



Технология программирования

Определение требований к
системе

Виды требований

Функциональные требования описывают желаемое поведение системы

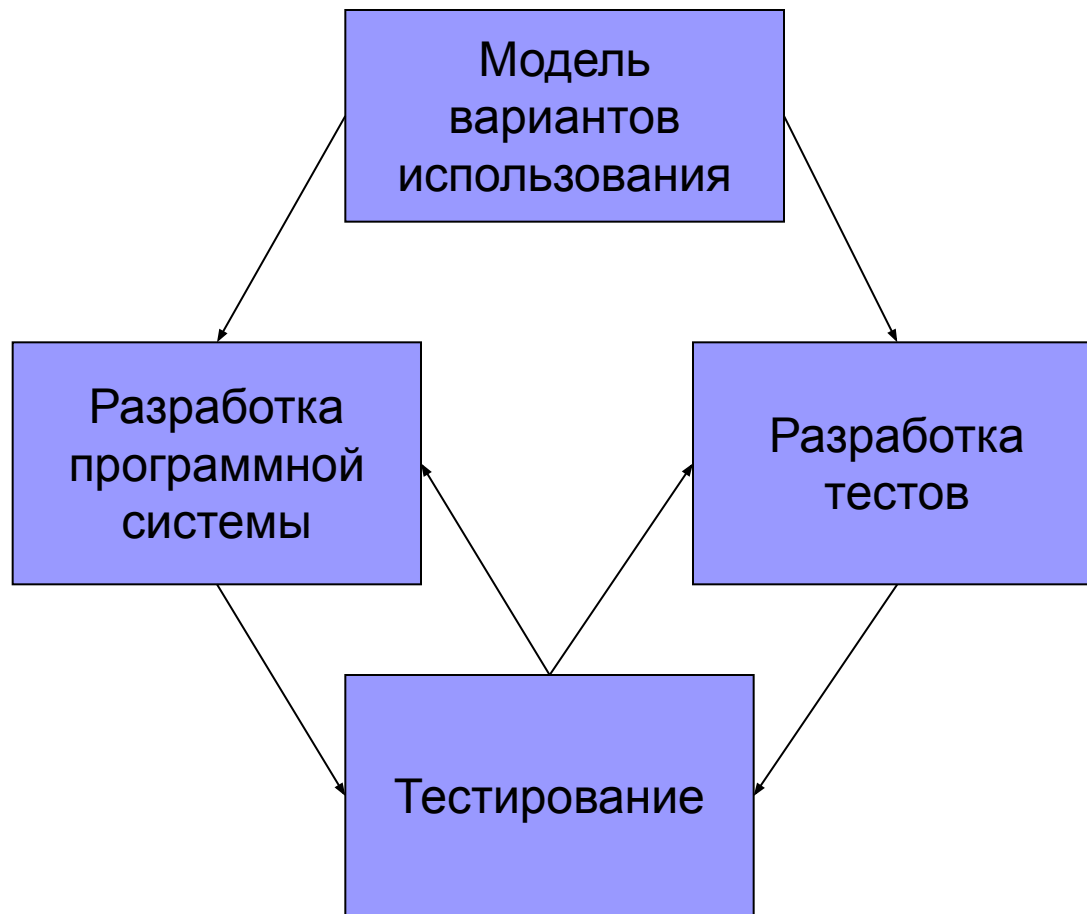
Нефункциональные требования – любые требования, не являющиеся функциональными, например:

- Эргономические требования
- Требования к надежности
- Требования к производительности
- Требования к сопровождению
- Ограничения на проектирование
- Ограничения реализации
- Требования к внешним интерфейсам

Модель вариантов использования

- **Актант** (внесистемный агент, Actor) – внешняя система, взаимодействующая с заданной системой. Актант может быть пользователем, либо технической системой.
- **Вариант использования (Use Case)** – это последовательность действий актанта и реакций системы, приводящая к полезному для актанта результату.
- **Модель вариантов использования** – совокупность функциональных требований к системе, описанных в форме вариантов использования. Она содержит описание актантов, спецификации вариантов использования и ассоциаций между ними. Модель вариантов использования является соглашением между заказчиком и разработчиком ПО.
- **Классификация вариантов использования**
 - По степени использования проектных решений: {идеальный ↔ реальный}
 - По детализации: {краткий ↔ развернутый}
 - По уровню абстракции: {абстрактный ↔ конкретный}

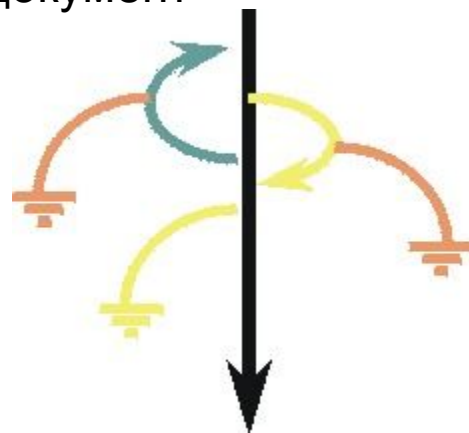
Использование модели вариантов использования в разработке ПО



Спецификация варианта использования

Спецификация варианта использования – это текстовый документ

Название варианта использования	Корректировать учебный материал
Актеры	Преподаватель
Краткое описание	Преподаватель формирует учебный материал по определенной дисциплине
Основной поток событий	Преподаватель выбирает нужный раздел. Система отображает имеющийся материал. Преподаватель вводит учебный материал. Система сохраняет изменения. Преподаватель добавляет контрольные вопросы по введенному материалу. Система сохраняет вопросы.
Альтернативные потоки событий	Преподаватель создает раздел дисциплины, если он отсутствует. Для этого преподаватель указывает название дисциплины и контрольные мероприятия (экзамен, зачет).
Предусловия	Вариант использования начинается, когда преподаватель запрашивает корректировку учебного материала
Постусловия	Материал для ответов на контрольные вопросы содержится в системе.
Точки расширения	Дисциплина выбрана
Особые требования	При вводе текста учебного материала должна контролироваться грамматика языка.

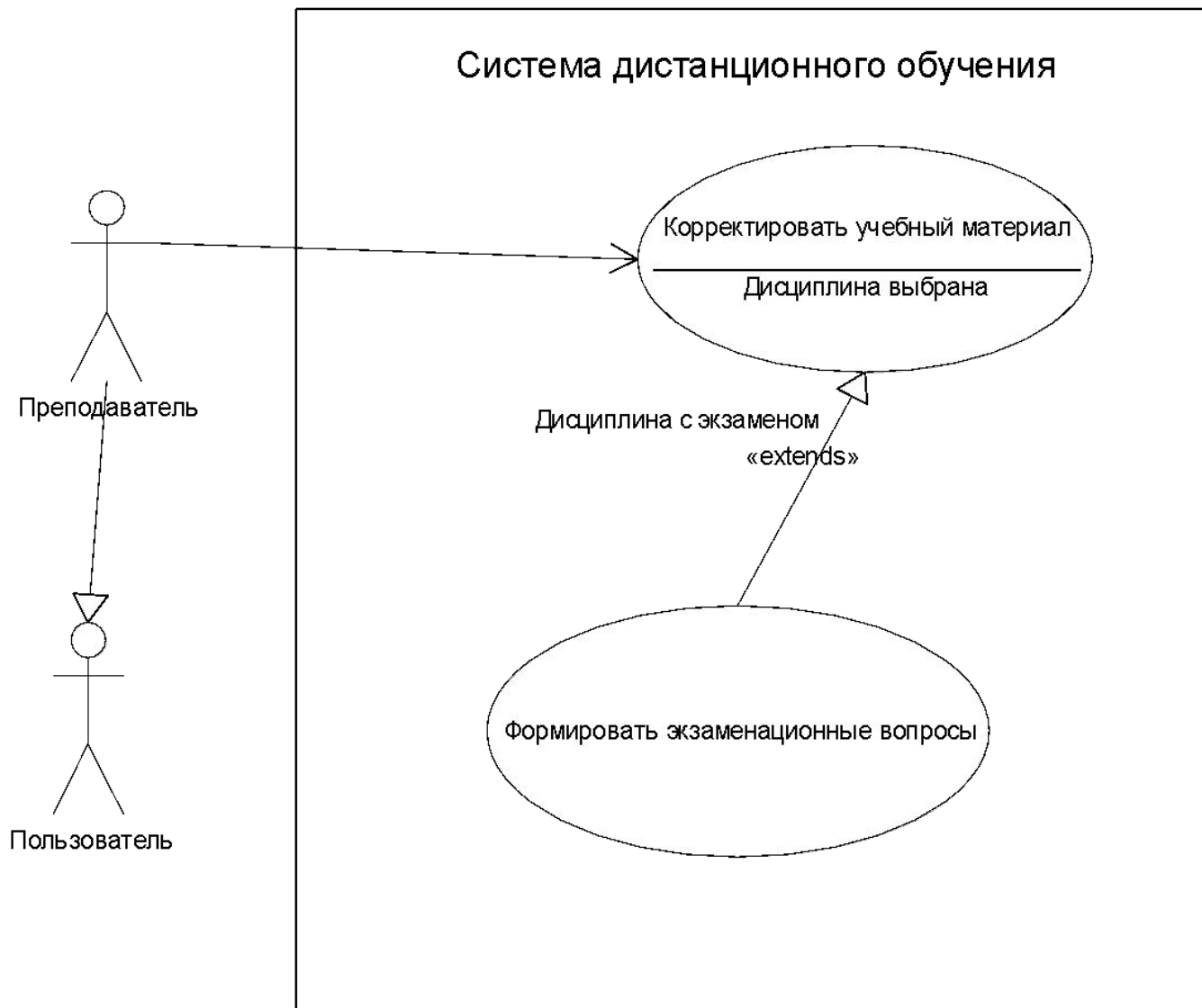


Handwritten red notes and scribbles, possibly indicating a correction or additional information related to the specification.

Спецификация варианта использования

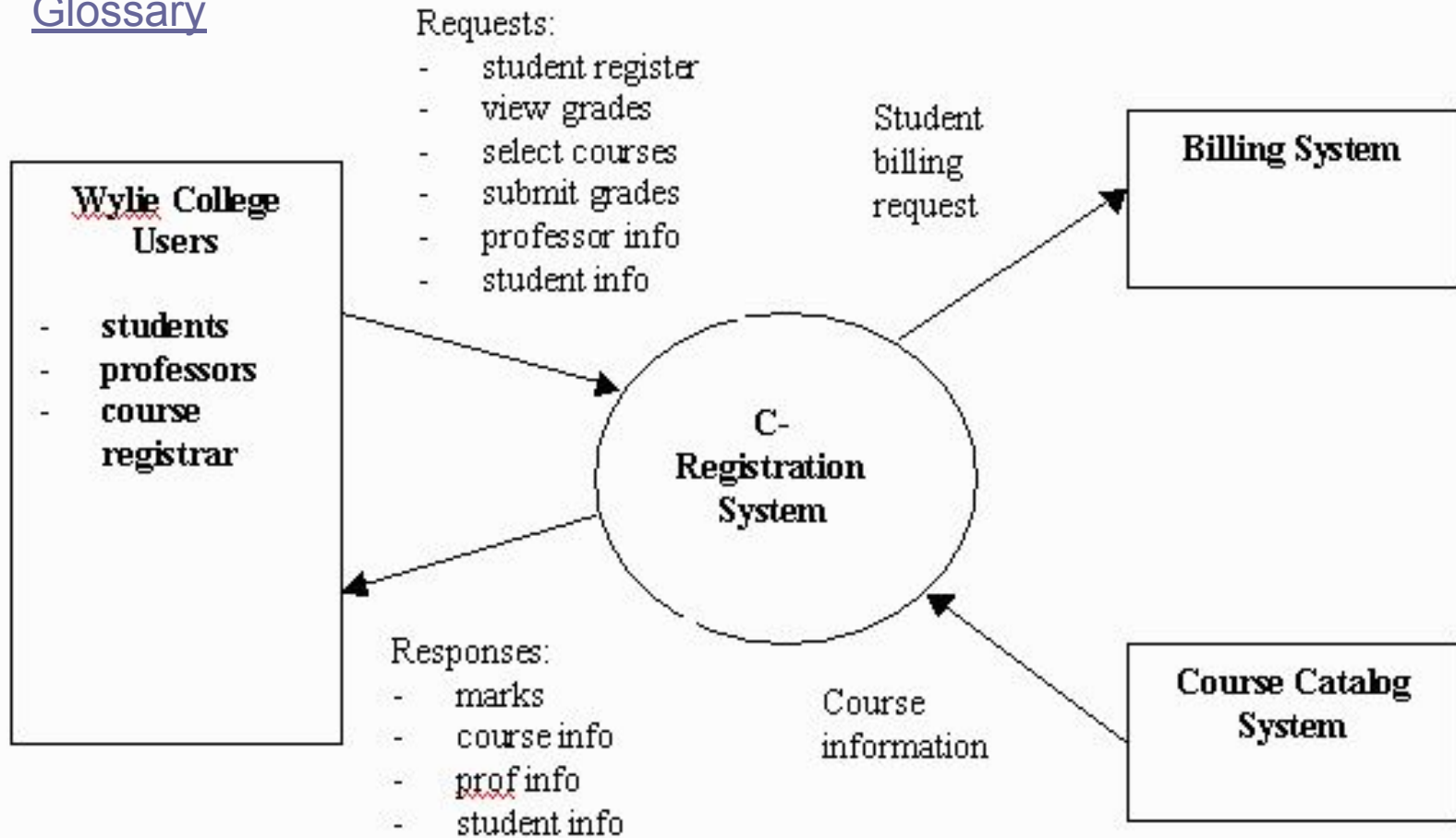
- **Основной поток событий** – типовая и, может быть, самая короткая *последовательность* действий, дающая актанту желаемый результат
- **Альтернативный поток событий** – вспомогательный вариант получения желаемого результата, который используется, если выполнение основного потока не возможно.

Графическая нотация

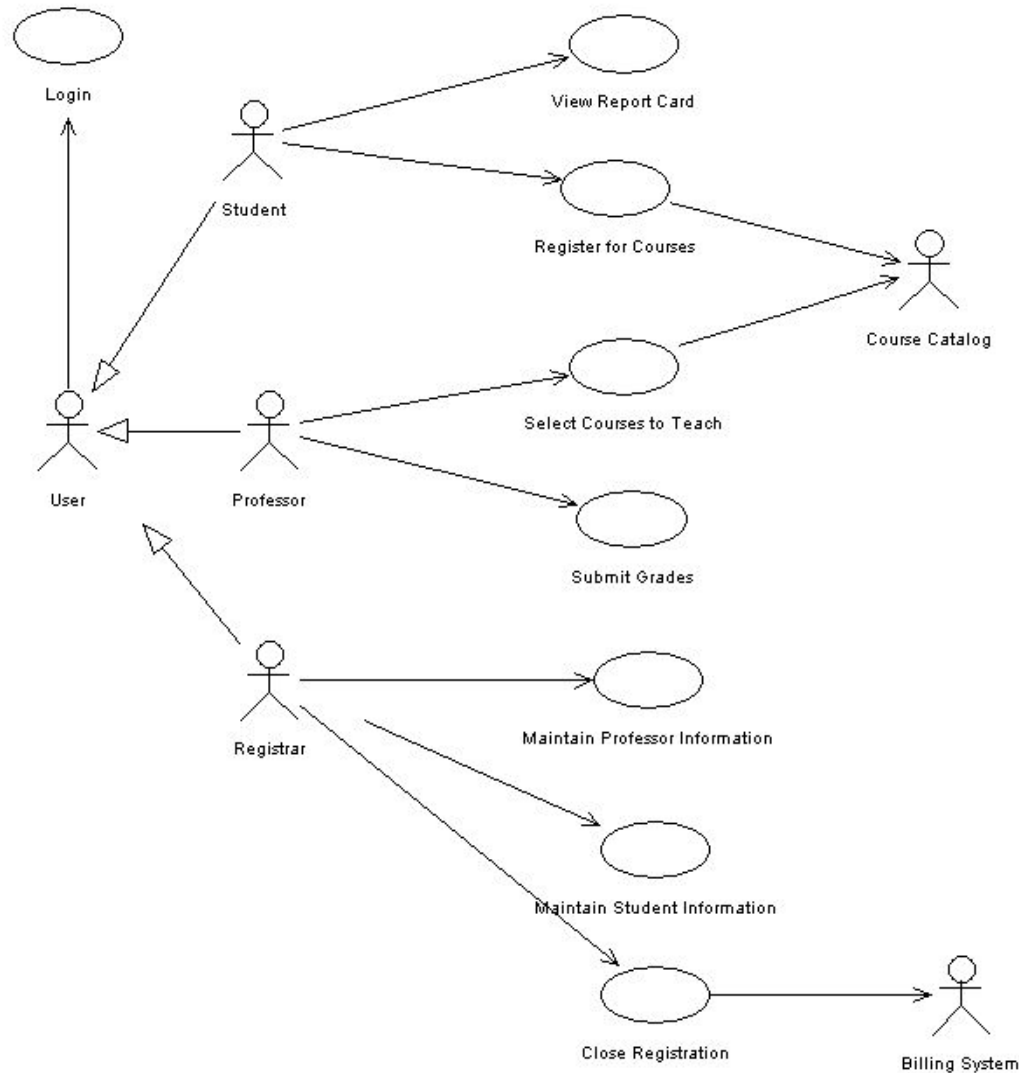


Пример: Wylie Course Registration System

Glossary



Пример: Wylie Course Registration System



[Close Registration](#)

[Login](#)

[Maintain Professor Information](#)

[Maintain Student Information](#)

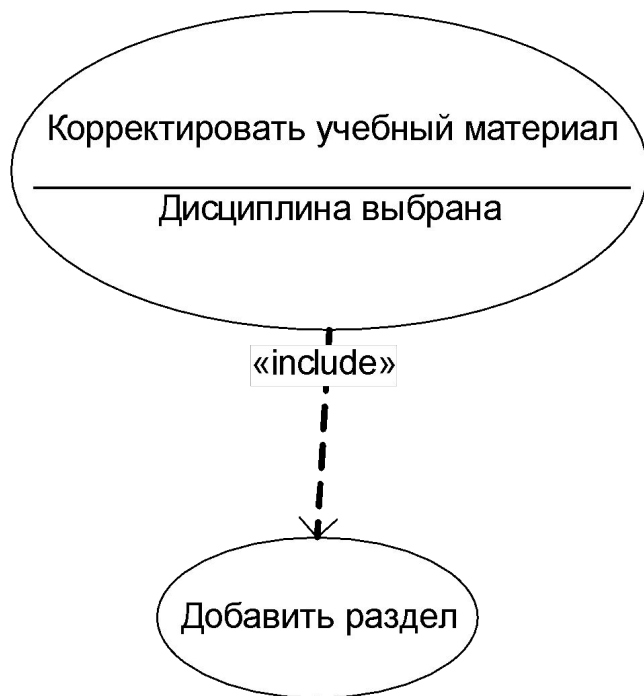
[Register for Courses](#)

[Submit Grades](#)

[Select Courses to Teach](#)

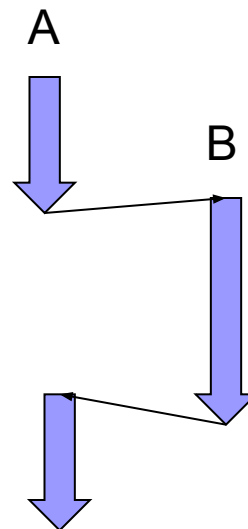
[View Report Cards](#)

Отношения между вариантами ИСПОЛЬЗОВАНИЯ



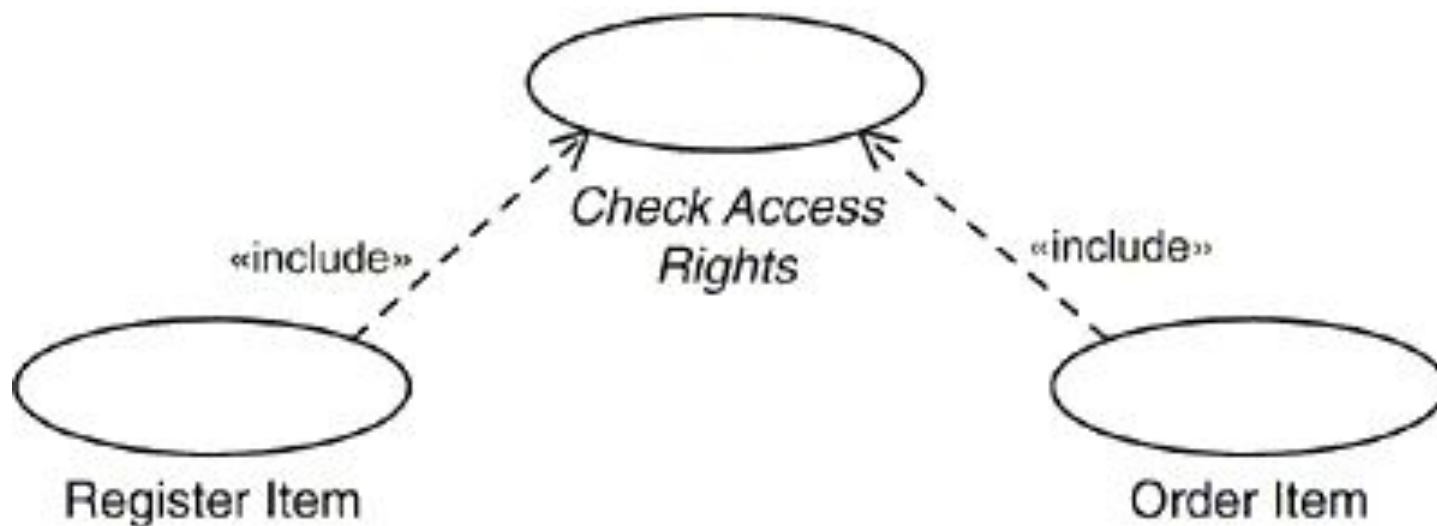
Отношение Include

Последовательность событий включаемого варианта использования становится последовательностью включающего варианта использования (аналогично вызову подпрограмм)



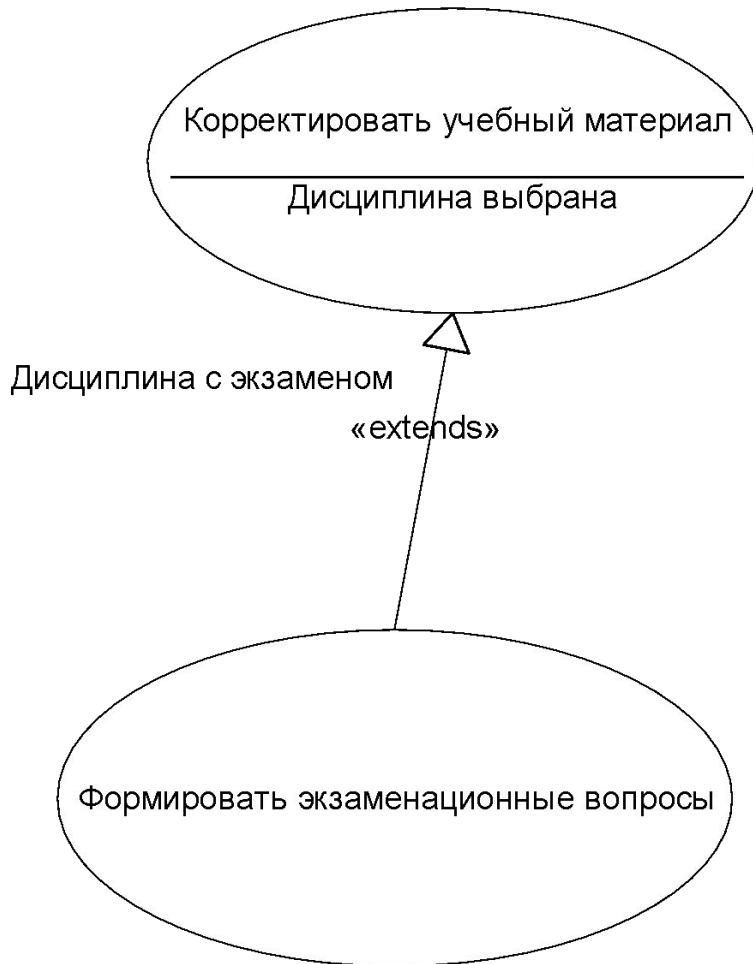
A включает B.
A и B – конкретные варианты
использования

Отношения между вариантами использования



Повторное использование

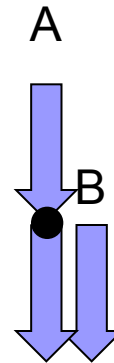
Отношения между вариантами использования



Отношение Extend

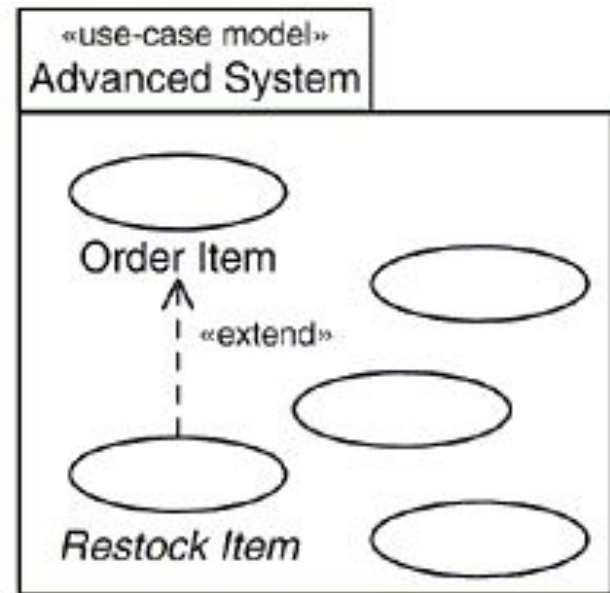
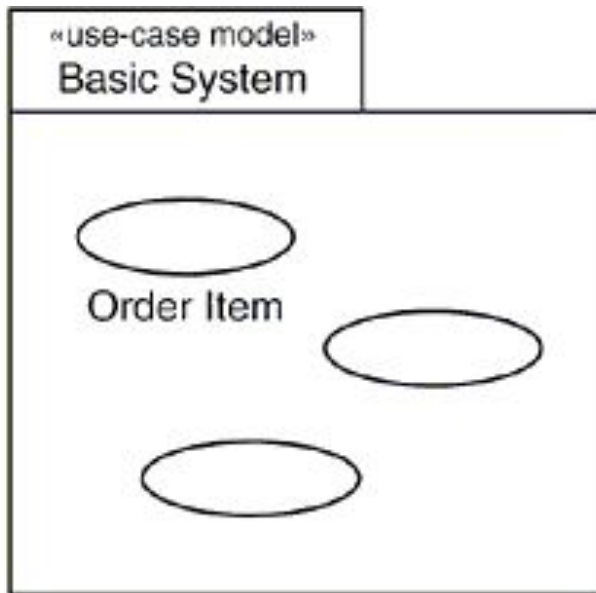
Если выполняется условие расширения, то расширяющий вариант использования становится частью расширяемого, добавляя потоки событий

Точка расширения

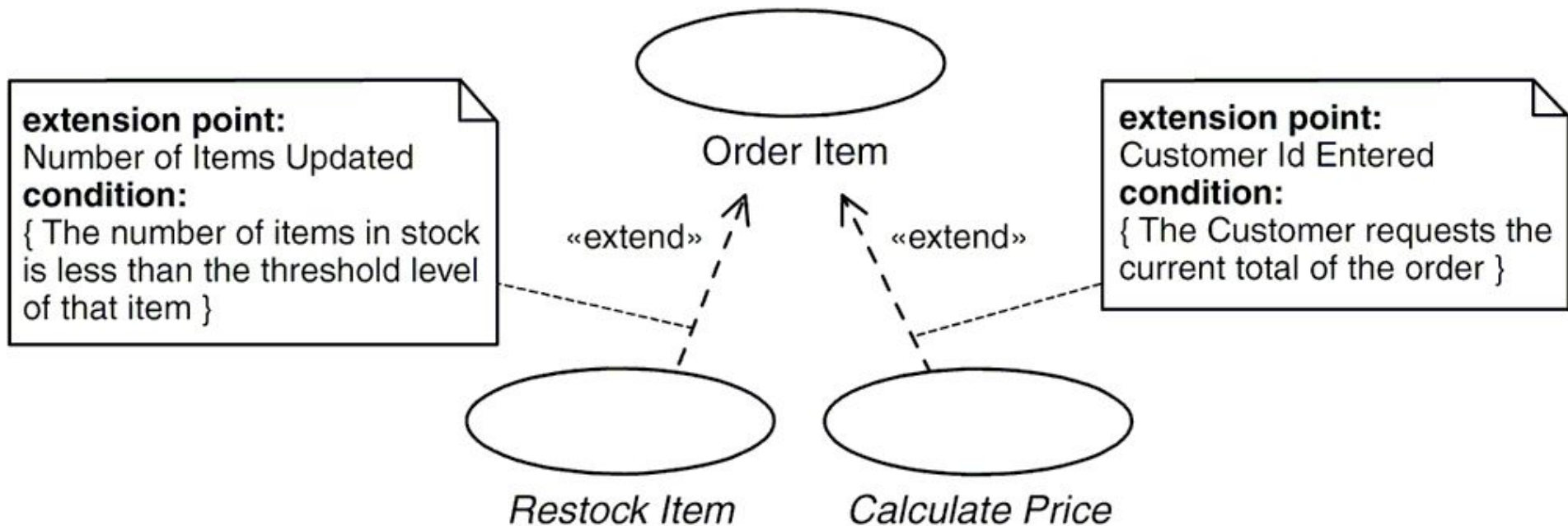


В расширяет А.
В – абстрактный вариант использования, А – как правило, конкретный (абстрактный, если он является расширением другого вариант использования)

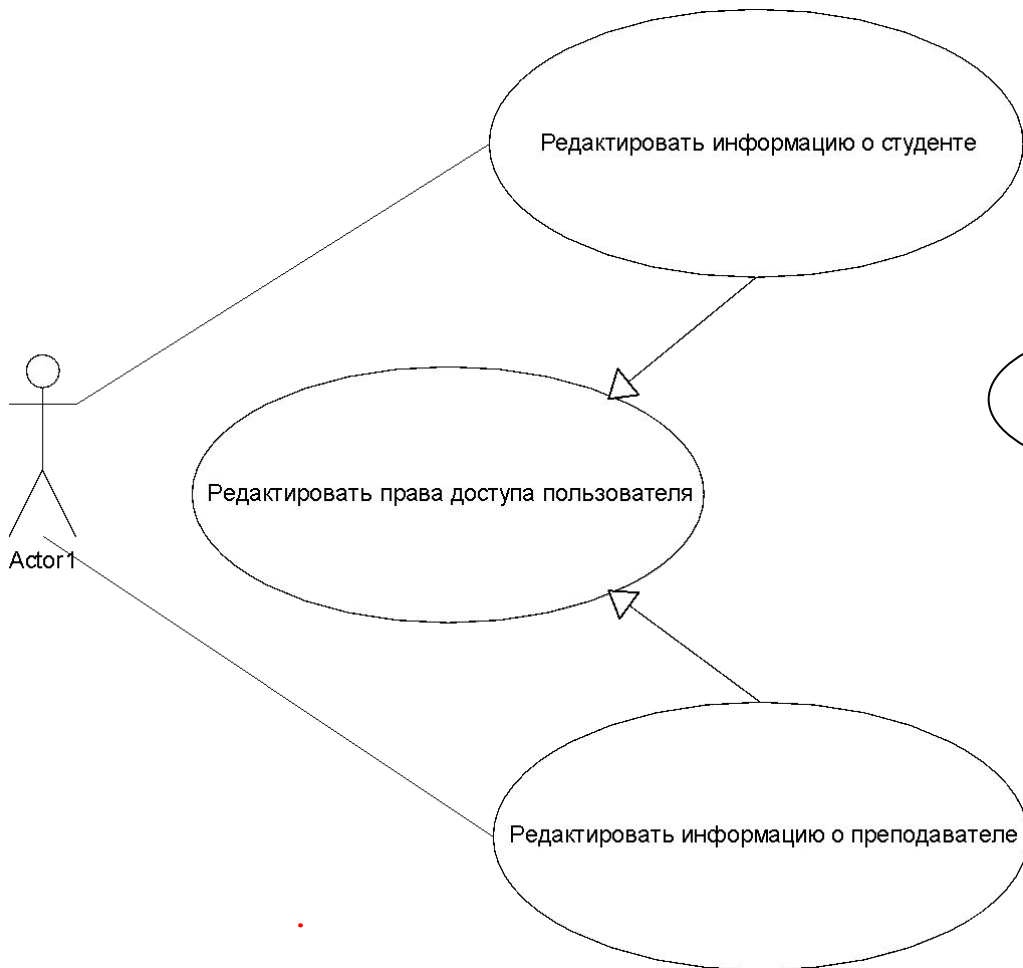
Отношения между вариантами использования



Отношения между вариантами использования

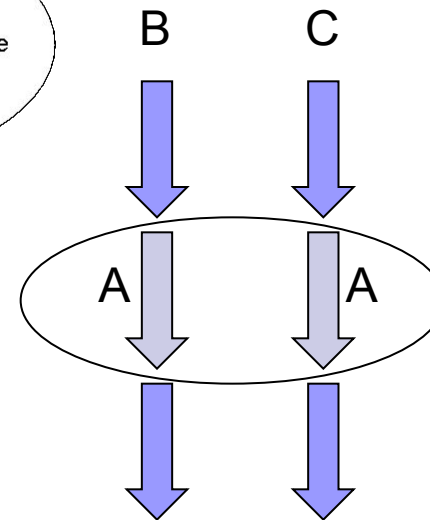


Отношения между вариантами использования



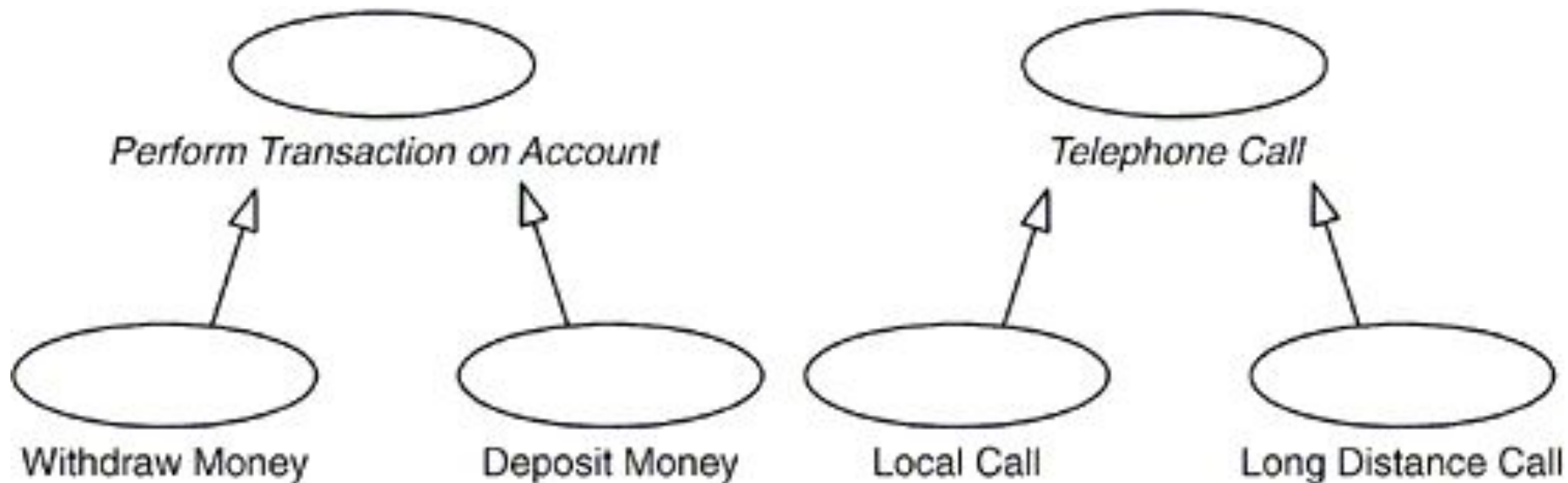
Отношение Обобщения

В абстрактном варианте использования обобщаются одинаковые потоки событий других вариантов использования



А обобщает варианты использования В и С. А – как правило, абстрактный вариант использования.

Отношения между вариантами использования



Отноц вариан

Withdraw Money

Perform Transaction on Account

Basic Flow

START OF THE FLOW

The use case starts as described in the Perform Transaction on Account use case.

PERFORMANCE OF A TRANSACTION

The ATM Customer selects to withdraw money. The use case asks for the amount to be withdrawn.

The ATM Customer enters the amount to be withdrawn. The use case sends a request to the Bank System that the amount is to be withdrawn together with the card information, and logs the communication.

The use case receives a transaction confirmation from the Bank System and logs the reception of the confirmation. The use case logs that the amount is withdrawn together with a timestamp and the card information, prints a receipt, dispenses the cash, and ejects the card.

END OF THE FLOW

The use case ends as described in the Perform Transaction on Account use case.

inserted here

Specialized here

inserted here

Basic Flow

START OF THE FLOW

The use case starts when the ATM Customer inserts the card into the ATM's card reader. The use case reads the card information and requests the PIN code.

The ATM Customer inserts the PIN code. The PIN code is checked according to the declaration in the included use case Check PIN Code. The use case logs the card information and that the entered PIN code is valid.

The use case checks that there is paper to print a receipt on. If not, the ATM Customer is notified. Then, the use case displays the possible kinds of transactions.

PERFORMANCE OF A TRANSACTION

The ATM Customer selects one alternative. The use case performs the selected transaction.

END OF THE FLOW

The use case checks that there is enough cash for another transaction. If not, an error message is displayed, the ATM changes state to Disabled, and a message is sent to the Bank system. Both of these actions are logged. Otherwise, the welcome message is displayed.

The use case ends.

Анализ требований

- Цель: добиться понимания требований в той степени, которая необходима для представления системы в целом, включая архитектуру.
- Результат: объектная модель анализа, начальный вариант архитектуры системы

Анализ требований

Классы модели анализа – определяют ответственности на высоком уровне абстракции (без детализации методов и конкретных атрибутов). Используются следующие стереотипы классов:

- **Граничные классы** – моделируют взаимодействие между системой и актантом (запросы и получение информации). Запросы актантов – системные операции. Для системных операций составляют диаграммы взаимодействия.
- **Классы сущностей** – моделируют информацию, существующую некоторое время. Как правило классы сущностей получаются из классов предметной области
- **Управляющие классы** – моделируют управление другими объектами (граничных классов и классов сущности). Управляющий класс может соответствовать одному или нескольким вариантам использования.

Анализ требований

