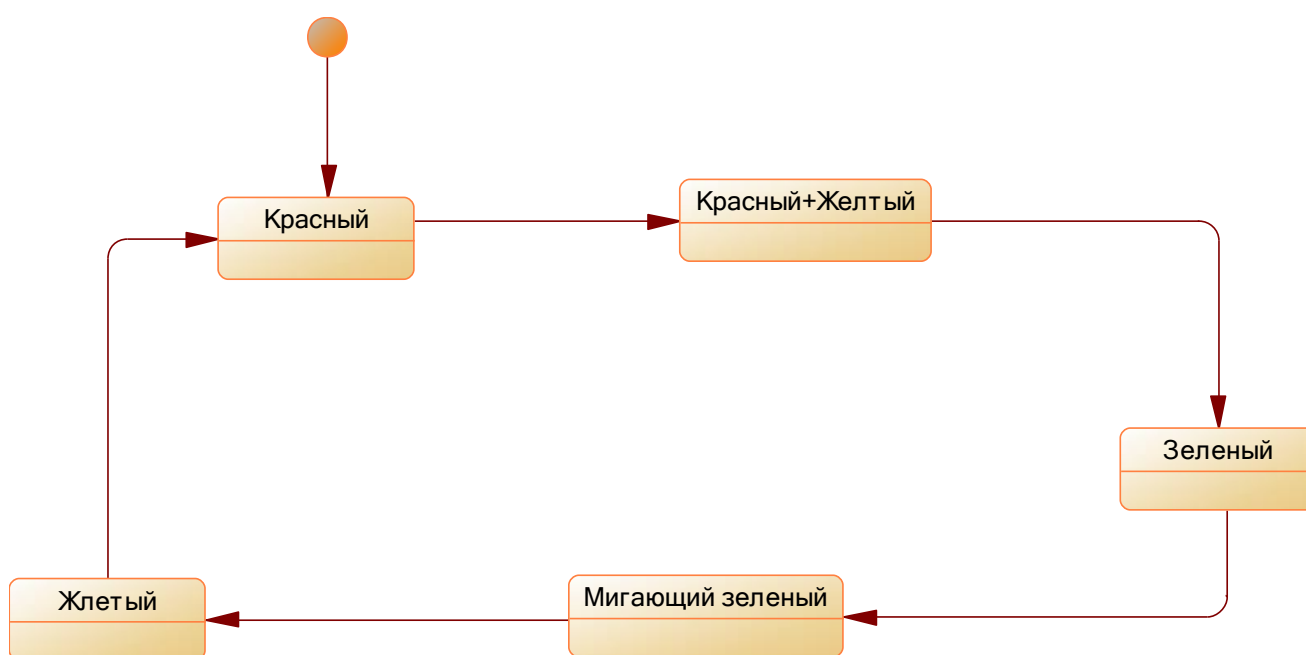


Рисунок 52 – Диаграмма состояния светофора

Обратим внимание, что даже такой просто пример при ближайшем рассмотрении не кажется очевидным.

Далее рассмотрим ситуацию, когда с помощью суперсостояний мы смоделируем переход между разными режимами работы светофора: Зеленый-Желтый-Красный и Мигающий Желтый.

Суперсостояние позволяет смоделировать более сложную логику без того, чтобы отражать переходы из каждого вложенного состояния. Не важно, какой горел свет на светофоре в момент, когда его переключили в режим мигающего желтого (рис.53).

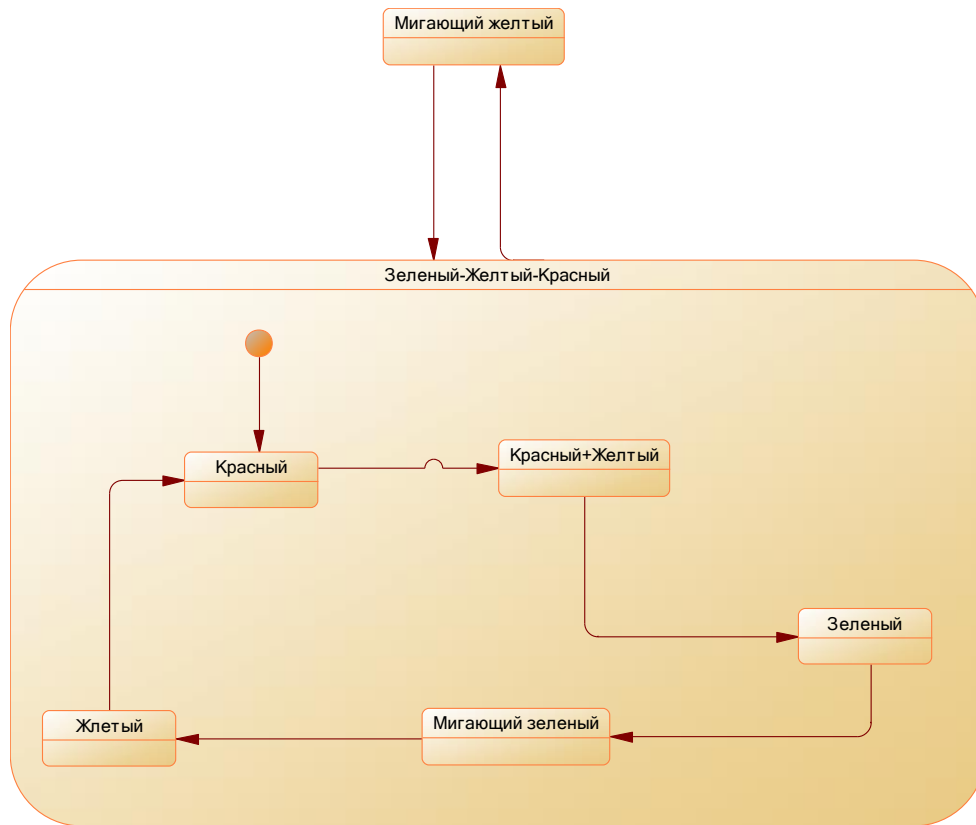


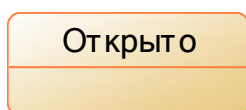
Рисунок 53 – Диаграмма суперсостояния

### САМОСТОЯТЕЛЬНО,

Реализовать и изобразить диаграмму состояний для светофора с боковой стрелкой.

⚠ Показать результаты работы преподавателю.

Рассмотрим более детально элементы диаграммы состояний.



Состояние. Система находится в нем в течение некоторого промежутка времени. Состояние может характеризоваться некоторыми атрибутами.

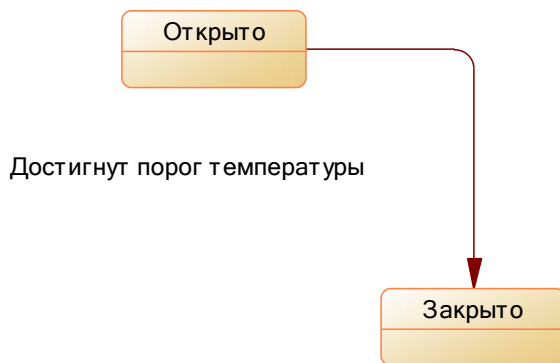


Переход. Происходит мгновенно и переводит систему из одного состояния в другое.

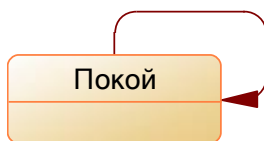
Если переход не мгновенный, то его тоже нужно моделировать с помощью состояния.

Можно задать триггеры, которые срабатывают и вызывают различные переходы

Триггерный переход вызван некоторым внешним воздействием на систему, заставляющим ее сменить состояние.



Переход системы в то же состояние, в котором она была, называется петлей.



Рассмотрим особые состояния.

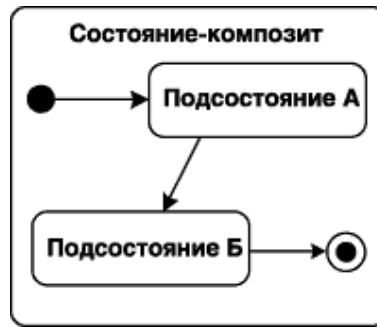


Начальное состояние системы. Может быть, только одно на всю рассматриваемую модель.



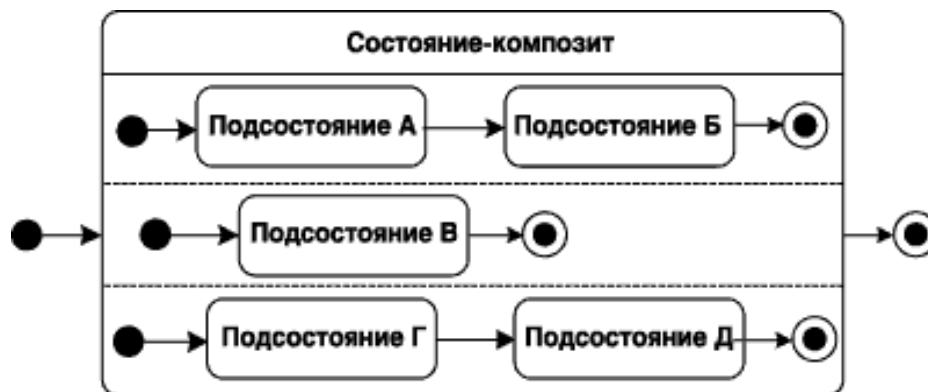
Конечное состояние.

Суперсостояние (или композитное состояние). Включает в себя ряд других состояний.



Возможны параллельные состояния. Ниже показано суперсостояние, в котором по отдельным Плавающим дорожкам протекает смена различных состояний. Причем они не пересекаются.

Система может находиться в разных такого рода состояниях. Например, автомобиль может двигаться или стоять, при этом может играть музыка из проигрывателя или радио, а фары могут быть выключены или включены на дневные ходовые огни или ближний свет или дальний свет и т.п. Однако выключение двигателя означает выход из суперсостояния и прекращение всей этой активности.



Состояние может иметь некоторые атрибуты. Например, набор телефонного номера предполагает хранение некоторого уже введенного значения.

Современные нотации, в том числе UML 2 дает возможность рассматривать и более сложные элементы, например, на рисунке ниже показана история (рис.54).

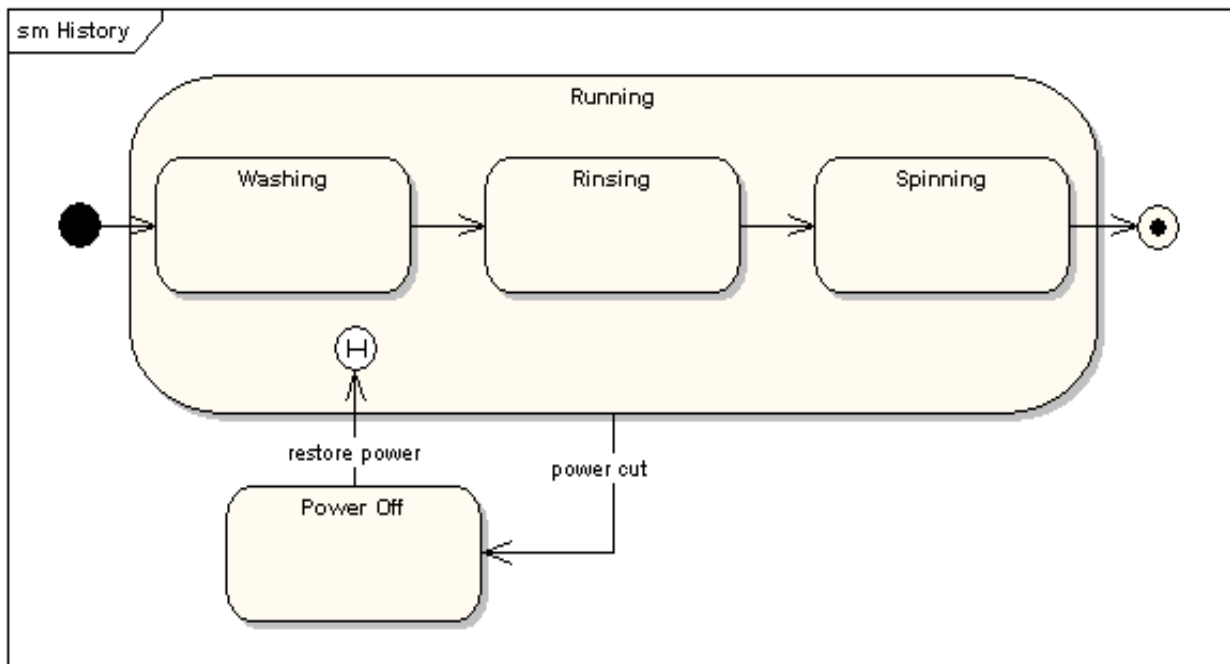
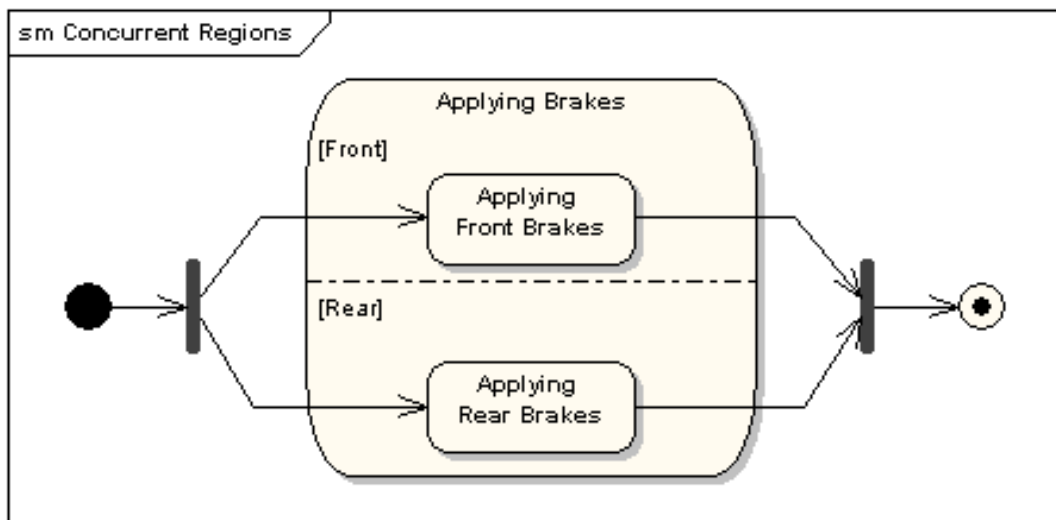


Рисунок 54 – Диаграмма состояния

Продолжая пример автомобиля, история позволяет нам, заведя двигатель, услышать или последнюю радиостанцию или последнюю композицию из плейлиста. То есть перед выходом из суперсостояния текущее вложенное состояние запоминается системой. При следующем входе в суперсостояние активно будет именно это вложенное состояние.

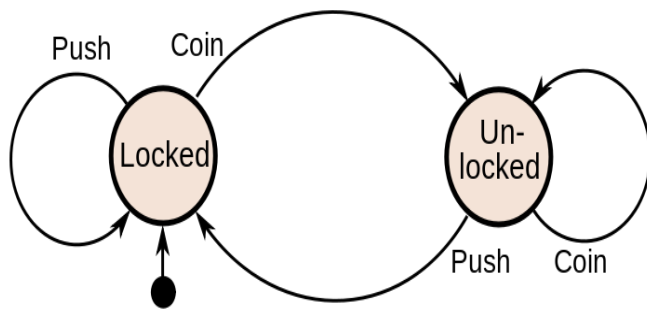


Наконец посмотрим на возможности разделения и наоборот объединения состояний системы по ходу выполнения задач.

С помощью псевдосостояния `fork` происходит переход системы в два параллельных состояния, после чего через какое-то времени, возможно, объединение этих потоков псевдосостоянием `join`.

Обратим внимание, диаграмма деятельности языка UML является частным случаем диаграммы состояний.

Одним из альтернативных способов задания диаграммы состояний является таблица переходов.



**Диаграмма**

Current State	Input	Next State
Locked	coin	Unlocked
	push	Locked
Unlocked	coin	Unlocked
	push	Locked

**Таблица**

### САМОСТОЯТЕЛЬНО

Построить таблицу переходов для примера светофора, рассмотренного ранее.

⚠ Показать результаты работы преподавателю.

Среди задач связанных с разработкой программного обеспечения интеллектуальных информационных систем одним из ключевых является процесс тестирования.

Цель тестирования – убедиться в отсутствии ошибок, точнее в снижение числа и критичности таких ошибок до приемлемого уровня.

Тестирование по машине состояний позволяет отслеживать и проверять сложное поведение программы.

Существуют два способ тестирования по машине состояний - тестировать систему во всех состояниях, проверяя при этом все переходы. Это более простой способ

Иногда нужно проверить не только отдельные состояния и переходы, но и сложные последовательности, пути: случай открыт -> переведен в другое отделение -> перевод отменен.

Таких путей очень много, поэтому тестируются сначала основные, по мере их важности и распространённости (рис.55).



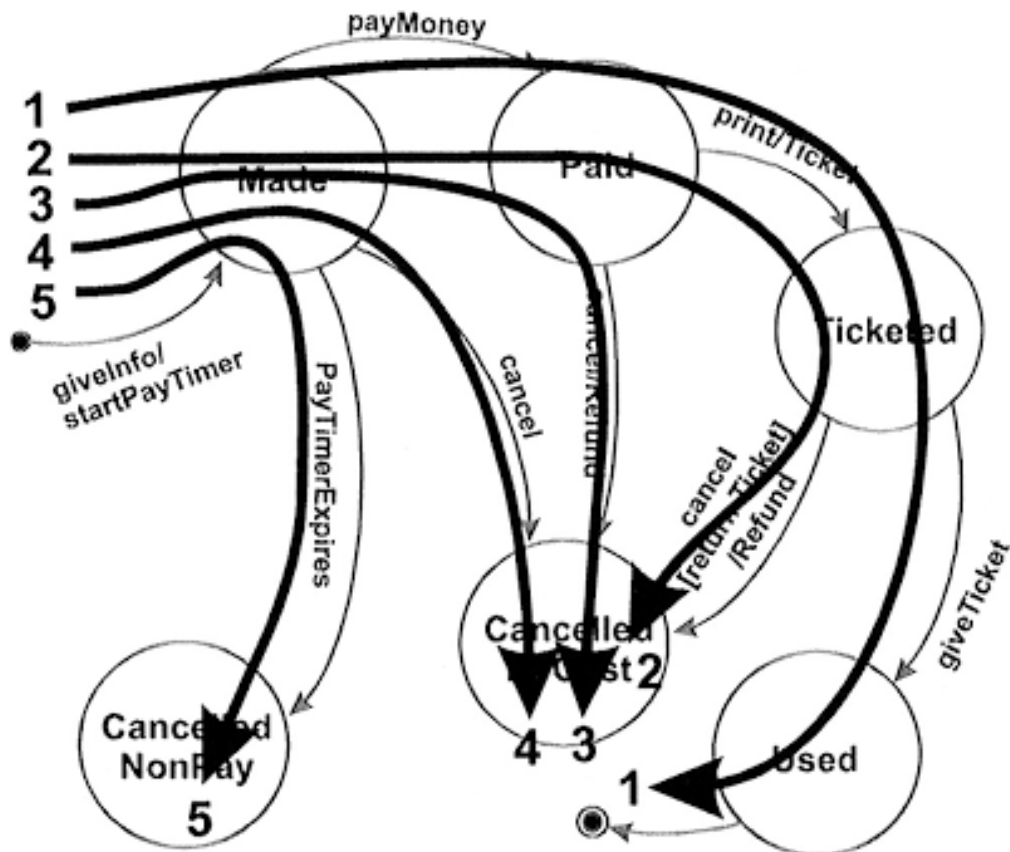


Рисунок 55 – Схема тестирования

### САМОСТОЯТЕЛЬНО

Разработать вариант диаграммы состояний и предложить способ тестирования программного обеспечения для одного выбранного из следующих объектов:

1. Домофон
2. Калькулятор
3. Мессенджер (например, Whatsapp)
4. Турникет метро
5. Банкомат
6. Холодильник
7. Лифт
8. Автомагнитола
9. Кофе-машина
10. Кондиционер

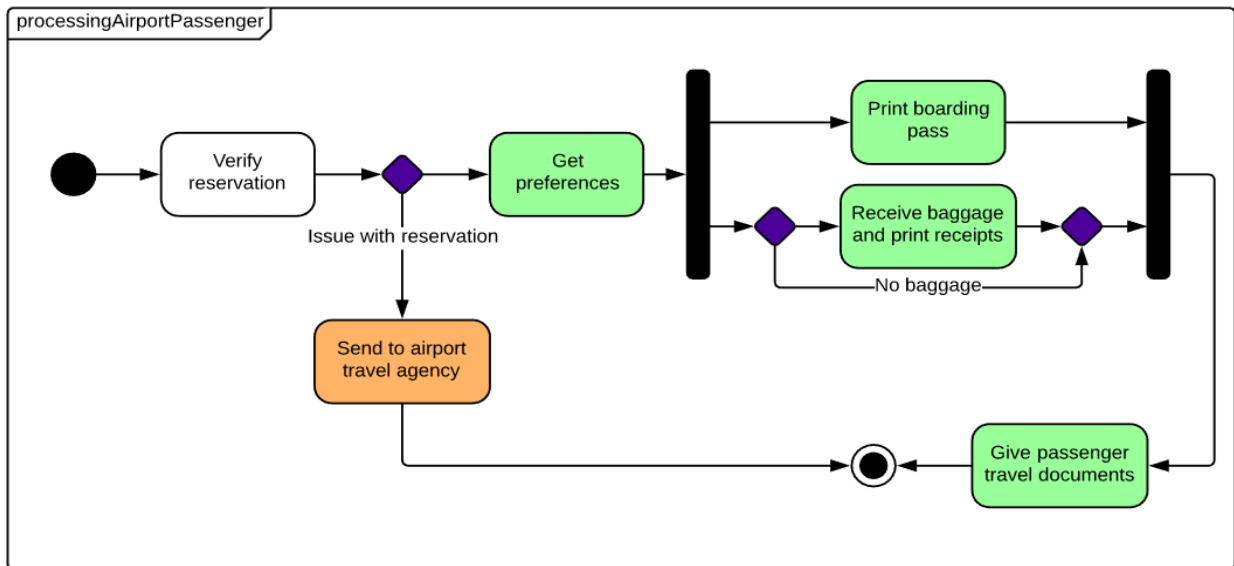


Показать результаты работы преподавателю.

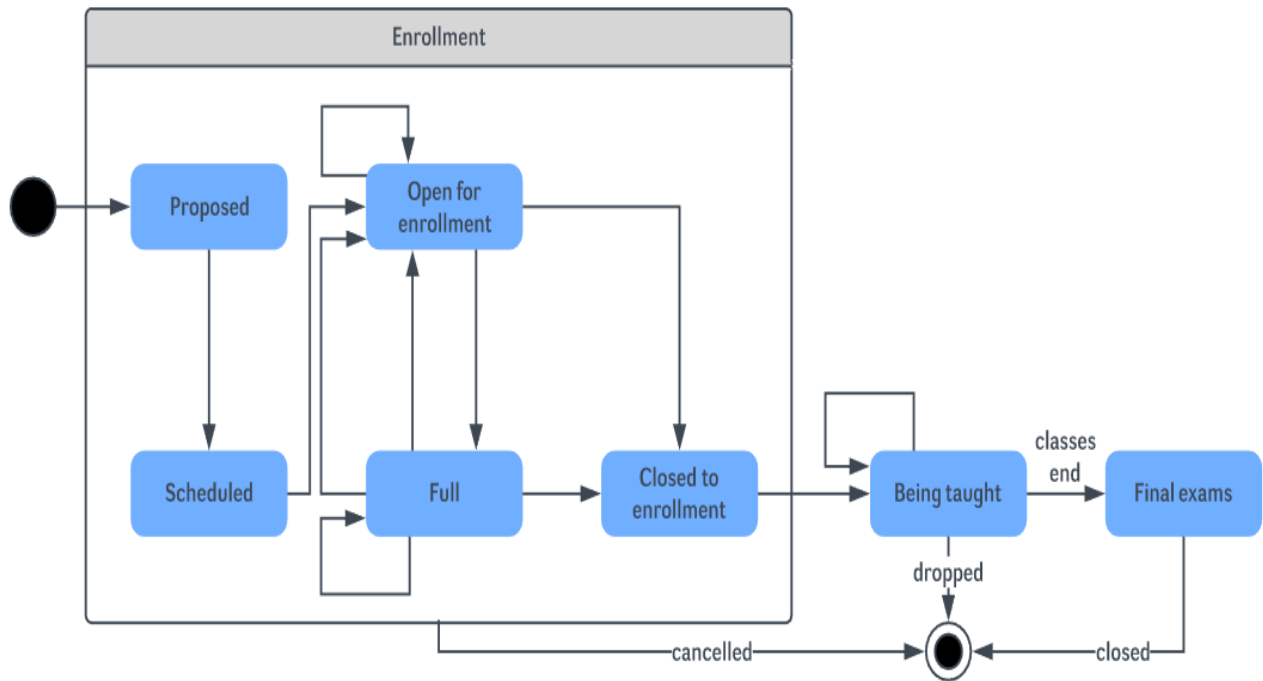
### САМОСТОЯТЕЛЬНО

Объяснить поведение системы в одной из выбранных диаграмм состояний из показанных ниже.

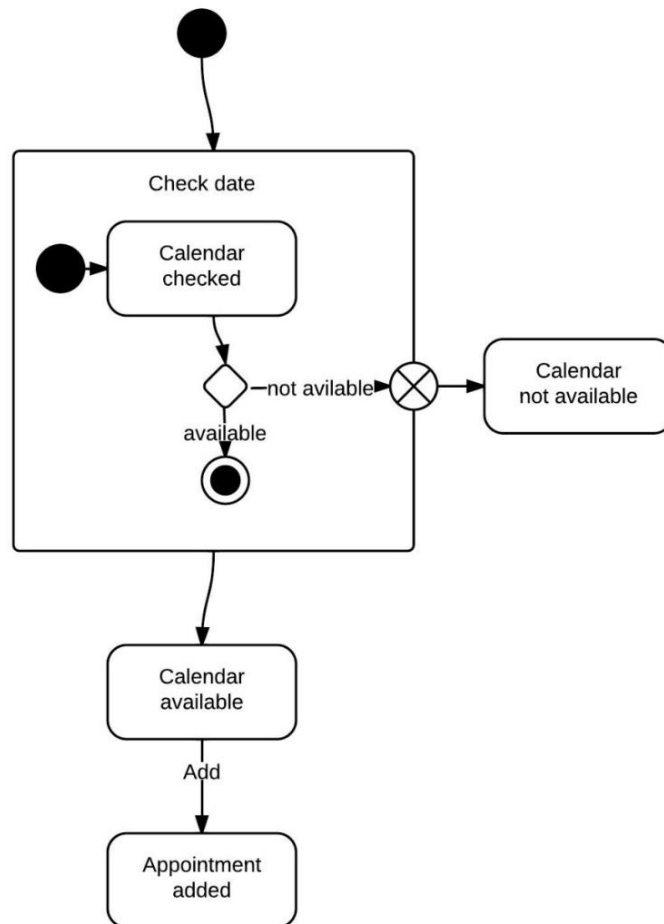
- 1 аэропорт, бронирование билетов пассажира



2 поступление



3 запись на прием



! Показать результаты работы преподавателю.

### САМОСТОЯТЕЛЬНО,

Разработать диаграмму 3 состояний для любого документа из выбранного варианта самостоятельного задания.

! Показать результаты работы преподавателю.

Диаграммы состояний тесно связаны с паттерном проектирования State или Состояние.

Состояние — это поведенческий паттерн проектирования, который позволяет объектам менять поведение в зависимости от своего состояния. Извне создаётся впечатление, что изменился класс объекта.

Основная идея в том, что программа может находиться в одном из нескольких состояний, которые всё время сменяют друг друга. Набор этих состояний, а также переходов между ними, предопределён и конечен. Находясь в разных состояниях, программа может по-разному реагировать на одни и те же события, которые происходят с ней.

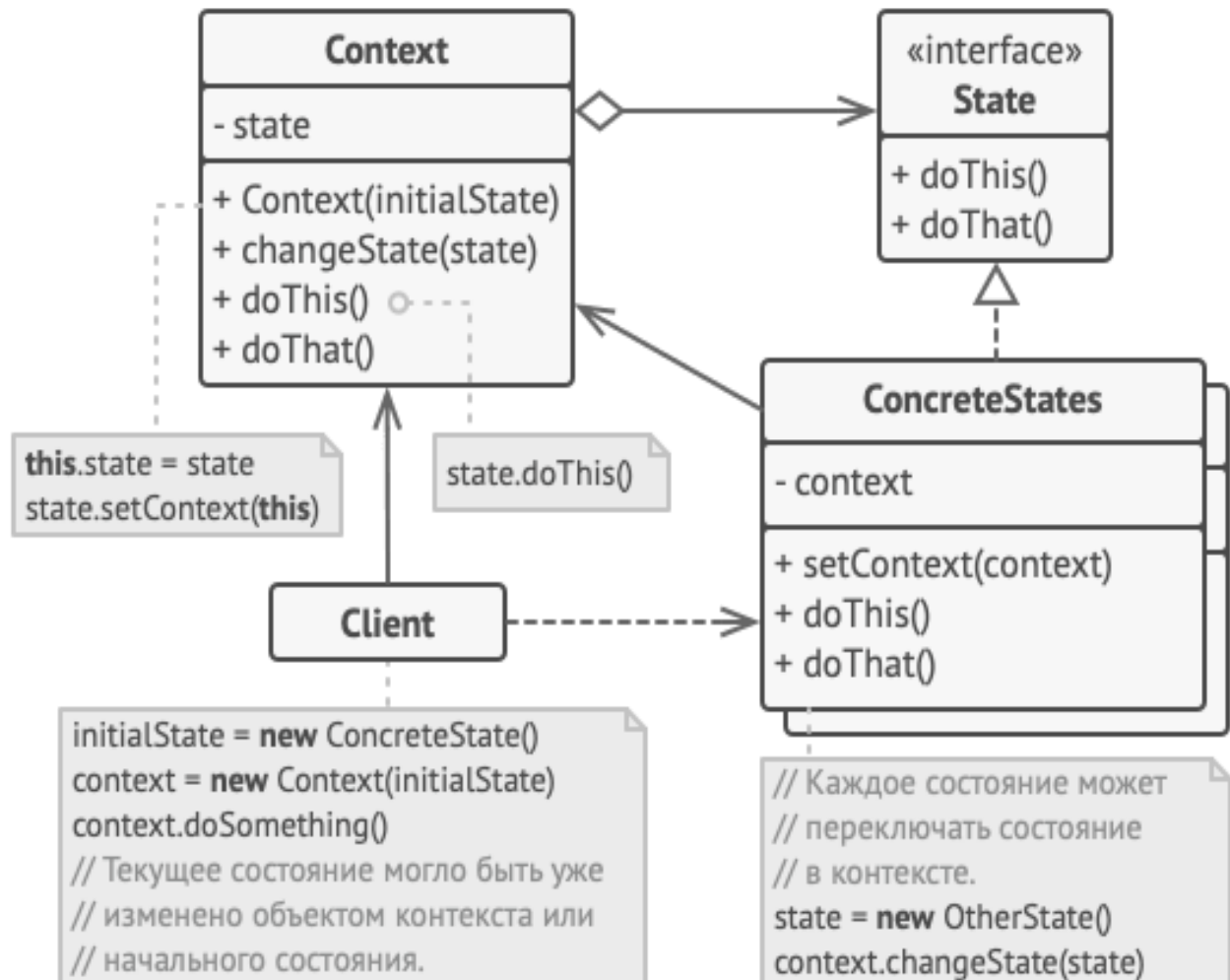
Такой подход можно применить и к отдельным объектам. Например, объект Документ может принимать три состояния: Черновик, Модерация или Опубликован. В каждом из этих состояний метод опубликовать будет работать по-разному:

Из черновика он отправит документ на модерацию.

Из модерации — в публикацию, но при условии, что это сделал администратор.

В опубликованном состоянии метод не будет делать ничего.

Схема паттерна на языке uml приведена на рисунке ниже.



Паттерн Состояние предлагает создать отдельные классы для каждого состояния, в котором может пребывать объект, а затем вынести туда поведения, соответствующие этим состояниям.

Вместо того чтобы хранить код всех состояний, первоначальный объект, называемый контекстом, будет содержать ссылку на один из объектов-состояний и делегировать ему работу, зависящую от состояния.

Класс контекста не реализует теперь все необходимые реакции, делегировав их классу состояние. Это абстрактный класс, он описывает свои возможности, но их реализация осуществлена в конкретных состояниях – его наследниках.

Рассмотрим пример из области фантастических космических кораблей. Класс Космический корабль реализует его основные характеристики. При этом часть реакции на события зависят от состояния корабля. Например, попадания вражеского выстрела вызовет разную реакцию с поднятыми или опущенными энергетическими щитами.

Конечно, можно реализовать это командами if .Но более гибким является подход, основанный на паттерне Состояние. Будет создан класс Состояние корабля. И несколько классов-наследников для конкретных состояний. Реакцию на выстрел hit – мы делегируем классу состояния.

# при поднятом щите энергия корабля сокращается на 10, а при опущенном - здоровье корабля снижается на силу атаки

```
class Spaceship_State:
    def hit(self, ship, power):
        pass

class Spaceship_State_off(Spaceship_State):
    def hit(self, ship, power):
        ship.hp-=power

class Spaceship_State_on(Spaceship_State):
    def hit(self, ship, power):
        ship.hp-=5

class Spaceship_State_hyper(Spaceship_State):
    def hit(self, ship, power):
        ship.energy+=10

class Space_object:
    def get_max_speed():
        pass

class Spaceship(Space_object):
    def __init__(self, name, max_hp, energy, max_speed):
        self.max_hp=max_hp
        self.hp=max_hp
        self.name=name
        self.energy=energy
        self.shield_state=Spaceship_State_off()
        self.speed=0
        self.max_speed=max_speed

    def shield_on(self):
        self.shield_state=Spaceship_State_on()
        self.energy-=30
```

```

def shield_off(self):
self.shield_state=Spaceship_State_off()

def hit(self,power):
self.shield_state.hit(self,power)

def __str__(self):
return f'Космический корабль {self.name} состояние
{self.hp}'

```

### САМОСТОЯТЕЛЬНО

Привести пример применения паттерна Состояние. Разобрать программный код на одном из языков программирования.

### Вопросы для самопроверки.

1. Сколько времени занимает переход?
2. Что такое триггерный переход?
3. Может ли система находиться в двух различных параллельных состояниях?
4. Что такое псевдосостояние?
5. Что такое суперсостояние?
6. Как и зачем реализуется история?
7. Может ли система переходить в то же состояние, в котором она была?
8. Что такое петля?
9. Какие существуют псевдосостояния?
10. Что делает пара `fork` и `join`,
11. Что такое паттерн `State`?

### Выполните тест:

1. Основными элементами диаграммы состояний являются:
  - a) состояние
  - b) блок
  - c) стрелка
  - d) переход
  
2. Переход системы в то же состояние, в котором она была, называется ....
  - a) переход
  - b) триггерный переход
  - c) петля
  - d) возврат состояния

- 3. Это состояние может быть только одно на всю рассматриваемую модель:**
- начальное состояние
  - конечное состояние
  - суперсостояние
- 4. Переход системы в два параллельных состояния происходит помощью...**
- Псевдосостояния join
  - псевдосостояния fork
  - таблицы переходов
- 5. Состояние системы может иметь некоторые...**
- Переходы
  - Петли
  - Триггерные переходы

**Таблица 13 – Правильные ответы по тесту**

Номер вопроса	Номер правильного ответа
1	a, d
2	c
3	a
4	b
5	d

### **1.1.3.6 Моделирование информационной системы на основе структурного подхода. Построение моделей IDEF, DFD диаграмм, ER-диаграмм.**

#### **Цель практической работы:**

- изучить принципы построения моделей IDEF;
- диаграмм в нотации DFD. Это, так называемые диаграммы потоков данных;
- получить навыки разработки диаграмм данного вида.

Рассмотрим еще один пример разработки модели IDEF, создавая диаграммы бизнес-процесса «Взаиморасчеты за оказанные услуги». Данный бизнес-процесс является результатом декомпозиции диаграммы «Финансово-экономической деятельности» на более мелкие процессы.

Оказание платных медицинских услуг регулируется следующими законодательными и нормативными документами:

- ФЗ от 21.11.2011 № 323-ФЗ «Об основах охраны здоровья граждан в Российской Федерации»;
- ФЗ от 07.02.1992 № 2300-1 «О защите прав потребителей»;

- Гражданским кодексом РФ.
- Лицензия на право медицинской деятельности

Предоставление платных медицинских услуг оформляется договором с пациентом, который может заключаться только в отношении видов деятельности, которые перечислены в лицензии.

Также в договоре присутствуют стандартные для этого вида документов разделы:

- условия и сроки получения платных медицинских услуг;
- порядок расчетов;
- права и обязанности сторон;
- ответственность сторон.

На предоставление платных медицинских услуг может быть составлена смета. Но в данном примере на основе договора выписывается документ «Счет на оплату».

Декомпозируем на диаграмме «Финансово-экономическая деятельность» работу «Ведение договоров с пациентами» (рис. 56).

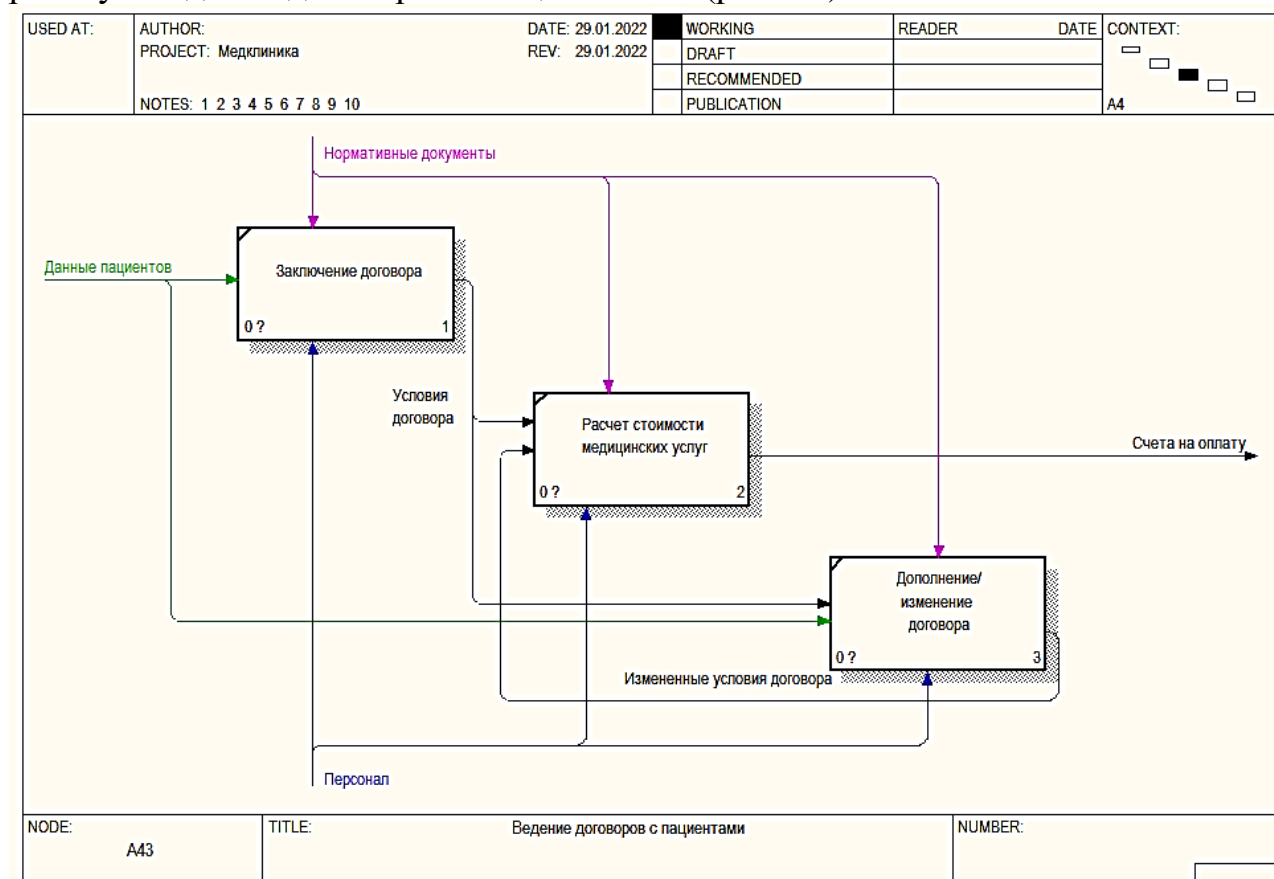


Рис. 56 – Диаграмма декомпозиции 2-го уровня бизнес-процесса «Ведение договоров с пациентами»

Перед декомпозицией работы «Взаиморасчеты за оказанные услуги» сохраните файл **ФИО студента.br1** и сделайте его копию. В дальнейшей работе процесс «Взаиморасчеты за оказанные услуги» будет еще раз декомпозирован, но в другой нотации, поэтому и понадобилась копия.



Откройте снова файл **ФИО студента.bp1** и выполните следующие действия:

1. Перейти на диаграмму «Финансово-экономическая деятельность» и декомпозировать работу «Взаиморасчеты за оказанные услуги».
2. В диалоге **Activity Box Count** установите число работ 2 и нотацию **IDEF** (рис. 57).

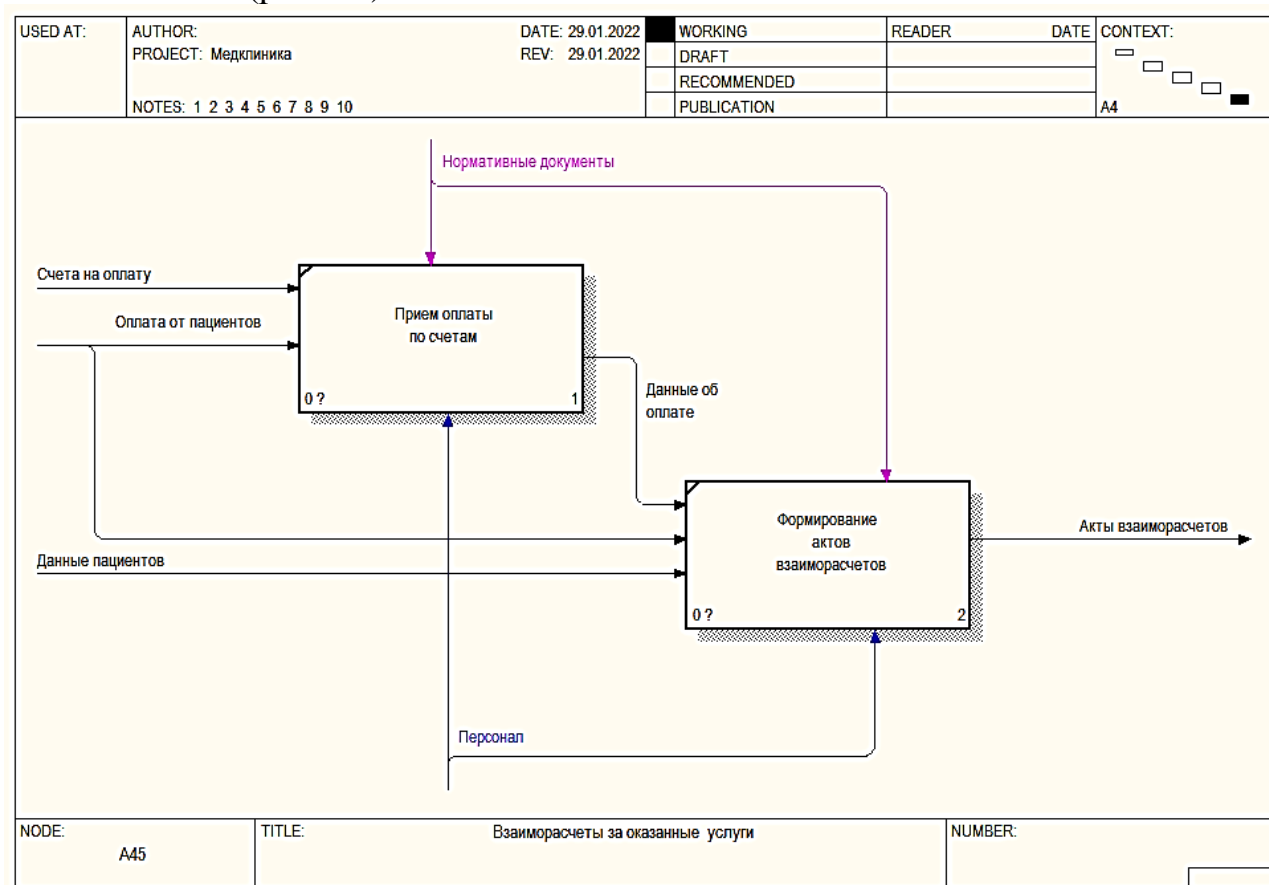


Рис. 57 – Диаграмма декомпозиции 2-го уровня бизнес-процесса «Взаиморасчеты за оказанные услуги»

Сохраните файл **ФИО студента.bp1**.

Откройте копию файла для создания диаграммы потоков данных, которая предназначена для наглядного представления какие потоки информации и по каким маршрутам перемещаются в системе. Эта диаграмма является дополнением к нотации IDEF, и в то же время – одним из основных инструментов структурного анализа бизнес-процессов.

Понятие и принципы разработки DVD диаграммы в предыдущих работах еще не рассматривались, поэтому сначала немного теории.

### Основные термины и определения.

**Диаграммы потоков данных (Data flow diagramming, DFD)** можно использовать как дополнение к модели **IDEF0** для более наглядного отображения текущих операций документооборота в системах обработки информации.

**Диаграммы потоков данных** используются для описания документооборота и обработки информации и представляют модельную систему как сеть связанных между собой работ.




**Диаграммы потоков данных (DFD)** показывают внешние источники и приемники данных, потоки данных и хранилища (накопители) данных, к которым осуществляется доступ.

Диаграмма **DFD** описывает:

- 1) функции обработки информации (**работы**);
- 2) документы (**стрелки, arrows**), объекты, сотрудников или отделы, которые участвуют в обработке информации;
- 3) внешние ссылки (**external references**), которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы;
- 4) таблицы для хранения документов (**хранилища данных, data store**).

Для изображения диаграмм потоков данных в **VRwin** используется нотация **Гейна-Сарсона** (таблица 14).


Таблица 14 - Нотация Гейна - Сарсона


Компонента	Изображение в нотации Гейна-Сарсона
Поток данных	Имя →
Управляющий процесс	
Хранилище данных	
Внешняя сущность	

#### Этапы построения модели

1. Выделение внешних объектов, с которыми система должна быть связана.
2. Формирование DFD диаграммы первого уровня.
3. Функциональная декомпозиция каждого подпроцесса с помощью диаграмм нижнего уровня.
4. Составление словаря данных.
5. Построение спецификаций процесса, если его нельзя выразить комбинацией подпроцессов.

При дополнении модели IDEF0 диаграммой DFD, в палитре инструментов на новой диаграмме **DFD** появляются новые кнопки:

 - добавить в диаграмму внешнюю ссылку (*External Reference*). Внешняя ссылка является источником или приемником данных извне модели.

 - добавить в диаграмму хранилище данных (*Data store*). Хранилище данных позволяет описать данные, которые необходимо сохранить в памяти прежде, чем использовать в работах.

В отличие от IDEF0, где система рассматривается как взаимосвязанные работы, DFD рассматривает систему как совокупность предметов.

**Работы** в DFD представляют собой функции системы, преобразующие *входы* в *выходы*. Хотя работы изображаются прямоугольниками со скругленными углами, смысл их совпадает со смыслом работ в IDEF0, они имеют входы и выходы, но не поддерживают управления и механизмы, как IDEF0 (рис. 58).

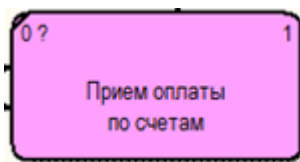


Рис. 58 - Изображение работы

**Внешние сущности** изображают входы в систему и/или выходы из системы. Внешние сущности изображаются в виде прямоугольника с тенью и обычно располагаются по краям диаграммы (рис. 35). Одна внешняя сущность может быть использована многократно на одной или нескольких диаграммах. Обычно такой прием применяют, чтобы не рисовать слишком длинных и запутанных стрелок.

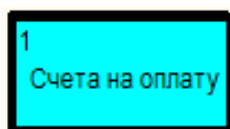


Рис. 35 - Внешняя сущность

**Стрелки (потоки данных)** описывают движение объектов из одной части системы в другую. Поскольку в DFD каждая сторона работы не имеет четкого назначения, как в IDEF0, стрелки могут подходить и выходить из любой грани прямоугольника работы. В DFD также применяются двунаправленные стрелки для описания диалогов типа команды-ответа между работами, между работой и внешней сущностью и между внешними сущностями (рис. 59).

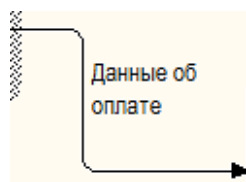


Рис. 59 - Потоки данных (стрелки)

**Хранилище данных** изображают объекты в покое. В материальных системах хранилища данных изображаются там, где объекты ожидают обработки, например в очереди. В системах обработки информации хранилища данных являются механизмом, который позволяет сохранить данные для последующих процессов (рис. 60).

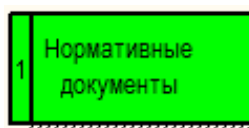


Рис. 60 - Хранилище данных

### Задание 11. Создание DFD диаграммы бизнес-процесса «Взаиморасчеты за оказанные услуги».

Для создания диаграммы DFD, например, бизнес-процесса «Взаиморасчеты за оказанные услуги» необходимо выполнить следующие действия:

3. Перейти на диаграмму «Финансово-экономическая деятельность» и декомпозировать работу «Взаиморасчеты за оказанные услуги».
4. В диалоге **Activity Box Count** установите число работ 2 и нотацию DFD (рис. 61).

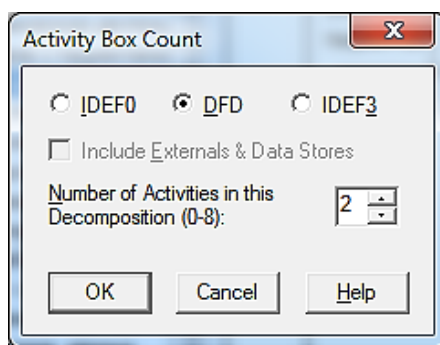


Рис. 61 - Диалог Activity Box Count

5. Появится диаграмма DFD, содержащая работы (рис.62).

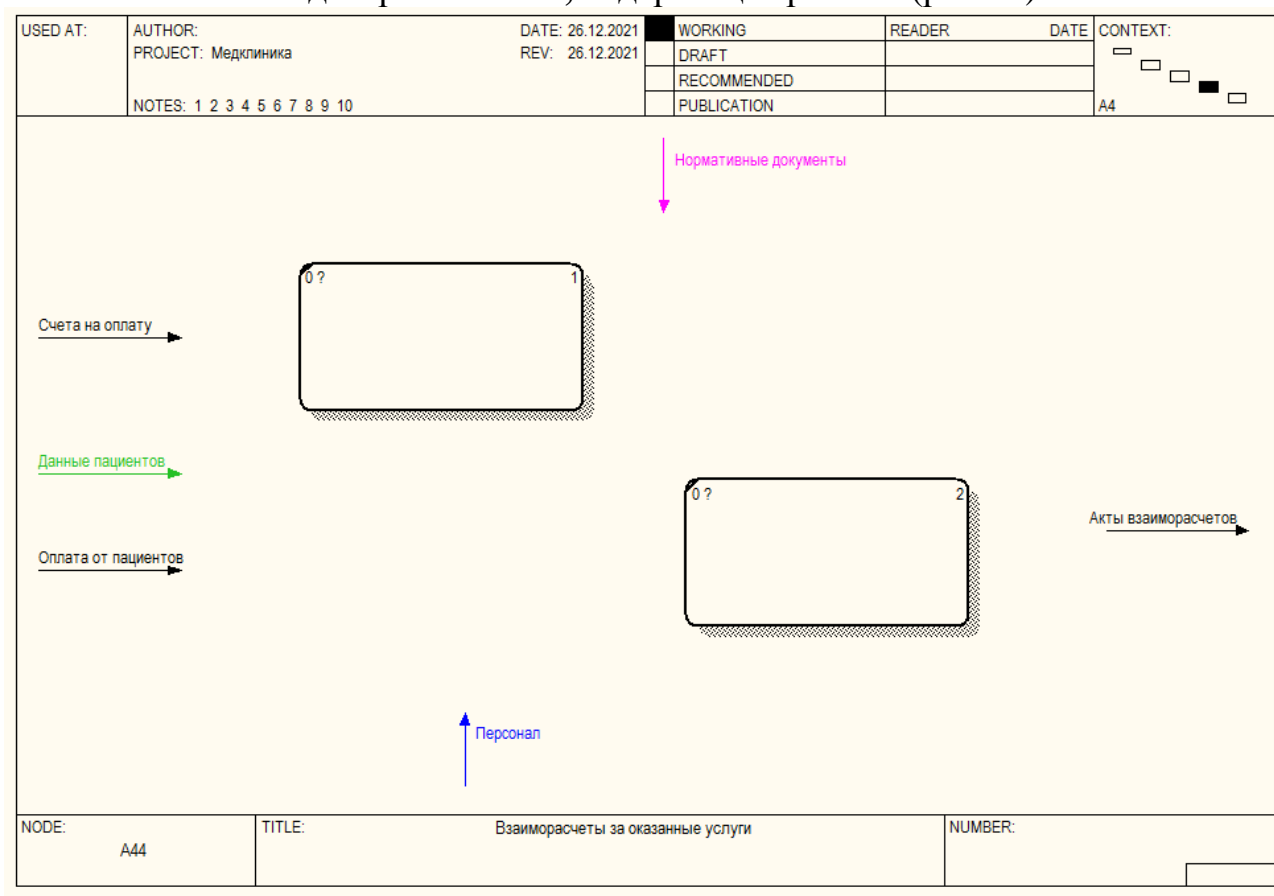


Рис. 62 – Диаграмма DFD

6. Дважды щелкнув левой клавишей мыши по работе, внести в окне **Activity Properties** имя работы (рис. 63).

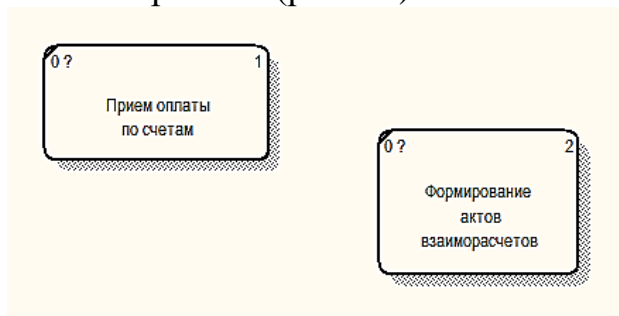


Рис. 63 – Ввод имен работ

7. Приграничные стрелки удаляются. Вместо них на диаграмму добавляются внешние сущности и хранилища данных.

В отличие от стрелок **IDEF0**, которые представляют собой жесткие взаимосвязи, стрелки **DFD** показывают, как объекты (включая данные) двигаются от одной работы к другой. Это представление потоков совместно с хранилищами данных и внешними сущностями делает модели **DFD** более похожими на физические характеристики системы - движение объектов (data flow), хранение объектов (data stores), поставка и распространение объектов (external entities) (рис. 64).

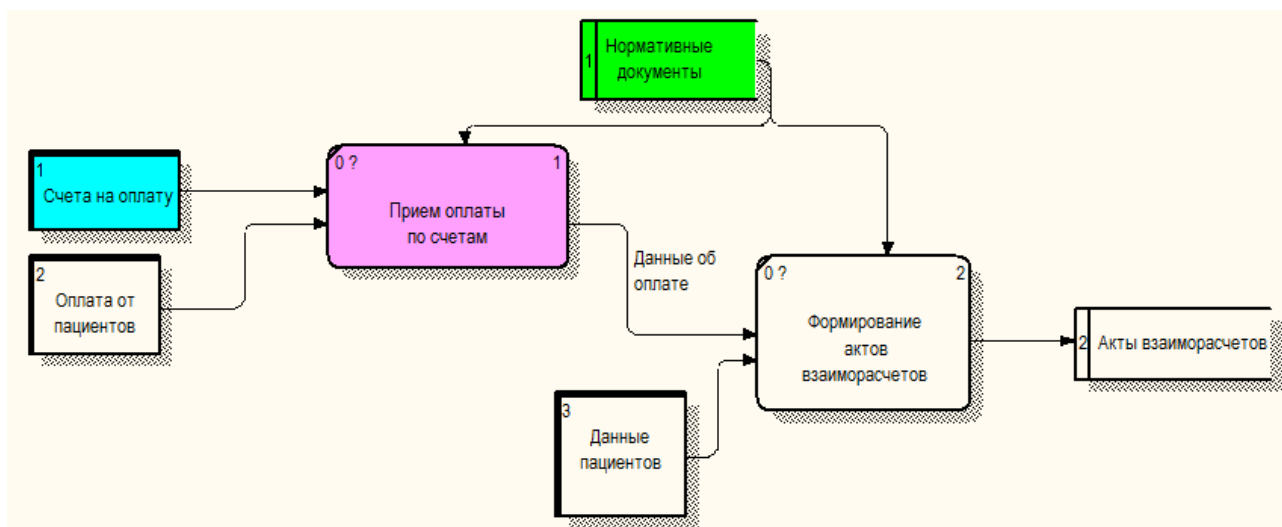


Рис. 64 – DFD диаграмма бизнес-процесса «Взаиморасчеты за оказанные услуги»

### САМОСТОЯТЕЛЬНО.

Построить диаграмму DFD любого бизнес-процесса из выбранного варианта самостоятельного задания.

⚠ Показать результаты работы преподавателю.

**Вопросы для самопроверки:**

1. Для чего служит DFD - диаграмма?
2. В чем отличие DFD - диаграммы от IDFE0?
3. Какой инструмент используется для построения внешних сущностей?
4. Каким инструментом можно построить Хранилище данных?
5. Какие графические элементы используются для обозначения на диаграмме Работы, Поточков данных, Хранилищ данных?

**Выполните тест:**

**1. DFD - это:**

- a) диаграмма бизнес - процесса;
- b) диаграмма потока данных;
- c) контекстная диаграмма;
- d) функциональная диаграмма;
- e) диаграмма сущность - связь.






**2. Какие объекты описываются на диаграмме потоков данных:**

- a) функции (работы);
- b) хранилища данных;
- c) внешние объекты;
- d) внешние ссылки;
- e) все ответы правильные.

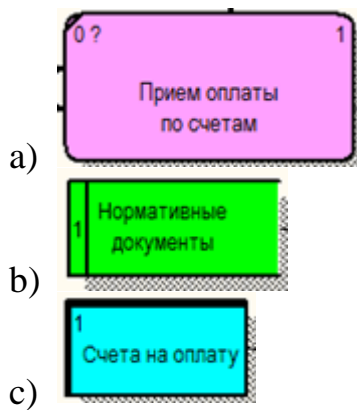
**3. Что показывают диаграммы потоков данных?**

- a) внешние источники и приемники данных;
- b) потоки данных;
- c) хранилища (накопители) данных;
- d) все ответы правильные;
- e) правильного ответа нет.

**4. Какой инструмент нужно выбрать для создания диаграммы DFD?**

- a) ;
- b) ;
- c) ;
- d) ;
- e) .

**5. Укажите номер рисунка, обозначающий на диаграмме Хранилище данных.**



6. Каким инструментом можно обозначить в диаграмме DFD Внешнюю сущность?

- a) ;
- b) ;
- c) ;
- d) ;
- e) .

Таблица 15 – Правильные ответы по тесту

Номер вопроса	Номер правильного ответа
1	b
2	e
3	d
4	a
5	b
6	c

### Разработка ER-модели.

**Цель практической работы:** Построение модели сущность-связь (ER-диаграммы).

Модель «Сущность-связь» или ER-диаграмма (Entity-Relationship Diagram) представляет собой диаграмму (блок-схемы), где показано, как разные сущности связаны и взаимодействуют между собой внутри системы. В качестве таких сущностей могут выступать люди, объекты, концепции и так далее.

В моделях «Сущность-связь» обычно выделяют три уровня детализации:

- **Концептуальный уровень** — это первый уровень, который отображает лишь общую структуру модели и всю архитектуру системы.

- **Логический уровень** содержит более подробную информацию: атрибуты и операции. Но при этом логическая модель не зависит от технологии, в которой она будет применяться.
- **Физический уровень** составляется на основе логической модели с техническими подробностями (типами данных атрибутов) для составления и внедрения самой базы данных.

Основные символы нотации ER представлены в таблице 16.

Таблица 16 - символы нотации ЕС

Графический символ	Описание
<b>Сущности</b>	
	Независимые сущности сопровождаются первичным ключом, который позволяет идентифицировать каждый экземпляр сущности.
	Сущность, которая зависит от сущности другого типа. Не сопровождается первичным ключом.
	Соединяет экземпляры сущностей разных типов. Также содержит атрибуты, характерные для связей между этими сущностями.
<b>Связи</b>	
	Отношение между сущностями.
	Связь между зависимой сущностью и сущностью, от которой она зависит.
<b>Атрибуты</b>	
	Характеризует сущность, а также отношения между двумя или более элементами.
	Атрибут, которому может быть присвоено несколько значений.
	Атрибут, чье значение можно вычислить, опираясь на значения связанных с ним атрибутов.

Этапы создания простейшей ER-диаграммы:

1. Определить сущности – это основные составляющие модели «сущность – связь».
2. Установить связи, т.е. показать, как сущности взаимодействуют между собой.

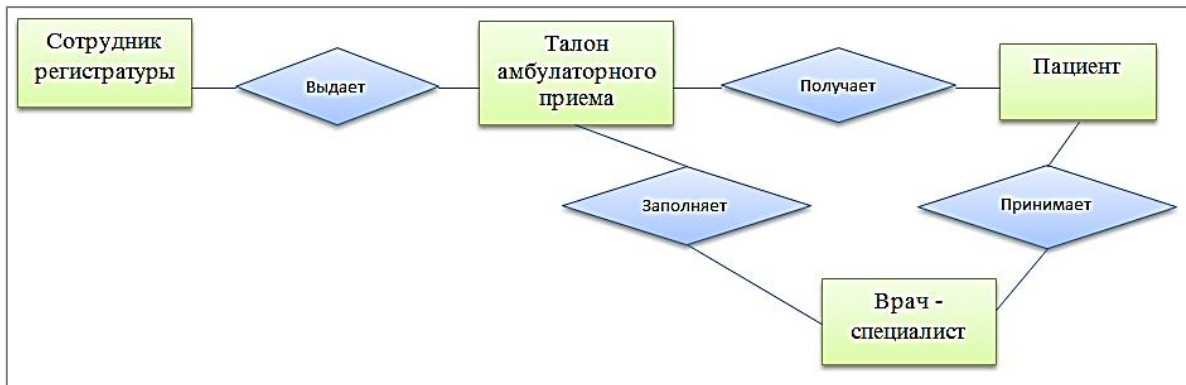


3. Определить атрибуты, необходимые для моделирования характеристик, которые будут сопровождать каждую сущность ER-диаграммы. Чаще всего встречаются атрибуты, как «ID-номер», «Наименование».

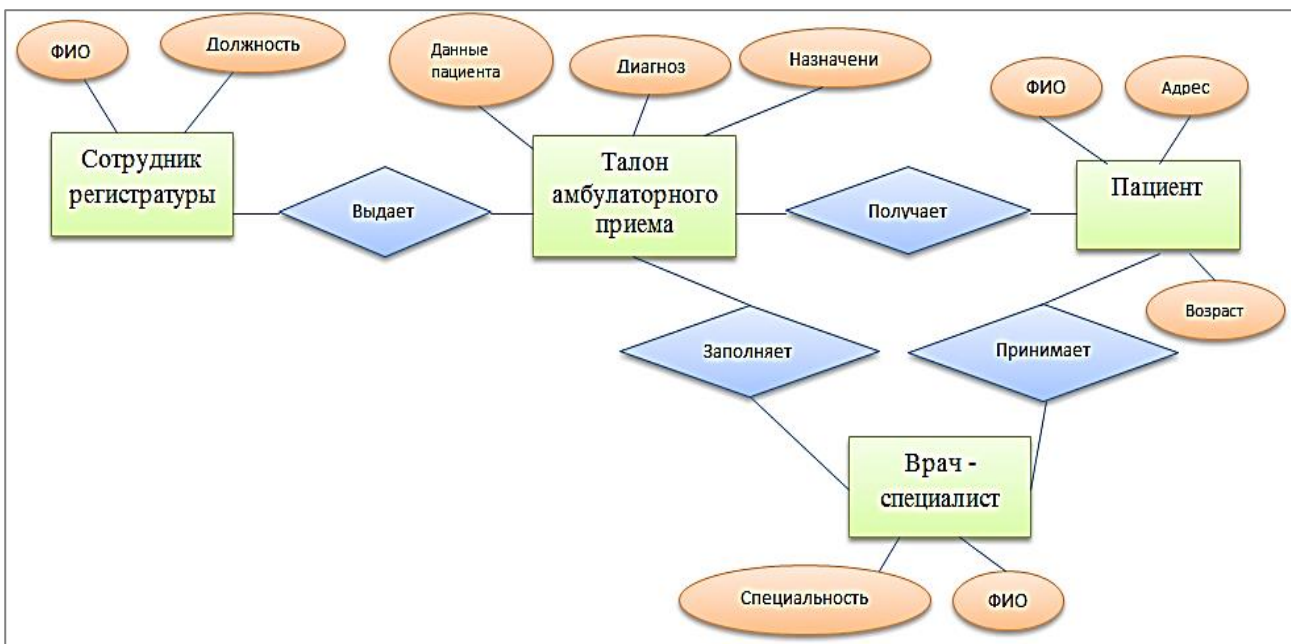
Рассмотрим построение ER-диаграммы на примере деятельности медклиники. Первый этап – определение сущностей (рис. 65).



Следующий этап – определить, как сущности взаимодействуют между собой (Рис. 66).



И, заключительный этап, определение атрибутов (рис. 67).



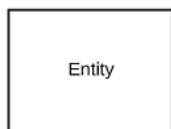
## САМОСТОЯТЕЛЬНО.

Построить модель «Сущность-связь» (ER-диаграмму) выбранного варианта самостоятельной работы.

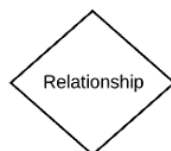
! Показать результаты работы преподавателю.

### Вопросы для самопроверки:

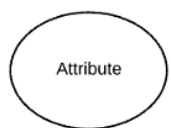
1. Что представляет собой ER-диаграмма?
2. Перечислите уровни детализации модели «Сущность-связь».
3. Чем отличаются уровни детализации модели «Сущность-связь».
4. Что представляет собой концептуальный уровень?
5. Что представляет собой логический уровень?
6. Что представляет собой физический уровень?
7. Этапы создания ER-диаграммы.



8. - что обозначает этот символ в ER-диаграмме?



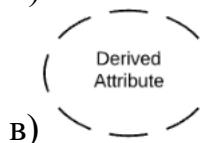
9. - для чего предназначен этот символ?



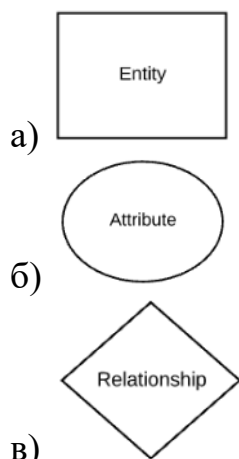
10. - какое описание вы дадите этому символу?

### Выполните тест:

1. Укажите графический символ атрибута, которому может быть присвоено несколько значений:



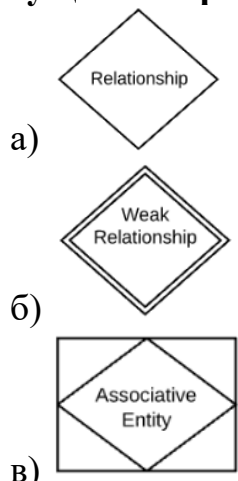
2. Характеризует сущность, а также отношения между двумя или более элементами:



3. Первый уровень, который отображает лишь общую структуру модели и всю архитектуру системы – это ...

- а) концептуальный уровень;
- б) логический уровень;
- в) физический уровень.

4. Этот символ нотации ER-диаграммы соединяет экземпляры сущностей разных типов:



5. Первичным ключом сопровождаются:

- а) сущность, которая зависит от сущности другого типа;
- б) независимая сущность;
- в) оба ответа правильные.

**Таблица 17 - Правильные ответы по тесту**

Номер вопроса	Правильный ответ
1	б
2	б
3	а
4	в
5	б

### 1.1.3.7 Моделирование информационной системы на основе объектно-ориентированного подхода. Построение моделей UML 2.0.

Объектно-ориентированное моделирование (Таблица 18) подразумевает описание процессов, как набора взаимодействующих объектов без детализации выполняемых операций, но с описанием условий и событий. Объект – это какой-либо предмет, который преобразуется при выполнении процессов. В основе – объектная модель, которая базируется на таких принципах, как инкапсуляция, абстрагирование, полиморфизм, наследование, параллелизм, устойчивость и т.д. При этом статическую структуру модели описывают объекты, а поведение модели – сообщения, которыми эти объекты обмениваются.

Таблица 18 – Методы и язык объектно-ориентированного моделирования

Вид моделирования	Метод	Язык графического описания
Объектно-ориентированное моделирование	Метод Гради-Буча	UML - Unified Modeling Language), или унифицированный язык моделирования
	Метод Джеймса Румбаха (OMT)	
	Метод Айвара Джекобсона (OOSE)	

Модель UML представляет собой совокупность специальных графических конструкций, получивших название диаграмм. Диаграммы подразделяются на структурные диаграммы (статические) и диаграммы поведения (динамические) (рис. 68).





Рис. 68 Виды диаграмм UML

Диаграмма вариантов использования (сценариев поведения, прецедентов) относится к динамическому типу диаграмм UML. Данная диаграмма состоит из

актеров, вариантов использования и отношений между ними (в таблице 19 представлены графические элементы и их описание). При построении диаграммы могут использоваться также общие элементы нотации: примечания и механизмы расширения.

Таблица 19 – Графические элементы диаграммы вариантов использования и их описание

Графический элемент	Описание
	Одно лицо или группа лиц или систем, взаимодействующих с моделируемой системой
	варианты использования (прецеденты, сценарии) - сервисы (функции), которые моделируемая система предоставляет актерам
	Отношение ассоциации - служит для обозначения взаимодействия актера с вариантом использования.
	Отношение обобщения - служит для обозначения взаимодействия актера с вариантом использования.
	Отношение включения указывает, что некоторое заданное поведение одного варианта использования обязательно включается в качестве составного компонента в последовательность поведения другого варианта использования.
	Отношение расширения определяет потенциальную возможность включения поведения одного варианта использования в состав другого

В п. 1.1.3.2 уже рассматривался процесс построения диаграммы вариантов использования (сценариев) в программе проектирования RSA. Диаграмма приведена на рисунке 69.

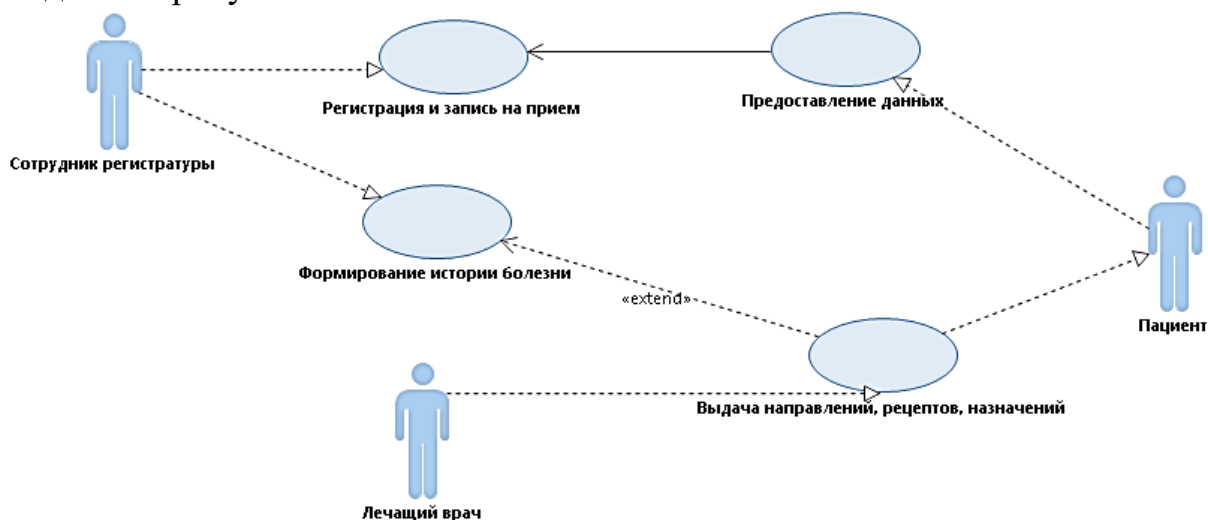


Рис. 69 – Диаграмма вариантов использования проведения приема пациента

В пункте 1.1.3.4 в целях создания удобного пользовательского интерфейса были разработаны диаграмма сценариев сотрудника регистратуры (рис. 70) и диаграмма сценариев врача-специалиста (рис.71).

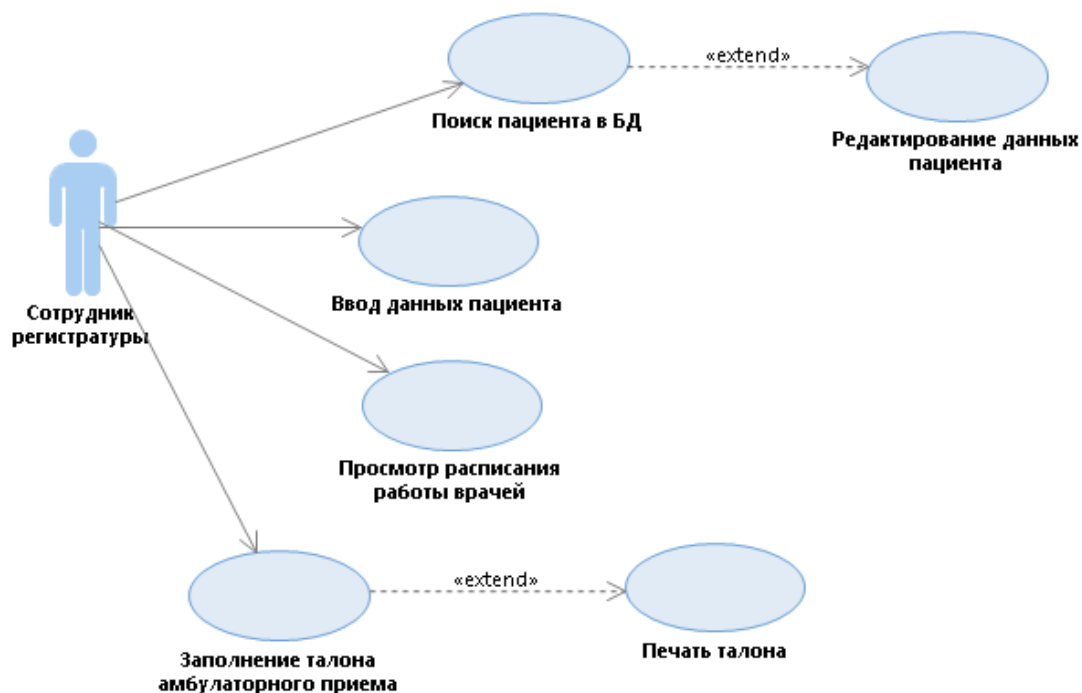


Рис. 70 – Диаграмма сценариев сотрудника регистратуры

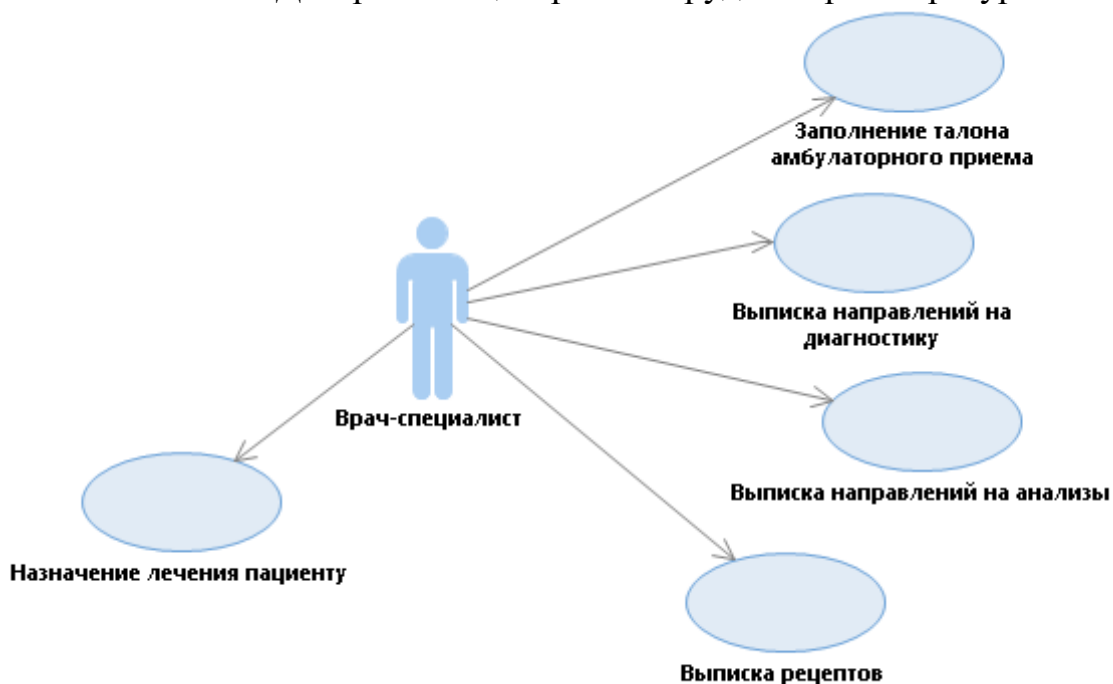
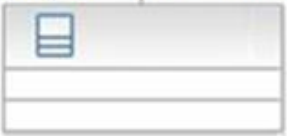






Рис. 71 – Диаграмма сценариев врача-специалиста

Основной структурной диаграммой является диаграмма классов - это одна из форм описания системы с точки зрения ее проектирования, показывающая её структуру. Диаграмма классов не отображает динамическое

поведение объектов, изображенных на ней классов. На диаграммах классов показываются классы, интерфейсы и отношения между ними (Таблица 20).

Таблица 20 - Графические элементы диаграммы классов и их описание

Графический элемент	Описание
	Класс (class) в языке UML служит для обозначения множества объектов, которые обладают одинаковой структурой, поведением и отношениями с объектами других классов. Графически класс изображается в виде прямоугольника, который дополнительно может быть разделен горизонтальными линиями на разделы или секции. В этих разделах могут указываться имя класса, атрибуты (переменные) и операции (методы).
	Отношение зависимости – семантически представляет собой связь между двумя элементами модели, в которой изменение одного элемента (независимого) может привести к изменению семантики другого элемента (зависимого).
	Отношение ассоциации – это структурная связь между элементами модели, которая описывает набор связей, существующих между объектами.
	Отношение обобщения – выражает специализацию или наследование, в котором специализированный элемент (класс-потомок) строится по спецификациям обобщенного элемента (класса-родителя).
	Отношение реализации – это семантическая связь между классами, когда один из них (поставщик) определяет соглашение, которого второй (клиент) обязан придерживаться.

На рисунке 72 показан пример класса «Врач».

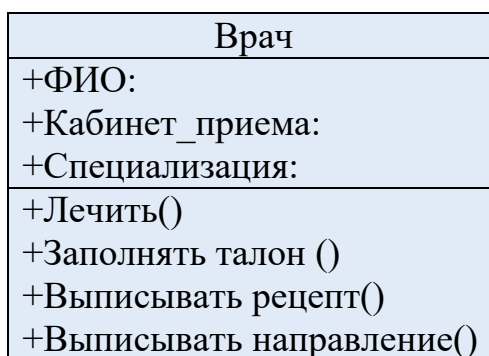


Рисунок 72 – Пример класса «Врач»

**САМОСТОЯТЕЛЬНО.** Создать диаграмму классов модели «Медклиника».

Для создания диаграммы классов откройте проект **ФИОстудентаUML** в окне Project Explorer, кликните правой кнопкой на My Interaction Diagram Model (под узлом Models), и кликните Add Diagram > Class Diagram. Откроется новая пустая диаграмма классов. На рис. 73 представлена панель инструментов для создания диаграммы классов.

Добавьте класс. Для добавления атрибутов класса кликните правой кнопкой на классе и выберите Add UML > Attribute. Для добавления операций класса - Add UML > Operation.

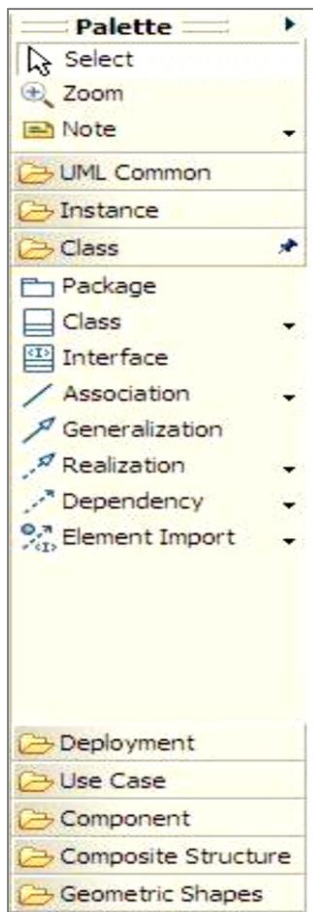


Рис. 73 – Панель инструментов для создания диаграммы классов

⚠ Показать результаты работы преподавателю.

⚠ Сохранить проект **ФИОстудентаUML**.

Далее, перейдем к рассмотрению динамических диаграмм. С двумя из них мы уже познакомились: в п. 1.1.3.4 с диаграммой взаимодействия (последовательностей) и в п. 1.1.3.5 с диаграммой состояния.

Диаграммы последовательностей используются для уточнения диаграмм прецедентов, более детального описания логики сценариев использования. Они обычно содержат объекты, которые взаимодействуют в рамках сценария, сообщения, которыми они обмениваются, и возвращаемые результаты, связанные с сообщениями. Графические элементы диаграммы последовательностей представлены в таблице 21.



Таблица 21 – Графические элементы диаграммы последовательностей

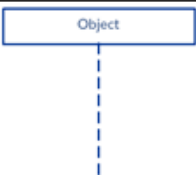
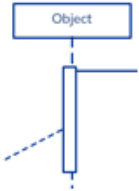



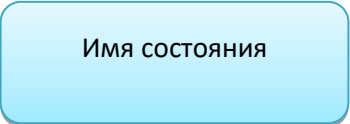

Графический элемент	Описание
	Линия жизни. Последовательность состоит из нескольких таких обозначений линии жизнеобеспечения, которые должны быть расположены горизонтально в верхней части диаграммы.
	Активационная планка – это коробка, расположенная на страховочной линии. Используется для указания на то, что объект активен (или инстанцирован) во время взаимодействия между двумя объектами.
	Синхронное сообщение используется, когда отправитель ждет, пока приемник обработает сообщение и вернется, прежде чем продолжить с другим сообщением.
	Асинхронное сообщение используется, когда вызывающая сторона не ждет, пока приемник обработает сообщение и вернется, прежде чем отправить другие сообщения другим объектам в системе.
	Возвращаемое сообщение используется для указания на то, что приемник сообщения закончил обработку сообщения и возвращает управление вызывающему абоненту.


Диаграмма последовательностей, разрабатываемая для проектирования пользовательского интерфейса, изображена на рисунке 74.

Диаграмма состояний подробно рассматривалась в п. 1.1.3.5. Но все же приведем краткие сведения о ней.

Диаграмма диаграммы состояний является одной из пяти диаграмм UML, используемых для моделирования динамической природы системы. Они определяют различные состояния объекта в течение его жизни, и эти состояния изменяются событиями. Графические элементы диаграммы состояний показаны в таблице 22.

Таблица 22 – Графические элементы диаграммы состояний и их описание

Графический элемент	Описание
	Состояние (state) - условие или ситуация в ходе жизненного цикла объекта, в течение которого он удовлетворяет логическому условию, выполняет определенную деятельность или ожидает события.
	Начальное состояние (start state) - разновидность псевдосостояния, обозначающее начало выполнения процесса изменения состояний конечного автомата

	или нахождения моделируемого объекта в составном состоянии.
	Конечное состояние (final state) - разновидность псевдосостояния, обозначающее прекращение процесса изменения состояний конечного автомата или нахождения моделируемого объекта в составном состоянии.

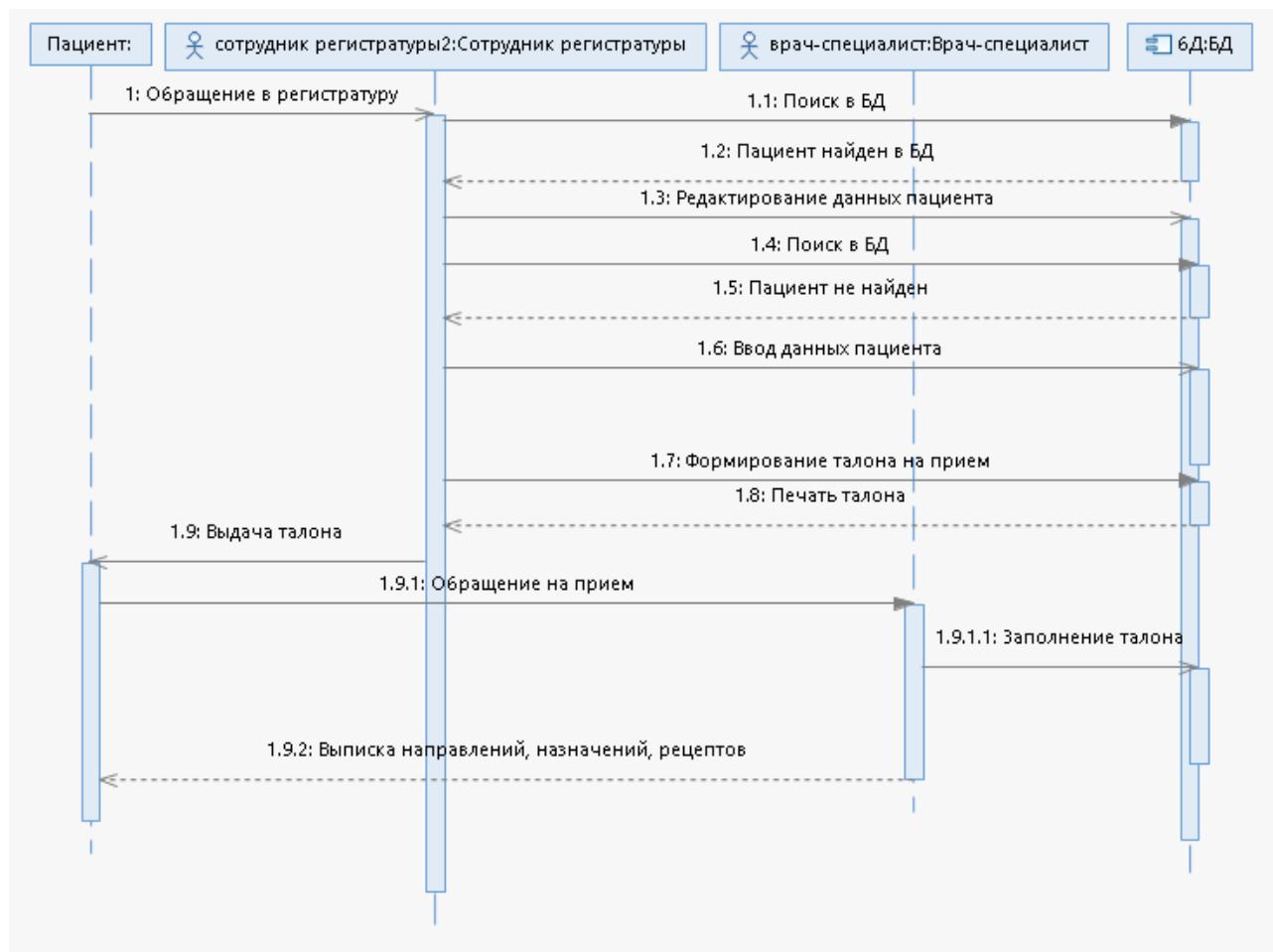


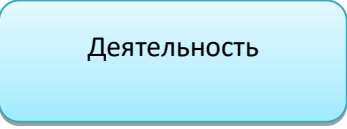
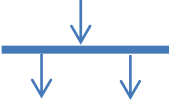
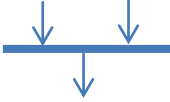



Рис. 74 – Диаграмма последовательностей бизнес-процесса «Оказание амбулаторной медицинской помощи»

Следующая рассматриваемая диаграмма - диаграмма деятельности (активностей), которая позволяет наглядно визуализировать происходящие бизнес-процессы и существующие потоки движения. Можно построить несколько диаграмм деятельности для одной и той же системы, причем каждая из них будет фокусироваться на разных аспектах системы, показывать различные действия, выполняющиеся внутри ее.

Диаграмма деятельности является разновидностью диаграммы состояний, где все или большая часть состояний являются некоторыми деятельностями, а все или большая часть переходов срабатывают при завершении определенной деятельности и позволяют перейти к выполнению следующей. В таблице 23 приведены графические элементы диаграммы деятельности (активностей).

Таблица 23 – Графические элементы диаграммы деятельности и их описание

Графический элемент	Описание
	Начальное состояние
	Конечное состояние
	Состояние деятельности (activity state) - состояние в графе деятельности, которое служит для представления процедурной последовательности действий, требующих определенного времени.
	Разделение (fork)
	Слияние (join)
	Ветвление или решение (decision)

Пример диаграммы деятельности бизнес-процесса «Оказание амбулаторной медицинской помощи» изображен на рисунке 75.

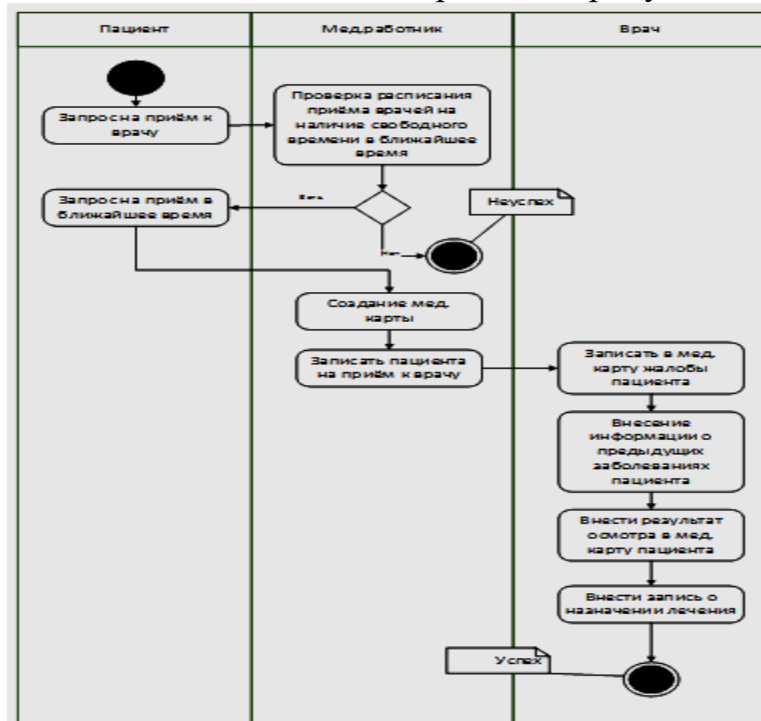


Рисунок 75 – Пример диаграммы деятельности бизнес-процесса «Оказание амбулаторной медицинской помощи»

Для создания диаграммы деятельности откройте проект **ФИОстудентаUML** в окне Project Explorer, кликните правой кнопкой на **My**

Interaction Diagram Model (под узлом Models), и кликните Add Diagram > Activity Diagram. Откроется новая пустая диаграмма деятельности (активностей). На рис. 76 представлена панель инструментов для создания диаграммы.

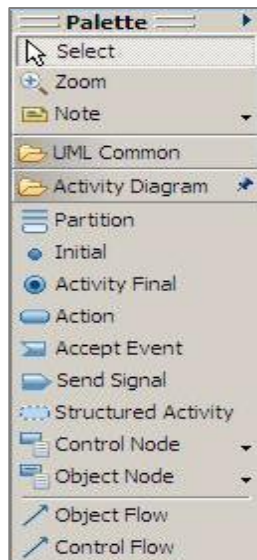


Рис. 76 – Панель инструментов для создания диаграммы деятельности

Добавьте в свою диаграмму разделители (partitions), начальное состояние (initial node), действия (action nodes), узлы решения (decision nodes), соединители (merge nodes), конечное состояние (final nodes), и элементы потоков управления (control flow), используя панель инструментов Palette (рис. 76).

### **САМОСТОЯТЕЛЬНО.**

Разработать диаграммы деятельности не менее 2-х бизнес-процессов деятельности медклиники (выбрать произвольно).

⚠ Показать результаты работы преподавателю.

⚠ Сохранить проект **ФИОстудентаUML**.

### **САМОСТОЯТЕЛЬНО.**

Разработать диаграммы деятельности не менее 2-х бизнес-процессов деятельности по ранее выбранному варианту самостоятельной работы..

⚠ Показать результаты работы преподавателю.

⚠ Сохранить проект **ФИОстудентаUML**.

### **Вопросы для самопроверки:**

1. Что представляет из себя модель UML?
2. На какие виды подразделяются диаграммы UML?
3. Перечислите, какие структурные диаграммы UML вы знаете.
4. Какие диаграммы UML относятся к поведенческому типу?
5. Какие диаграммы UML относятся к диаграммам взаимодействия?

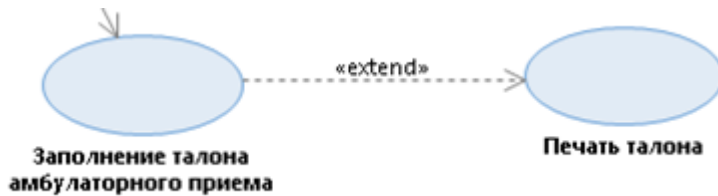
6. Что представляет собой диаграмма вариантов использования?
7. Что представляет собой диаграмма классов?
8. Что представляет собой диаграмма последовательностей?
9. Что представляет собой диаграмма состояния?
10. Что представляет собой диаграмма деятельности?
11. Что представляет собой линия жизни в UML?
12. Какого вида могут быть сообщения в диаграмме последовательностей?

**Выполните тест:**

- 1. Укажите, какие диаграммы UML относятся к структурным:**
  - a) Диаграмма классов
  - b) Диаграмма конечных автоматов
  - c) Диаграмма последовательности
  - d) Диаграмма развертывания
- 2. Укажите, какие диаграммы UML относятся к поведенческим:**
  - a) Диаграмма развертывания
  - b) Диаграмма прецедентов
  - c) Коммуникационная диаграмма
- 3. Диаграммы взаимодействия – это ...**
  - a) Диаграмма составных структур
  - b) Временная диаграмма
  - c) Диаграмма деятельности
- 4. Отношение ... определяет потенциальную возможность включения поведения одного варианта использования в состав другого.**
  - a) Отношение включения
  - b) Отношение обобщения
  - c) Отношение расширения
- 5. В какой диаграмме используется этот графический элемент?**



- a) Диаграмма вариантов использования
  - b) Диаграмма последовательностей
  - c) Диаграмма состояний
  - d) Диаграмма деятельности
- 6. Фрагмент, какой диаграммы представлен на рисунке ниже?**



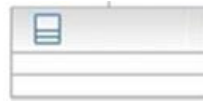
- a) Диаграмма вариантов использования
- b) Диаграмма состояний
- c) Диаграмма последовательностей

**7. Основной структурной диаграммой в UML является:**

- a) Диаграмма составных структур
- b) Диаграмма пакетов
- c) Диаграмма классов

**8. Выберите правильное утверждение:**

- a) На диаграммах классов показываются классы, интерфейсы и динамические отношения между ними.
- b) Диаграмма классов не отображает динамическое поведение объектов, изображенных на ней классов.
- c) Диаграмма классов не отображает динамическое поведение объектов, изображенных на ней классов, а только поведение интерфейсов.



**9. Что содержится в разделах класса?**

- a) В 1-м разделе – имя класса, во 2-м – отношения, в 3-м – атрибуты
- b) В 1-м разделе – операции класса, во 2-м – имя класса, в 3-м – атрибуты
- c) В 1-м разделе – имя класса, во 2-м – атрибуты, в 3-м – операции

**10. Отношение в диаграмме классов - семантическая связь между классами, когда один из них (поставщик) определяет соглашение, которого второй (клиент) обязан придерживаться – это ...**

- a) Отношение реализации
- b) Отношение зависимости
- c) Отношение обобщения

**11. Выберите правильный вариант примера класса «Врач»**

a)

Врач
+Лечить() +Заполнять талон () +Выписывать рецепт() +Выписывать направление()
+ФИО: +Кабинет_приема: +Специализация:

Врач
+ФИО() +Кабинет_приема()

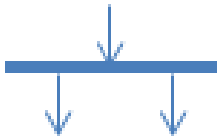
b)

+Специализация()
+Лечить
+Заполнять талон
+Выписывать рецепт
+Выписывать направление

c)

Врач
+ФИО:
+Кабинет приема:
+Специализация:
+Лечить()
+Заполнять талон ()
+Выписывать рецепт()
+Выписывать направление()

12. Выберите описание, которое подходит к этому графическому элементу



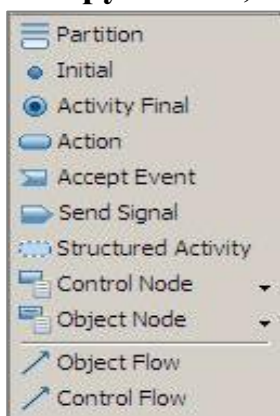
- a) Разделение
- b) Ветвление
- c) Активационная планка

13. В нотацию, какой диаграммы UML входит этот графический элемент



- a) Диаграммы вариантов использования
- b) Диаграммы деятельности
- c) Диаграммы состояний

14. Какую диаграмму UML можно создать с помощью панели инструментов, представленной ниже:



- a) Диаграмму вариантов использования
- b) Диаграмму последовательностей
- c) Диаграмму деятельности

**15. Выберите правильное утверждение:**

- a) Диаграмма деятельности является разновидностью диаграммы состояний.
- b) Диаграмма состояний является разновидностью диаграммы деятельности.
- c) Диаграмма последовательностей является разновидностью диаграммы деятельности.

**Таблица 24 - Правильные ответы по тесту:**

Номер вопроса	Номер ответа
1	a, d
2	b, c
3	b
4	c
5	c, d
6	a
7	c
8	b
9	c
10	a
11	c
12	a
13	b
14	c
15	a