*Занятие № 14*

*Номер учебной группы:* П-16

*Фамилия, инициалы учащегося:* Язубец Е.В.

*Дата выполнения работы:* 23.11.2022

*Тема работы:* «Разработка диаграмм деятельности и состояний»

*Результат выполнения работы*

**Задание 1**

Изучила теоретический материал по теме «Построение диаграмм деятельности в Rational Rose».

**Задание 2**

Изучила теоретический материал по теме «Построение диаграмм состояния в Rational Rose».

**Задание 3**

Построила диаграммы деятельности и состояния для своего варианта задания в Rational Rose.

**Диаграммы деятельности(рис-1)**

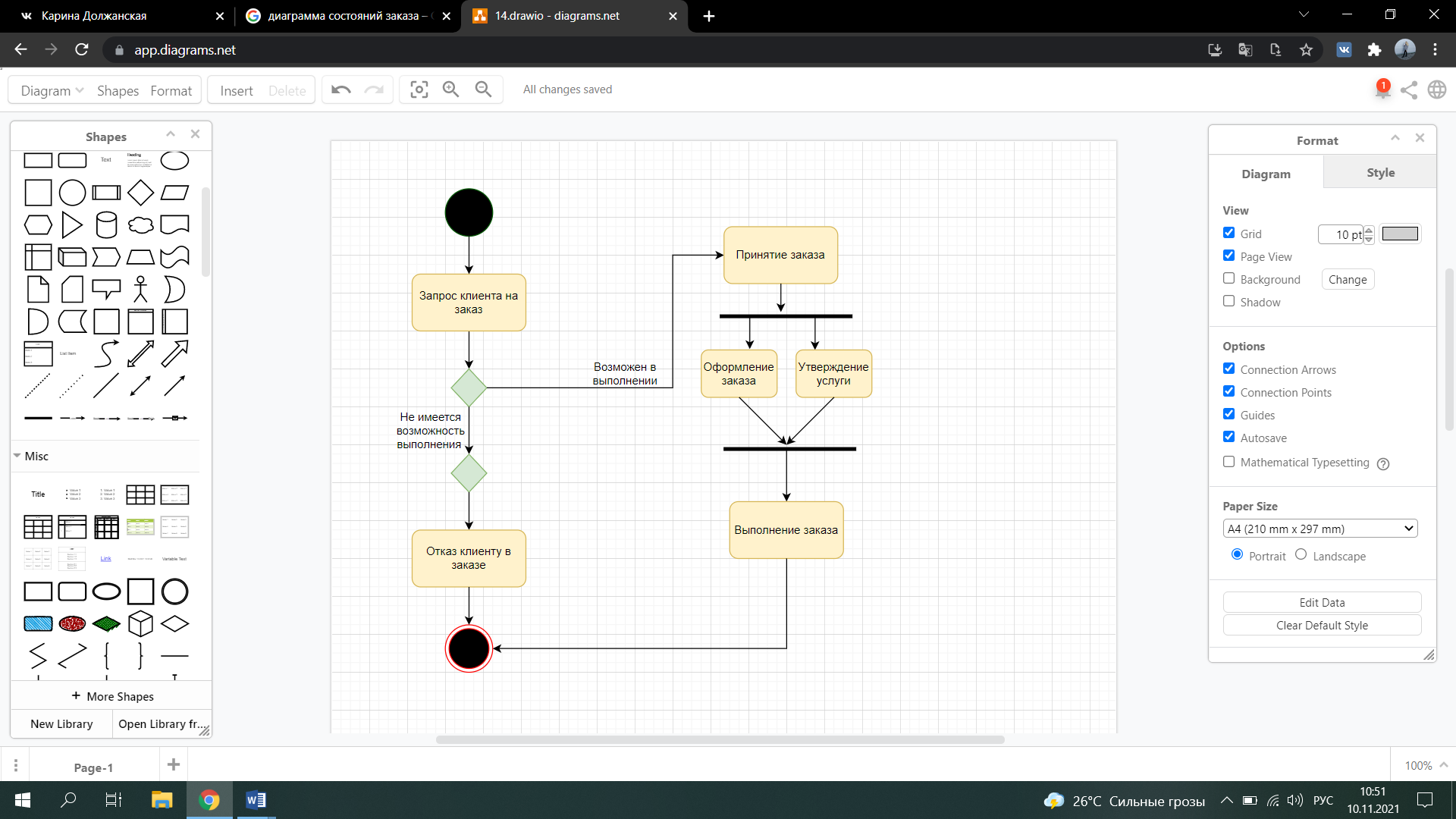


Рисунок-1

На рисунке-1 изображена диаграмма деятельности, выполненная в Rational Rose. В диаграмме отображена работа программного продукта.

В ней выполняется заказ путём определения возможно ли выполнить его или же нет.

Если не имеет возможности выполнения, то на этом работа программы прекращается.

Если имеет возможность выполнения, значит заказ принимается сотрудником. После этого происходит оформление заказа, а также утверждение услуги. И в итоге заказ выполняется, и программа на этом заканчивается.

**Диаграмма состояний(рис-2)**

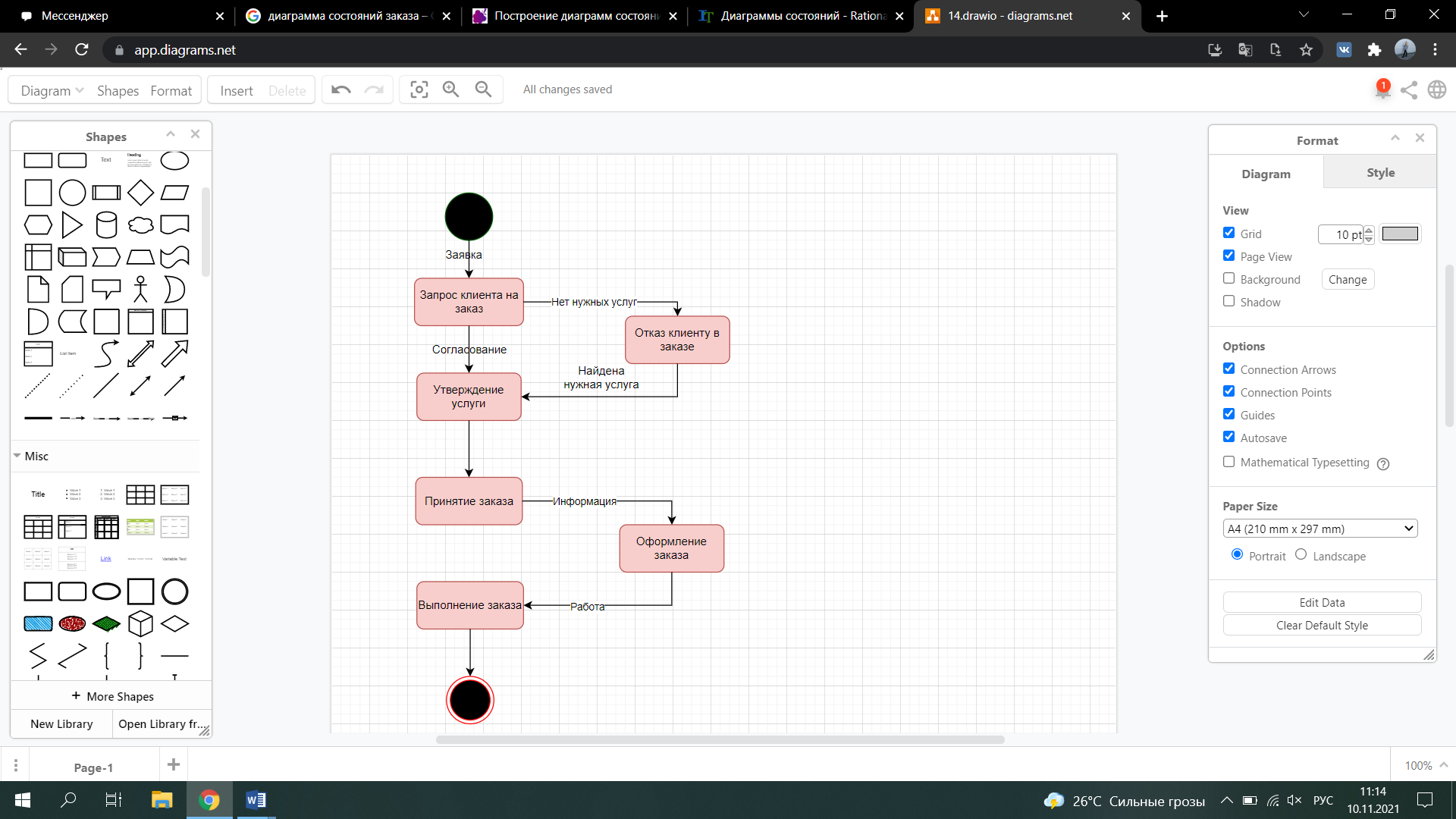


Рисунок-2

На рисунке-2 изображена диаграмма состояний, выполненная в Rational Rose. В диаграмме отображена работа программного продукта.

В ней выполняется сначала заявка на запрос клиента на заказ. Далее если нет нужных услуг, идёт отказ клиенту в заказе.

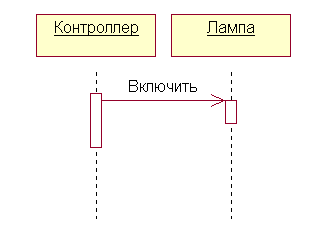
Сотрудник находит нужную услугу и происходит утверждение услуги.

Если нужная услуга есть, то она сразу согласовывается.

После чего происходит принятие заказа, далее благодаря информации оформляется заказ. Сотрудник делает работу и происходит выполнение заказа. На этом программа заканчивает свою работу.

1. Диаграммы действий в программе Rational Rose

**Ответ:** Диаграммы последовательностей действий (Sequence diagram)



Взаимодействие объектов в системе происходит посредством приема и передачи сообщений объектами-клиентами и обработки этих сообщений объектами-серверами. При этом в разных ситуациях одни и те же объекты могут выступать и в качестве клиентов, и в качестве серверов.

Данный тип диаграмм позволяет отразить последовательность передачи сообщений между объектами.

Этот тип диаграммы не акцентирует внимание на конкретном взаимодействии, главный акцент уделяется последовательности приема/передачи сообщений. Для того чтобы окинуть взглядом все взаимосвязи объектов, служит Collaboration diagram.

2. Действия и переходы в программе прецедентов в программе

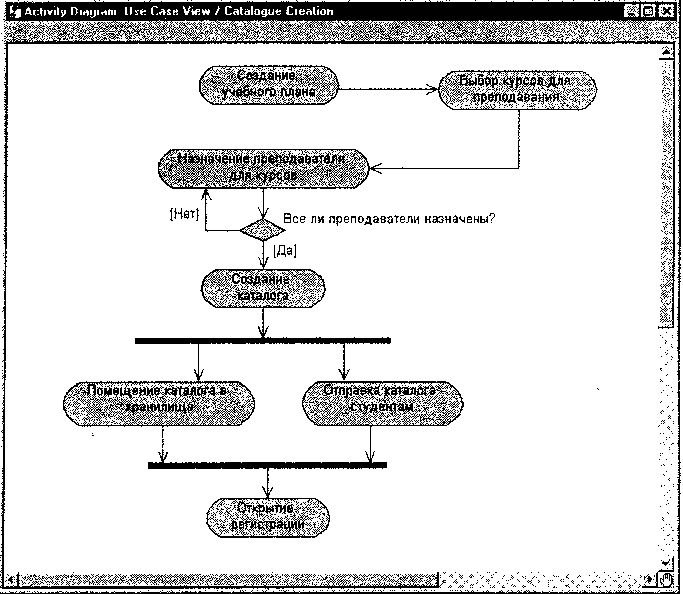
**Ответ:**

3. Элементы выбора, условные переходы, прямолинейные линии переходов

**Ответ:**

4. Линии синхронизации

**Ответ:** В потоке обычно существуют действия, выполняемые параллельно. *Линия синхронизации* (synchronization bar) позволяет указать на необходимость их одновременного выполнения, а также обеспечивает единое выполнение действий в потоке (то есть указывает на необходимость завершения определенных действий для перехода к следующему) — см. рис. 3.18. Таким образом, линии перехода могут иметь несколько входящих линий переходов и одну исходящую либо одну входящую и несколько исходящих.

*Рис. 3.18. Линии синхронизации*

5. Секции в программе

**Ответ:** *Секции* (swimlanes) делят диаграммы действий на несколько участков. Это нужно для того, чтобы показать, кто отвечает за выполнение действий на каждом участке.

Алгоритм создания секций в программе Rational Rose:

1. Щелкните по кнопке Swimlane (Секция) на панели инструментов.

2. Щелкните по диаграмме действий, чтобы создать на ней новую секцию с названием New Swimlane.

3. Дважды щелкните по названию новой секции, чтобы открыть диалоговое окно Specification (Параметры).

4. Введите нужное название секции в поле ввода Name (Название).

5. Щелкните по кнопке ОК, чтобы закрыть диалоговое окно.

6. Для изменения размеров секции переместите ее границу с помощью мыши.

7. Переместите все необходимые действия и переходы на диаграмме в новую секцию, где сразу сможете их создавать

6. Диаграммы состояний в программе Rational Rose

**Ответ:** Диаграммы состояний (State Maсhine diagram)

Каждый объект системы, обладающий определенным поведением, может находится в определенных состояниях, переходить из состояния в состояние, совершая определенные действия в процессе реализации сценария поведения объекта. Поведение большинства объектов реальных систем можно представить с точки зрения теории конечных автоматов, то есть поведение объекта отражается в его состояниях, и данный тип диаграмм позволяет отразить это графически. Для этого используется два вида диаграмм: Statechart diagram (дмаграмма состояний) и Activity diagram (диаграмма активности).

Диаграмма состояний (Statechart) предназначена для отображения состояний объектов системы, имеющих сложную модель поведения. Это одна из двух диаграмм State Machine, доступ к которой осуществляется из одного пункта меню.

7. Принцип построения диаграмм состояния в Rational Rose.

**Ответ:** Разработка диаграммы состояний в среде Rational Rose

Переходя к рассмотрению диаграммы состояний, следует отметить, что в среде Rational Rose этот тип диаграмм относится только к отдельному классу. Для того чтобы построить диаграмму состояний для класса, его вначале необходимо создать и специфицировать. После этого выделить на диаграмме классов или в браузере. Начать построение диаграммы состояний для выбранного класса можно одним из следующих способов:

• Раскрыть логическое представление в браузере (Logical View), выделить рассматриваемый класс и выбрать пункт контекстного меню Open State Diagram (Открыть диаграмму состояний), раскрывающегося по щелчку правой кнопкой мыши.

• Через пункт меню Browse-»State Diagram (Браузер-»Диаграмма состояний).

После выполнения указанных действий в окне диаграммы появится чистое изображение для размещения элементов этой диаграммы, выбираемых с помощью специальной панели инструментов (рис. 12.12).

https://storage.yandexcloud.net/wr4img/69851_136_pict_146.jpeg

Рис. 12.12. Внешний вид специальной панели инструментов для диаграммы состояний