

**АНАЛИЗ ТРЕБОВАНИЙ И
ОПРЕДЕЛЕНИЕ
СПЕЦИФИКАЦИЙ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ПРИ ОБЪЕКТНОМ ПОДХОДЕ**

Задачи этапа анализа:

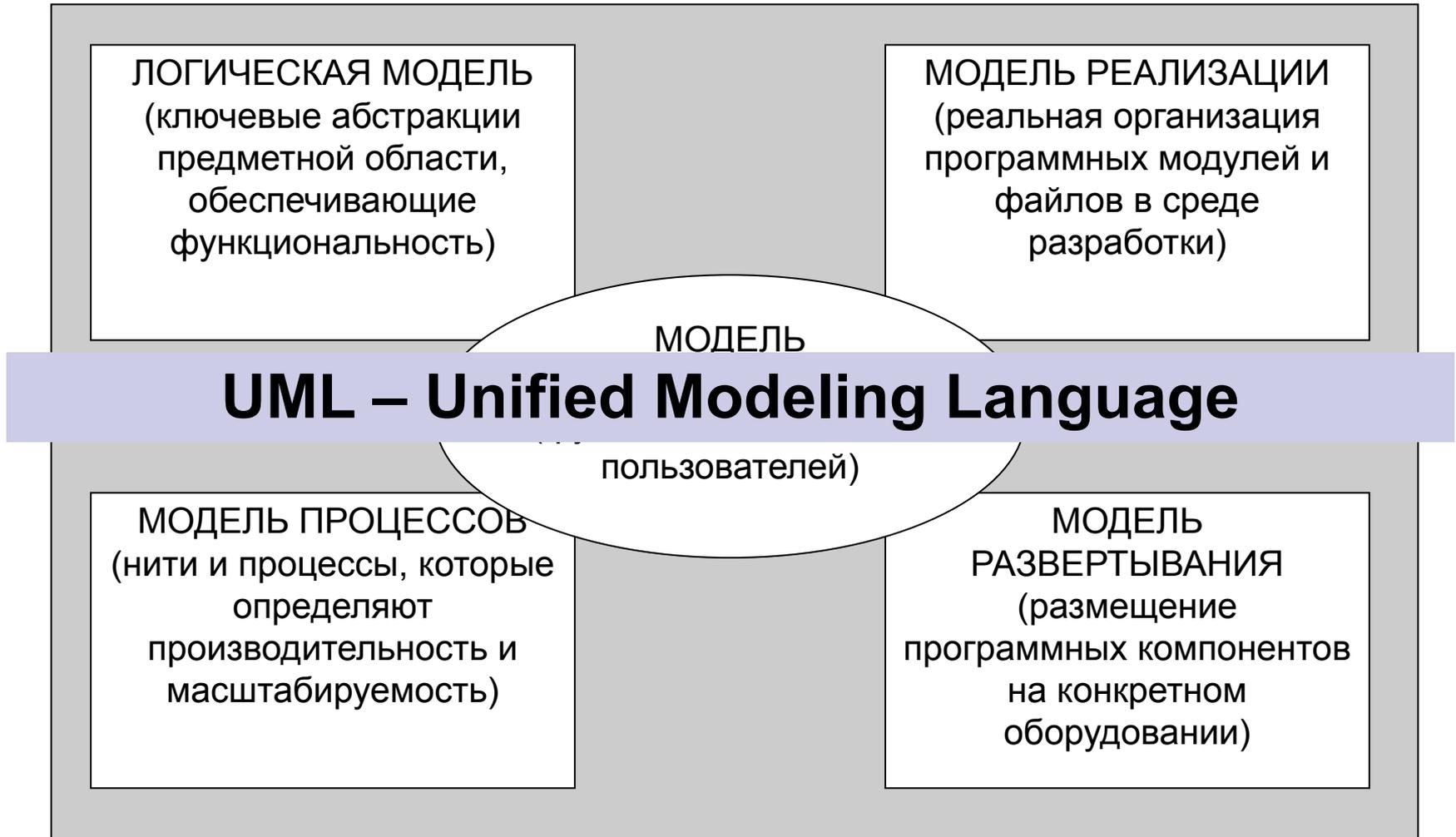
- *уточнить требуемое поведение разрабатываемого программного обеспечения;*
- *разработать концептуальную модель его предметной области с точки зрения поставленных задач.*

Использование UML при объектном подходе



**Объектная декомпозиция программы построения
таблиц и графиков**

Использование UML при объектном подходе



Полная спецификация разрабатываемого программного обеспечения при объектном подходе (UML)

Использование UML при объектном подходе

Модель использования представляет собой описание функциональности программного обеспечения с точки зрения пользователя.

Логическая модель описывает ключевые абстракции программного обеспечения (классы, интерфейсы и т. п.), т. е. средства, обеспечивающие требуемую функциональность.

Модель реализации определяет реальную организацию программных модулей в среде разработки.

Модель процессов отображает организацию вычислений и оперирует понятиями «процессы» и «нити». Она позволяет оценить производительность, масштабируемость и надежность программного обеспечения.

Модель развертывания показывает особенности размещения программных компонентов на конкретном оборудовании.

Использование UML при объектном подходе

Разновидности диаграмм UML

- *диаграммы вариантов использования;*
- *диаграммы классов;*
- *диаграммы пакетов;*
- *диаграммы последовательностей действий;*
- *диаграммы кооперации;*
- *диаграммы деятельности;*
- *диаграммы состояний объектов;*
- *диаграммы компонентов;*
- *диаграммы размещения.*

Определение «вариантов использования»

Аспекты поведения программного обеспечения называют **«вариантами использования»** или **«прецедентами»**

Вариант использования представляет собой характерную процедуру применения разрабатываемой системы конкретным действующим лицом, в качестве которого могут выступать не только люди, но и другие системы или устройства.

Разновидности вариантов использования:

- **основные** - обеспечивают требуемую функциональность разрабатываемого программного обеспечения;
- **вспомогательные** - обеспечивают выполнение необходимых настроек системы и ее обслуживание (например, архивирование информации и т. п.);
- **дополнительные** - обеспечивают дополнительные удобства для пользователя (как правило, реализуются в том случае, если не требуют серьезных затрат каких-либо ресурсов ни при разработке, ни при эксплуатации).

Определение «вариантов использования»

Краткое описание варианта использования *Выполнение задания*

<i>Название варианта</i>	<i>Выполнение задания</i>
Цель Действующи е лица Краткое описание	<i>Получение результатов решения задачи</i> <i>Пользователь</i> <i>Решение задачи предполагает выбор задачи, выбор алгоритма, задание данных и получение результатов решения.</i>
Тип варианта	<i>Основной</i>

Типичный ход событий

<i>Действия исполнителя</i>	<i>Отклик системы</i>
<p>1. Пользователь инициирует новое задание</p> <p>3. Пользователь выбирает тип задачи</p> <p>5. Пользователь выбирает способ задания данных</p> <p>а) Если выбран ввод с клавиатуры, см. раздел <u>Ввод данных</u></p> <p>б) Если выбран ввод из базы данных, см. раздел <u>Выбор данных из базы</u></p> <p>7. Пользователь выбирает алгоритм</p> <p>9. Пользователь инициирует процесс решения</p>	<p>2. Система регистрирует новое задание и предлагает список типов задач</p> <p>4. Система регистрирует тип задачи и предлагает список способов задания данных</p> <p>6. Система регистрирует данные и предлагает список алгоритмов решения</p> <p>8. Система регистрирует алгоритм и предлагает начать решение</p> <p>10. Система проверяет полноту определения задания и запускает подпрограмму решения задачи</p>

Типичный ход событий (окончание)

<i>Действия исполнителя</i>	<i>Отклик системы</i>
<i>11. Пользователь ожидает</i>	<i>12. Система демонстрирует пользователю результаты и предлагает сохранить их в базе данных</i>
<i>13. Пользователь анализирует результаты и выбирает, сохранять их в базе или нет</i>	<i>14. Если выбрано сохранение данных, то система выполняет запись данных задания в базу 15. Система переходит в состояние ожидания</i>

Альтернатива

11. Если время выполнения программы с точки зрения пользователя велико, то он прерывает процесс выполнения.

12. Система прерывает расчеты, предлагает список алгоритмов решения и возвращается на шаг 7.

Раздел *Ввод данных*

Типичный ход событий

<i>Действия исполнителя</i>	<i>Отклик системы</i>
<i>1. Пользователь выбрал Ввод данных 3. Пользователь вводит данные 5. Пользователь отвечает на запрос.</i>	<i>2. Система последовательно запрашивает ввод данных 4. Система проверяет данные и запрашивает, сохранять ли данные в базе 6. Если выбран вариант сохранения данных, то система выполняет запись данных в базу и регистрирует их в текущем задании</i>

Альтернатива

4. Если обнаружены некорректные данные, то система выдает сообщение об ошибке и предлагает их исправить, возвращаясь на предыдущий шаг.

Раздел *Выбор данных из базы*

Типичный ход событий

<i>Действия исполнителя</i>	<i>Отклик системы</i>
<i>1. Пользователь выбрал Выбор данных из базы 3. Пользователь выбирает данные</i>	<i>2. Система демонстрирует список данных в базе 4. Система читает данные и регистрирует их в текущем задании</i>

Диаграммы вариантов использования

Диаграммы вариантов использования позволяют наглядно представить ожидаемое поведение системы. Основными понятиями диаграмм вариантов использования являются: *действующее лицо*, *вариант использования*, *связь*

Действующее лицо - внешняя по отношению к разрабатываемому программному обеспечению сущность, которая взаимодействует с ним с целью получения или предоставления какой-либо информации. *Действующими лицами* могут быть пользователи, другое программное обеспечение или какие-либо технические средства, взаимодействующие с разрабатываемым программным обеспечением.

Вариант использования - некоторая очевидная для действующего лица процедура, решающая его конкретную задачу. Все варианты использования, так или иначе, связаны с требованиями к функциональности разрабатываемой системы и могут значительно отличаться по объему выполняемой работы.

Связь - взаимодействие действующих лиц и соответствующих вариантов использования.

Диаграммы вариантов использования

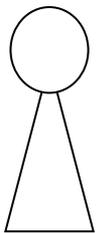
Варианты использования также могут быть связаны между собой. При этом фиксируют связи использования и расширения.

Использование подразумевает, что существует некоторый фрагмент поведения разрабатываемого программного обеспечения, который повторяется в нескольких вариантах использования. Этот фрагмент оформляют, как отдельный вариант использования и указывают связь с ним типа «*использование*».

Расширение применяют, если имеется два подобных варианта использования, различающиеся наличием в одном из них некоторых дополнительных действий. В этом случае дополнительные действия определяют как отдельный вариант использования, который связан с основным вариантом связью типа «*расширение*».

Диаграммы вариантов использования

Условные обозначения, применяемые при изображении диаграмм вариантов использования



а



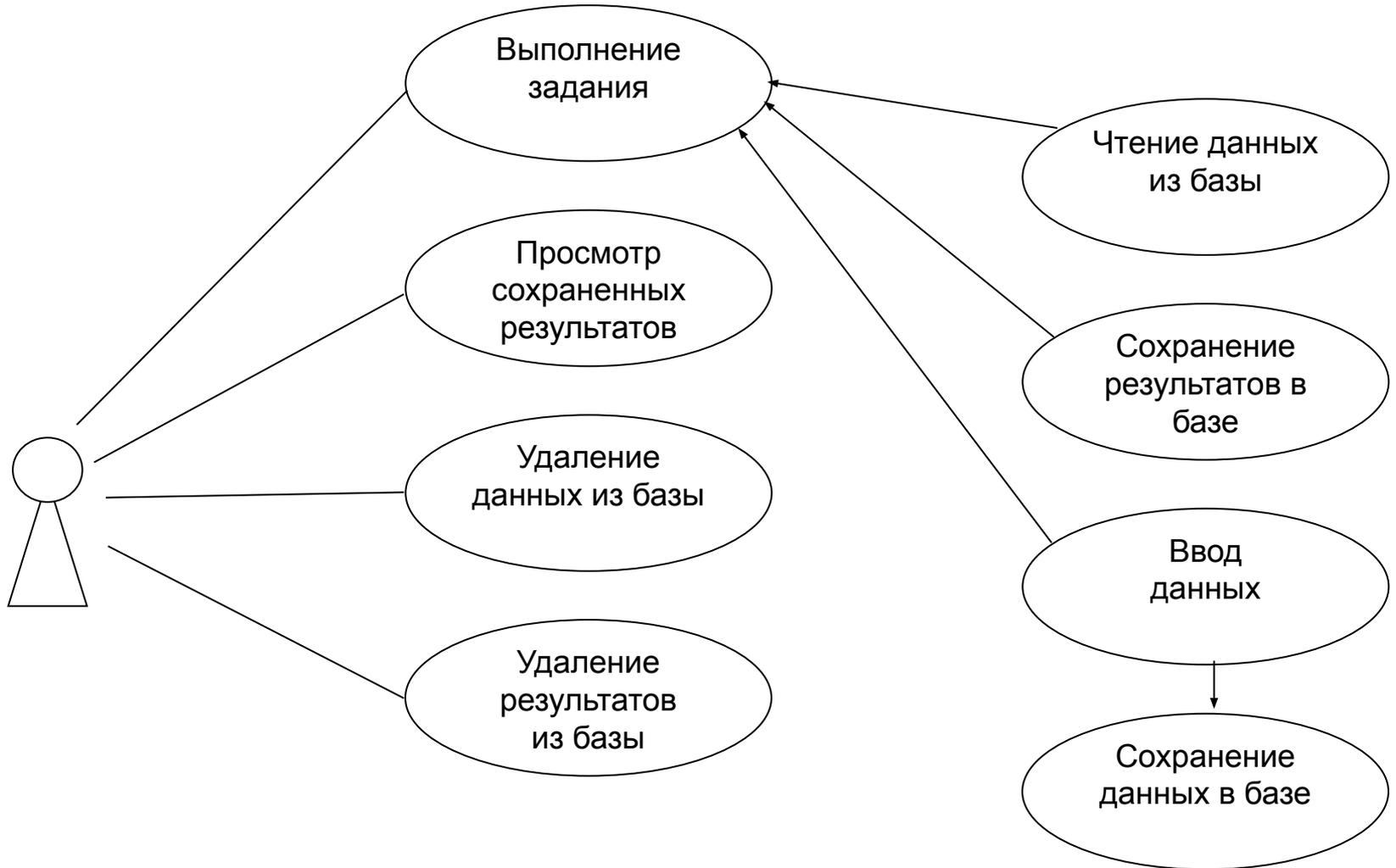
б



в

а - действующее лицо; б - вариант использования; в - связь

Пример 1. Построить диаграмму вариантов использования для системы решения комбинаторно-оптимизационных задач.



Пример 2. Построить диаграмму вариантов использования для системы учета успеваемости студентов.

