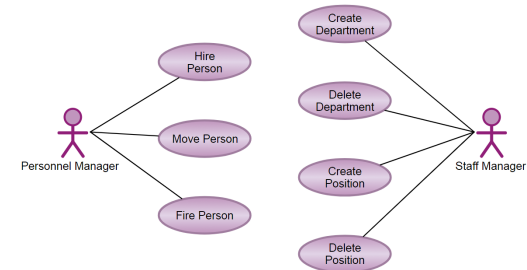


Что мы узнали к этому моменту?

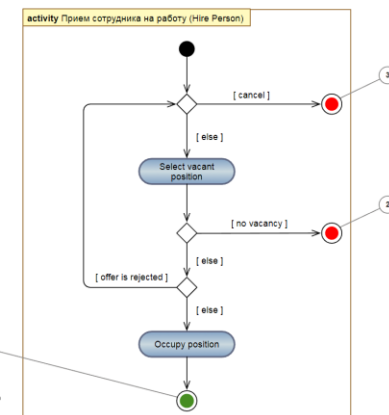
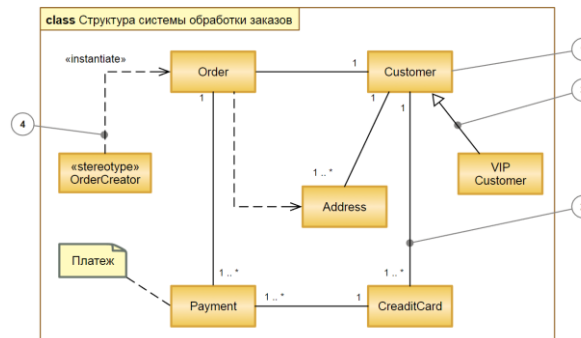
- UML –графический язык моделирования систем
 - Помогает описать модель какой либо системы в различных аспектах:

- Использование системы (зачем?)
- Структура системы (что?)
- Поведение системы (как работает?)



- Нотация языка:

- Сущности
- Связи
- Диаграммы



- Итого – научились **читать** диаграммы UML

Сегодня учимся писать :)

- Применение UML для анализа предметной области и проектирования ПО:
 - Моделирование вариантов использования
 - Моделирование структуры
 - Моделирование поведения



Моделирование использования

- Моделирование использования призвано ответить на вопрос **что полезного делает система во внешнем мире?**, и этот вопрос – один из первых, на которые необходимо дать ответ.
- Инструмент – диаграммы вариантов использования, ценится за:
 - Простоту утверждений
 - ТЗ отображается «в картинках», описывающих простые утверждения;
 - Абстрагирование от реализации и декларативное описание
 - ЧТО система делает, а не КАК и ЗАЧЕМ
 - Выявление границ
 - Четко определяется, что система делает, а что нет;
 - Выявление действующих лиц
 - Четкое понимание кто и как использует систему, позволяет построить роли пользователей
- Диаграммы вариантов использования – особенность UML, которая часто вызывает вопросы, т.к. не имеет прямых аналогий в отличие от:
 - Диаграммы классов (ER-диаграмма)
 - Диаграммы деятельности (блок-схемы алгоритмов)
 - Диаграммы состояний (автоматные модели)

Учебный пример - ТЗ

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Информационная система «Отдел кадров» (сокращенно ИС ОК) предназначена для ввода, хранения и обработки информации о сотрудниках и движении кадров.

Система должна обеспечивать выполнение следующих основных функций:

1. Прием, перевод и увольнение сотрудников.
2. Создание и ликвидация подразделений.
3. Создание вакансий и сокращение должностей.

Действующие лица (ДЛ)

- Крайне важно правильно выделить ДЛ
 - Экстремумы:
 - 1 ДЛ – «Внешний мир»
 - по 1 ДЛ на каждый вариант использования
 - Формального критерия выделения ДЛ нет
 - Неформальные критерии:
 - действующие лица находятся вне проектируемой системы (или рассматриваемой части системы)
 - отдельные ДЛ для понятных заранее категорий пользователей
 - пользователи участвуют в разных (независимых) бизнес-процессах;
 - пользователи имеют различные права на выполнение действий и доступ к информации;
 - пользователи взаимодействуют с системой в разных режимах: от случая к случаю, регулярно, постоянно.
 - отдельные ДЛ для внешних программных и аппаратных средств (если система взаимодействует с таковыми)

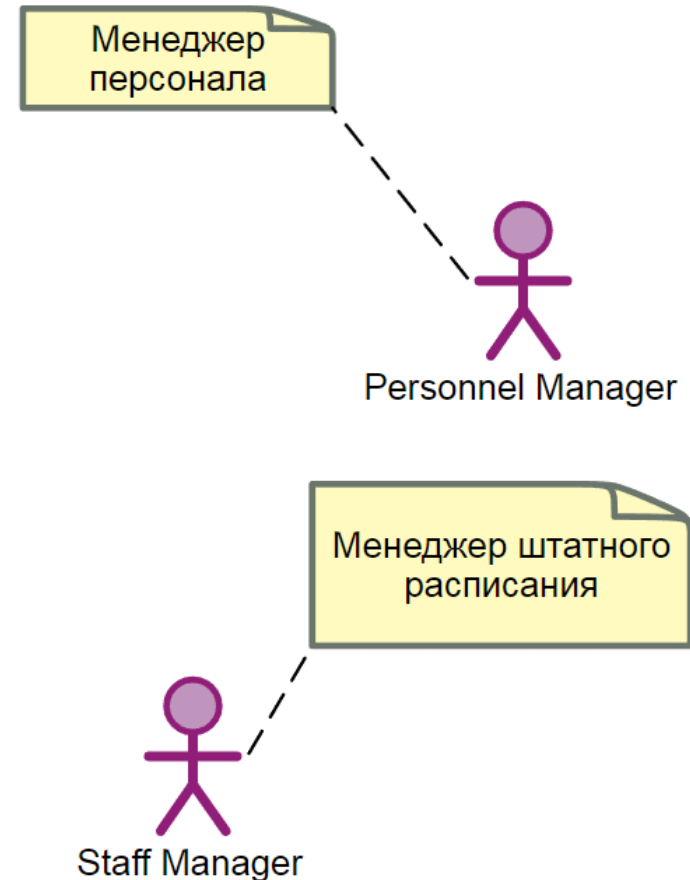
Пример – Действующие лица

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Информационная система «Отдел кадров» (сокращенно ИС ОК) предназначена для ввода, хранения и обработки информации о сотрудниках и движении кадров.

Система должна обеспечивать выполнение следующих основных функций:

1. Прием, перевод и увольнение сотрудников.
2. Создание и ликвидация подразделений.
3. Создание вакансий и сокращение должностей.



Варианты использования (ВИ)

- Выделение вариантов использования:
 - Среди всех последовательностей действий, которые могут произойти при работе приложения, выделяются такие, в результате которых получается **явно видимый и достаточно важный для действующего лица** результат.
 - Формальных методов нет
 - Неформальный метод:
 - поискать в тексте отглагольные существительные и глаголы с прямым дополнением: зачастую, в них зашифрованы варианты использования
- Вариант использования != сценарию
 - Вариант использования может содержать набор сценариев (как позитивных, так и негативных), направленных на достижение одного результата.

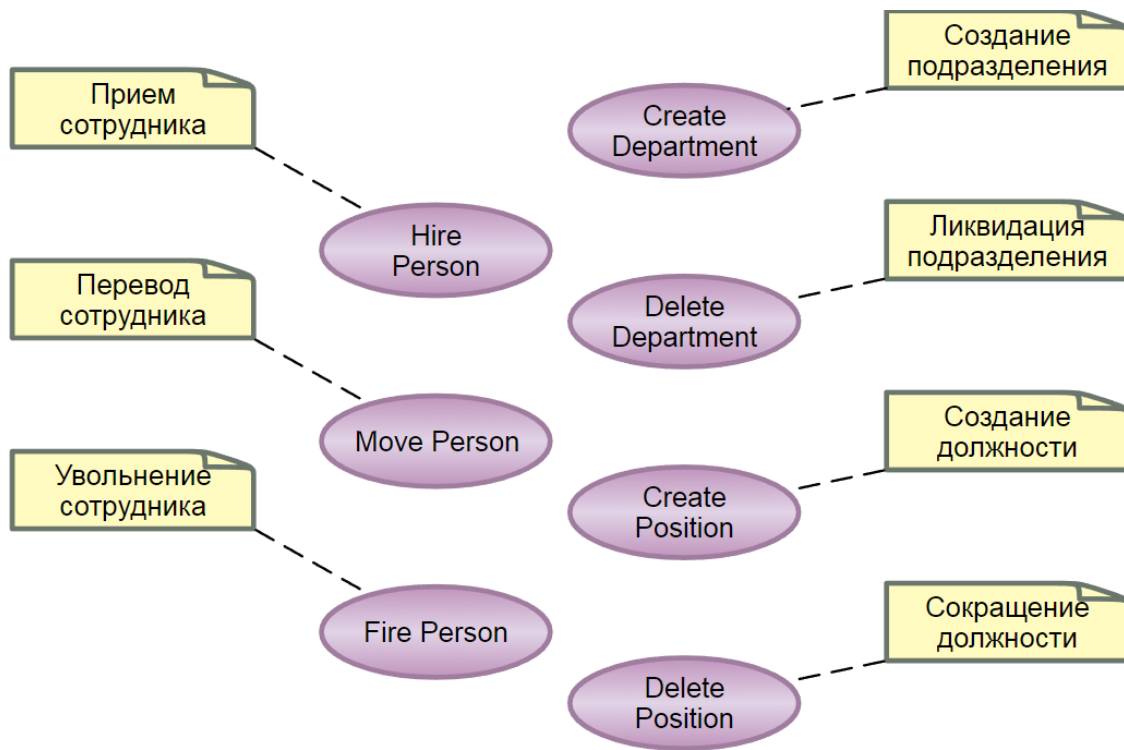
Пример – Варианты использования

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

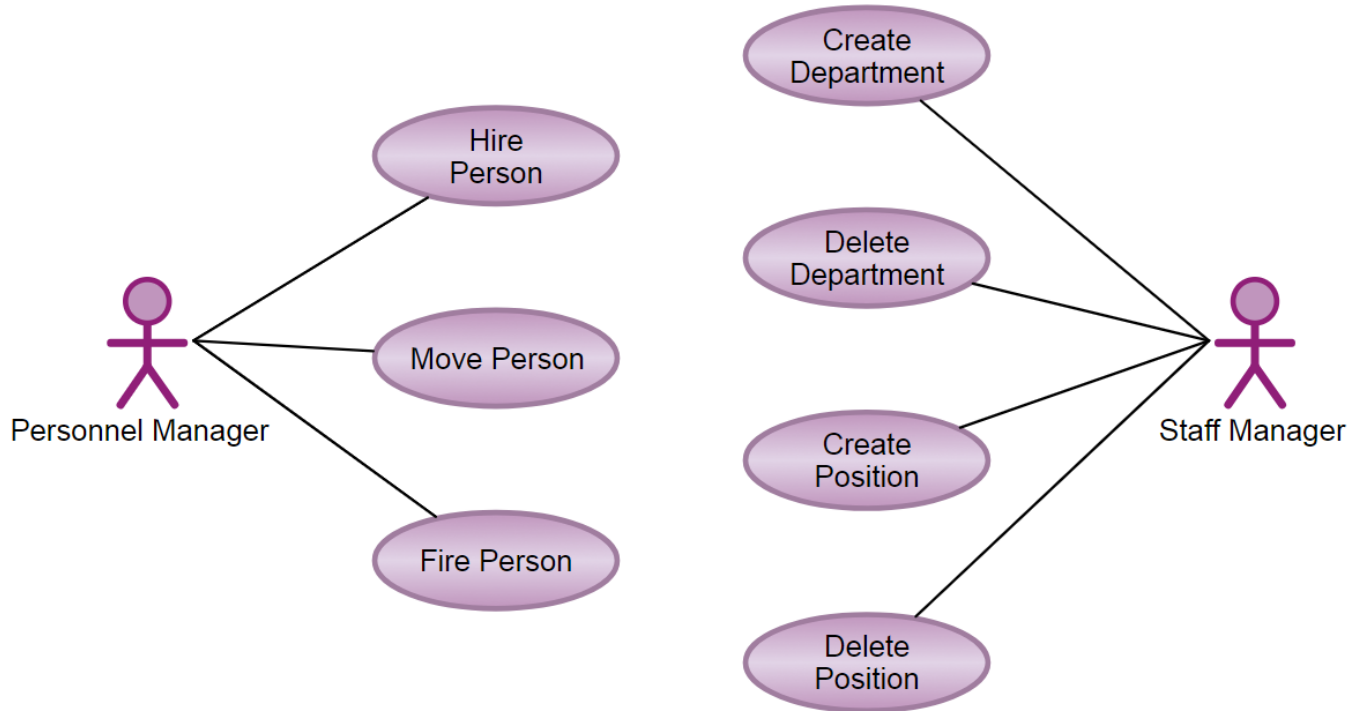
Информационная система «Отдел кадров» (сокращенно ИС ОК) предназначена для ввода, хранения и обработки информации о сотрудниках и движении кадров.

Система должна обеспечивать выполнение следующих основных функций:

1. Прием, перевод и увольнение сотрудников.
2. Создание и ликвидация подразделений.
3. Создание вакансий и сокращение должностей.



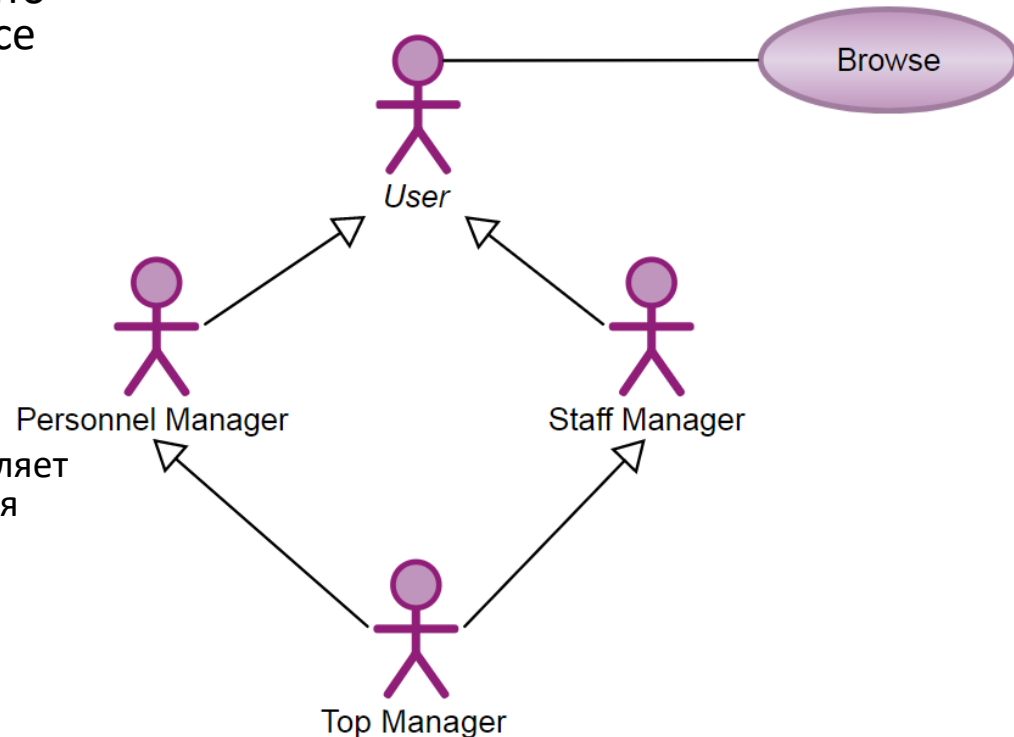
Ассоциация между ДЛ и ВИ



- **Ассоциация между ДЛ и ВИ** показывает, что ДЛ тем или иным способом взаимодействует (предоставляет исходные данные, получает результат) с ВИ.
- Ассоциация является наиболее важным и, фактически, обязательным отношением на диаграмме использования.
 - если на диаграмме использования нет ассоциаций между действующими лицами и вариантами использования, то это означает, что система не взаимодействует с внешним миром.

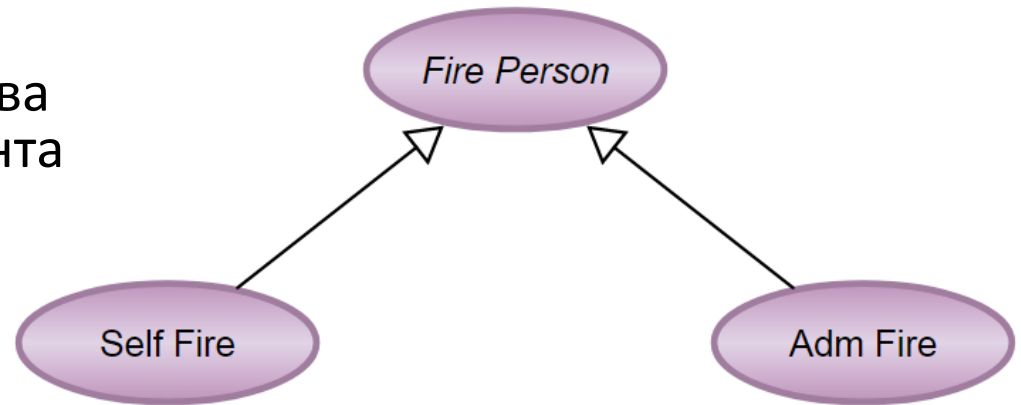
Обобщение между ДЛ

- **Обобщение между ДЛ** показывает, что одно действующее лицо наследует все свойства (в частности, участие в ассоциациях) другого действующего лица.
 - с помощью обобщения между действующими лицами легко показать иерархию категорий пользователей системы (в частности, иерархию прав доступа к выполняемым функциям и хранимым данным).
 - ДЛ может быть абстрактным, что позволяет сократить количество ассоциаций, делая модель лаконичнее и понятнее
- Пример - ИЗМЕНЕНИЯ В ТЕХНИЧЕСКОМ ЗАДАНИИ:
 - Среди всех пользователей информационной системы следует выделить особую категорию пользователей (высшее руководство), которой разрешен доступ к любым данным и операциям.
 - Информационная система должна предоставлять возможность просматривать данные без внесения в них каких-либо изменений.



Обобщение между ВИ

- **Обобщение между вариантами использования** показывает, что один вариант использования является частным случаем (подмножеством множества сценариев) другого варианта использования.
- **Пример - ИЗМЕНЕНИЯ В ТЕХНИЧЕСКОМ ЗАДАНИИ**
 - Система должна поддерживать два способа увольнения сотрудника: по инициативе администрации и по собственному желанию.



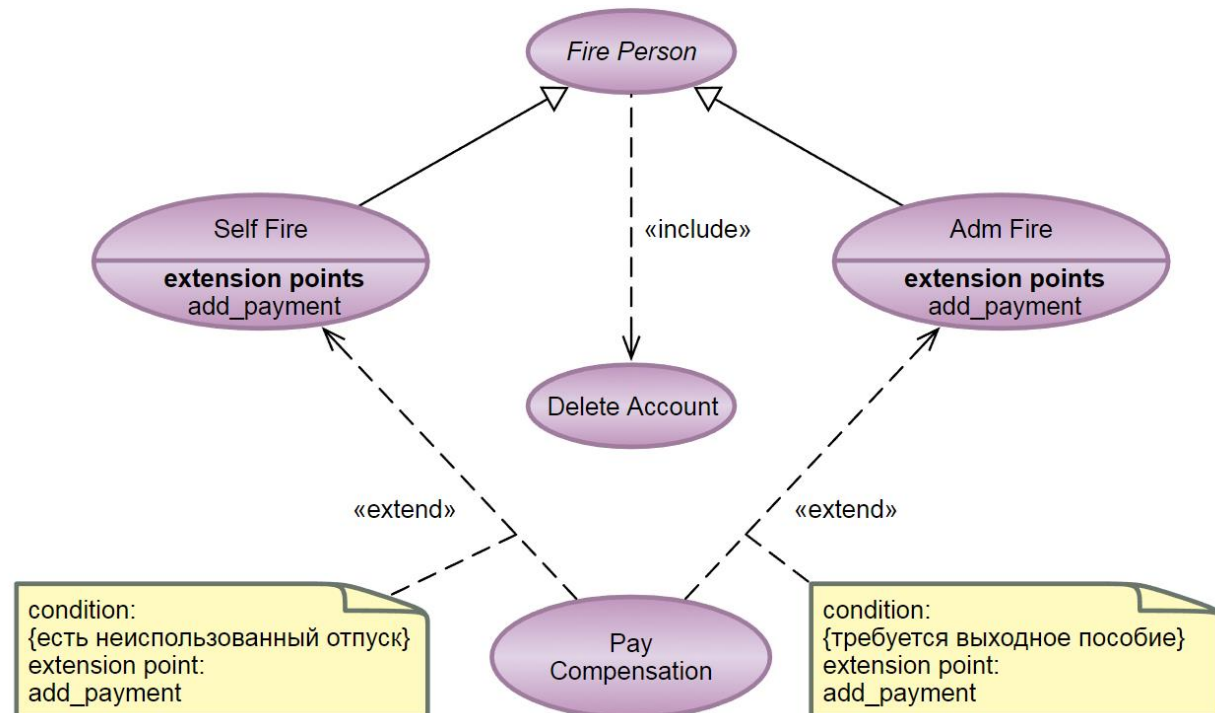
Зависимость между ВИ

- Зависимость между вариантами использования показывает, что один вариант использования зависит от другого варианта использования.

- **«include»** - каждый сценарий зависимого ВИ включает в себя сценарий независимого ВИ
- **«extend»** - независимый ВИ является **возможным** вариантом выполнения зависимого ВИ

ИЗМЕНЕНИЯ В ТЕХНИЧЕСКОМ ЗАДАНИИ:

- При увольнении сотрудника должна быть осуществлена выплата денежной компенсации за неиспользованный отпуск. В случае вынужденного сокращения возможна выплата выходного пособия.
- Учетная запись сотрудника при увольнении должна быть заблокирована.



Моделирование структуры

- Моделирование структуры призвано ответить на вопрос **из чего состоит система?**
 - в центре внимания находятся отношения "часть–целое" и статические свойства частей и целого.
- Инструменты
 - Основной инструмент – диаграммы классов (объектов)
 - Для предметной области
 - Для проектируемой системы
 - Дополнительные диаграммы – используются реже, и, зачастую, уже на финальных стадиях проектирования
 - Компонентов
 - Размещения
 - Внутренней структуры

Выделение классов

- Универсального формального метода нет
- Есть ряд неформальных методов
 - **словарь предметной области** - это набор основных понятий (сущностей) данной предметной области
 - выделите в ТЗ **имена существительные** – все они являются кандидатами на то, чтобы быть названиями классов (или атрибутов классов) проектируемой системы.
 - **реализации вариантов использования** - это описание всех или некоторых сценариев, составляющих вариант использования.
 - При анализе реализации ВИ могут быть выявлены дополнительные участники, которых необходимо смоделировать в виде классов.
 - **шаблоны (паттерны) проектирования** – это типовые решения типовых задач проектирования
 - если в процессе проектирования принимается решение использовать определенный шаблон, то зачастую приходится добавлять в модель классы, участвующие в нем.

Пример – Выделение классов

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Информационная система «Отдел кадров» (сокращенно ИС ОК) предназначена для ввода, хранения и обработки информации о сотрудниках и движении кадров.

Система должна обеспечивать выполнение следующих основных функций:

1. Прием, перевод и увольнение **сотрудников**.
2. Создание и ликвидация **подразделений**.
3. Создание **вакансий** и сокращение **должностей**.

«utility»
Company

Person

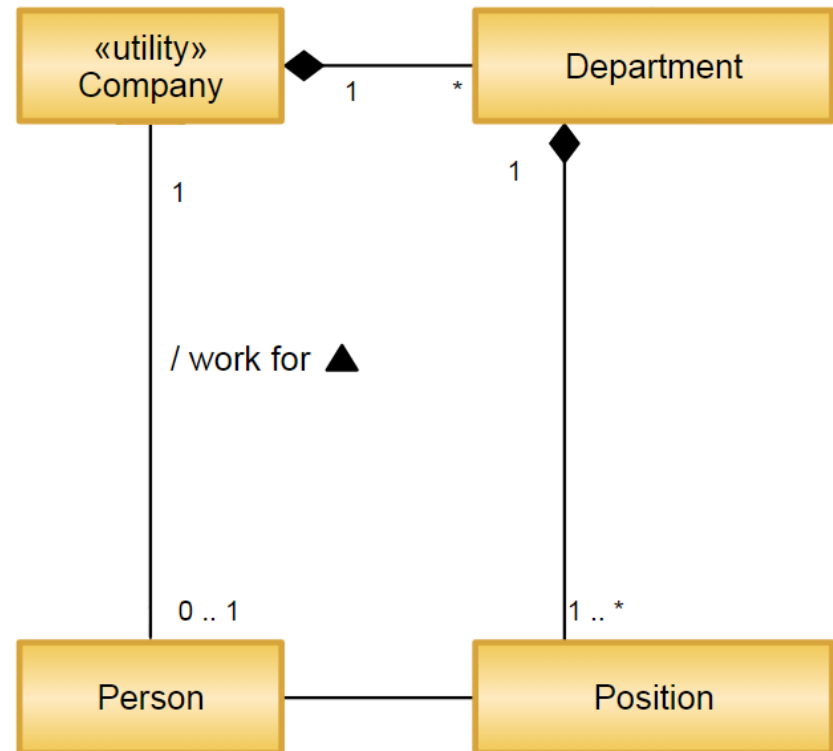
Department

Position

Вакансия == Должность, атрибут «Вакантна» = true

Отношения на диаграмме классов

- Сущности на диаграммах классов связываются главным образом отношениями
 - ассоциации (в том числе агрегирования и **композиции**)
 - и обобщения.
- Отношения зависимости и реализации на диаграммах классов применяются реже
- Выделение отношений
 - Формального метода выделения отношений нет
 - Неформальный метод требует исследования взаимоотношений классов в предметной области

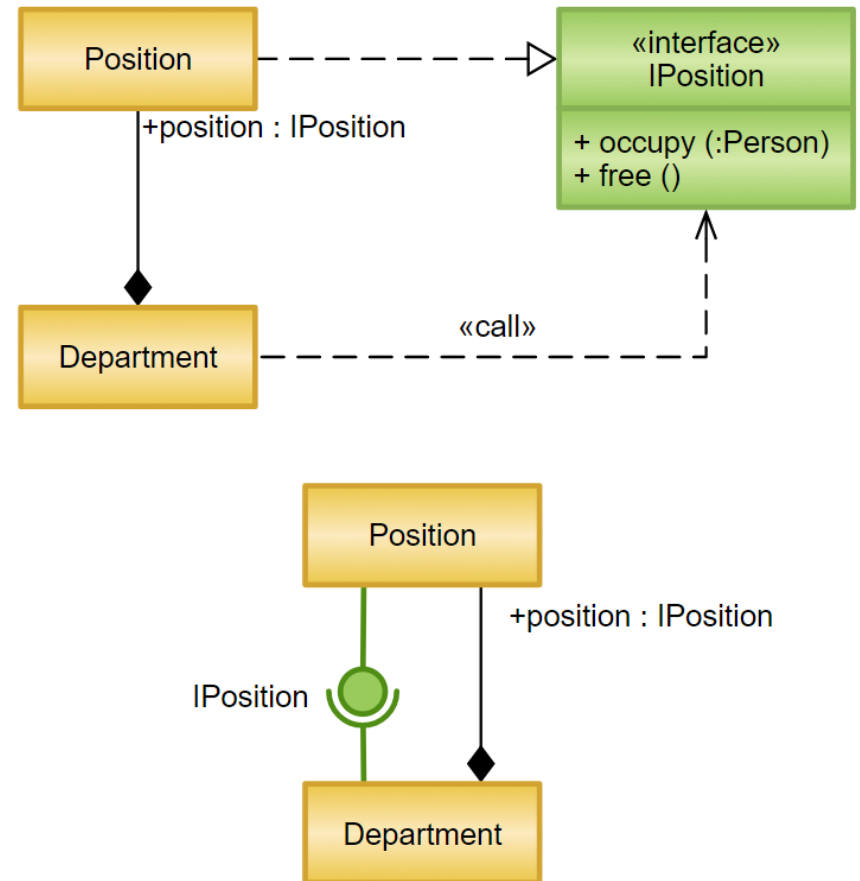


Уточнение модели

- Уточнение модели – итеративный процесс, идущий
 - параллельно с уточнением требований к разрабатываемой системе
 - параллельно с уточнением деталей технической реализации разрабатываемой системы
- И снова нет формального критерия, когда этот процесс можно считать завершенным.
- Неформально можно считать, что :
 - Уточнения модели по технической части завершаются тогда, когда уже полностью понятно, как реализовывать систему
 - Уточнения модели по требованиям потенциально, возможны до конца жизненного цикла системы, т.е. до момента вывода ее из эксплуатации.

Пример - уточнение модели

- Допустим, что класс **Department** для реализации операций связанных с движением кадров, использует операции класса **Position**, позволяющие занимать и освобождать должность – другие операции класса **Position** классу **Department** не нужны.



Пример - уточнение модели

• ИЗМЕНЕНИЯ В ТЕХНИЧЕСКОМ ЗАДАНИИ

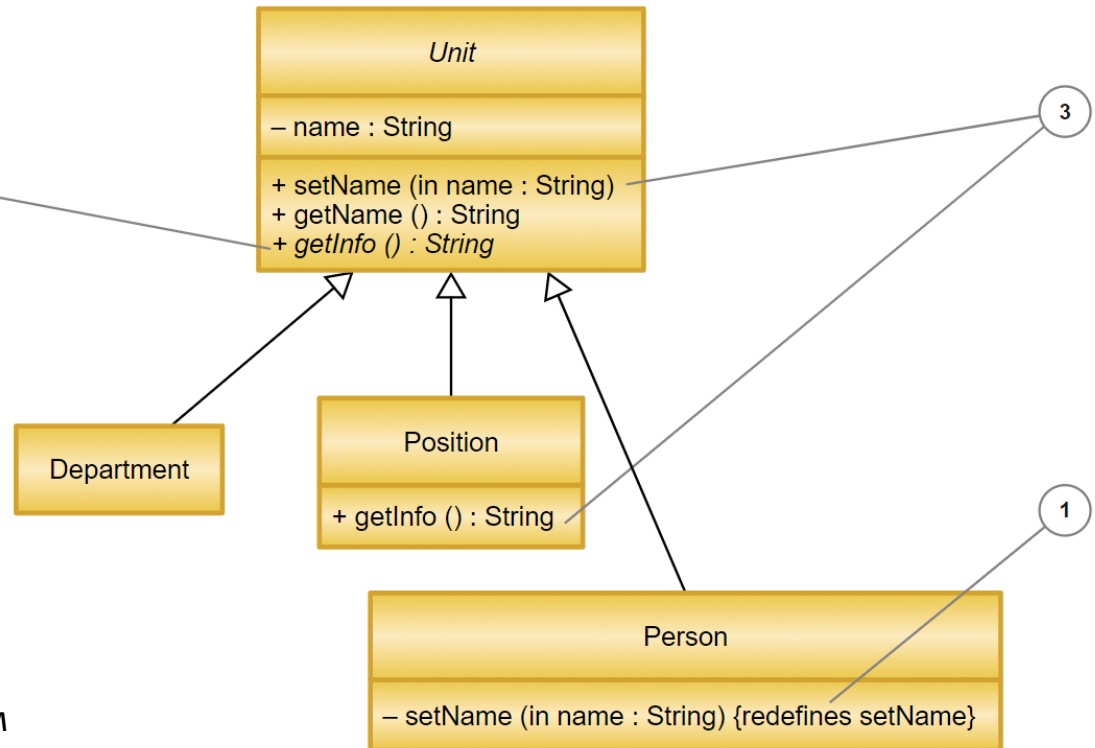
- Каждая структурная единица предприятия (подразделение, должность) должна иметь свое название.

• Вводим суперкласс **Единица (Unit)**, хранящий имя структурной единицы.

- От него порождаем **Отделы**, **Должности** и даже **Сотрудников** (у них тоже есть имя)

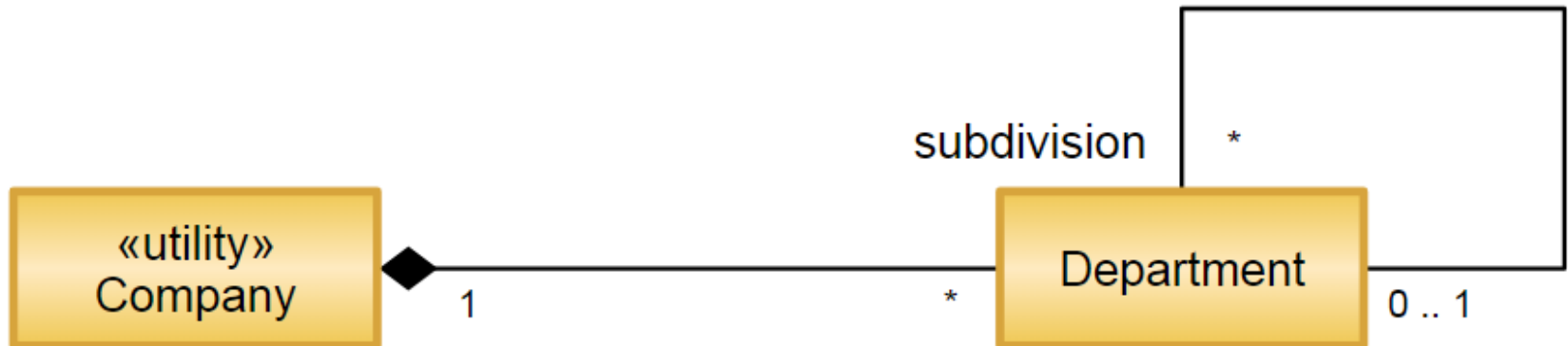
• Работа с собственным именем для выделенных классов производится не совсем одинаково

- назначение и изменение собственных имен подразделениям и должностям находится в пределах ответственности информационной системы отдела кадров
- назначение (тем более, изменение) собственного имени сотрудника явно выходит за эти пределы.



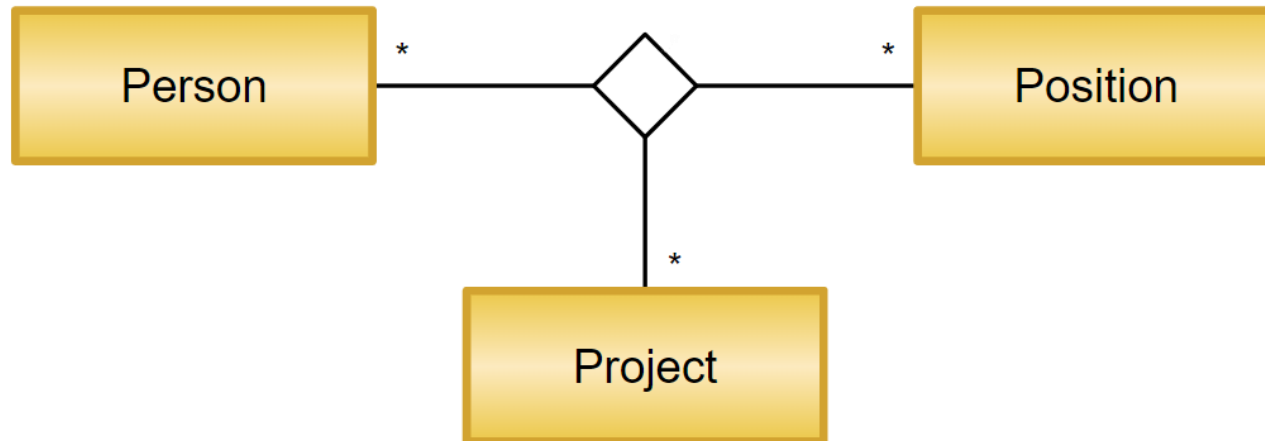
Пример - уточнение модели

- ИЗМЕНЕНИЯ В ТЕХНИЧЕСКОМ ЗАДАНИИ
 - Информационная система отдела кадров должна поддерживать иерархическую структуру подразделений на предприятии.



Пример - уточнение модели

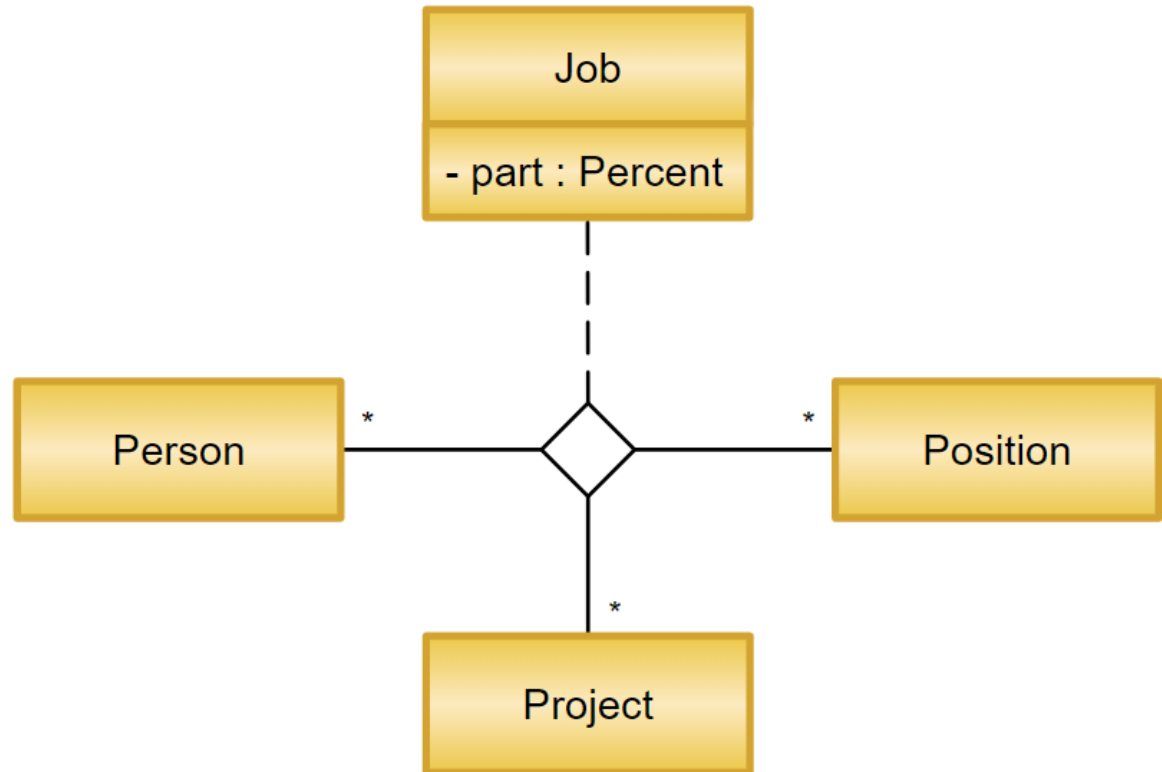
- ИЗМЕНЕНИЯ В ТЕХНИЧЕСКОМ ЗАДАНИИ
 - Информационная система должна поддерживать матричную структуру управления на предприятии и уметь оперировать таким понятием, как проект, в следующем контексте: один и тот же сотрудник может участвовать во многих проектах, выполняя различные обязанности (т.е. занимая различные должности).



Пример - уточнение модели

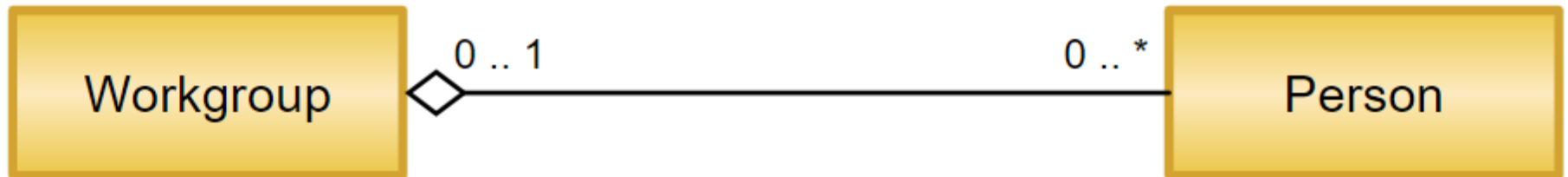
- ИЗМЕНЕНИЯ В ТЕХНИЧЕСКОМ ЗАДАНИИ

- Допускается ситуация, когда сотрудник может работать на нескольких должностях в разных проектах, а также возможно, чтобы одну и ту же должность в одном проекте занимало несколько сотрудников (дробление ставки). Размер заработной платы зависит от того, сколько конкретно времени проработал данный сотрудник в данной должности в данном проекте.



Пример - уточнение модели

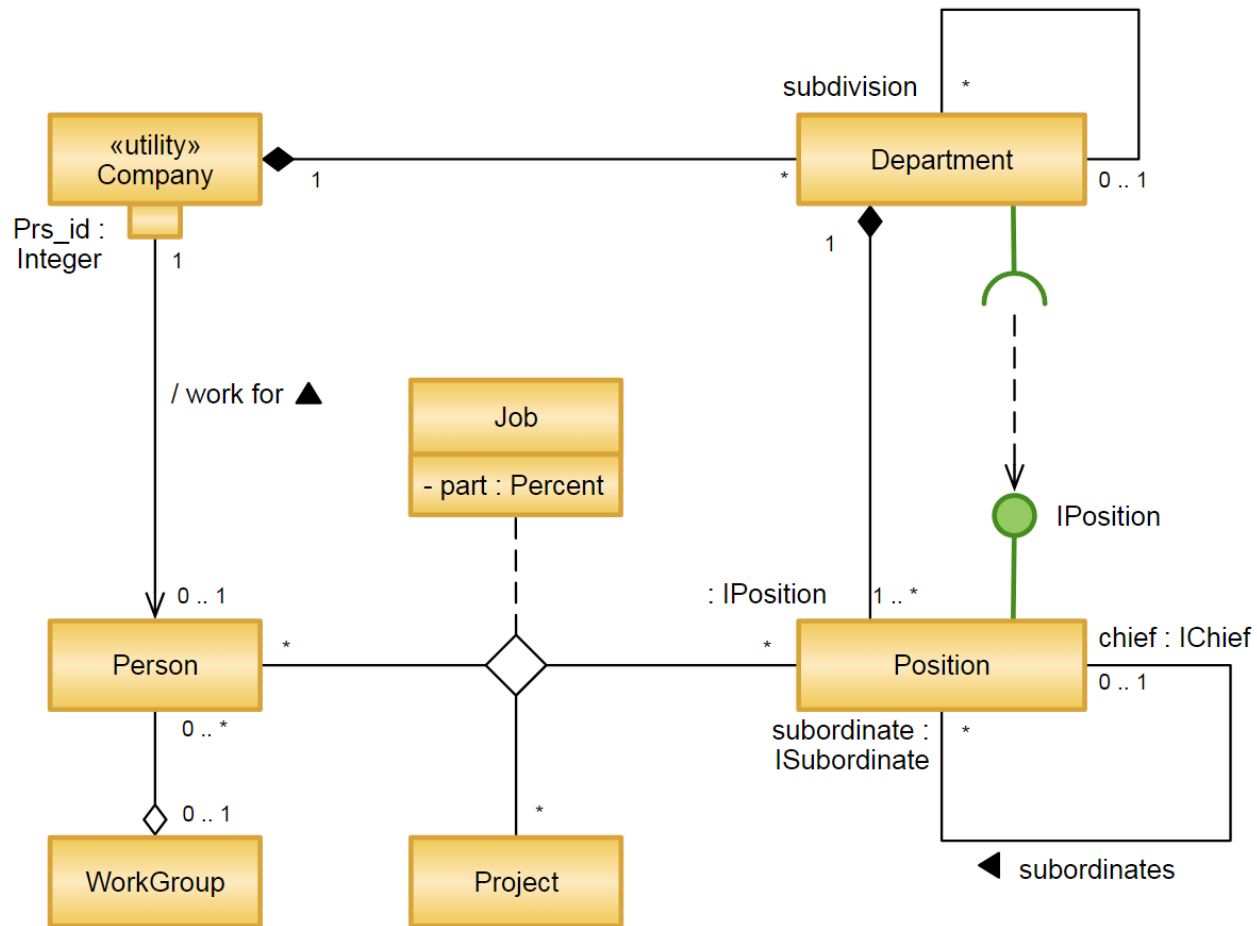
- ИЗМЕНЕНИЯ В ТЕХНИЧЕСКОМ ЗАДАНИИ
 - Информационная система отдела кадров должна поддерживать организацию временных рабочих групп на предприятии



Советы по моделированию

- **Описывать структуру удобнее параллельно с описанием поведения.** Каждая итерация должна быть небольшим уточнением, как структуры, так и поведения.
- **Не обязательно включать в модель все классы сразу.** На первых итерациях достаточно идентифицировать очень небольшую (~10%) долю всех классов системы.
- **Не обязательно определять все составляющие класса сразу.** Начните с имени класса – операции и атрибуты постепенно выявятся в процессе моделирования поведения.
- **Не обязательно показывать на диаграмме все составляющие класса и их свойства.** В процессе работы диаграмма должна легко охватываться одним взглядом.
- **Не обязательно определять все отношения между классами сразу.** Пусть класс на диаграмме "висит в воздухе" – ничего с ним не случится.

Объединяем и дополняем



Моделирование поведения

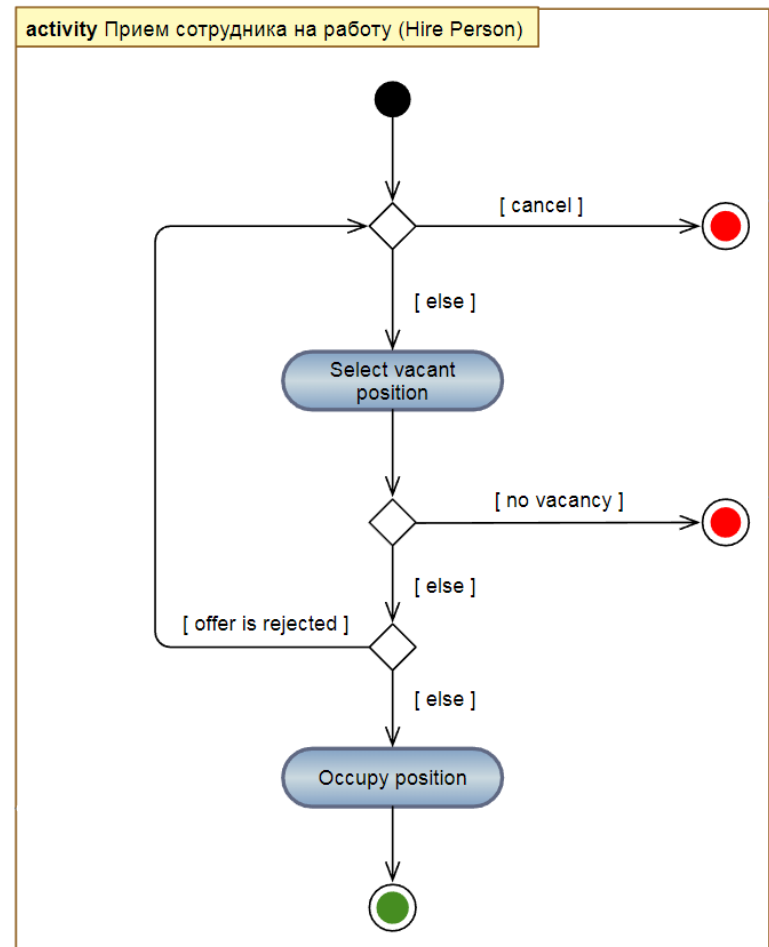
- Моделирование поведения призвано ответить на вопрос **как работает система?**
 - Модель поведения – это описание алгоритма работы системы.
- Модель поведения
 - **должна быть достаточно детальной** для того, чтобы послужить основой для составления компьютерной программы
 - **должна быть компактной и обозримой**, чтобы служить средством общения между людьми в процессе разработки системы и для обмена идеями.
 - **не должна зависеть от особенностей реализации** конкретных компьютеров, средств программирования и технологий, чтобы не сужать область применения языка UML.
 - должна быть построена знакомыми и привычными для большинства пользователей средствами и не должна противоречить требованиям наиболее ходовых парадигм программирования.
- Формальных методов, как обычно, нет.
- Неформальный метод – продумываем алгоритм работы и описываем его различные аспекты:
 - **Поток действий или данных** – диаграмма деятельности и обзорная диаграмма взаимодействия
 - **Обмен сообщениями между участниками** – диаграммы коммуникации и последовательности.
 - **Переходы между состояниями системы** – автоматная диаграмма
 - **Точная временная привязка действий** – диаграмма синхронизации

Текстовые описания

- Исторически самый заслуженный и до сих пор один из самых популярных способов: составить текстовое описание типичного сценария варианта использования.
- Рассмотрим следующий ниже текст в качестве примера одного из возможных сценариев.
- Сценарий варианта использования **Увольнение по собственному желанию**
 - 1. Сотрудник пишет заявление
 - 2. Начальник подписывает заявление
 - 3. Если есть неиспользованный отпуск, то бухгалтерия рассчитывает компенсацию
 - 4. Бухгалтерия рассчитывает выходное пособие
 - 5. Системный администратор удаляет учетную запись
 - 6. Менеджер персонала обновляет базу данных
- Казалось бы, что здесь неясного? А неясно, например, вот что: как должна вести себя система, если на шаге 2 начальник не подписывает заявление. Из текста сценария не только не ясен ответ, но, хуже того, при невнимательном чтении можно и не заметить, что есть вопрос.
- Текстовые описания сценариев всем хороши: просты, всем понятны, легко и быстро составляются. Плохи они тем, что могут быть неполны и неточны, и эти недостатки незаметны.

Диаграммы деятельности

- Еще один вариант реализации варианта использования – описать алгоритм с помощью диаграммы деятельности.
- С одной стороны, диаграмма деятельности – это полноценная диаграмма UML, с другой стороны, диаграмма деятельности немногим отличается от блок-схемы.
- Таким образом, реализация варианта использования диаграммой деятельности является компромиссным способом ведения разработки – в сущности, это проектирование сверху вниз в терминах и обозначениях UML.
- ПРИМЕР: как выглядит процесс приема на работу?



Диаграммы деятельности

- Применение диаграмм деятельности для реализации вариантов использования не слишком приближает к появлению целевого артефакта – программного кода, однако может привести к более **глубокому пониманию** существа задачи и даже открыть неожиданные возможности улучшения приложения, которые было трудно усмотреть в первоначальной постановке задачи.
- Вариант использования **должен доставлять значимый результат**, значит, если результата нет, то что-то спроектировано не так, как нужно.
- ПРИМЕР: ИС отдела кадров обычно накапливают статистическую информацию обо всех проведенных кадровых операциях. Такая статистика совершенно необходима для так называемого анализа движения кадров – важной составляющей процесса управления организацией.

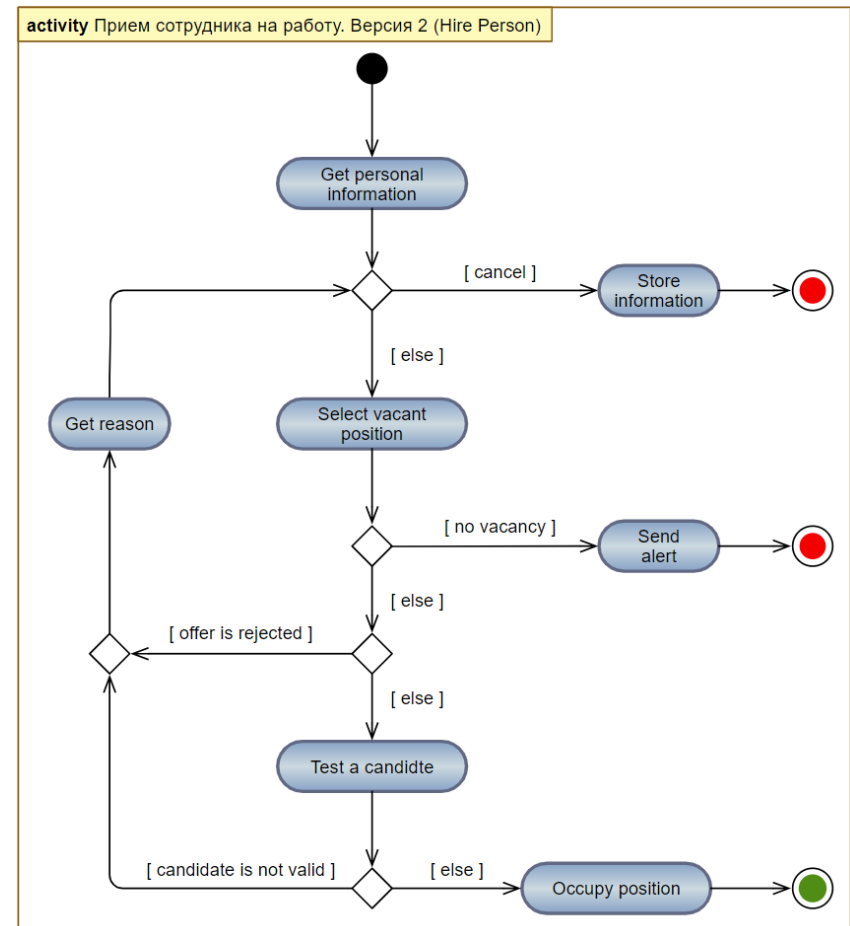


Диаграмма взаимодействия

- На диаграммах взаимодействия представлено взаимодействие объектов, т. е. экземпляров некоторых классов. Т.е., построение такой диаграммы с приводит к **выявлению некоторых классов**, которые должны существовать в модели и **некоторых операций** этих классов.
- Т.к. сообщения передаются от объекта к объекту вдоль связей (ассоциаций), то оказываются выявленными и некоторые необходимые **ассоциации**.
- Таким образом, реализация варианта использования какой-либо диаграммой взаимодействия обеспечивает **органичный переход** от моделирования использования к моделированию структуры и моделированию поведения.

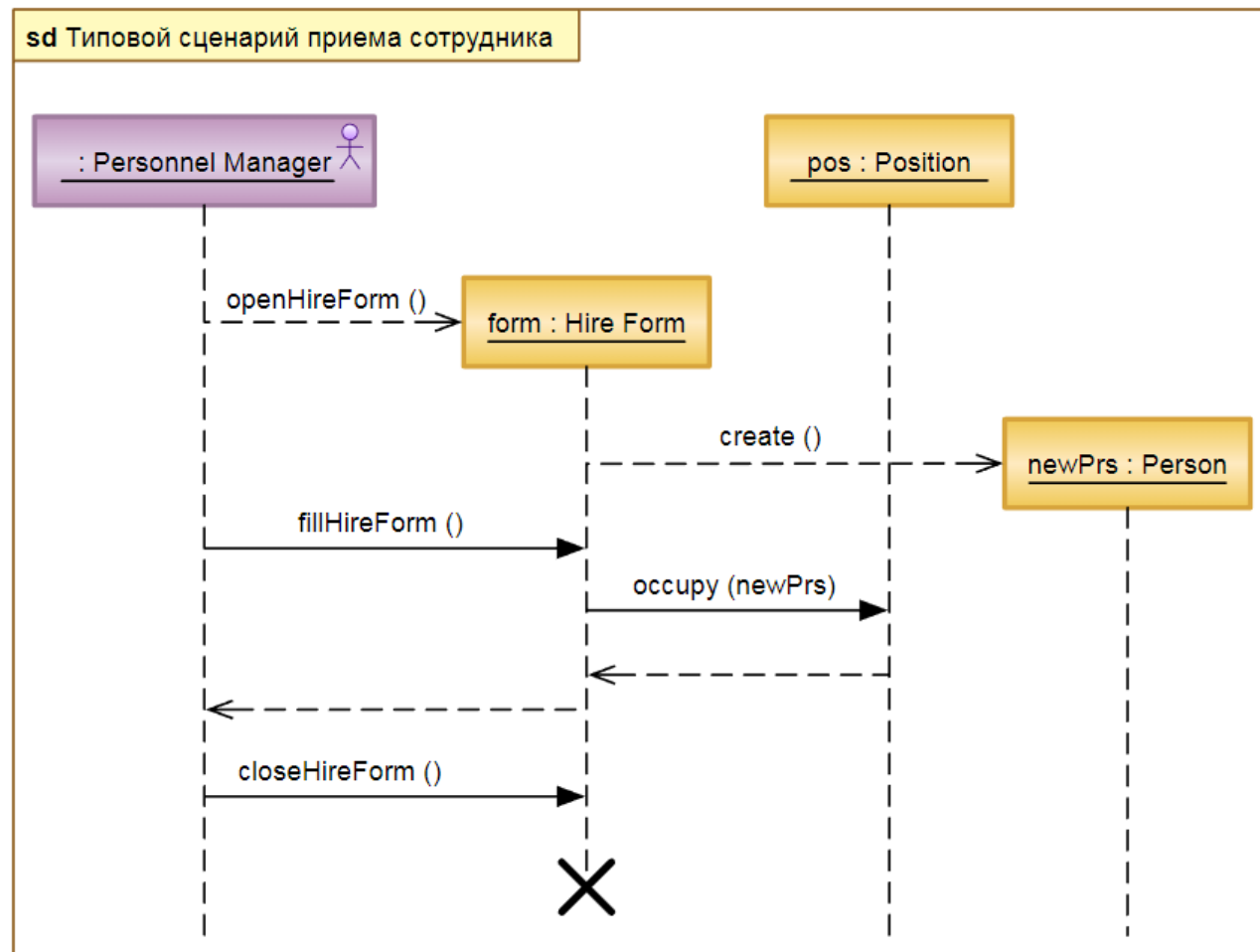


Диаграмма коммуникации

- Предыдущая диаграмма последовательности семантически не полна: она не отражает **все сценарии** варианта использования, которые мы выявили.
- В этом случае можно составить дополнительные диаграммы взаимодействия, реализующие **альтернативные сценарии** варианта использования.
- Здесь показан сценарий приема сотрудника, соответствующий **исключительной ситуации**, когда нет вакантных должностей (в виде диаграммы коммуникации).
- Построение этой диаграммы выявило необходимость включения в модель еще одного класса – Exceptions Handler

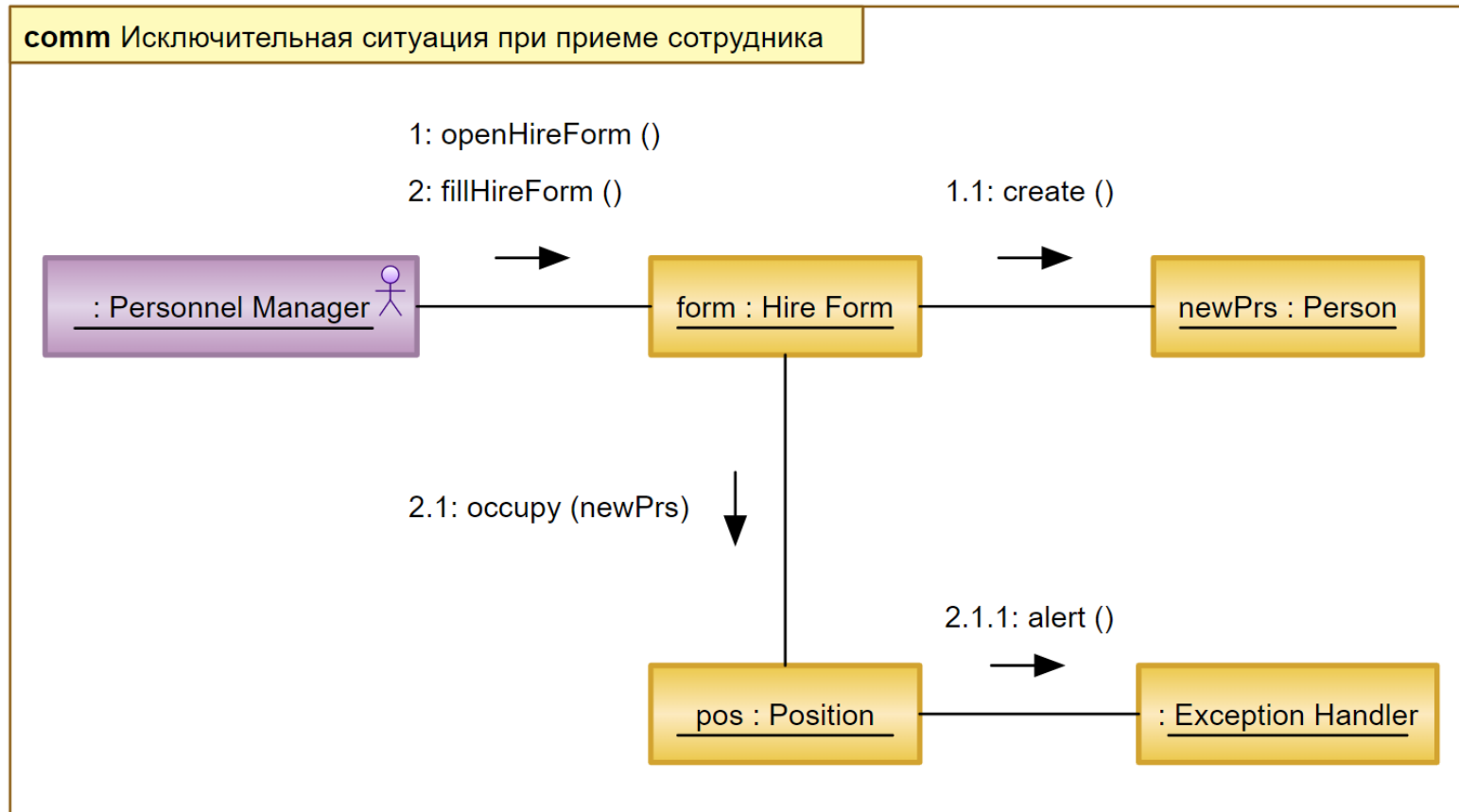
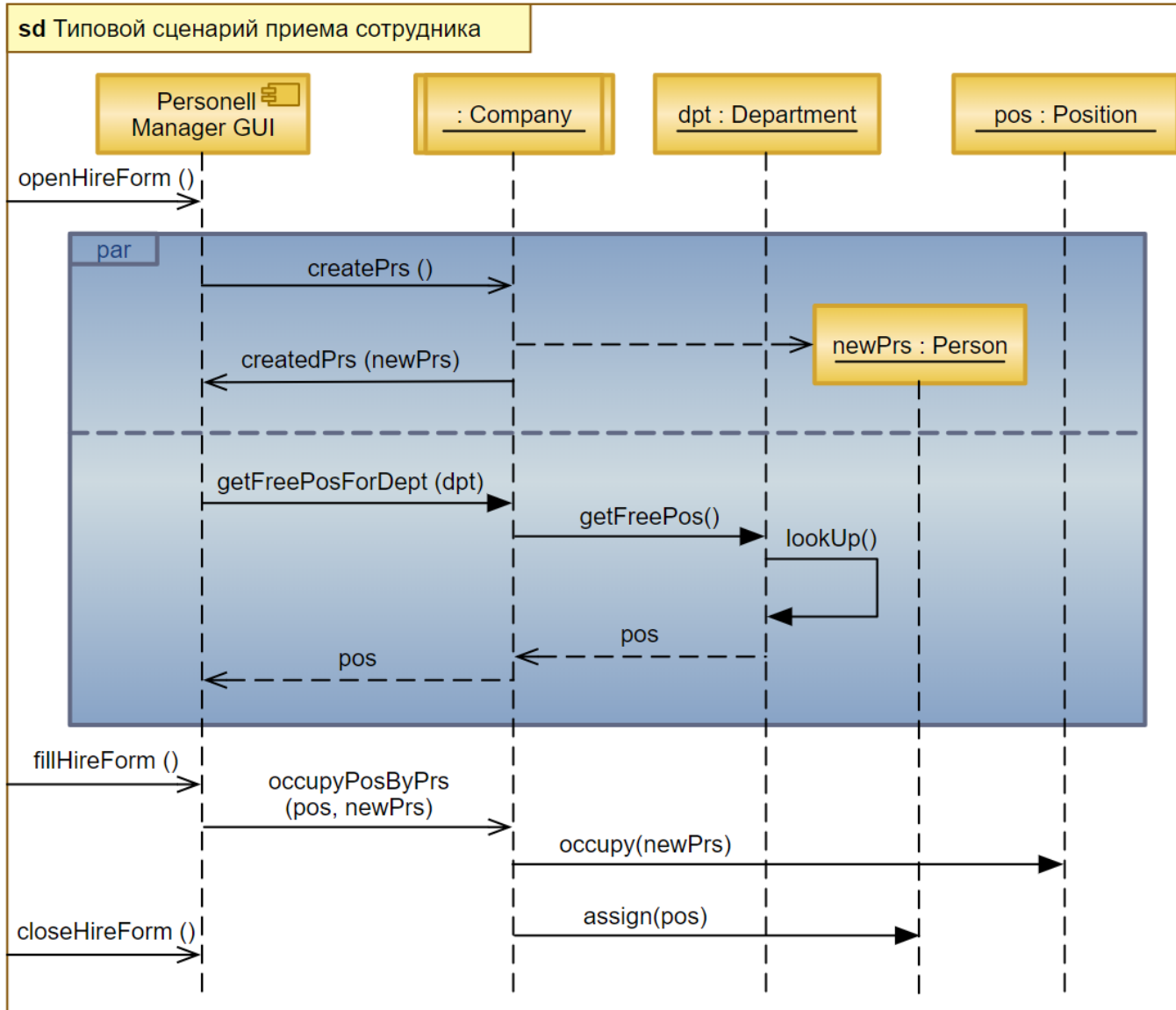


Диаграмма взаимодействия (разделение модели и представления)



Итого

- Модель UML отвечает на вопросы:
 - **ДЛЯ ЧЕГО?**
 - Набор **диаграмм вариантов использования**, описывающих **использование** системы действующими лицами
 - **ЧТО?**
 - **Диаграммы классов** (и, возможно, частные **диаграммы объектов** в определенных сценариях вариантов использования), описывающие **устройство системы**.
 - На более поздних стадиях – **диаграммы компонентов и развертывания**.
 - **КАК?**
 - **Диаграммы деятельности, коммуникаций, последовательности**, возможно – **автоматные и синхронизации** в специфических случаях, описывающие **поведение системы** при реализации вариантов использования.
- Разработка модели – процесс **итеративный**, и продолжается до тех пор, пока мы не поймем разрабатываемую систему **достаточно**, чтобы переходить к **реализации (кодированию)**.
- Тонкая грань между **анализом** и **проектированием** – анализ касается моделирования **предметной области**, проектирование – **реализации**, используя выбранный набор инструментов.