**ФКПОУ «НТТИ» Минтруда России**

**Практическая работа 9**

**по МДК 03.01 профессионального модуля ПМ.03Участие в интеграции программных модулей**

|  |  |  |
| --- | --- | --- |
| Выполнил |  | студент(ка) гр. \_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Фамилия, инициалы) |
| Принял |  | преподаватель  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Фамилия, инициалы) |
| Оценка  Дата сдачи отчета |  | \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_\_ 20\_ г. |

2020

**Практическая работа № 9**

Название работы: Статическое представление программной системы с использованием диаграмм классов и объектов UML.

Цель работы: изучить основные принципы построения визуальных моделей на основе унифицированного языка моделирования UML.

Продолжительность работы: 4 часа.

**Требования к знаниям и умениям студентов:**

*знать:*

* модели процесса разработки программного обеспечения;

*уметь:*

* выполнять системный анализ и проектирование компонент ПО на основе существующих методологий с использованием автоматизированных программных (CASE) средств

**Литература и средства обучения:**

1. Рудаков А.В. Технология разработки программных продуктов: Практикум (3-е изд., стер.) учебное пособие, 2015 г.
2. Перлова О.Н. Проектирование и разработка информационных систем (2-е изд., стер.) . -М.: Академия, 2018 г.

**Технические средства и программное обеспечение:**

1. StarUML

2. MS Visio 2010 или MS Visio 2007;

3. MS Word 2010 или MS Word 2007;

**КРАТКАЯ ТЕОРИЯ:**

Унифицированный язык моделирования UML (Unifed Modeling Language) предлагает общую методику построения визуальных моделей сложных объектно – ориентированных систем.

Основу языка составляет словарь UML (набор терминов, а также их обозначений), а также синтаксис UML , направленный на формирование диаграмм.

Диаграммы UML являются основными структурообразующими компонентами визуальной модели.

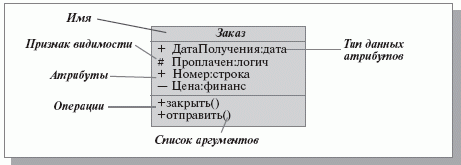
**Концептуальное и логическое моделирование информационных систем**

Классы — это базовые элементы любой объектно-ориентированной системы. Классы представляют собой описание совокупностей однородных объектов с присущими им свойствами — атрибутами, операциями, отношениями и семантикой.

В рамках модели каждому классу присваивается уникальное имя, отличающее его от других классов. Если используется составное имя (в начале имени добавляется имя пакета, куда входит класс ), то имя класса должно быть уникальным в пакете.

Атрибут — это свойство класса, которое может принимать множество значений. Множество допустимых значений атрибута образует домен. Атрибут имеет имя и отражает некоторое свойство моделируемой сущности, общее для всех объектов данного класса. Класс может иметь произвольное количество атрибутов.

Операция — реализация функции, которую можно запросить у любого объекта класса. Операция показывает, что можно сделать с объектом. Исполнение операции часто связано с обработкой и изменением значений атрибутов объекта, а также изменением состояния объекта.



**Диаграммы классов**

Классы в UML изображаются на диаграммах классов, которые позволяют описать систему в статическом состоянии — определить типы объектов системы и различного рода статические связи между ними. Классы отображают типы объектов системы.

Между классами возможны различные отношения:

* зависимости, которые описывают существующие между классами отношения использования;
* обобщения, связывающие обобщенные классы со специализированными;
* ассоциации, отражающие структурные отношения между объектами классов.

**ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:**

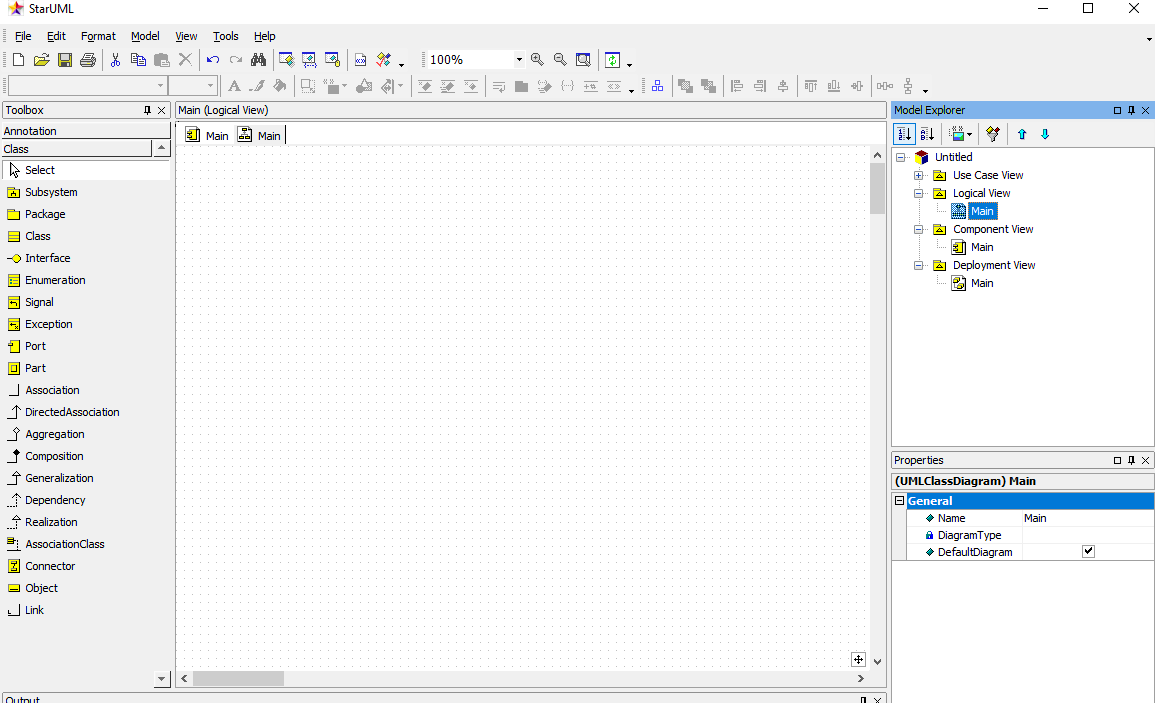
**Часть I. Концептуальное моделирование информационной системы с использованием диаграммы классов (Class Diagrams) средствами StarUML**

Диаграммы классов (class diagrams) – логическая модель базовой структуры системы, отражает статическую структуру системы и связи между ее элементами.

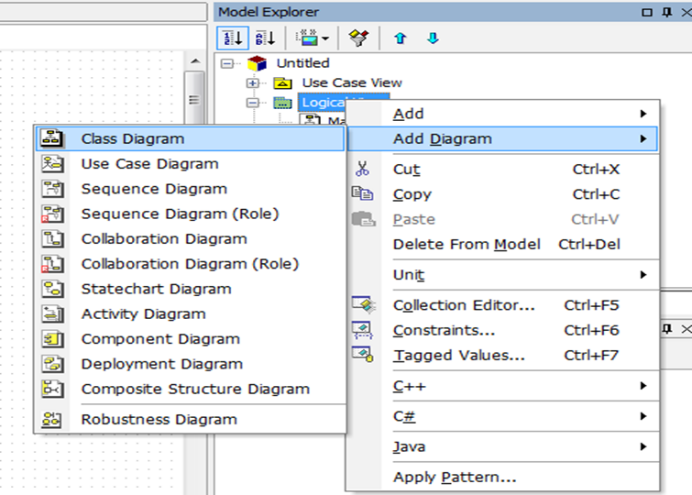
1. Запустите StarUML и выбрать подход (Approaches) Rasional Approache.

Откройте ранее созданную модель (см. Практические работы №7-8) кадрового учёта.

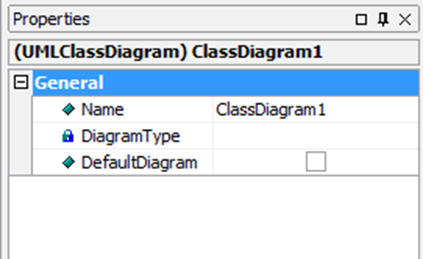
Для создания **ClassDiagram** раскройте логическое представление (Logical View) в навигаторе модели (Model Explorer).



На диаграмме Main представления Logical View обычно размещают главную диаграмму пакетов, а диаграммы классов помещают на другие листы этого представления. Для создания новой диаграммы классов выполним следующие шаги: щелкнуть правой кнопкой мыши по папке представления Logical View в навигаторе модели, в контекстном меню выбрать пункт Add Diagram, в списке выбрать диаграмму классов Class Diagram.

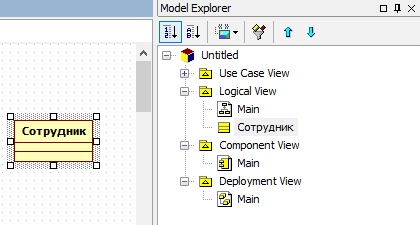


Будет создана новая диаграмма классов со стандартным именем ClassDiagram1.

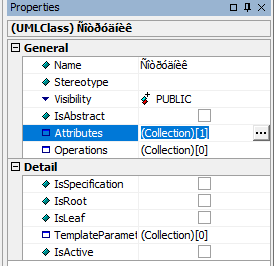


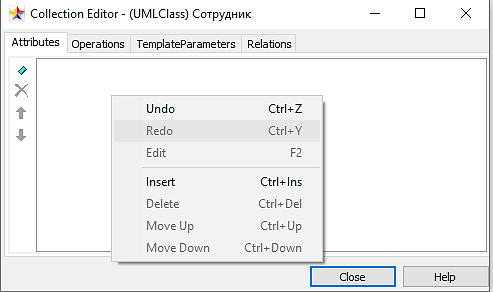
**2. Добавление компонентов на диаграмму классов.**

Щёлкните по кнопке Class на панели инструментов (Toolbox) или щелкните правой кнопкой мыши по Logical View в навигаторе модели, в контекстном меню выберите пункт Add (Добавить), затем выберите пункт Class (Класс). Новый класс будет создан и отобразится в навигаторе модели. На вкладке Properties (Свойства) измените имя класса на **Сотрудник**.

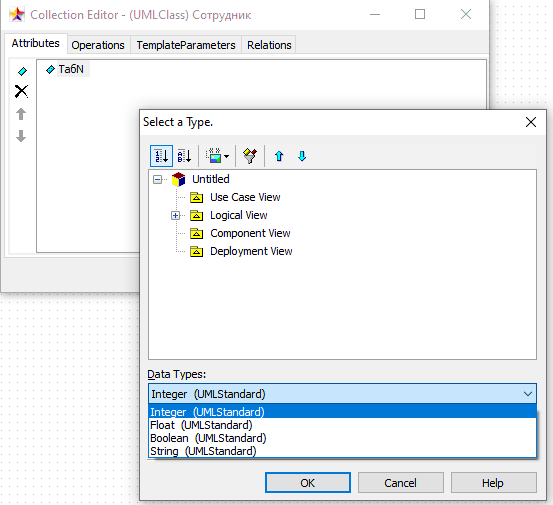


**3. Добавление атрибута класса.**

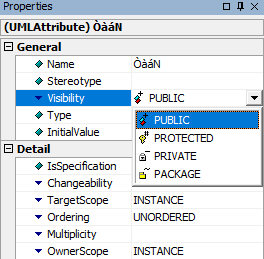
Для добавления атрибута дважды щёлкните на классе, в Панели свойств (Properties) щелкнуть на кнопку . В появившемся окне выбрать вкладку **Atributes.** Затем щёлкните правой кнопкой мыши в поле атрибутов и выберите в контекстном меню команду **Insert (Добавить).**



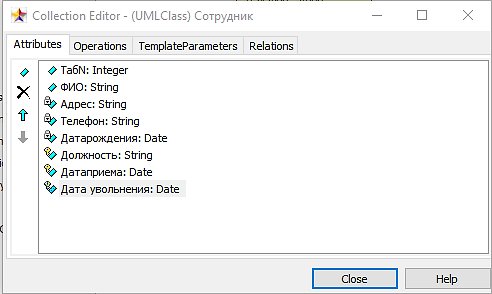
Введите имя атрибута: **ТабN.** Для того, чтобы установить тип этого атрибута выделите атрибут, откроется его редактор свойств, открыть в нем раздел Type. В появившемся диалоговом окне выберите один из стандартных типов. Если нет, то ввести вручную.



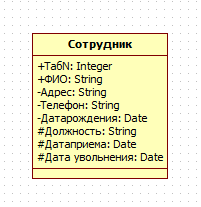
Для задания признака видимости выделите его и в Панели свойств (Properties) выберите Visibility.



Затем добавьте следующий атрибут, щёлкнув правой кнопкой мыши и выбрав команду Insert. Таким образом сформируйте все атрибуты класса Сотрудник, показанные на рисунке:

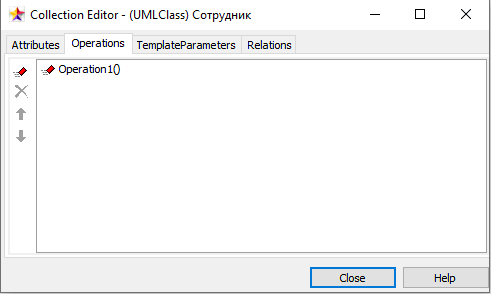


После того, как все атрибуты созданы, щёлкните по кнопке ОК и убедитесь, что класс Сотрудник на диаграмме сформирован следующим образом:

****

**4. Добавление операции к классу.**

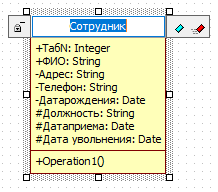
Чтобы создать операцию класса в StarUML, щелкните один раз по этому классу, в редакторе свойств Properties откройте раздел Operations, нажав кнопку . Откроется редактор коллекций (Collection Editor), в котором нужно перейти в редактор операций, выбрав вкладку Operations (Операции).



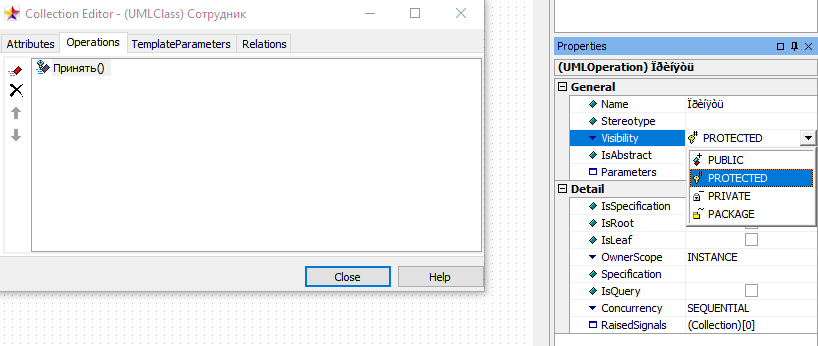
Если в редакторе операций нажать кнопку то будет создана новая операция и откроется редактор ее свойств, в котором можно изменить имя созданной операции (раздел Name). Введите имя новой операции – **Принять**.



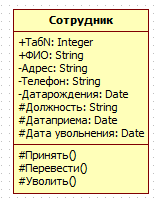
Более быстрый способ создать атрибут или операцию – это щелкнуть два раза левой кнопкой мыши по классу.



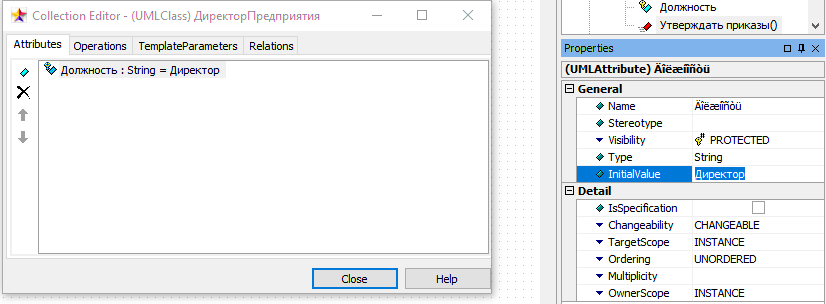
Для задания признака видимости в Панели свойств Properties задать значение.



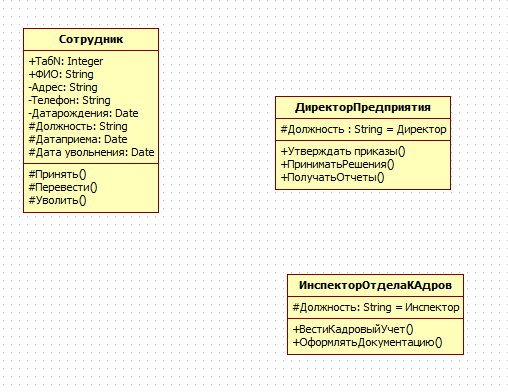
Таким же образом добавьте операции – **Перевести** и **Уволить.** После создания всех операций щёлкните на кнопке OK и убедитесь, что созданный класс Сотрудник имеет вид:

****

5. **Далее создайте классы Директор и Инспектор ОК**, которые являются дочерними для класса Сотрудник и, следовательно, наследуют все его атрибуты и операции. Добавьте только уникальные атрибуты и операции классов – наследников. Для задания значений Директор и Инспектор ввести в панели свойств в InitialValue.



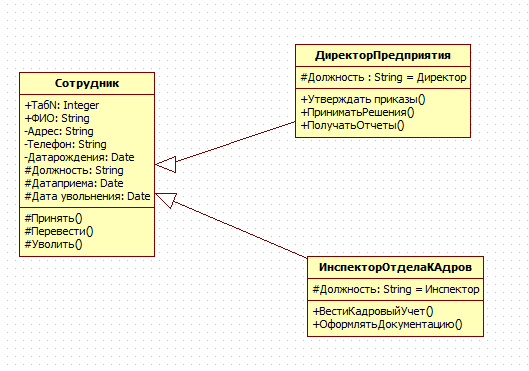
Должно получиться, как на рисунке.



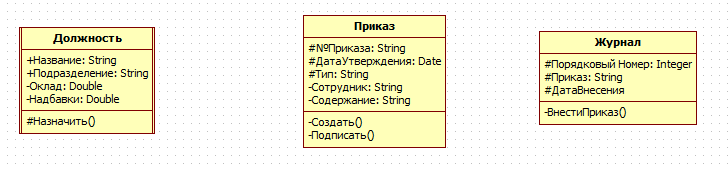
**6. Связывание классов**

Свяжем между собой классы Директор Предприятия и Сотрудник отношением обобщения (родитель-потомок). Для этого щёлкните на панели инструментов по кнопке Generalization и, удерживая левую кнопку мыши, протяните стрелку от дочернего класса (Директор Предприятия) к родительскому (Сотрудник). Аналогично свяжите отношением обобщения классы Инспектор Отдела Кадров (потомок) и Сотрудник (родитель).

Результат показан на рисунке:

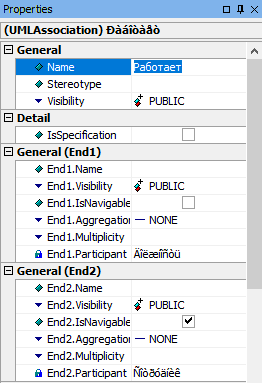


7. Добавьте на диаграмму классов созданные самостоятельно классы **Должность, Приказ** и **Журнал**, как показано на рисунке:

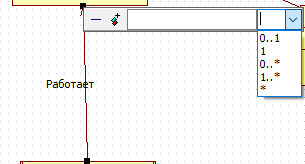


**8.** Создадим теперь отношение типа *ассоциация* между классами **Должность** и **Сотрудник** (на 1-й должности работают 0 или много сотрудников)

Для создания ассоциации щёлкните по кнопке  на Панели инструментов и протяните стрелку между классами. Затем выделите стрелку и в панели свойств введите имя (Name) **Работает.**



Для установления кратности ассоциации нажмите один раз на линию ассоциации, к которой вы хотите добавить кратность. Наведите курсор мыши на один из квадратных блоков в конце строки, чтобы получить курсор мыши и дважды щелкнуть мышью. Затем вы увидите всплывающее меню с выпадающим меню со списком справа.

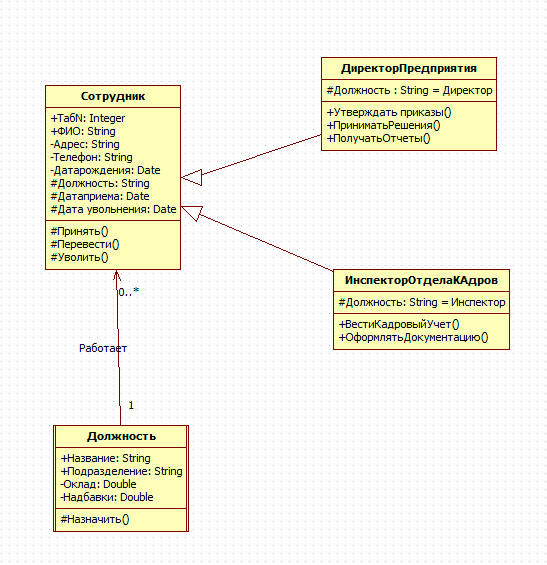


Выберите множественность из выпадающего меню, множественность можно выбрать из: 0..1, 1, 0.., 1.. и \* или ввести напрямую.

Выбрать кратность 0..n.

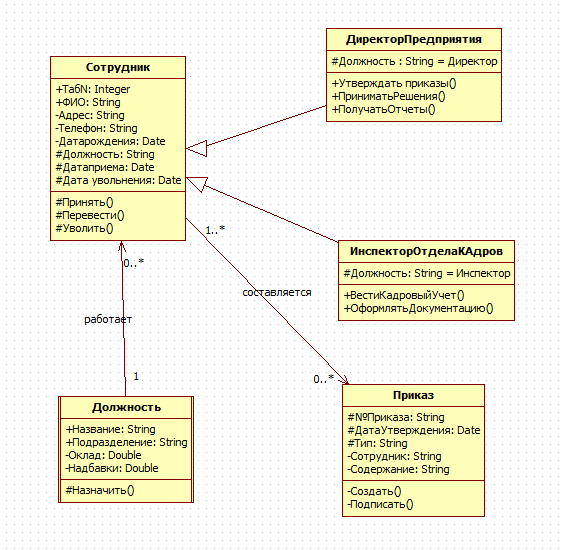
Повторите это на квадратном блоке на другом конце линии. Выбрать 1.

Теперь диаграмма классов будет иметь вид:



Создайте самостоятельно ассоциацию между классами **Сотрудник и Приказ**, показывающую, что на класс Сотрудник составляется класс Приказ.

После этого формирование диаграммы классов показано на рисунке:



Создайте самостоятельно отношение между классами **ИнспекторОтделаКадров** и **Журнал**.

Представить итоговый вид диаграммы.

**Контрольные вопросы:**

1. Современные технологии разработки программных продуктов. Назначение, характеристика и обзор современных CASE– средств.
2. Построение объектной модели. Принципы объектно – ориентированного проектирования и анализа программных систем.
3. Визуальное моделирование программных систем. Особенности, назначение и преимущества визуальных моделей.
4. Назначение и функциональная характеристика языка визуального моделирования UML.
5. Основные компоненты UML, формирующие базовый словарь языка. Предметы в UML: классификация, краткая характеристика и графическое представление на диаграммах.
6. Правила формирования отношений, связывающих предметы. Классификация, графическое обозначение, а также кратность отношений в UML.