

# Зачем это нужно?

- **Диаграммы деятельности**  
= блок схемы
- **Диаграммы состояний**  
= конечные автоматы
- **Диаграммы классов**  
= код в рамочке
- **Ассоциации**  
= диаграммы «сущность – связь»
- **Диаграммы размещения**  
= топология сети
- ... и так далее
- **Диаграммы использования**  
(Use case) = ... ???

# Пример ТЗ: Отдел кадров

- ИС «Отдел кадров» предназначена для ввода, хранения и обработки информации о сотрудниках и движении кадров
- Прием, перевод и увольнение сотрудников
- Создание и ликвидация подразделений
- Создание вакансий и сокращение должностей

# Подходы к проектированию

- **Top-down: система – подсистемы – модули ...**
  - Структура соответствует команде, а не задаче
- **БД: схема = таблицы + связи**
  - Табельный номер – атрибут сотрудника
- **ОО: словарь системы = классы**
  - Полнота и адекватность словаря

# **Недостатки традиционных ПОДХОДОВ**

- **Первый шаг выполняется в терминах проектируемой системы**
- **Только одна структура выбирается за основу:**
  - **Структурное проектирование – структура кода**
  - **Моделирование данных – структура хранения**
  - **Объектно-ориентированный подход – структура межмодульных интерфейсов**

# **Преимущества моделирования использования**

- **Простые утверждения**
  - **Субъекты, предикаты (и объекты)**
- **Абстрагирование от реализации**
  - **ЧТО** делает система (но не **КАК** это делается и не **ЗАЧЕМ** это делать)
- **Декларативное**
  - **но не императивное описание**
- **Выявление границ**
  - **но не черный ящик**

# **Зачем это нужно – вывод**

- **Традиционные подходы достаточны**
- **Опытный архитектор может с успехом использовать любой подход**
- **Моделирование использования позволяет неопытному архитектору совершать меньше грубых ошибок на ранних этапах проектирования**

# Диаграммы использования

- **Элементы и нотация**
- **Действующие лица и их идентификация**
- **Варианты использования и их идентификация**
- **Отношения элементов диаграмм использования**

# Элементы диаграмм использования

## ■ Сущности

- Действующие лица
- Варианты использования
- Примечания
- Пакеты

## ■ Отношения

- Ассоциации между действующими лицами и вариантами использования
- Обобщения между действующими лицами
- Обобщения и зависимости между вариантами использования



# Нотация

Действующее лицо

Границы системы

System



Association1

UseCase1

Имя ассоциации

Вариант использования



Association2

UseCase2

Имя действующего лица

Ассоциация

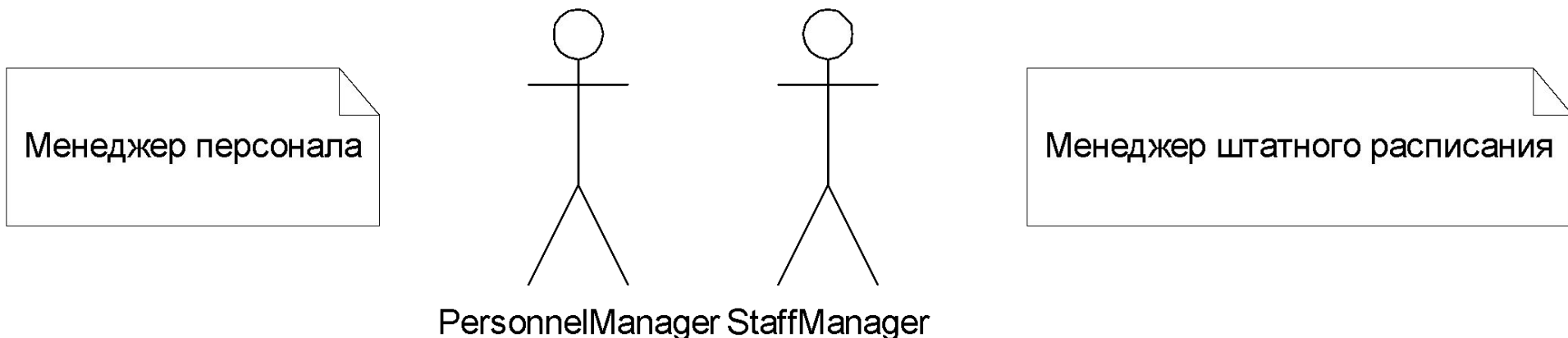


# **Действующие лица и их идентификация**

- **Действующие лица находятся ВНЕ проектируемой системы**
- **Действующее лицо – это множество логически взаимосвязанных РОЛЕЙ**
- **Действующее лицо – это стереотипный КЛАСС**
- **Типовые случаи: категории пользователей, внешние программные и аппаратные средства**

# Пример: действующие лица ИС ОК

- **Менеджер персонала**
  - Работает с конкретными людьми
- **Менеджер штатного расписания**
  - Работает с абстрактными должностями и подразделениями



# **Варианты использования и их идентификация**

- **Вариант использования – множество возможных последовательностей событий/действий (сценариев), приводящих к значимому для действующего лица результату**
- **Типичные случаи: пункты ТЗ**
- **Если ТЗ смутное, его можно (и нужно!) попробовать переписать фразами субъект – предикат – объект**

# Пример: варианты использования ИС ОК

- Менеджер персонала выполняет действия
- Прием сотрудника
- Перевод сотрудника
- Увольнение сотрудника
- Менеджер штатного расписания выполняет действия
- Создание подразделения
- Ликвидация подразделения
- Создание вакансии (=должности)
- Сокращение должности

# Пример: варианты использования ИС ОК

прием сотрудника

HirePerson

CreateDepartment

создание подразделения

перевод сотрудника

MovePerson

DeleteDepartment

ликвидация подразделения

увольнение сотрудника

FirePerson

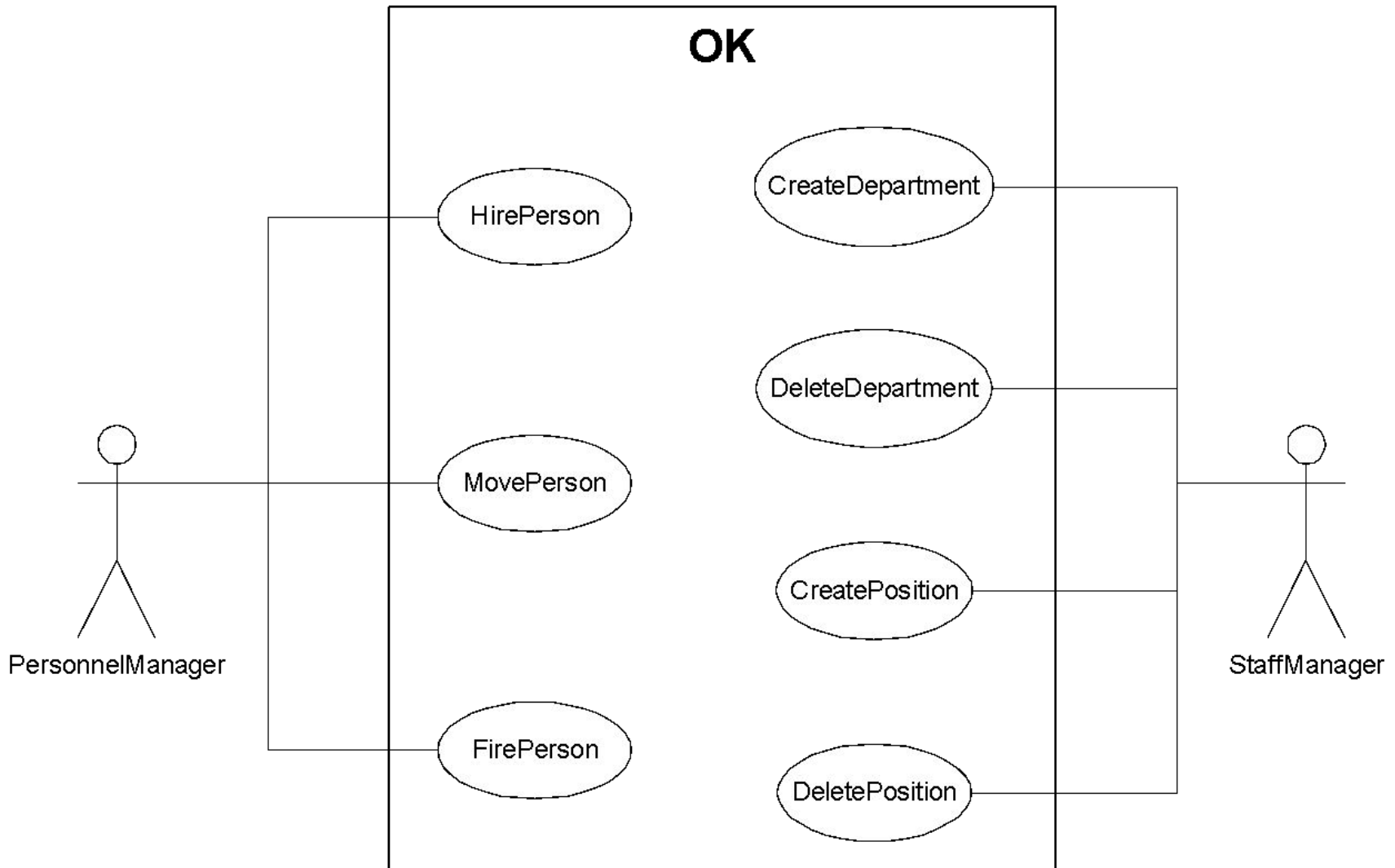
CreatePosition

создание вакансии

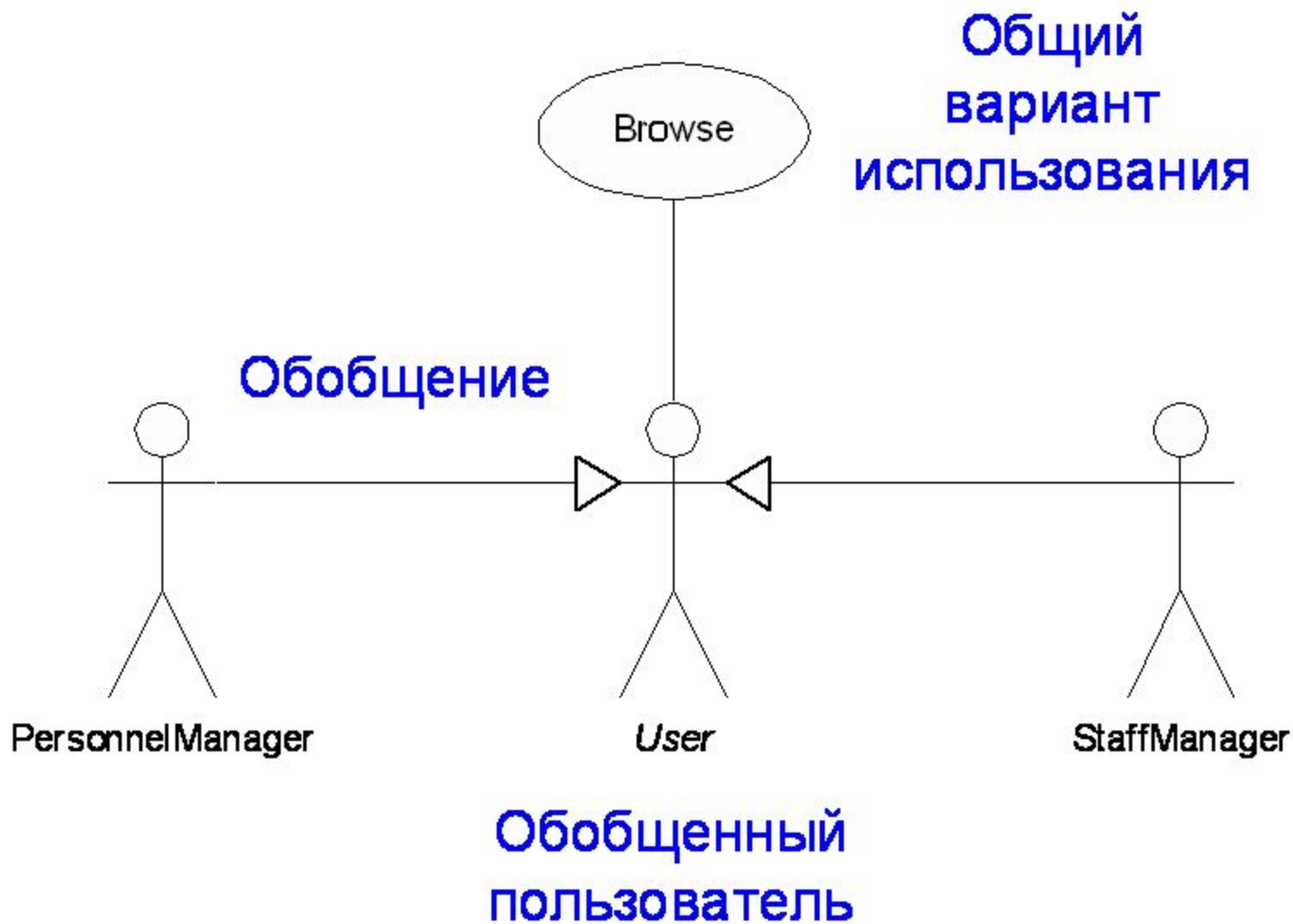
DeletePosition

сокращение должности

# Ассоциации между действующими лицами и вариантами использования

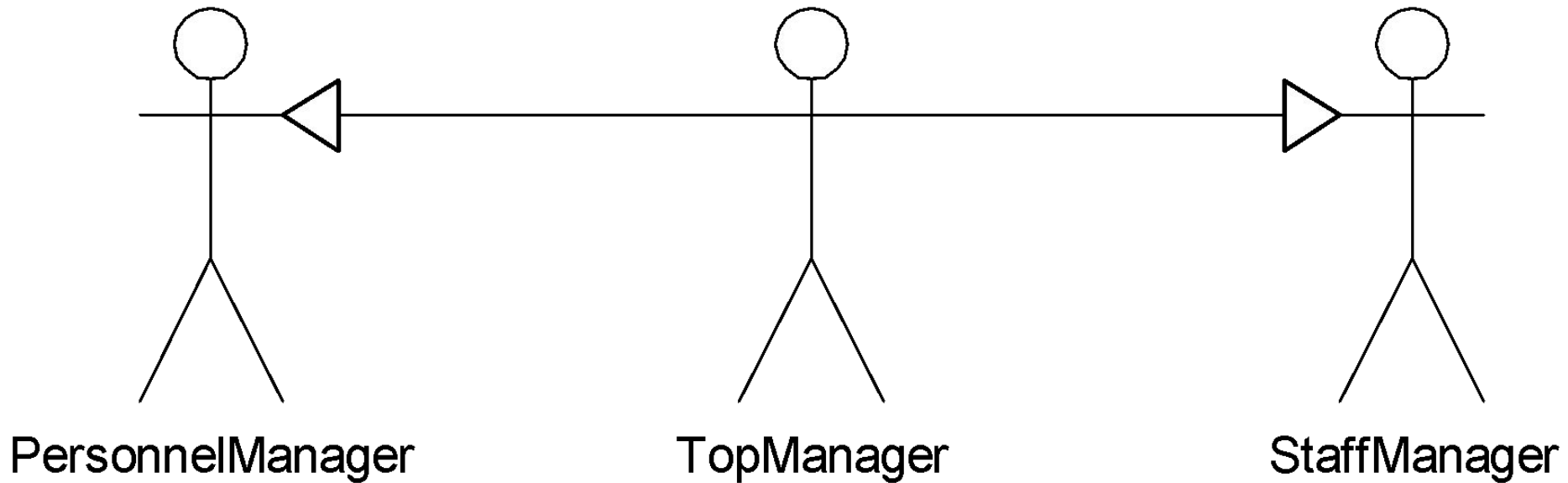


# Обобщение вариантов использования (1)

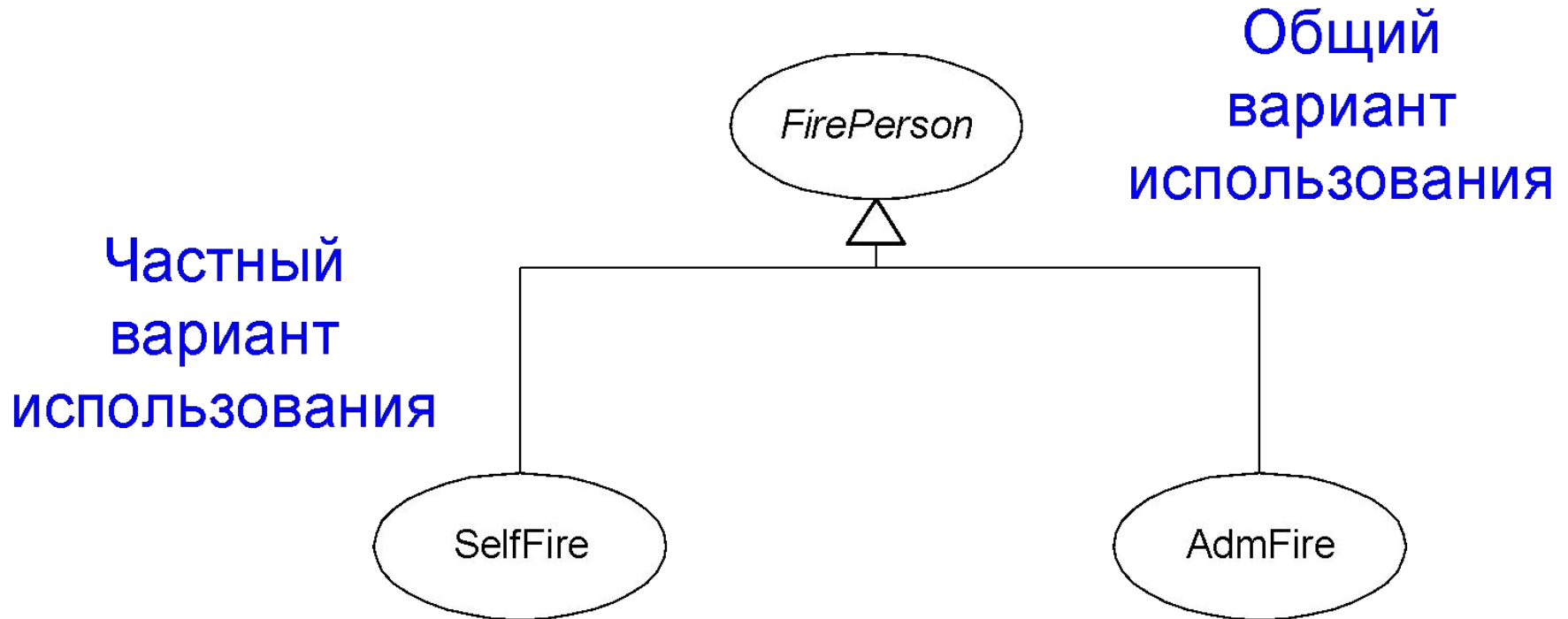




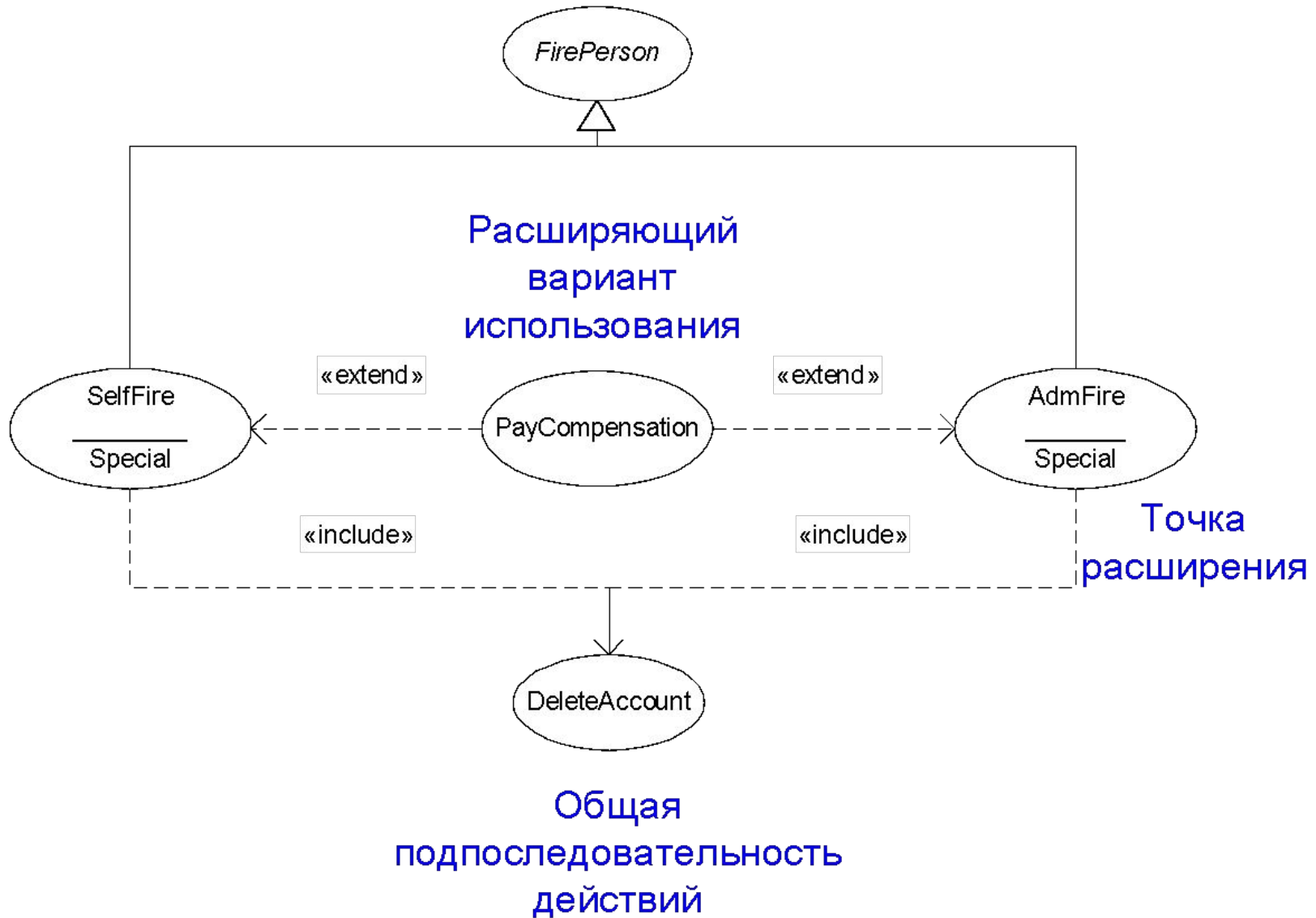
# Обобщение вариантов использования (2)



# Обобщение вариантов использования



# Зависимости между вариантами использования



# Текстовые описания

- 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 штатного расписания обновляет базу данных

# Программы на псевдокоде

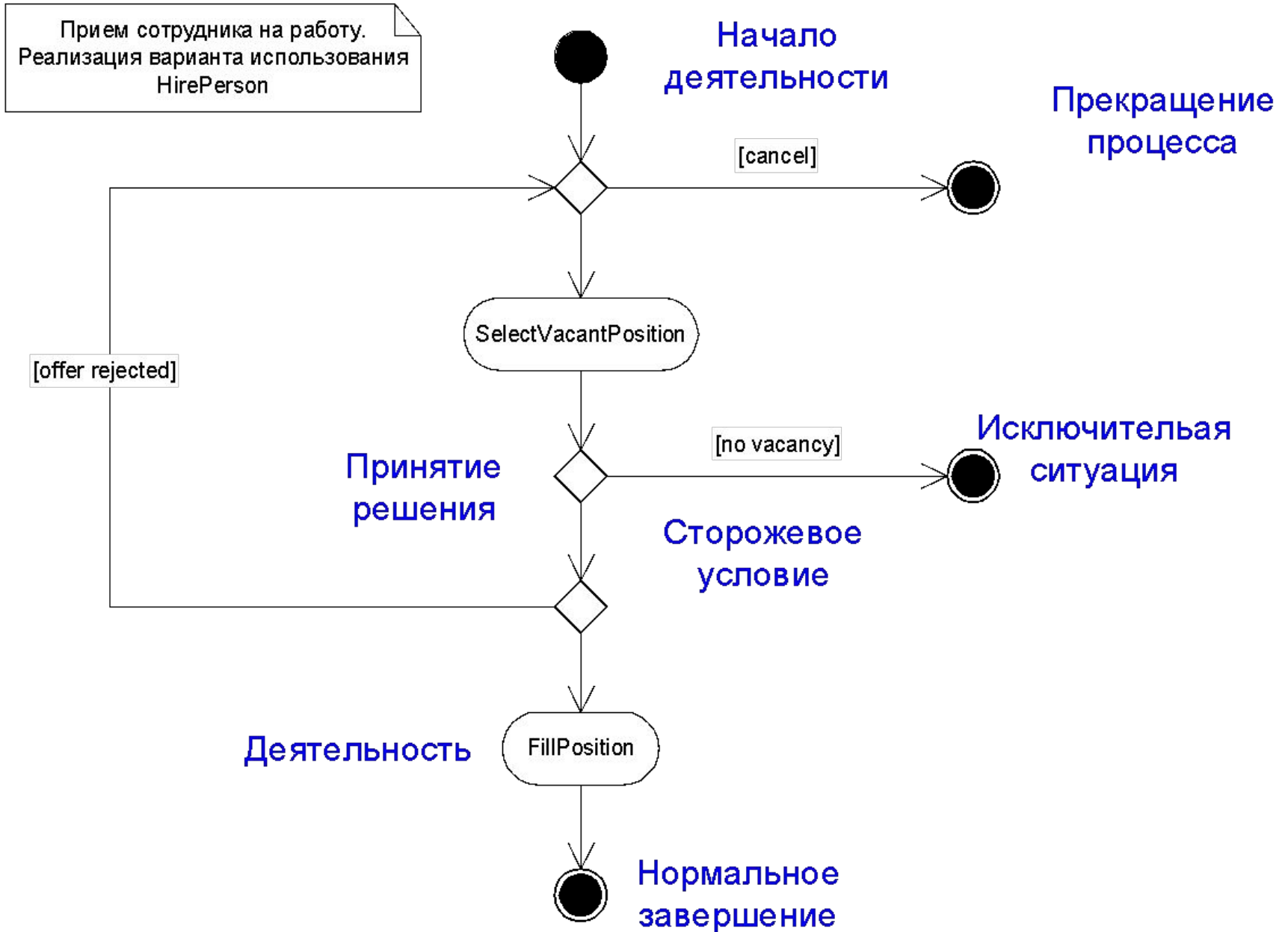
## ■ SelfFire

- Получить заявление
- Special:
- Рассчитать сотрудника
- include DeleteAccount
- Обновить информацию в базе данных

## ■ AdmFire

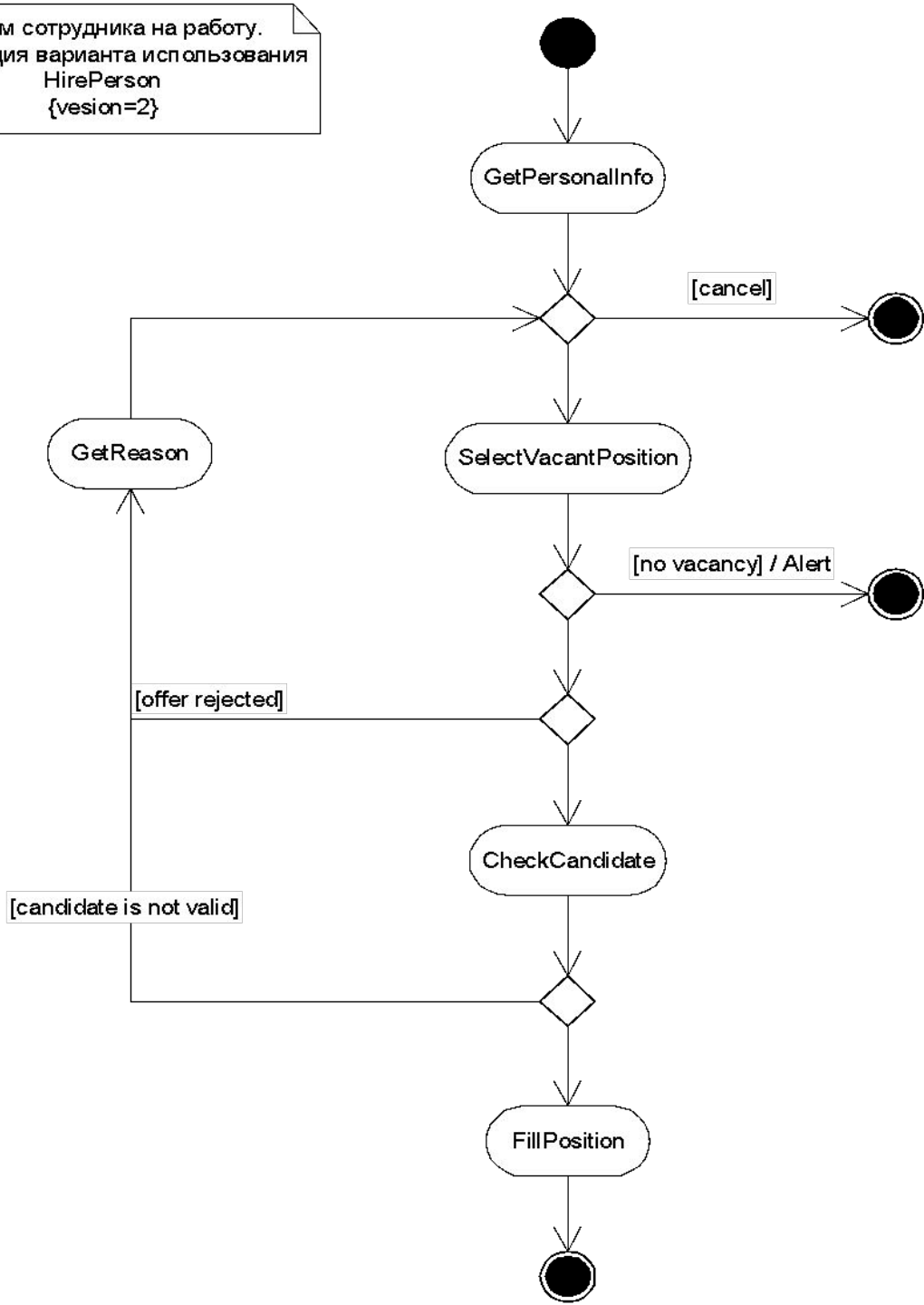
- Получить приказ
- Special:
- Рассчитать сотрудника
- include DeleteAccount
- Обновить информацию в базе данных

# Реализация диаграммами деятельности

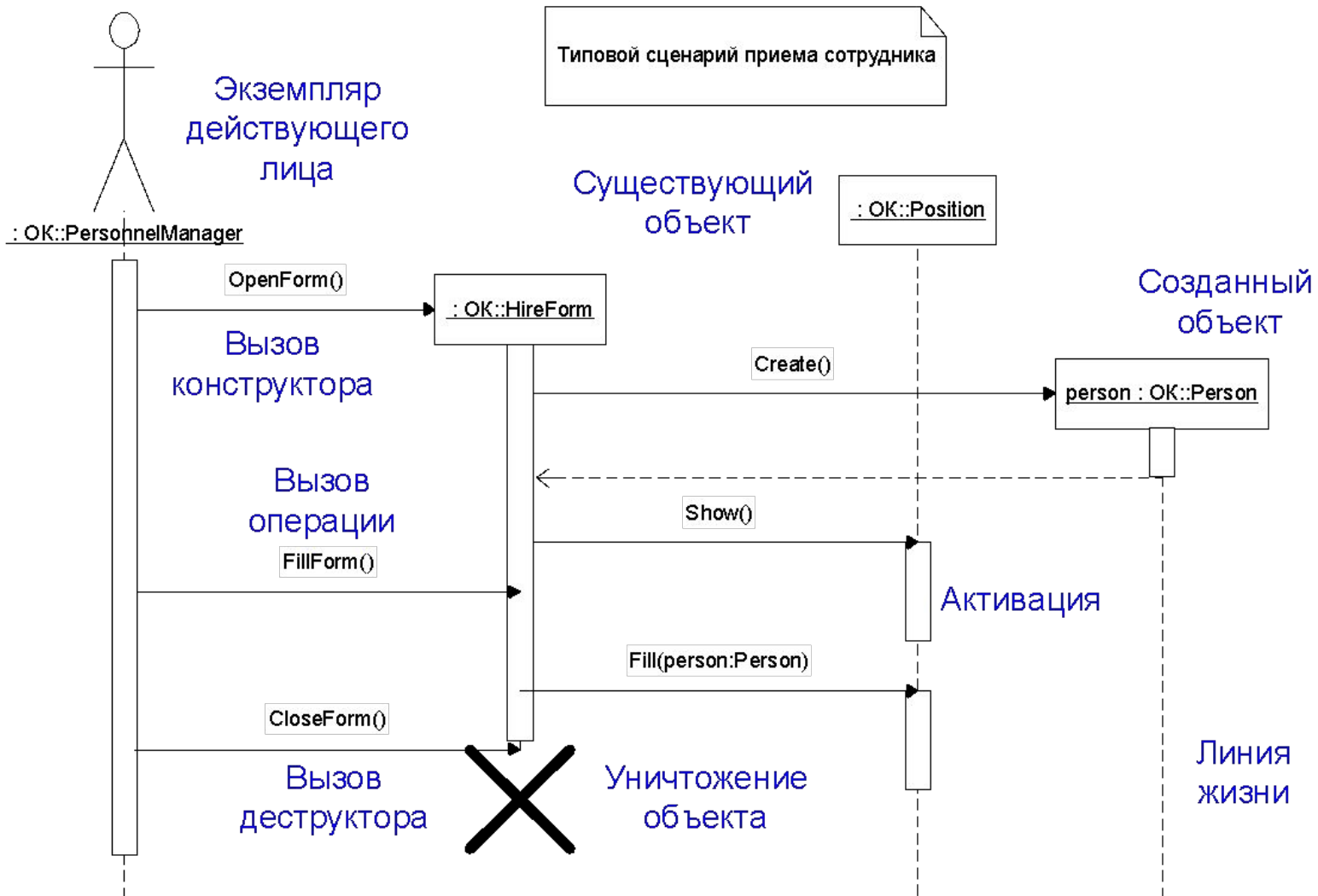


# Усовершенствование реализации

Прием сотрудника на работу.  
Реализация варианта использования  
HirePerson  
{vesion=2}



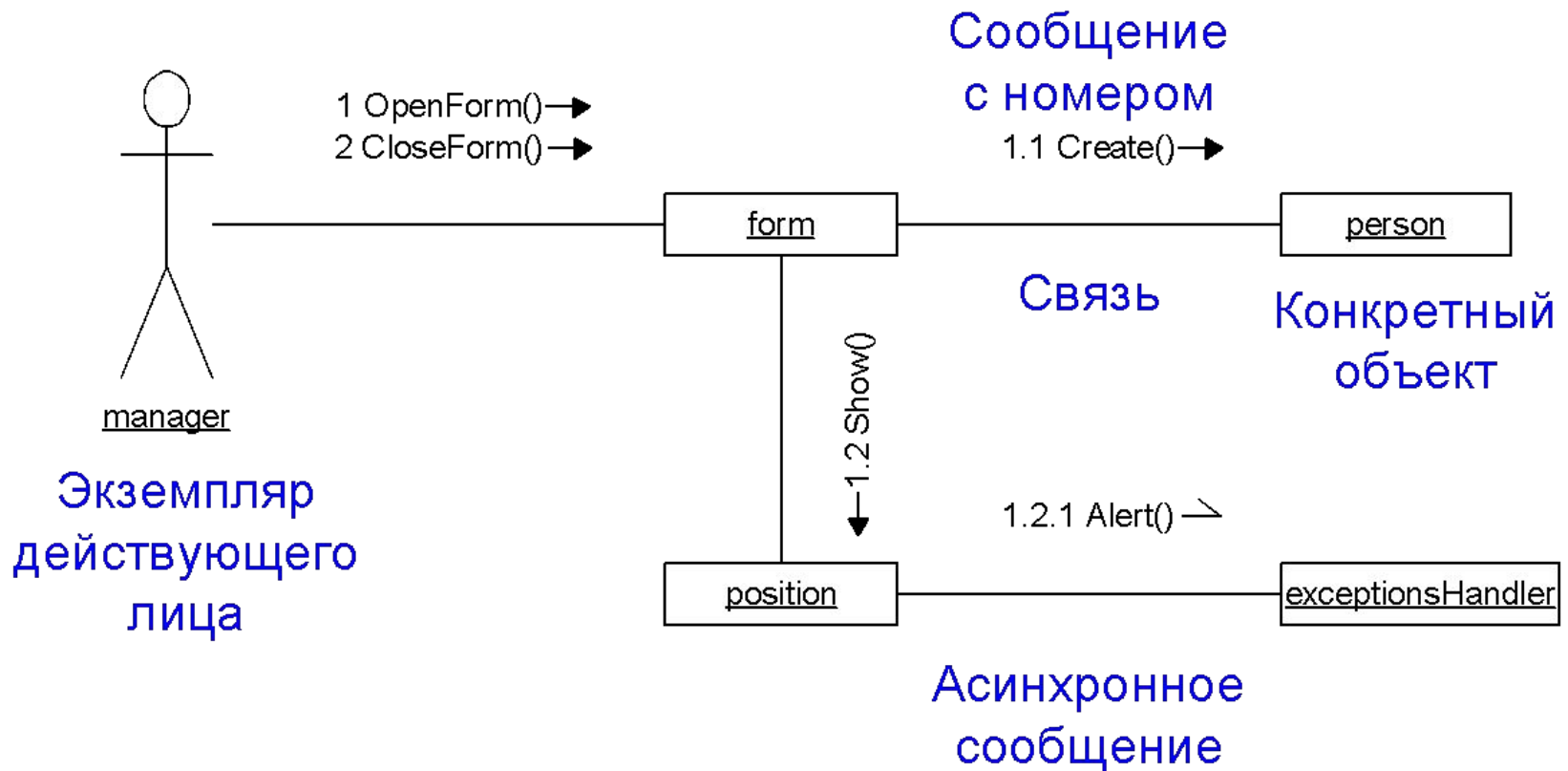
# Реализация диаграммами последовательности





# Реализация диаграммами кооперации

Исключительная ситуация при приеме сотрудника



# Сравнение способов реализации вариантов использования (1)

- **Текстовые описания**
  - Всем понятно, привычно и удобно
  - Длинно и неточно, пропуски и ошибки
  - Есть трансляторы в варианты использования (!)
- **Программы на псевдокоде**
  - Традиционное средство программистов
  - Компактнее текстового описания
  - Навязывают структуру реализации
  - Не приближает к объектной модели

# Сравнение способов реализации вариантов использования (2)

- **Диаграммы деятельности**
  - Псевдокод эквивалентен блок-схемам (с точностью до параллелизма)
  - Наглядно, но менее компактно
  - Почти не приближают к объектной модели
- **Диаграммы взаимодействия**
  - Сложная и непривычная нотация
  - Диаграммы объектного уровня – описывают ОДИН сценарий – нужно МНОГО диаграмм
  - Прямо ведут к объектной модели

# Выводы

- **Диаграмма использования – первый шаг моделирования**
- **Основное назначение – показать, что делает система во внешнем мире**
- **Не обязательно соответствует структуре классов, модулей и компонентов**
- **Адекватная идентификация действующих лиц и вариантов использования – ключ к успеху**
- **Способ реализации – дело вкуса**