**Занятие № 14**

**Номер учебной группы:** П-16.

**Фамилия, инициалы учащегося:** Пачко Н.Н.

**Дата выполнения работы:** 23.11.2022.

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

**Ход работы**

**Задание 1**

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

Диаграммы действий иллюстрируют действия, переходы между ними, элементы выбора и линии синхронизации. В языке UML действие изображается в виде прямоугольника с закругленными углами, переходы - в виде направленных стрелок, элементы выбора - в виде ромбов, линии синхронизации - в виде толстых горизонтальных или вертикальных линий.

Диаграммы действий в программе Rational Rose создаются следующим образом:

1. Щелкните правой кнопкой мыши по разделу Use Case View (Представле-ние прецедентов) в списке браузера

2. В появившемся контекстно-зависимом меню выберите команду New => Activity Diagram (Создать => Диаграмма действий). В список будет добавлена новая диаграмма с именем New Diagram.

3. Введите название диаграммы.

4. Чтобы открыть диаграмму, дважды щелкните по ней мышью в браузере.

**Задание 2**

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

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.

После добавления состояния или перехода на диаграмму состояний можно открыть спецификацию выбранных элементов и определить их специальные свойства, доступные на соответствующих вкладках. При необходимости можно визуализировать вложенность состояний и подключить историю отдельных состояний.

**Задание 3**

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

Диаграмма деятельности

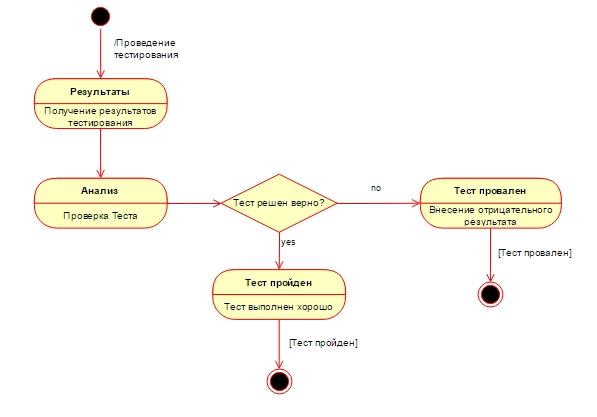
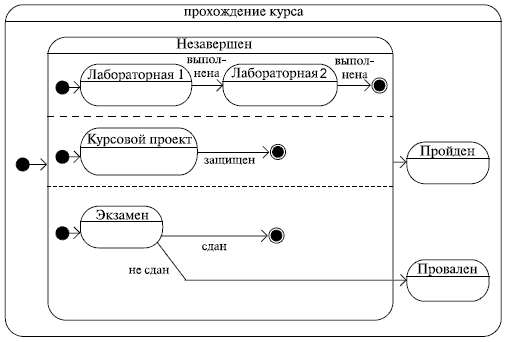


Диаграмма состояния



**Задание 4**

Ответил на контрольные вопросы

**1. Диаграммы действий в программе Rational Rose**

Ответ: Диаграммы действий отражают динамику проекта и представляют собой схемы потоков управления в системе от действия к действию, а также параллельные действия и альтернативные потоки.

В конкретной точке жизненного цикла диаграммы действий могут представлять потоки между функциями или внутри отдельной функции. На разных этапах жизненного цикла они создаются для отражения последовательности выполнения операции.

**2. Действия и переходы в программе прецедентов в программе**

Ответ: Действием называется исполнение определенного поведения в потоке управления системы (см. рис. 3.13).

Для создания действий в программе Rational Rose:

1. Щелкните по кнопке Activity (Действие) на панели инструментов.

2. Щелкните по диаграмме действий, чтобы поместить элемент, изображающий действие, на диаграмму.

3. Введите имя нового действия.

Переходы используются для изображения пути потока управления от действия к действию. Они обычно осуществляются по завершении очередного действия.

Чтобы получить переходы в программе Rational Rose:

1. Щелкните по кнопке State Transition (Переход) на панели инструментов.

2. Щелкните по начальному действию на диаграмме и переместите стрелку перехода на последующее действие.

**3. Элементы выбора, условные переходы, прямолинейные линии переходов**

Ответ: Для создания элементов выбора в программе Rational Rose выполните следующие действия:

1. Щелкните по кнопке Decision (Элемент выбора) на панели инструментов.

2. Щелкните по диаграмме действий, чтобы поместить на нее элемент выбора.

3. Введите имя нового элемента.

4. Щелкните по кнопке State Transition на панели инструментов.

5. Щелкните по начальному действию на диаграмме и переместите стрелку перехода на элемент выбора.

Последовательность создания условных переходов в программе Rational Rose:

1. Щелкните по кнопке State Transition на панели инструментов.

2. Щелкните по элементу выбора на диаграмме и переместите стрелку перехода на последующее действие.

3. Дважды щелкните по стрелке перехода, чтобы открыть диалоговое окно Specification (Параметры).

4. Щелкните по вкладке Detail (Подробно).

5. В поле ввода Guard Condition (Условие) введите условие перехода.

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

Чтобы получить прямолинейные линии переходов в программе Rational Rose:

1. Выберите линии переходов, которые вы хотите сделать прямолинейными (для выбора нескольких линий можно использовать клавишу Shift).

2. Выберите команду меню Format => Style => Rectilinear (Формат => Стиль => Прямолинейный).

3. Расположите линии нужным образом на диаграмме действий, перетаскивая их с помощью мыши.

**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.

Для создания линий синхронизации в программе Rational Rose:

1. Щелкните по кнопке Horizontal Synchronization (Горизонтальная линия синхронизации) или Vertical Synchronization (Вертикальная линия синхронизации) на панели инструментов.

2. Щелкните по диаграмме действий, чтобы поместить на нее линию синхронизации.

3. Щелкните по кнопке State Transition (Переход) на панели инструментов и добавьте необходимые входящие и исходящие линии переходов к 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: Для обозначения начального и конечного состояний в потоке управления системы используются специальные символы. Начальное состояние изображается в виде закрашенного круга, а конечное — в виде закрашенного круга, обведенного дополнительной окружностью. Обычно в потоке существуют одно начальное и несколько конечных состояний — для каждого альтернативного направления.

Последовательность создания начального и конечного состояний в программе Rational Rose:

1. Щелкните по кнопке Start State (Начальное состояние) или End State (Конечное состояние) на панели инструментов.

2. Щелкните по диаграмме действий, чтобы поместить на нее символ конечного или начального состояния.

3. Если вы добавили начальное состояние, щелкните по кнопке State Transition (Переход) на панели инструментов, а затем на символе начального состояния и выполните переход к первому действию в потоке.

4. Если вы добавили конечное состояние, щелкните по кнопке State Transition на панели инструментов, а затем на предшествующем действии и выполните переход к символу конечного состояния на диаграмме.

**7. Принцип построения диаграмм состояния в Rational Rose.**

Ответ: Процесс начинается с начальной точки, затем следует самый первый переход в состояние «Проверка даты отчета». В поведении объекта в системе можно выделить действия, отображаемые переходами, и деятельности, отображаемые состояниями. Хотя и то и другое - это процессы, реализуемые, как правило, некоторым методом класса «Отчет», они трактуются различным образом. Действия связаны с переходами и рассматриваются, как мгновенные

и непрерываемые. Деятельности связаны с состояниями и могут длиться достаточно долго. Деятельность может быть прервана в результате наступления некоторого события.

Переход может содержать метку. Синтаксически метка перехода состоит из трех частей, каждая из которых является необязательной: <Событие> [<Условие>]/<Действие>. Если метка перехода не содержит никакого события, это означает, что переход происходит, как только завершается какая-либо деятельность, связанная с данным состоянием. Из состояния «Проверка даты отчета» возможны два перехода. Метка одного из них включает условие. Условие - это логическое условие, которое может принимать два значения: «истина» или «ложь». Условный переход выполняется только в том случае, если условие принимает значение «истина», в противном случае выполняется переход, не помеченный условием.

Из конкретного состояния в данный момент времени может быть осуществлен только один переход; таким образом, условия являются взаимно исключающими для любого события. Существует два особых состояния: вход и выход. Любое действие, связанное с событием входа, выполняется, когда объект входит в данное состояние. Событие выхода выполняется в том случае, когда объект выходит из данного состояния. Диаграммы состояний хорошо использовать для описания поведения некоторого объекта в нескольких различных вариантах использования. Они не слишком пригодны для описания поведения ряда взаимодействующих объектов.

Рекомендуется строить диаграммы состояний только для тех классов, поведение которых влияет на общее поведение системы, например для классов пользовательского интерфейса и управляющих объектов.