Лабораторная работа №1 Разработка диаграммы вариантов использования

**1 Цель работы**

Изучить инструментальные средства разработки и разработать диаграммы вариантов использования.

**2 Задание**

Создайте диаграмму вариантов использования

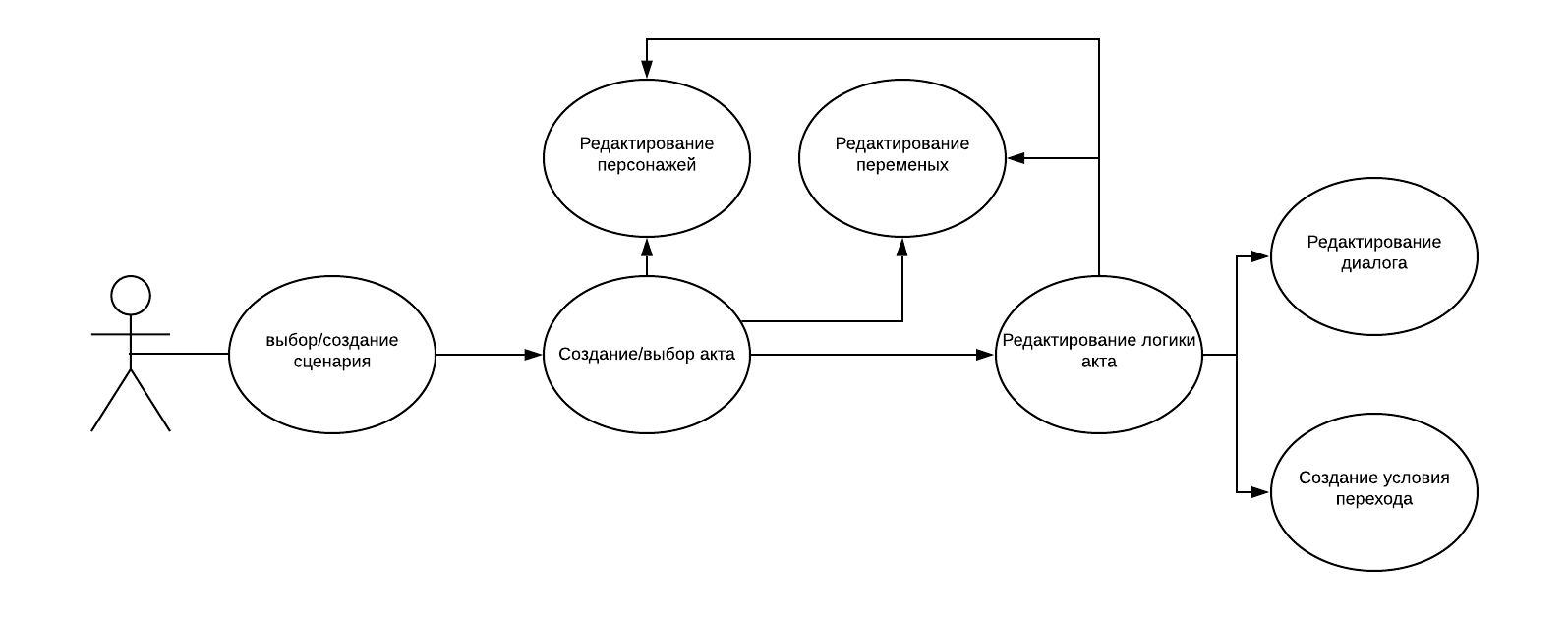
Диаграмма вариантов использования является самым общим представлением функциональных требований к системе. Для последующего проектирования системы требуются более конкретные детали, которые описываются в документе, называемом сценарием варианта использования или потоком событий (flow of events).

Сценарий подробно документирует процесс взаимодействия действующего лица с системой, реализуемого в рамках варианта использования. Основной поток событий описывает нормальный ход событий (при отсутствии ошибок). Альтернативные потоки описывают отклонения от нормального хода событий (ошибочные ситуации) и их обработку.

Достоинства модели вариантов использования заключаются в том, что она:

* определяет пользователей и границы системы;
* определяет системный интерфейс;
* удобна для общения пользователей с разработчиками;
* используется для написания тестов;
* является основой для написания пользовательской документации;
* хорошо вписывается в любые методы проектирования (как объектно-ориентированные, так и структурные).

Каждый вариант использования определяет последовательность действий, которые должны быть выполнены проектируемой системой при взаимодействии ее с соответствующим актером.





Лабораторная работа №2 Разработка диаграмм взаимодействия

**1 Цель работы**

Изучить инструментальные средства разработки и разработать диаграммы взаимодействия.

**2 Задание**

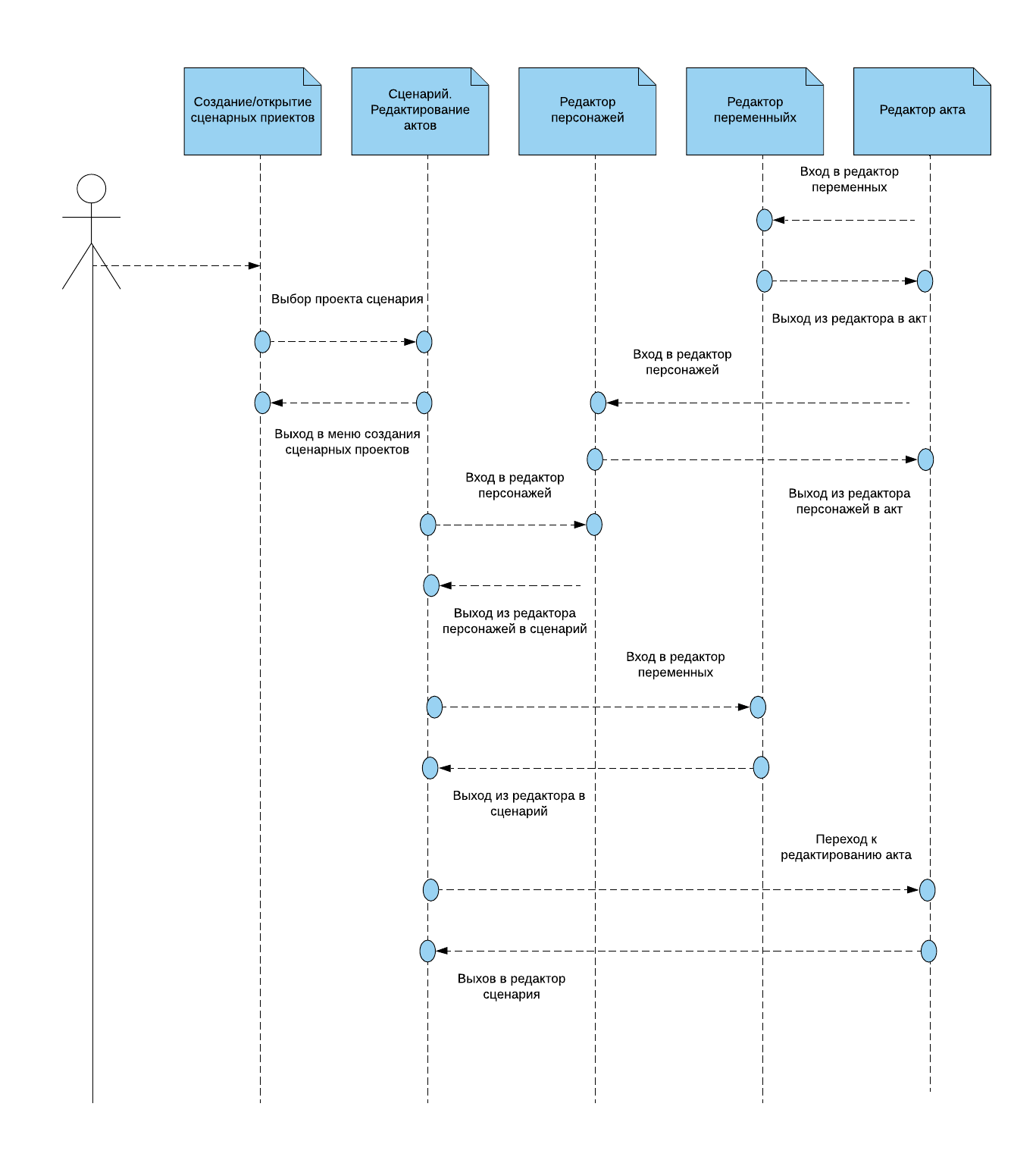
Создайте диаграмму взаимодействия

Реализация отдельного варианта использования требует участия и взаимодействия определенных экземпляров актеров и классов. Наиболее подходящий инструмент для описания такого взаимодействия – это диаграммы последовательности и коммуникации, которые, по сути, отображают одну и ту же информацию. В связи с этим большинство Case-средств позволяет после построения одной из диаграмм автоматически получить другую, а также выполнять синхронизацию этих диаграмм между собой.

Общими элементами диаграмм являются:

* экземпляры актеров и объекты, участвующие во взаимодействии;
* сообщения, передаваемые между экземплярами актеров и объектами.

Экземпляры сущностей отображаются стандартно (экземпляр актера – человечком, экземпляр класса (объект) – прямоугольником или графическим стереотипом класса анализа). В то же время следует помнить, что экземпляр – это конкретная реализация соответствующей сущности (актера, класса, узла и т. д.).



Лабораторная работа № 3 Разработка диаграмм классов

**1 Цель работы**

Изучить инструментальные средства разработки и разработать диаграммы классов.

**2 Задание**

Создайте диаграмму классов

Диаграмма классов определяет типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами. Вид и интерпретация диаграммы классов существенно зависит от точки зрения (уровня абстракции): классы могут представлять сущности предметной области (в процессе анализа) или элементы программной системы (в процессах проектирования и реализации).

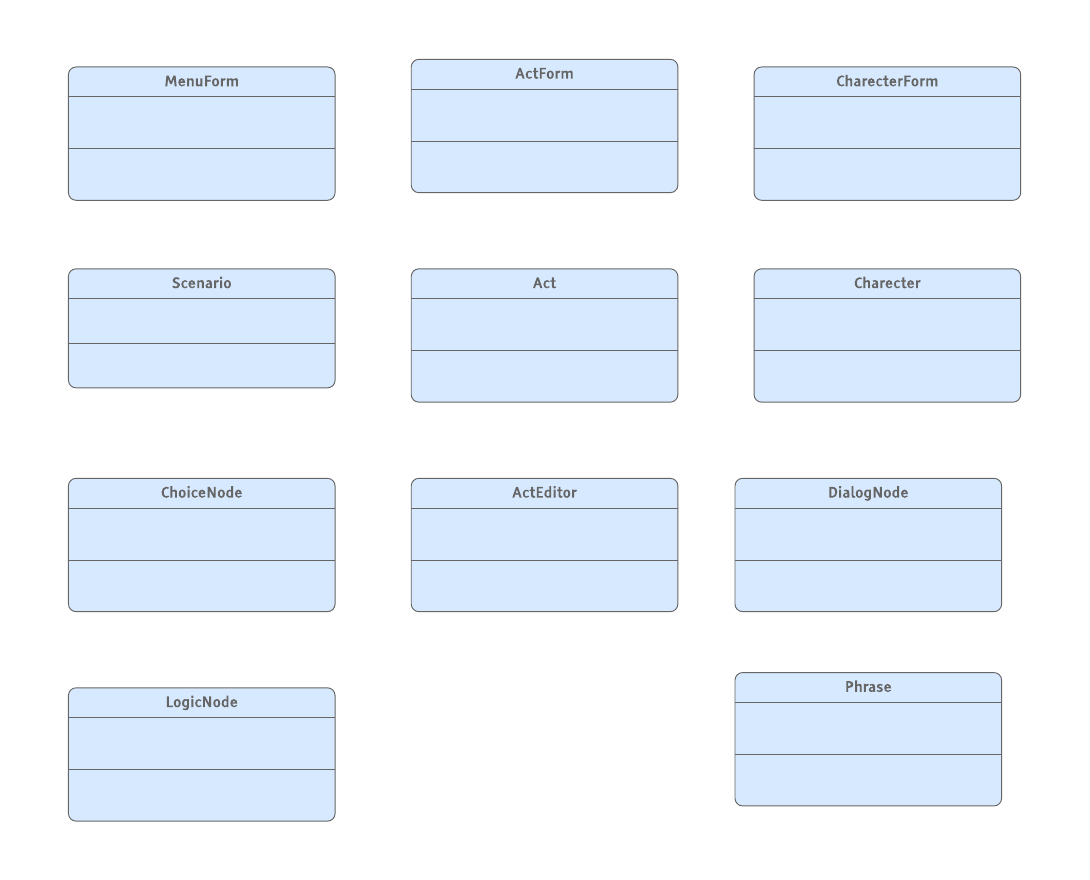
Основными элементами являются классы и связи между ними. Классы характеризуются при помощи атрибутов и операций.

Атрибуты описывают свойства объектов класса. Большинство объектов в классе получают свою индивидуальность из-за различий в их атрибутах и взаимосвязи с другими объектами. Однако, возможны объекты с идентичными значениями атрибутов и взаимосвязей. Т.е. индивидуальность объектов определяется самим фактом их существования, а не различиями в их свойствах. Имя атрибута должно быть уникально в пределах класса. За именем атрибута может следовать его тип и значение по умолчанию.

Операция есть функция или преобразование. Операция может иметь параметры и возвращать значения.

Виды связей:

* ассоциация
* агрегация
* наследование.



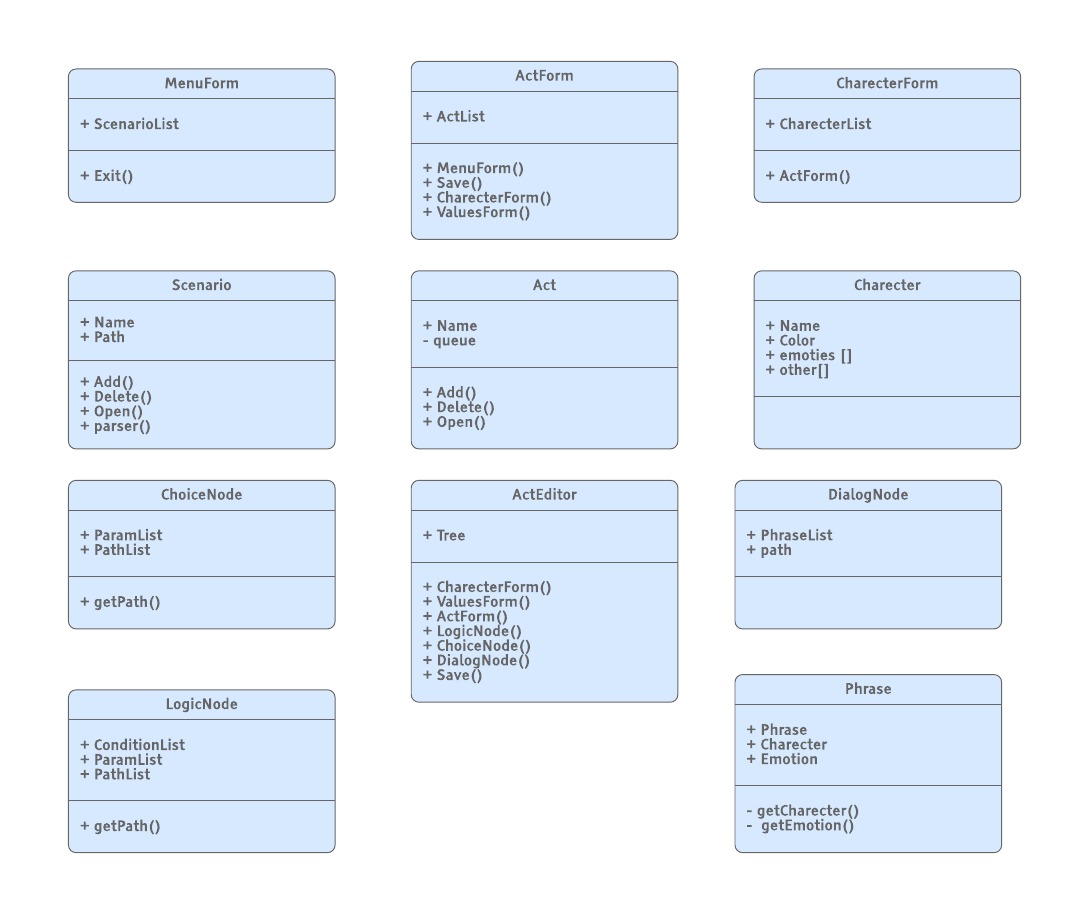
Лабораторная работа №4 Разработка диаграмм классов (добавление атрибутов)

**1 Цель работы**

Изучить инструментальные средства разработки и разработать диаграммы классов.

**2 Задание**

Дополнить диаграмму атрибутами



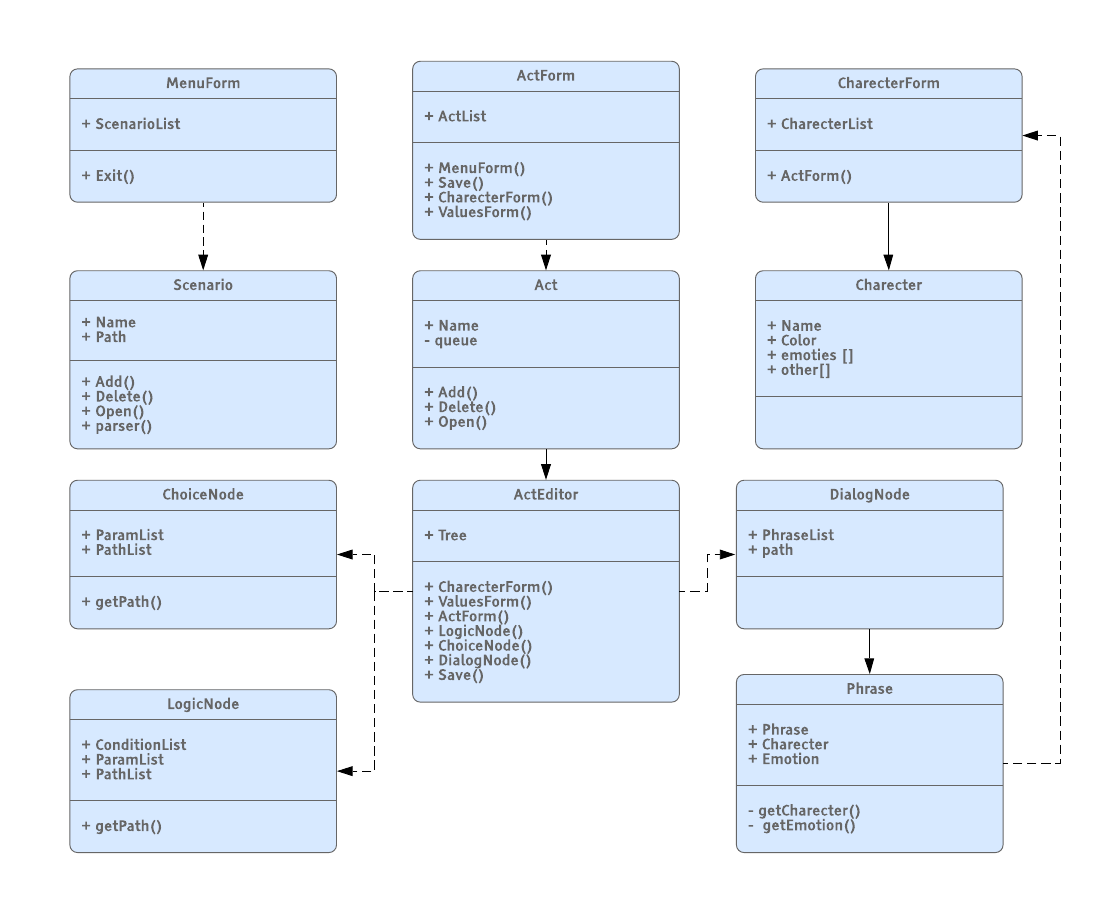
Лабораторная работа №5 Разработка диаграмм классов (добавление связей между классами)

**1 Цель работы**

Изучить инструментальные средства разработки и разработать диаграммы классов.

**2 Задание**

Дополнить диаграмму связями между классами



Лабораторная работа №6 Разработка диаграммы состояний

**1 Цель работы**

Изучить инструментальные средства разработки и разработать диаграммы состояний.

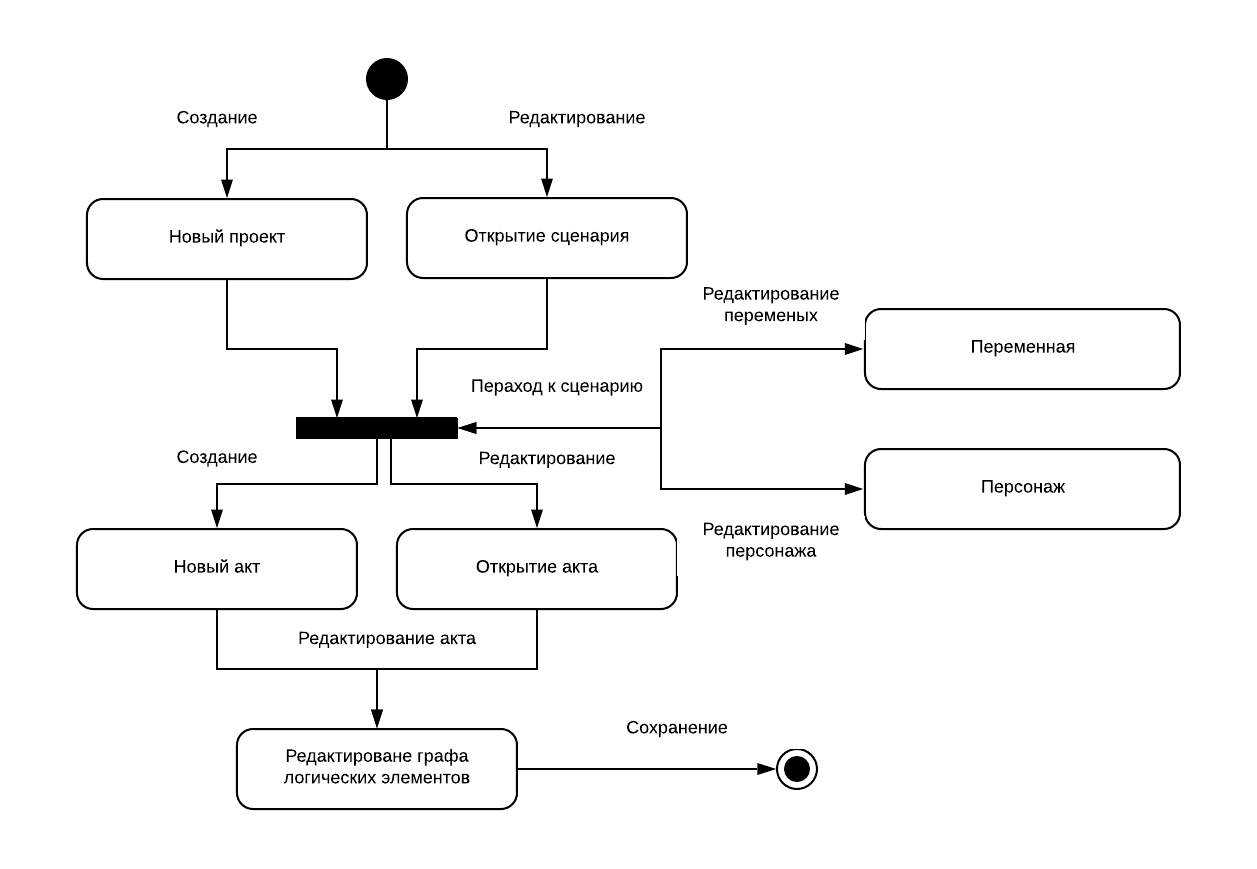
**2 Задание**

Создайте диаграмму состояний.

Диаграмма состояний показывает, как объект переходит из одного состояния в другое. Диаграммы состояний служат для моделирования динамических аспектов системы. Данная диаграмма полезна при моделировании жизненного цикла объекта. От других диаграмм диаграмма состояний отличается тем, что описывает процесс изменения состояний только одного экземпляра определенного класса - одного объекта, причем объекта реактивного, то есть объекта, поведение которого характеризуется его реакцией на внешние события.

Также уже из класса формы сценария порождаются акты, в которых в свою очередь выстраивается граф.

Изменение состояний проекта сценария отражены на рисунке 10.



Лабораторная работа №7 Разработка диаграммы компонентов

**1 Цель работы**

Изучить инструментальные средства разработки и разработать диаграммы компонентов.

**2 Задание**

Создайте диаграмму компонентов.

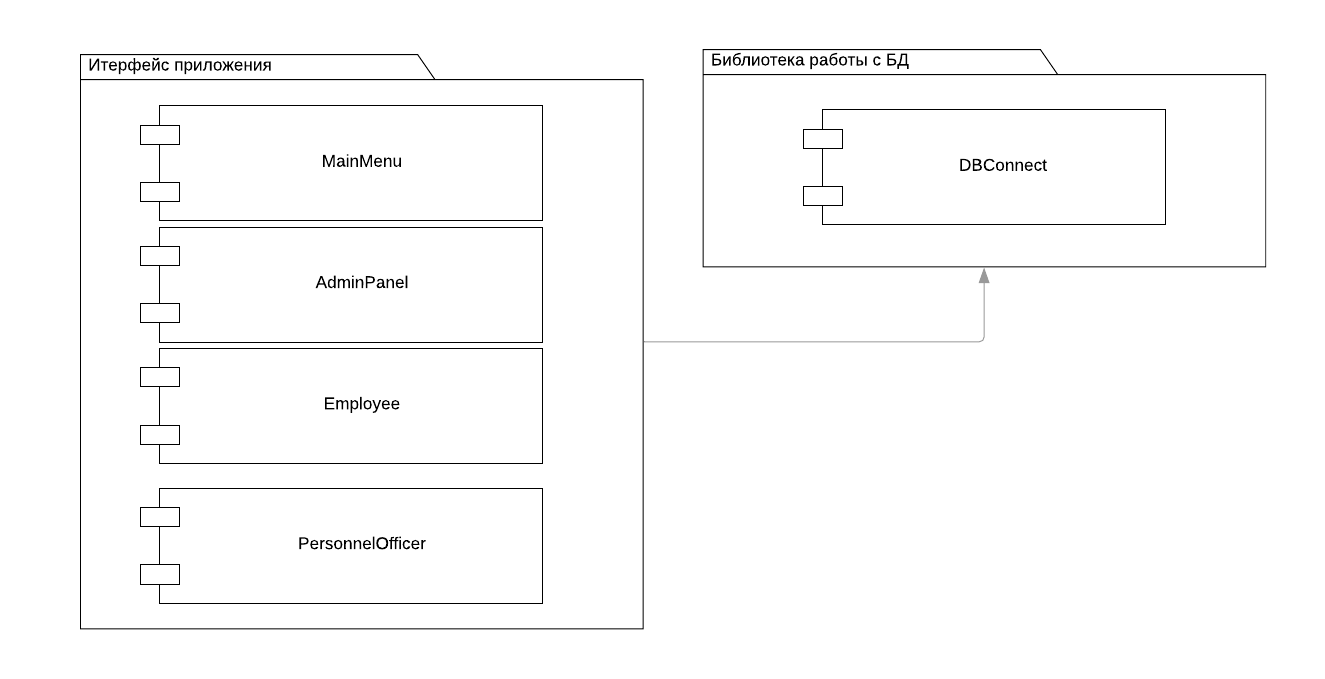
Диаграмма компонентов позволяет определить состав программных компонентов, в роли которых может выступать исходный, бинарный и исполняемый код, а также установить зависимости между ними.

При разработке диаграмм компонентов преследуются цели:

* спецификация общей структуры исходного кода системы;
* спецификация исполнимого варианта системы.

Данная диаграмма обеспечивает согласованный переход от логического к физическому представлению системы в виде программных компонентов. Одни компоненты могут существовать только на этапе компиляции программного кода, другие – на этапе его исполнения. Основными элементами диаграммы являются компоненты, интерфейсы и зависимости между ними. Кроме этого, на ней могут отображаться ключевые классы, входящие в компоненты.

Компонент (англ. component) – это физическая часть системы. Компоненты, представляющие собой файлы с исходным кодом классов, библиотеки, исполняемые модули и т.п., которые должны обладать согласованным набором интерфейсов.



Лабораторная работа №8 Разработка диаграммы размещения

**1 Цель работы**

Изучить инструментальные средства разработки и разработать диаграммы размещения.

**2 Задание**

Создайте диаграмму размещения.

Диаграмма размещения (deployment diagram) отражает физические взаимосвязи между программными и аппаратными компонентами системы. Она является хорошим средством для того, чтобы показать маршруты перемещения объектов и компонентов в распределенной системе.

Каждый узел на диаграмме размещения представляет собой некоторый тип вычислительного устройства, в большинстве случаев – часть аппаратуры. Эта аппаратура может быть простым устройством или датчиком, а может быть и мэйнфреймом.

