

## Лабораторная работа №4

### Составление диаграмм с помощью программы StarUML

Цель работы: Изучение методов и приемов объектно-ориентированного проектирования, моделирования и программирования с помощью программы StarUML

#### *План занятия:*

- Изучение языка UML.
- Изучение методов разработки диаграмм на языке UML.
- Разработка модели информационной системы АСУ ВУЗ на языке UML.
- Составление диаграмм на языке UML с помощью программы StarUML.
- Изучение модели информационной системы АСУ ВУЗ с помощью программы StarUML.

#### *Порядок выполнения работы:*

1. Выбрать вариант индивидуального задания в соответствии с порядковым номером в списке группы, по последней цифре
2. Изучить теоретические сведения о программе StarUML.
3. Разработать модель информационной системы АСУ ВУЗ.
4. Составить диаграммы на языке UML.
5. Создать проект с помощью программы StarUML.
6. Разработать диаграммы с помощью программы StarUML.
7. Сохранить диаграммы в проекте.
8. Оформить отчет о лабораторной работе.

#### *Содержание отчета о лабораторной работе:*

1. Титульный лист (с указанием названия кафедры, названия дисциплины, ФИО и номера группы студента, ФИО преподавателя)
2. Содержание с номерами стр.
3. Название лабораторной работы
4. Вариант индивидуального задания
5. Скриншоты диаграмм, подготовленных с помощью программы StarUML
6. Описание диаграмм
7. Выводы
8. Ответы на контрольные вопросы
9. Список литературы

### Описание подсистем АСУ ВУЗ

Вариант задания	Название	Описание
1	Подсистема «Ректорат»	Модель работы ректора ВУЗа, проректора по учебной работе, проректора по научной работе, проректора по АХЧ, секретаря ректора, секретаря проректора
2	Подсистема «Ученый Совет ВУЗа»	Модель работы Ученого Совета ВУЗа, председателя (ректора), ученого секретаря, членов Ученого Совета ВУЗа
3	Подсистема «Деканат»	Модель работы декана, зам. декана, секретаря
4	Подсистема «Кафедра АСУ»	Модель работы зав. кафедрой, секретаря, преподавателей (профессоров, доцентов, старших преподавателей, преподавателей, ассистентов), Зав. учебной лабораторией, инженера, лаборанта, техника
5	Подсистема «Кафедра физвоспитания»	Модель работы зав. кафедрой, секретаря, преподавателей (профессоров, доцентов, старших преподавателей, преподавателей, ассистентов), инженера, лаборанта, техника
6	Подсистема «Дворец культуры»	Модель работы зав. Дворцом культуры, секретаря, руководителя кружка
7	Подсистема «Профком студентов»	Модель работы председателя профкома, бухгалтера, секретаря, членов профкома, профторгов студенческих групп, студентов
8	Подсистема «Учебная библиотека»	Модель работы зав. библиотекой, библиографов, сотрудников библиотеки, читателей (студентов и преподавателей)
9	Подсистема «Отдел кадров»	Модель работы начальника отдела кадров, секретаря, сотрудников отдела кадров, сотрудников, преподавателей и студентов ВУЗа
10	Подсистема «Канцелярия»	Модель работы зав. канцелярией, секретаря, сотрудников канцелярии, сотрудников, преподавателей и студентов ВУЗа

### *Теоретические сведения*

Язык и методика объектно-ориентированного моделирования UML (universal modeling language). UML – универсальный язык моделирования (universal modeling language). UML используется для создания моделей сложной системы.

Цель разработки UML – предоставить в распоряжение пользователей легко воспринимаемый и выразительный язык визуального моделирования,

предназначенный для разработки и документирования моделей сложных систем. В основе UML лежат диаграммы.

### Диаграммы UML

Определены следующие виды диаграмм:

1. Диаграмма вариантов использования, диаграмма прецедентов (usecase diagram),
2. Диаграмма классов (class diagram)
3. Диаграммы поведения (behavior diagrams)
  - а) диаграммы состояний (state chart diagram)
  - б) диаграмма деятельности (activity diagram)
  - в) диаграммы взаимодействия (interaction diagrams), диаграмма последовательности (sequence diagram), диаграмма кооперации (collaboration diagram)
4. Диаграммы реализации (implementation diagram)
  - а) диаграмма компонентов (component diagram)
  - б) диаграмма развертывания (deployment diagram).

### Диаграмма прецедентов

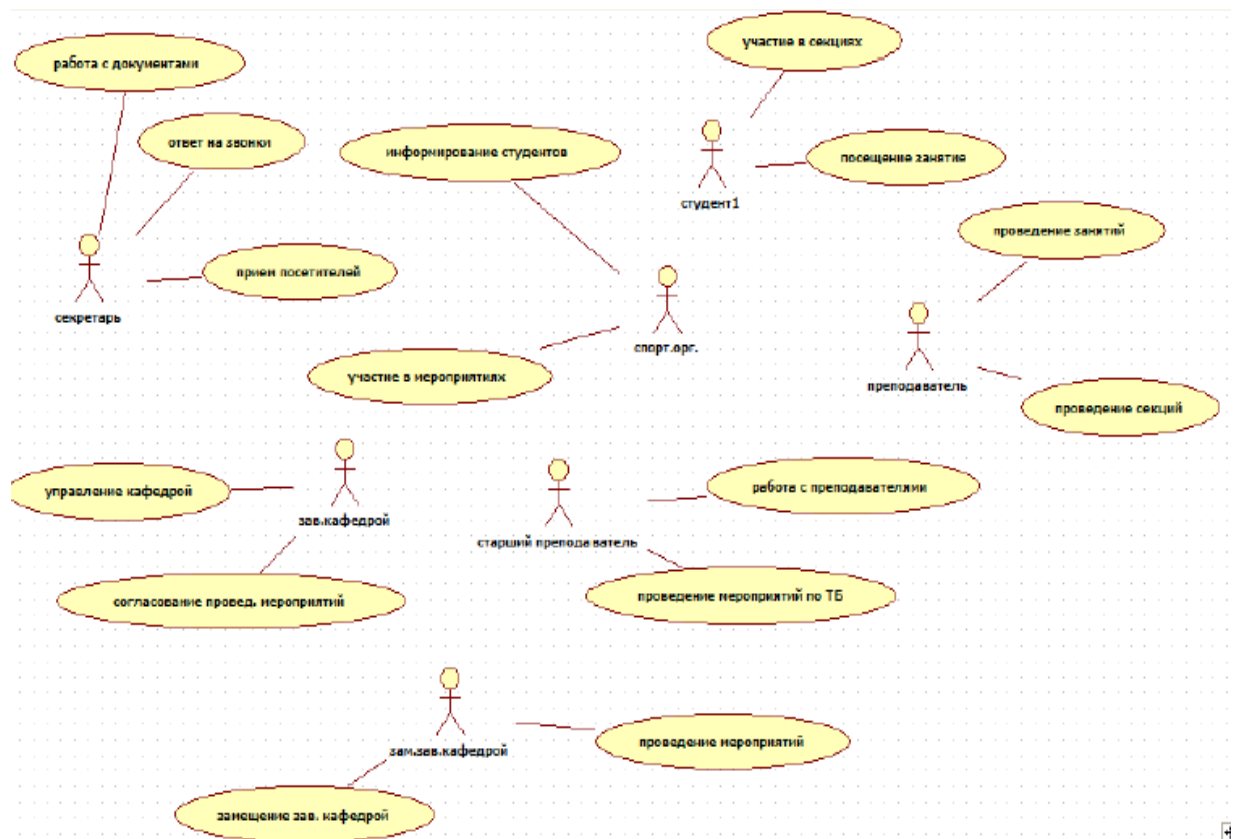


Рис. 1. Диаграмма прецедентов

Диаграмма прецедентов включает актеров (действующих лиц) и действия (прецедент). Каждому актеру ставят в соответствие одно или несколько действий. Пара «актер-действие» образует роль.

## Описание диаграммы прецедентов

Табл.1.

Актер	Краткое описание
Зав. кафедрой	Сотрудник, который занимается организацией работы кафедры. В том числе ее управлением и согласованием мероприятий, проводимых на кафедре
Зам. зав. кафедрой	Сотрудник, который замещает заведующего, проводит мероприятия.
Секретарь	Сотрудник, который отвечает на звонки по телефону, принимает посетителей, работает с документами, выполняет поручения Зав. кафедрой
Старший преподаватель	Сотрудник, который отвечает за технику безопасности и взаимодействие с преподавателями, проводит занятия со преподавателями взаимодействие с преподавателями, проводит занятия со студентами, проводит занятия специализированных секций.
Преподаватель	Сотрудник, который проводит занятия со студентами, проводит занятия специализированных секций.
Спорт.орг.	Сотрудник, который отвечает за взаимодействие со студентами, помощь в проведение мероприятий и своевременное оповещение об их проведении.
Студент	Посещает занятия по общей физической подготовке, посещает спортивные секции, принимает участие в мероприятиях.

Прецедент	Краткое описание
Управление кафедрой	Запускается зав. Кафедрой. Позволяет вносить изменение в работе кафедры.
Согласование проводимых мероприятий	Запускается зав. Кафедрой. Позволяет вносить изменение в проведение мероприятий, которые проводятся на кафедре.
Замещение зав. кафедрой	Запускается зам. Зав. Кафедрой. Позволяет вносить изменение в работу кафедры во время отсутствия зав. кафедрой
Проведение мероприятий	Запускается зам. Зав. Кафедрой. Позволяет составлять отчет о проведенных мероприятиях, составление планов по мероприятиях
Работа с документами	Запускается секретарем. Позволяет вносить изменения во входных-выходных и внутренних документах.

### Цели:

- Определение общих принципов и контекста моделируемой предметной области на начальных этапах проектирования.
- Сформулировать общие требования к функциональному поведению

- проектируемой области.
- Разработать исходную концептуальную модель для ее следующей детализации.
- Подготовка исходной документации для взаимодействия с заказчиком.

*Используются обозначения:*

Действующее лицо (actor, актер) – тот кто со своим запросом обращается к проектируемому программному комплексу.

Действующее лицо имеет 3 разновидности: человек, какое-то техническое устройство, для управления которым создается ПК, другой ПК, внешний к данному.

Варианты использования описывают отношения:

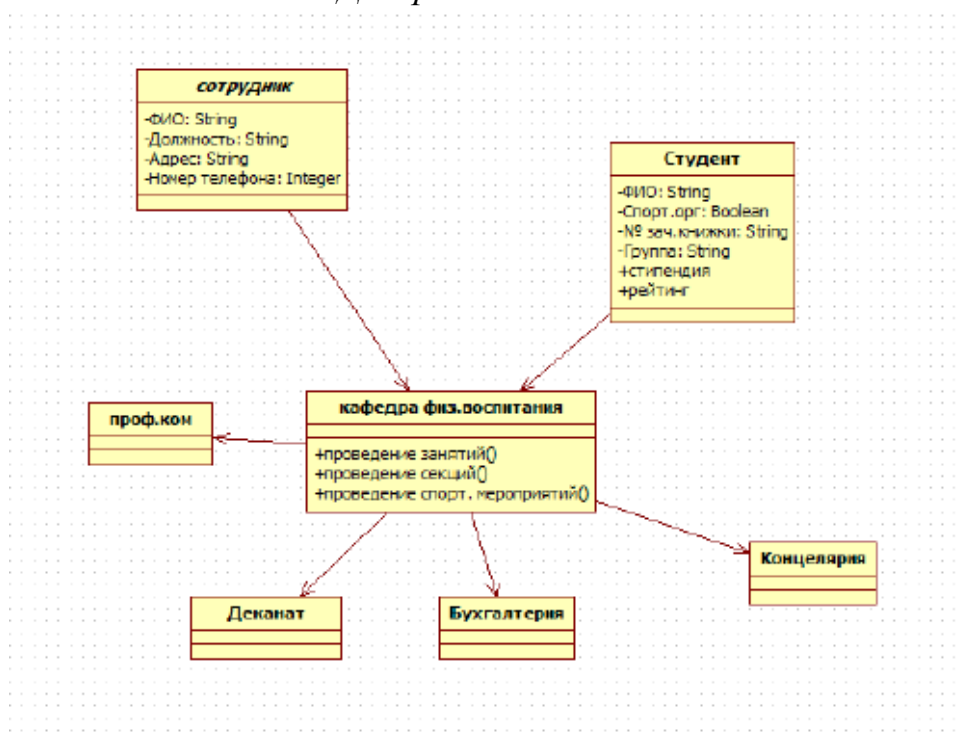
Отношение ассоциации: служит для обозначения специфической роли актера в отдельном варианте использования, т.е. какую конкретную роль играет актер при взаимодействии с экземпляром варианта использования.

Отношение обобщения: служит для указания факта, что некоторый вариант использования может быть обобщен до другого варианта использования.

Отношение расширения: определяет связь одного варианта использования с более общим.

Отношение включения: показывает, что некоторый вариант использования включает в себя подварианты

### *Диаграмма классов*



*Рис. 2. Диаграмма классов*

Класс – это абстрактное описание множества объектов с одинаковыми свойствами. Класс включает атрибуты (свойства, характеристики объектов) и методы (действия над объектами этого класса). Модификатор доступа для атрибутов и методов – public, privat.

Таблица описания класса «кафедра физ. воспитания»

Табл.2

Параметр	Значение
комментарий	Класс, который представляет собой описание работы кафедрой.
операции	Проведение занятий() – занесение информации о проводимом занятии. Проведение секций() – занесение информации о проводимой секции. Проведение спорт. мероприятий() – занесение информации о проводимом мероприятии. Все операции имеют модификатор доступа – public.

Класс «Сотрудник»

Параметр	Значение
Комментарий	Класс, представляющий собой сотрудника кафедры
Атрибуты	-ФИО: string – фамилия, имя отчество сотрудника кафедры. -Должность: string – должность сотрудника. -Адрес: string - домашний адрес сотрудника -номер телефона: string – телефон сотрудника для связи с ним. Все атрибуты имеют модификатор доступа -privat.

Класс «Студент»

Параметр	Значение
Комментарий	Класс, представляющий собой студента
Атрибуты	-ФИО: string – ФИО студента -спорт. орг: boolean – идентификатор спорт.орга -№ зачетной книжки: string – номер -№ зачетной книжки: string – номер зачетной книги студента -группа: string – Наименование группы студента -стипендия – назначенная стипендия -рейтинг – рейтинг студента Все атрибуты имеют модификатор доступа -privat.

### Диаграмма деятельности

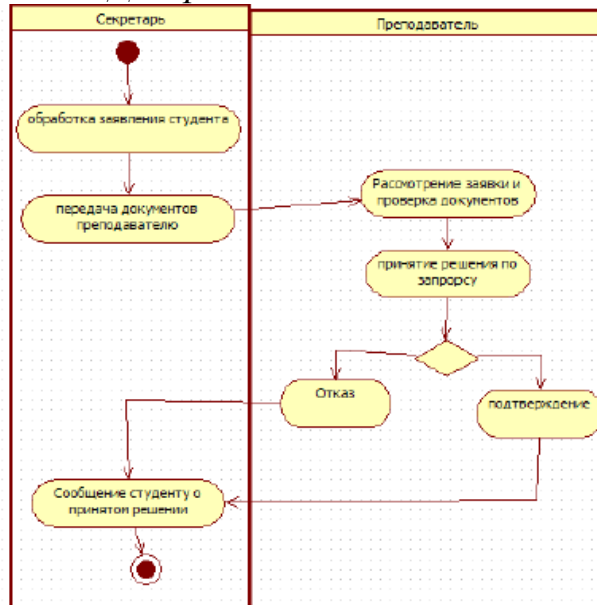


Рис.3. Диаграмма деятельности

Диаграмма действий – специальная форма диаграммы состояний, которая отображает последовательность выполнения действий во времени. Диаграмма действий в общем случае используется для отображения любых последовательных действий для обработки данных, но чаще всего применительно к объектам, классам, пакетам и операциям.

В нашем случае диаграмма действий показывает процесс обработки поданного студентом заявления на зачисление в спортивную секцию.

### Диаграмма взаимодействий

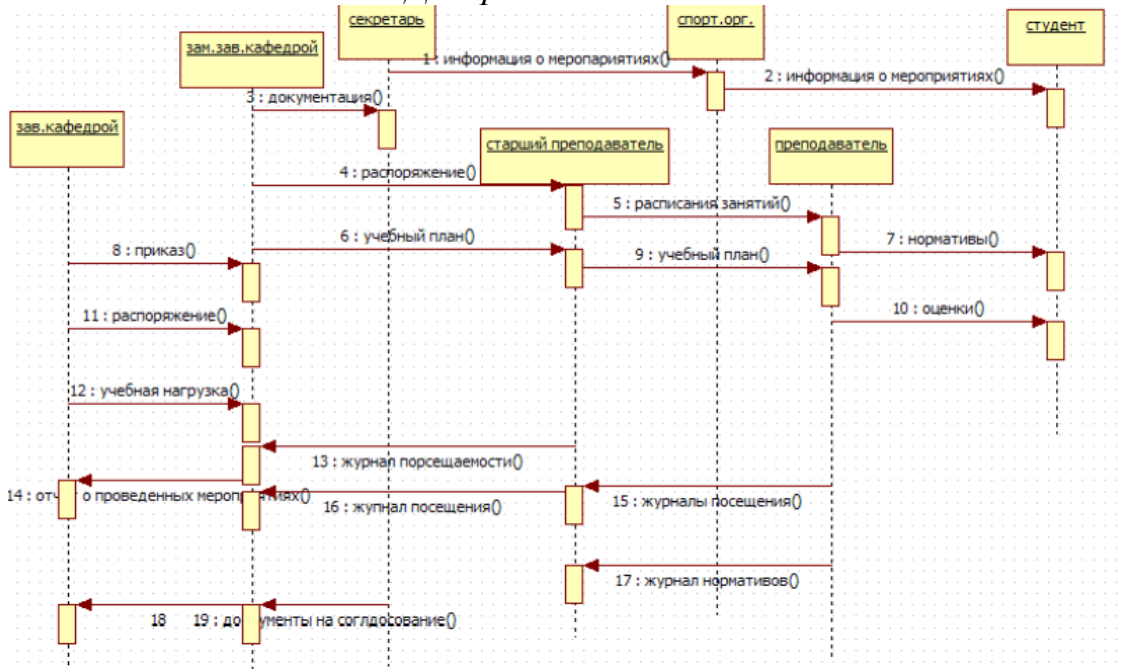


Рис. 4. Диаграмма взаимодействий

Диаграмма сообщений отображает взаимодействие объектов.  
В нашем случае данная диаграмма отображает взаимодействие (передачу сообщений) между сотрудниками кафедры физического воспитания.

### Диаграмма состояний

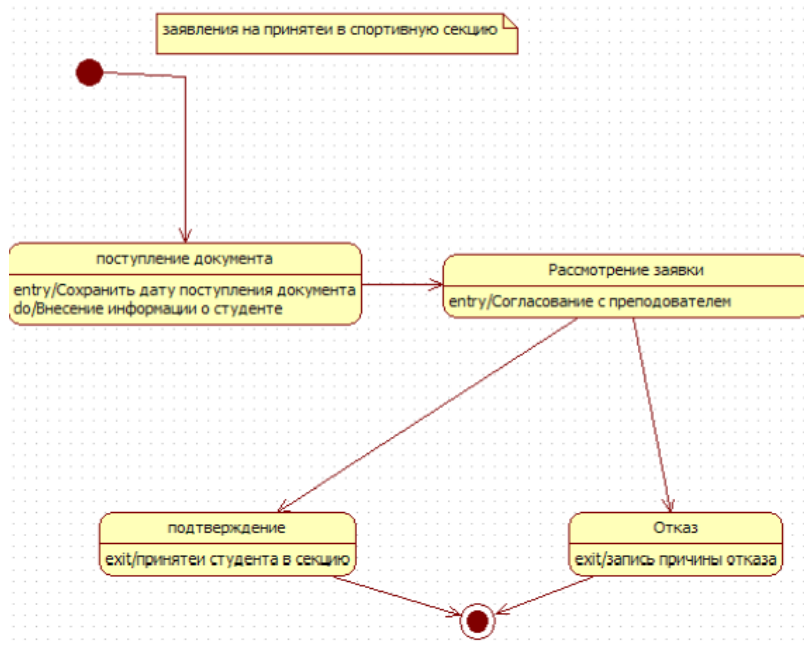


Рис.5. Диаграмма состояний

Диаграмма состояний выражает поведение объекта через состояния и переходы состояний.

В нашем случае данная диаграмма показывает состояние заявления, начиная с его подачи и заканчивая заверением (подтверждение или отказ).

Диаграмма состояний необходима для описания динамического поведения классов, если эти классы могут находиться в разных состояниях. Одна диаграмма соответствует одному классу.

В любой момент времени класс может находиться в одном состоянии. Переход от одного состояния в другое - скачкообразный. Переход должен сопровождаться изменением значения хотя бы одной переменной из данных класса. Классы могут переходить из одного состояние в другое самостоятельно или под внешним воздействием. Переход класса из одного состояния в другое – событие (event).

Событие = условие возникновения + параметры.



## Диаграмма взаимодействия ролей

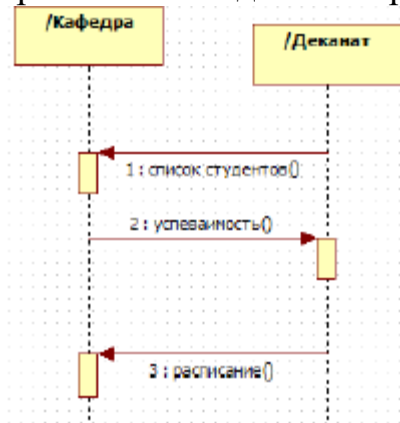


Рис.61. Диаграмма сообщений роли

Диаграмма сообщений роли отображает взаимодействия в концепции ролей. В нашем случае диаграмма показывает взаимодействие кафедры с деканатом.

## Диаграмма коопераций

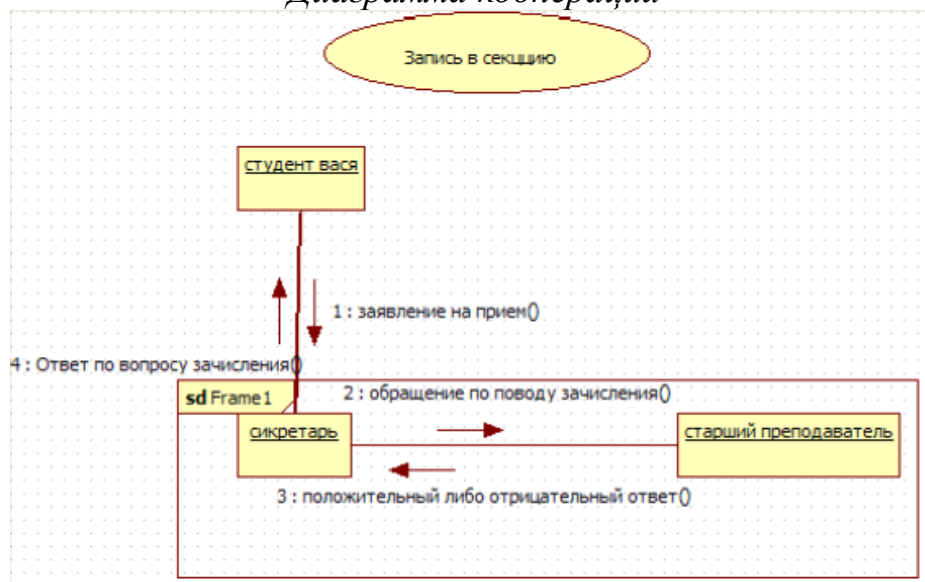


Рис.7. Диаграмма коопераций

Диаграмма коопераций отображает взаимодействие между объектами. Она является прямым отображением модели взаимодействия объектов.

В нашем случае диаграмма показывает взаимодействие кафедры во время обработки заявления на запись в секцию.

В ходе работы программы объекты обмениваются сообщениями. Сообщение – это передача информации от одного объекта к другому. Рассмотрим 2 аспекта:

- 1) временной – в какой очередности сообщения передаются между объектами;
- 2) структурный – как сообщения могут быть переданы между объектами. По сути передача сообщения означает, что объект одного класса вызывает метод объекта другого класса.

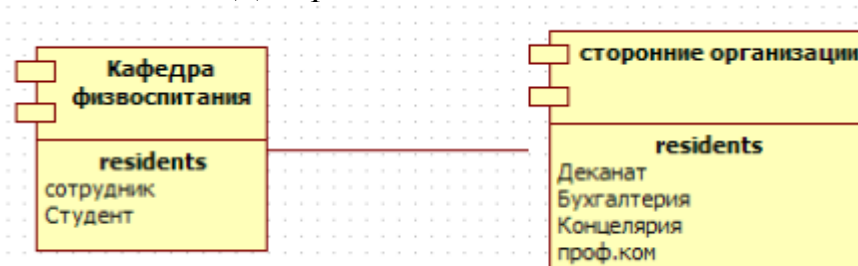
Обычно на диаграмме указывают объекты, а не классы. Но если все объекты ведут себя идентично, то можно написать имя класса. Каждый объект обладает линией жизни. Если она заканчивается крестиком, то в этот момент времени объект уничтожается. Если на ней нарисован прямоугольник, то это значит, что объект в это время действует. Диаграмма коопераций – вариант диаграммы последовательности откуда исключено время.



*Рис.8. Диаграмма коопераций ролей*

Диаграмма коопераций ролей отображает взаимодействия между ролями. Она является прямым отображением модели взаимодействия классификаторов-ролей внутри кооперации. В нашем случае диаграмма описывает взаимодействие и роли каждого участника спортивной секции, образованной кафедрой физ. воспитания.

*Диаграмма компонентов*



*Рис.9. Диаграмма компонентов*

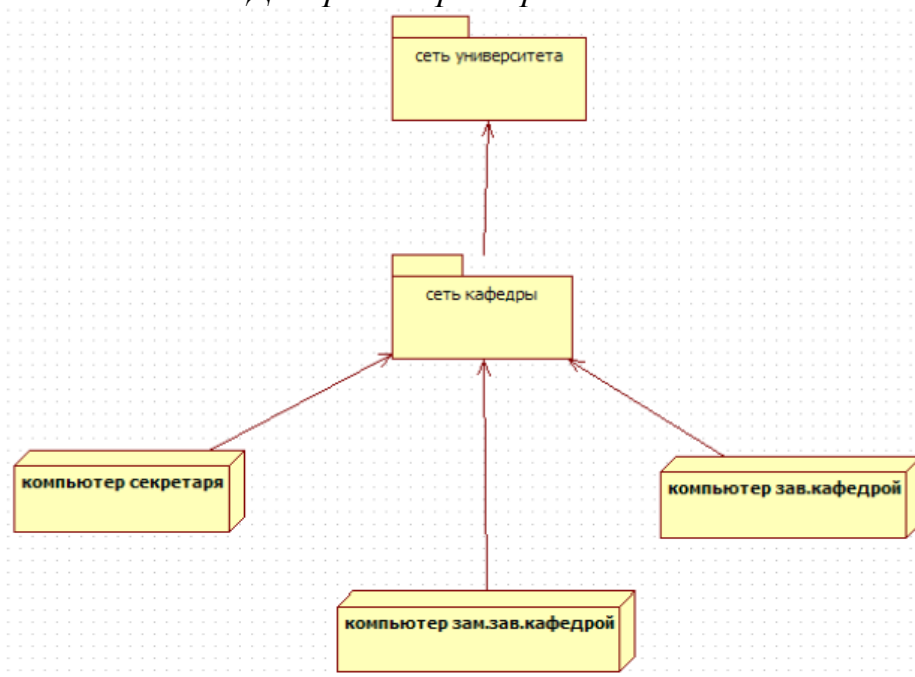
Диаграмма компонентов отображает зависимость между программными компонентами.

Элементы, которые составляют программные компоненты и элементы, которые реализуют эти компоненты, могут быть отображены на диаграмме компонентов. В нашем случае диаграмма компонентов отображает взаимодействие кафедры физ. воспитания со сторонними организациями внутри университета.

Диаграмма компонентов описывает особенности физического представления системы, позволяет определить архитектуру разрабатываемой системы.

Компонента – единица физической реализации системы. Все классы нужно прикрепить к компонентам и все компоненты к узлам обработки. Компоненты взаимодействуют с собой через интерфейс (обозначается кружочком, присоединяется сплошной линией).

#### *Диаграмма развертывания*



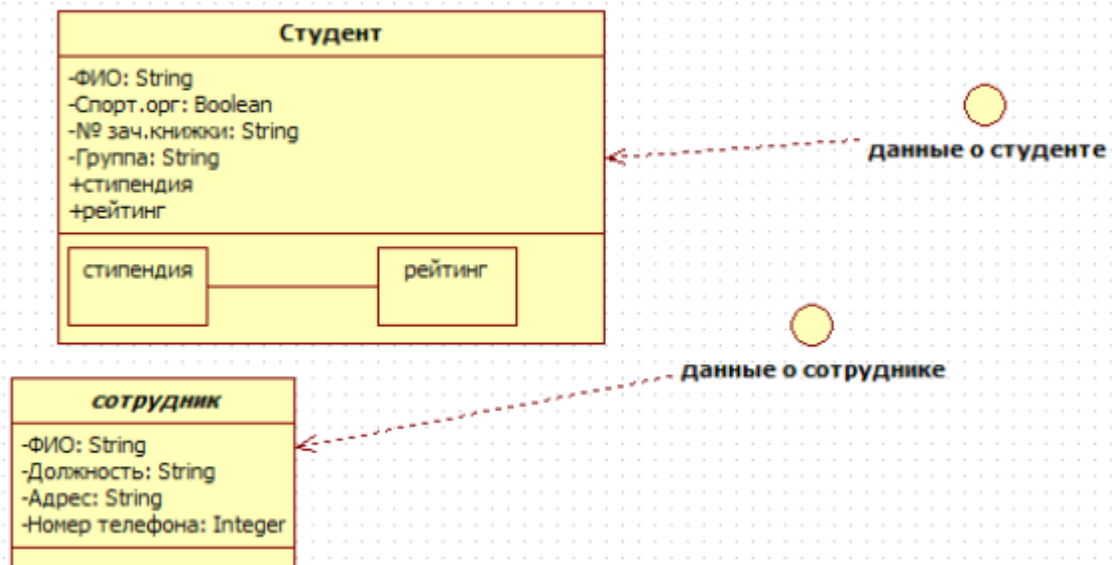
*Рис.10. Диаграмма развертывания*

Диаграмма развертывания отображает аппаратные элементы компьютера, другие устройства и программные компоненты, а также процессы и объекты, которые им назначены. В нашем случае данная диаграмма показывает связь имеющихся на кафедре компьютеров и их связь с общей сетью университета.

Диаграмма развертывания применяется для представления общей конфигурации системы и содержит распределение компонентов системы по отдельным узлам системы. Кроме того, показывает наличие физических соединений.

Узел – некоторый физически существующий элемент системы, обладающий некоторым вычислительным ресурсом. Узел изображается в виде куба.

#### *Композиционная структурная диаграмма*



*Рис.11. Композиционная структурная диаграмма*

Композиционная структурная диаграмма - диаграмма, выражающая внутреннюю структуру классификатора. Она показывает его точки зрения взаимодействия с другими частями системы. В нашем случае диаграмма показывает структуру элементов и связь между ними.

#### Контрольные вопросы

1. Что такое модель в программе StarUML?
2. Что такое представление?
3. Что такое диаграмма?
4. Что такое проект?
5. Какие элементы входят в структуру проекта?
6. Какое расширение имеют файлы проекта в StarUML?
7. Что такое фреймворк?
8. Какие профили используются в StarUML?
9. Какие базовые модули использует StarUML?
10. Перечислите основные типы диаграмм в программе StarUML?
11. Что означает название UML?
12. Что означает название MDA?
13. Перечислите типы диаграмм в языке UML?
14. В каком формате должны готовиться файлы для программы StarUML?
15. В каких областях науки и техники могут использоваться модели, созданные с помощью программы StarUML?
16. Перечислите основные концепции StarUML.