

Определение спецификаций программного обеспечения при объектном подходе



История UML. Этапы большого пути...*

- **1994**: Grady Booch & James Rumbaugh (Rational Software) объединили методы **Booch** (проектирование) и **OMT** (анализ) -> **Unified method**
- **1995**: присоединился Ivar Jacobson (**OOSE** метод)

"Three amigos"



James Rumbaugh



Grady Booch



Ivar Jacobson

• *Источник:* www.wikipedia.org; www-306.ibm.com/software/rational/bios; www.wikipedia.org; <http://www-306.ibm.com/software/rational/bios>; <http://www.ivarjacobson.com>

Структурные диаграммы

- **Диаграмма классов**
Показывает классы, их атрибуты и связи между классами.
- **Диаграмма компонентов**
Показывает компоненты и связи между ними
- **Структурная диаграмма**
Показывает внутреннюю структуру классов и связи с внешним миром
- **Диаграмма развертывания**
Показывает, как ПО размещается на аппаратуре (серверах, рабочих станциях...)
- **Диаграмма объектов**
Показывает структуру системы в конкретный момент времени, объекты, их атрибуты...
- **Диаграмма пакетов**
Показывает, как система раскладывается на крупные составные части и связи между этими частями

Диаграммы поведения

- **Диаграмма действия**

Показывает потоки информации в системе.

- **Диаграмма состояния**

Представляет собой конечный автомат, показывающий функционирование системы.

- **Диаграмма вариантов использования**

Показывает работу системы с точки зрения пользователей.

Диаграммы взаимодействия

- **Диаграмма кооперации**

Показывает структурную организацию участвующих во взаимодействии объектов

- **Диаграмма взаимодействия**

(новация UML 2.0)

- **Диаграмма последовательности**

Показывает временную упорядоченность событий

- **Временная диаграмма**

Диаграмма связана с временными рамками

Понятия UML

- **Для описания структуры:**

Актер, Атрибут, Класс, Компонент, Интерфейс, Объект, Пакет.

- **Для описания поведения:**

Действие, Событие, Сообщение, Метод, Операция, Состояние, Вариант использования.

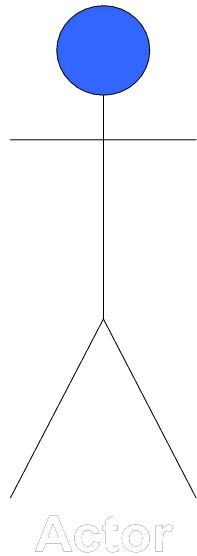
- **Для описания связей:**

Агрегация, Ассоциация, Композиция, Зависимость, Наследование.

- **Некоторые другие понятия:**

Стереотип, Кратность, Роль.

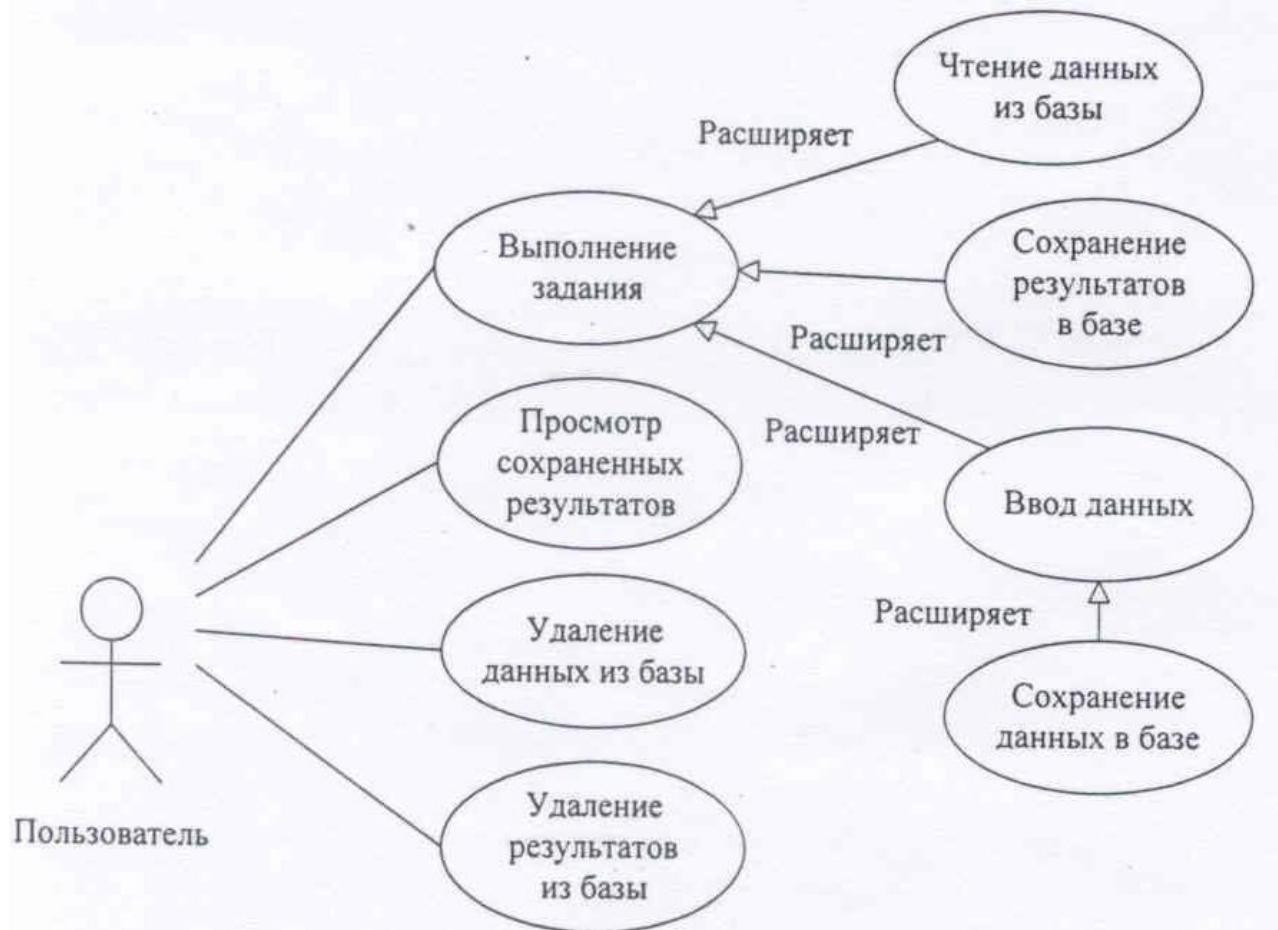
Актеры и Варианты использования в UML



Актер в UML – человек, машина или программа, воздействует на систему, является внешним по отношению к ней.



Вариант использования в UML – описание последовательности действий – (часто с вариантами – сценариями).



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

Вариант использования **Выполнение задания**

Типичный ход событий

Действие исполнителя	Отклик системы
1. Пользователь инициирует новое задание	2. Система регистрирует новое задание и предлагает список типов задач
3. Пользователь выбирает тип задачи	4. Система регистрирует тип задачи и предлагает список способов задания данных
5.Пользователь выбирает способ задания данных: а) Если выбран ввод с клавиатуры, см. раздел Ввод данных б) Если выбран ввод из базы данных, см. раздел Выбор данных из базы	6.Система регистрирует данные и предлагает список алгоритмов решения
7.Пользователь выбирает алгоритм	8.Система регистрирует алгоритм и предлагает начать решение
9.Пользователь инициирует процесс решения	10. Система проверяет полноту определения задания и запускает подпрограмму решения задачи
11.Пользователь ожидает	12. Система демонстрирует пользователю результаты и предлагает сохранить их в базе данных
13. Пользователь анализирует результаты и выбирает, сохранять их в базе или нет	14.Если выбрано сохранение данных, то система выполняет запись данных задания в базу 15.Система переходит в состояние ожидания

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

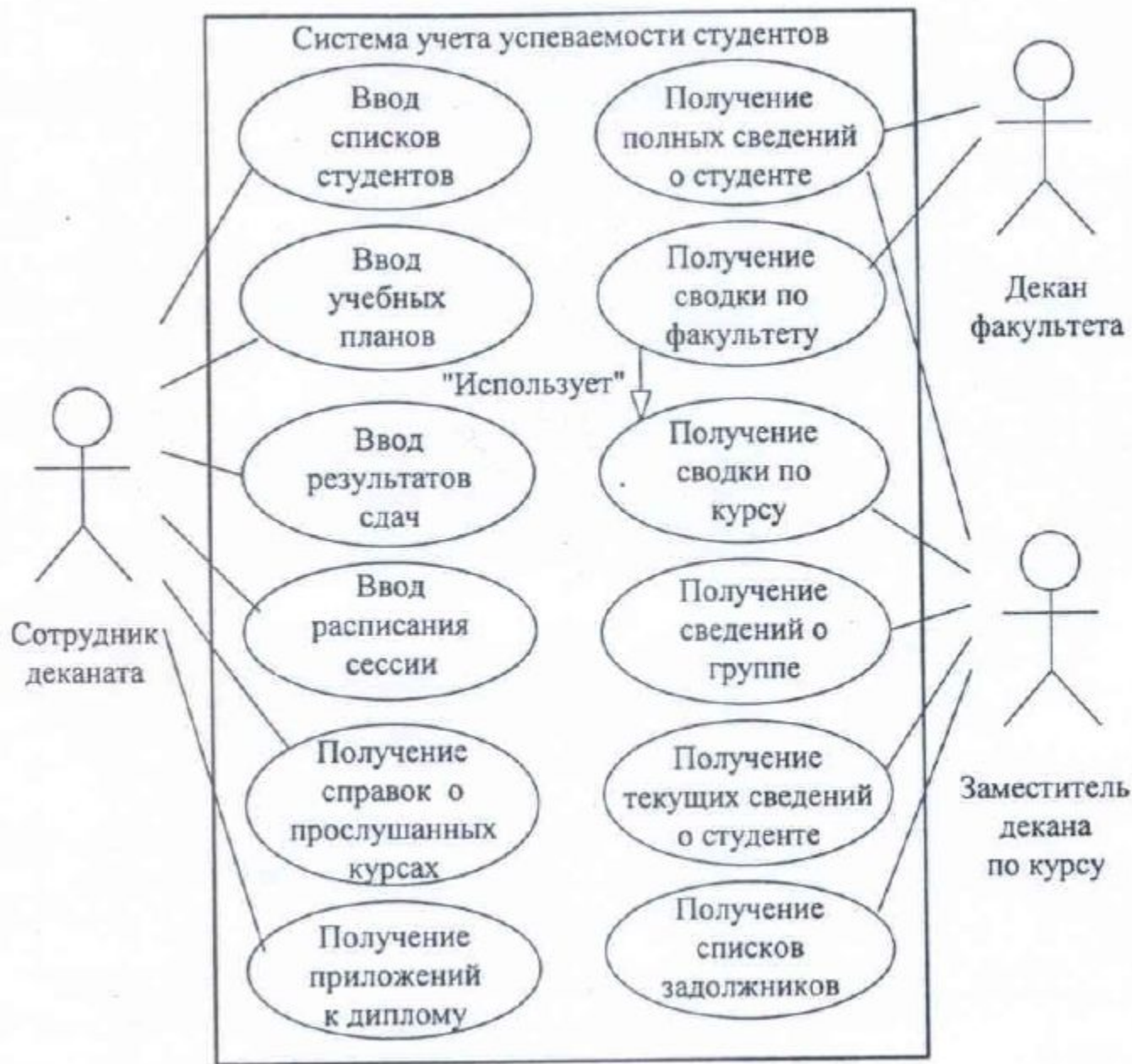
Действие исполнителя	Отклик системы
<p>1. Пользователь выбрал Ввод данных</p> <p>3. Пользователь вводит данные</p> <p>5. Пользователь отвечает на запрос</p>	<p>2. Система последовательно запрашивает ввод данных</p> <p>4. Система проверяет данные и запрашивает, сохранять ли данные в базе</p> <p>6. Если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

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

Раздел **Выбор данных из базы**

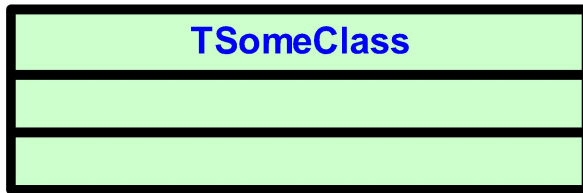
Типичный ход событий

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

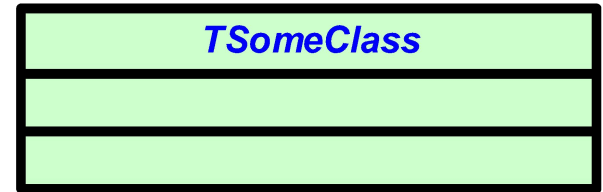


Концептуальная модель предметной области

Класс



Абстрактный класс

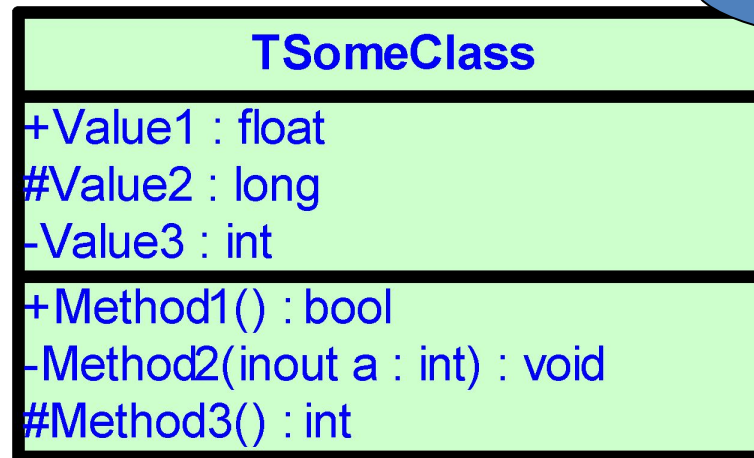


Классы в UML

Поля

Методы

ы



Имя класса

+ public
protected
- private

Отношения классов

Отношение ассоциации означает наличие связи между экземплярами классов или объектами, например, класс Студент ассоциирован с классом Институт. Ассоциация может иметь имя, например Обучается. Рядом с именем ассоциации обычно ставят стрелку, указывающую направление чтения имени («Студент обучается в институте», а не наоборот).



Обобщением называют такое отношение между классами, при котором любой объект одного класса (подтипа) обязательно является также и объектом другого класса, называемого в данном контексте супертипом.



Задание, тип задачи, список типов задач, способ задания данных, ввод данных, выбор данных из базы, алгоритм решения задачи, список конкретных алгоритмов решения задачи, полнота описания задания, результаты, данные, база да

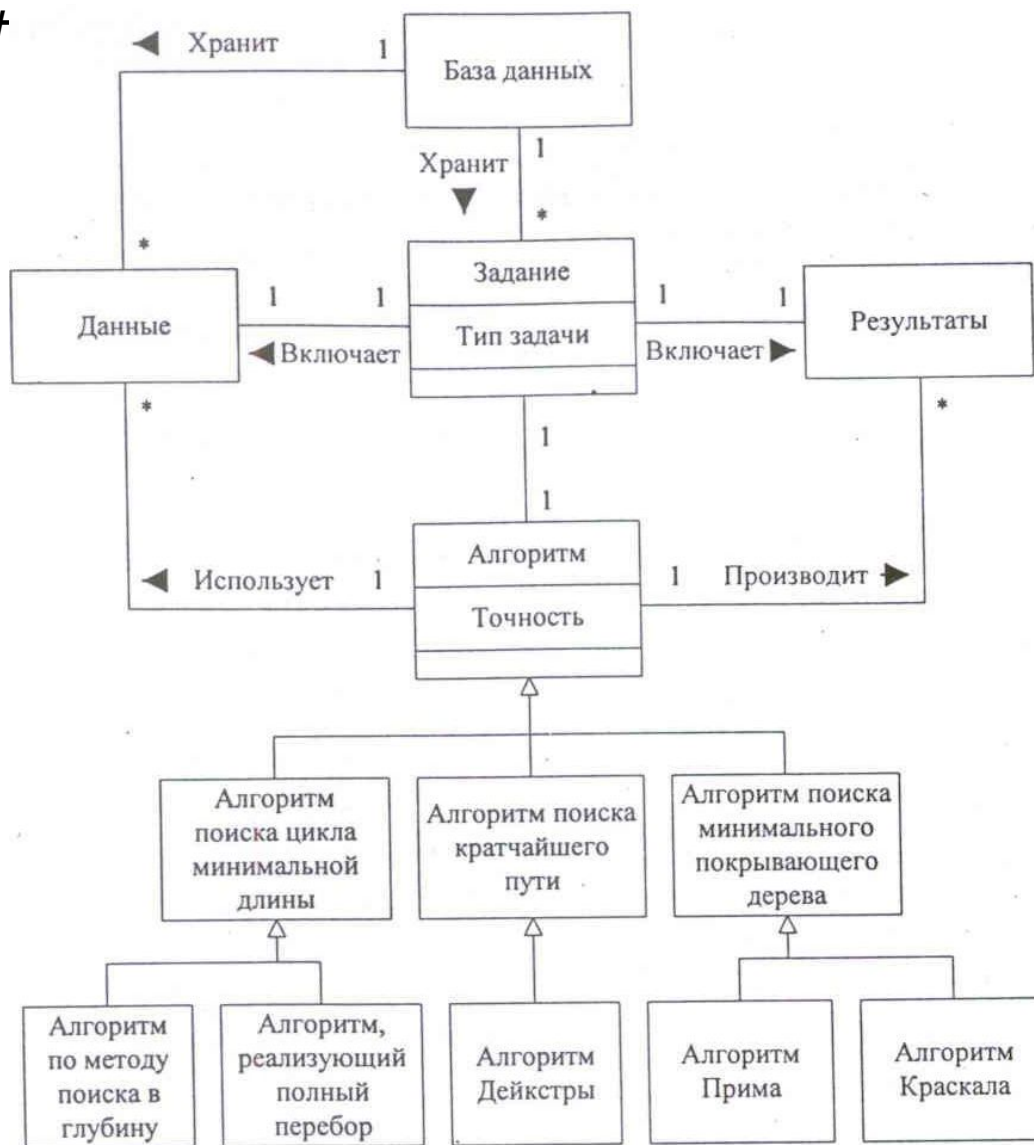
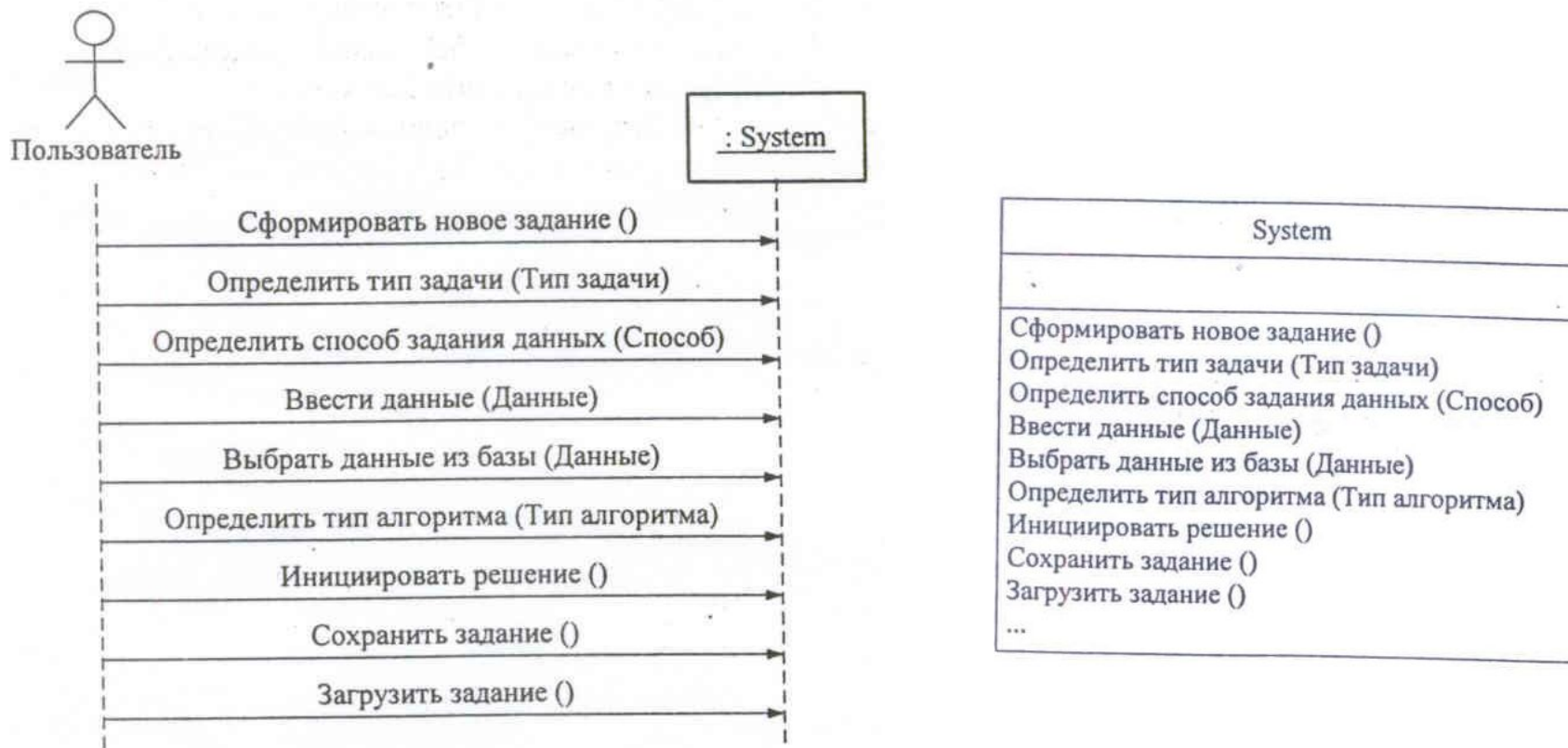


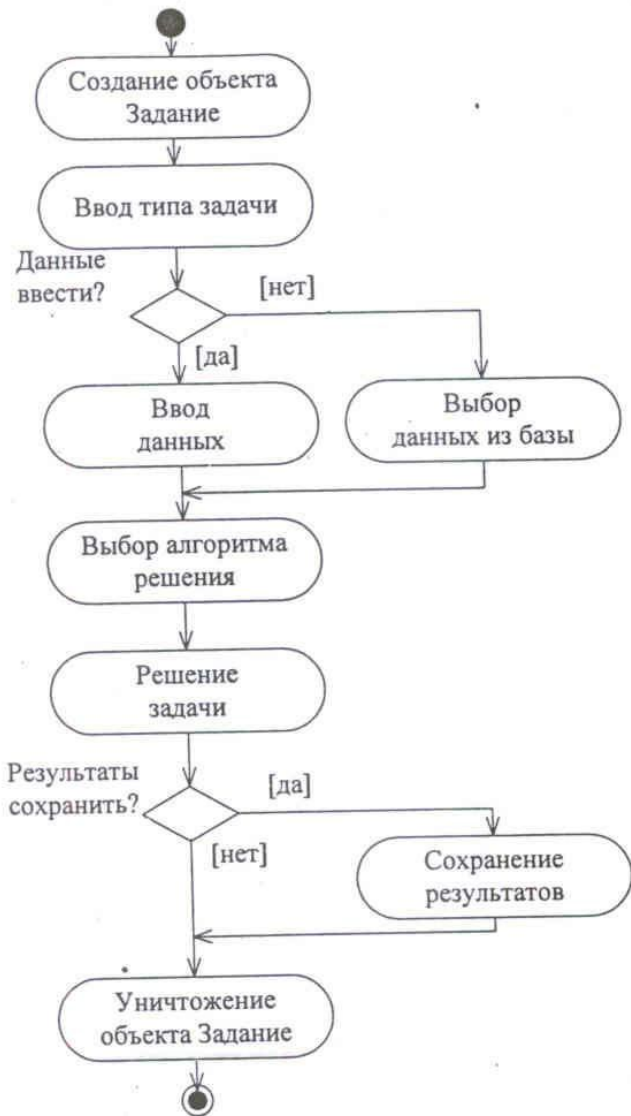
Диаграмма последовательностей системы. Системные события и операции

Диаграмма последовательностей системы — графическая модель, которая для определенного сценария варианта использования показывает генерируемые действующими лицами события и их порядок.

При этом система рассматривается как единое целое.



Диаграммы деятельностей



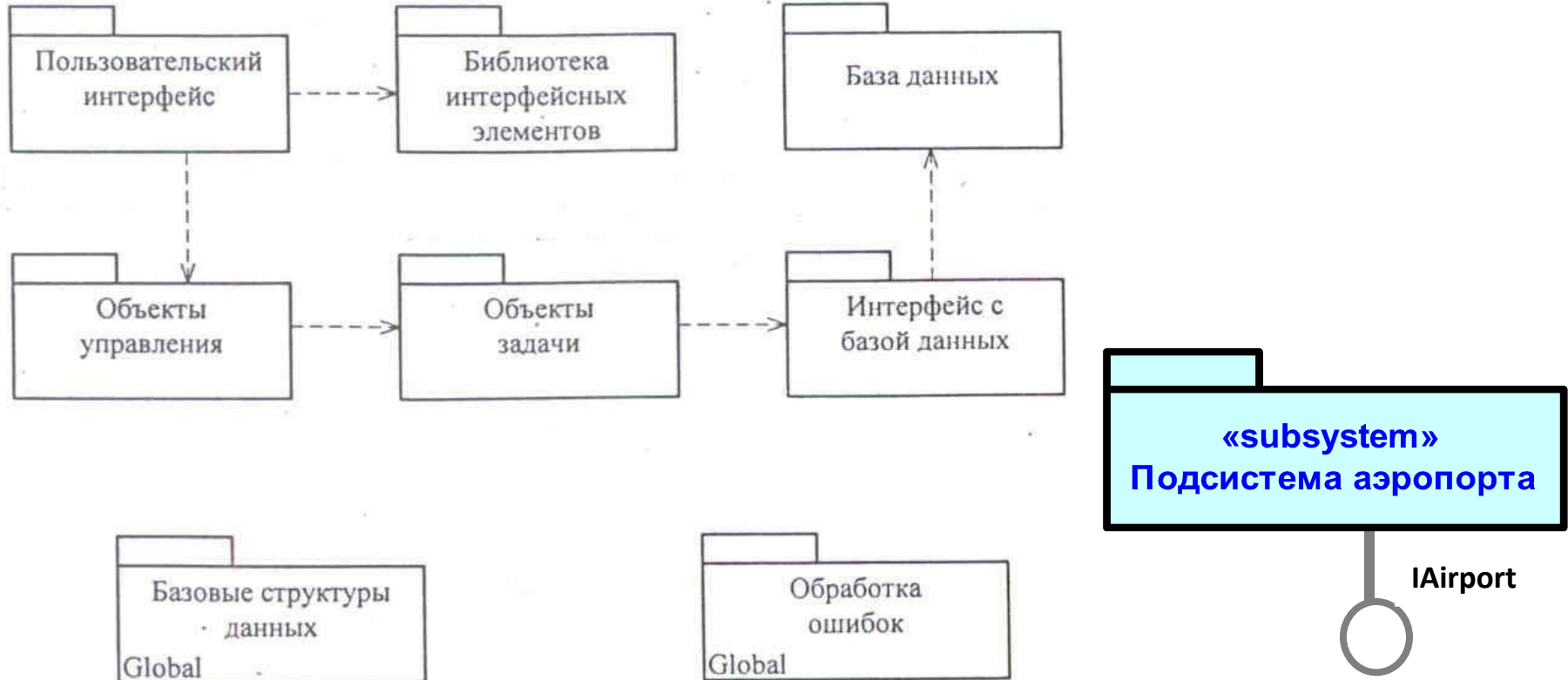
диаграммы деятельности являются обобщенным представлением алгоритма, реализующего анализируемый вариант использования.

Разработка структуры программного обеспечения

Пакетом при объектном подходе называют совокупность описаний классов и других программных ресурсов, в том числе и самих пакетов.

При этом в один пакет обычно собирают классы и другие ресурсы *единого назначения.*

Диаграмма пакетов показывает, из каких частей состоит проектируемая программная система, и как эти части связаны друг с другом.



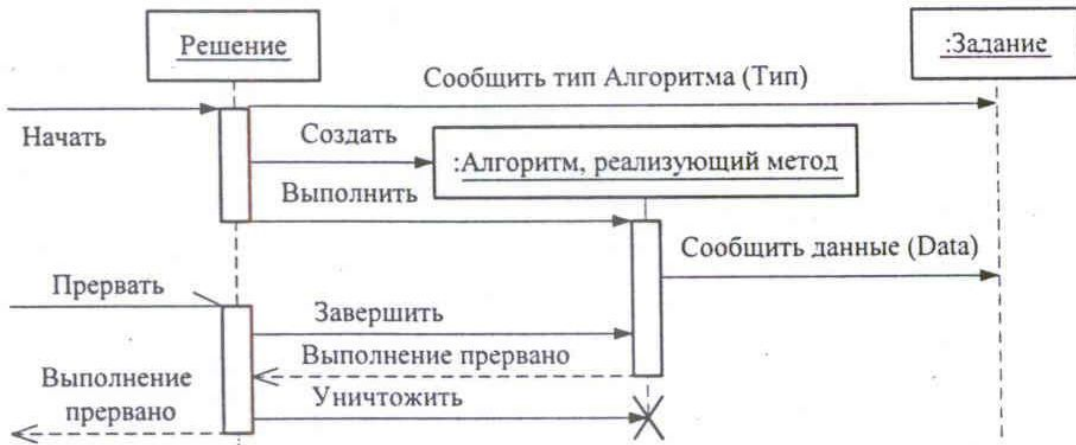
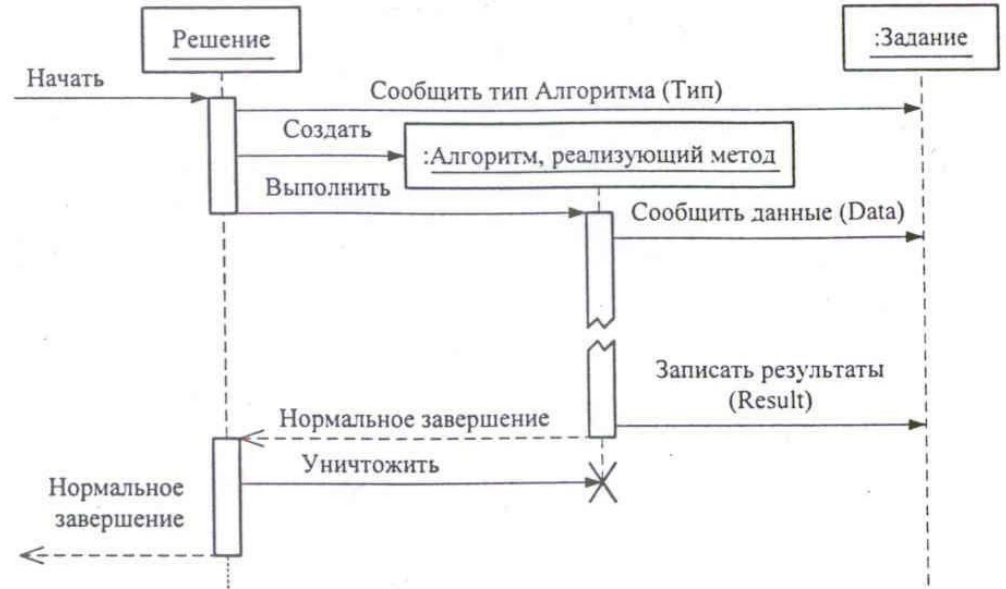
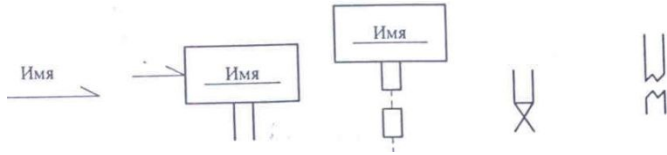
Диаграммы последовательностей этапа

НИЯ

Имя объекта

Имя объекта: Имя класса

:Имя класса



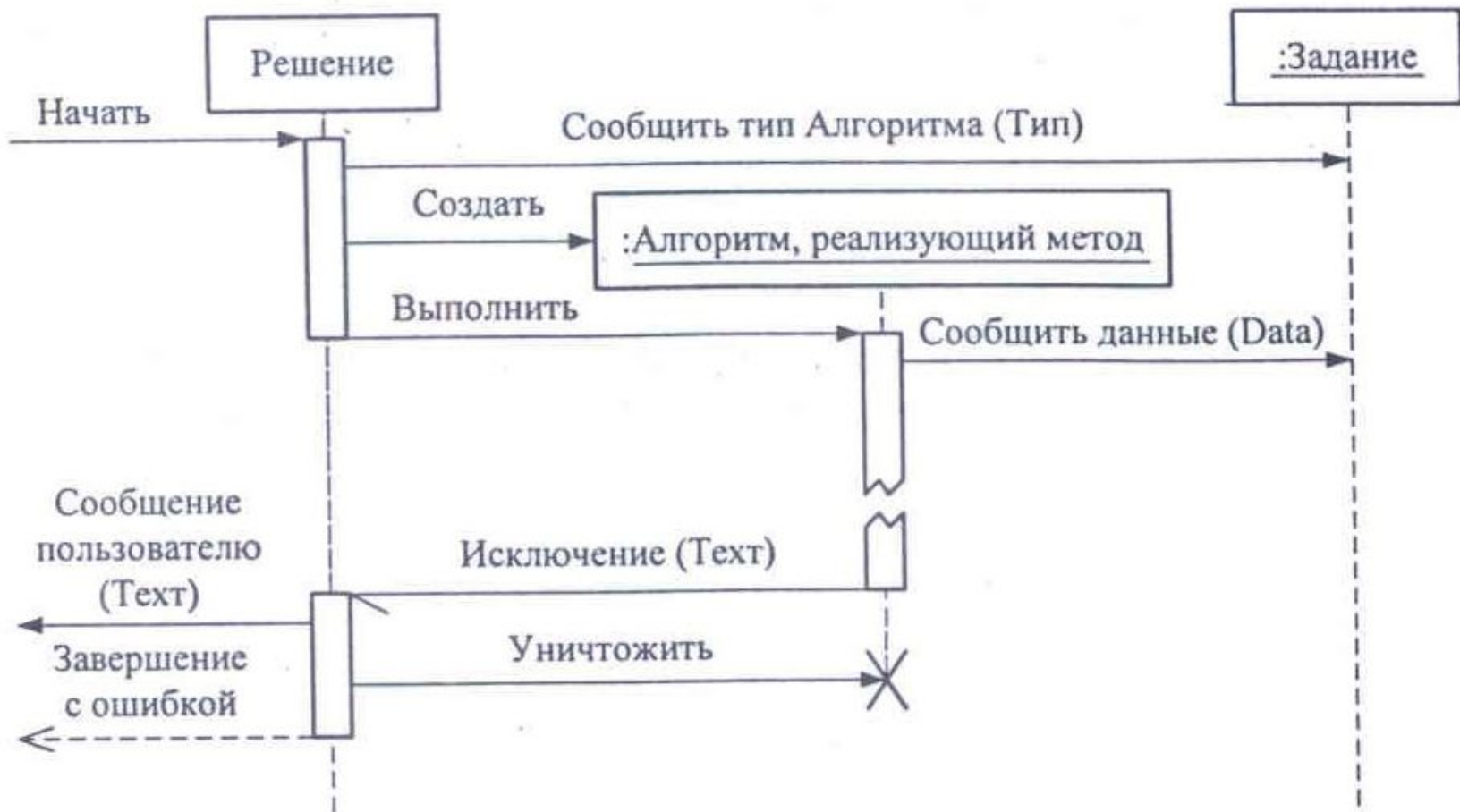
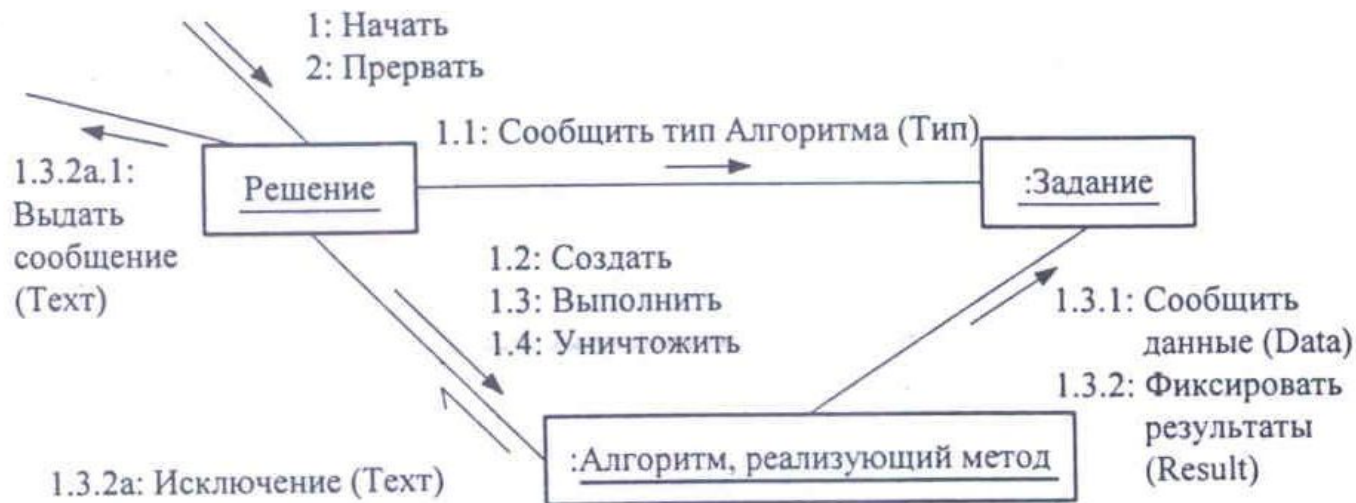


Диаграмма кооперации - это альтернативный способ представления взаимодействия объектов в процессе реализации сценария, который позволяет по-другому взглянуть на ту же информацию.



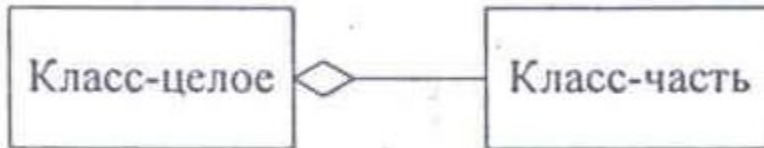
Уточнение отношений

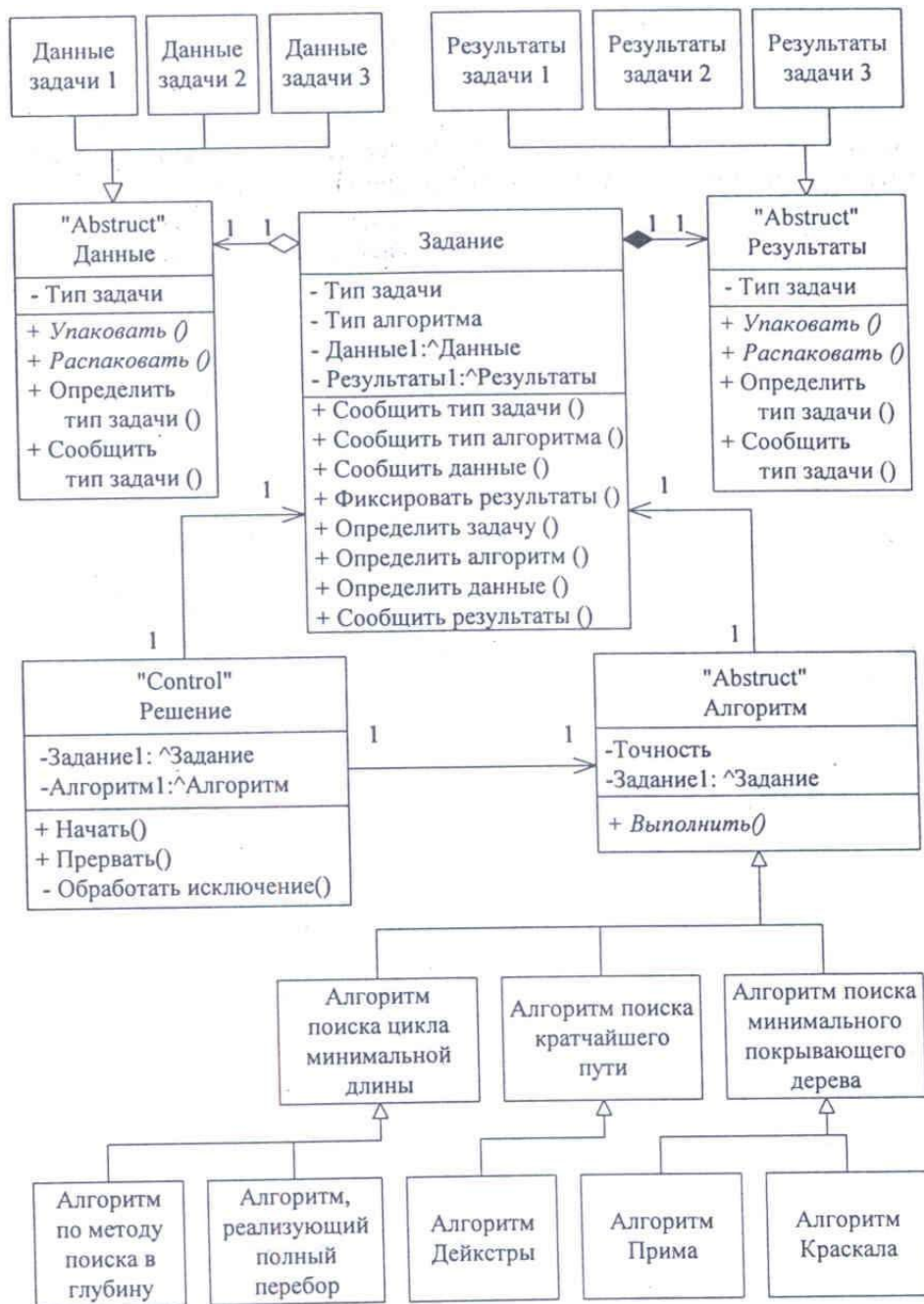
классов

Агрегацией называют ассоциацию между целым и его частью или частями. Агрегацию вместо ассоциации указывают, если отношение «целое-часть» в конкретном случае существенно. Например, если колесо нас интересует только как часть автомобиля, то между соответствующими классами целесообразно указать отношение агрегации, а если колесо - товар, также как и автомобиль, то связь целое-часть не существенна.,

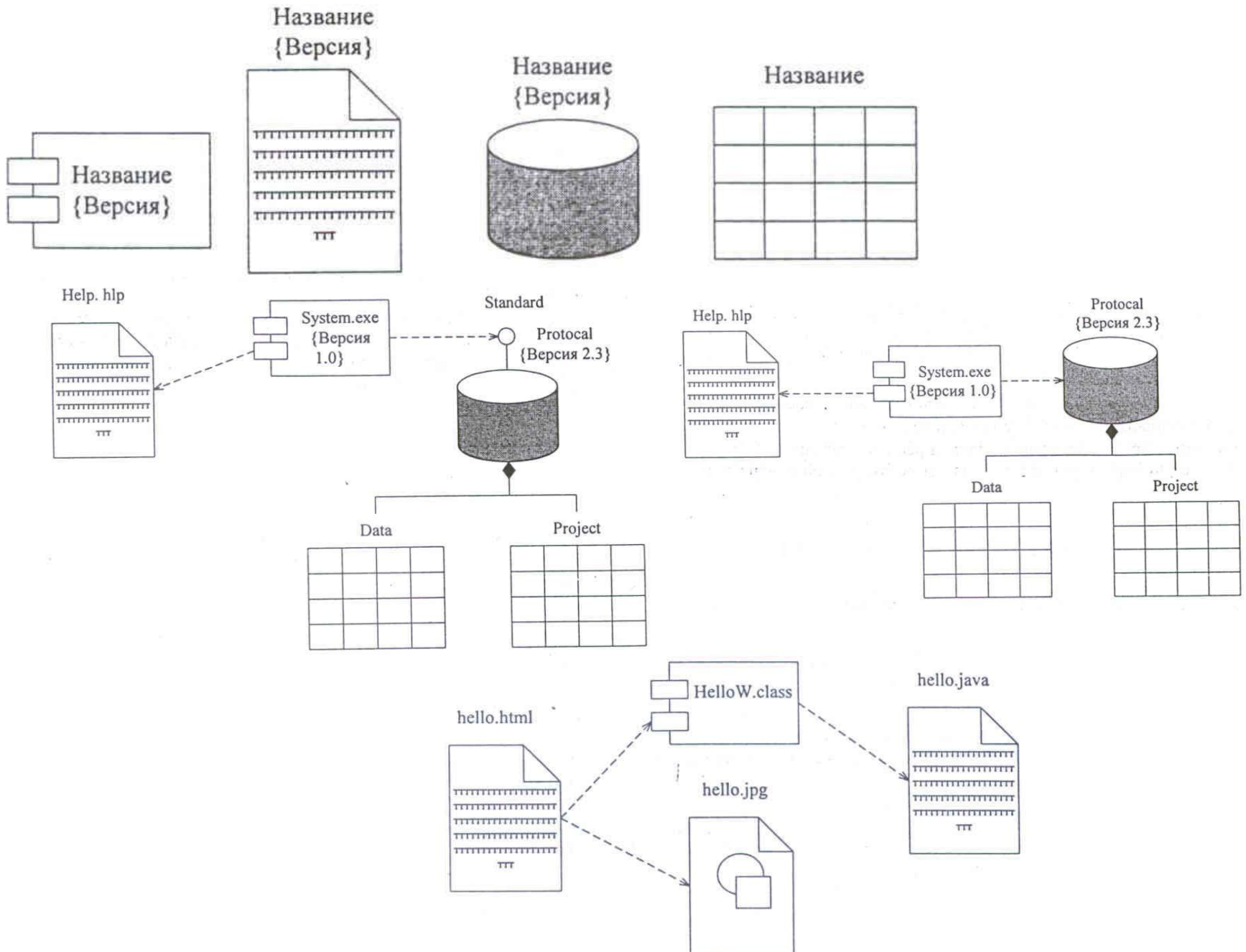


Композиция - более сильная разновидность агрегации, которая подразумевает, что объект-часть может принадлежать только единственному целому. Объект-часть при этом создается и уничтожается только вместе со своим целым.



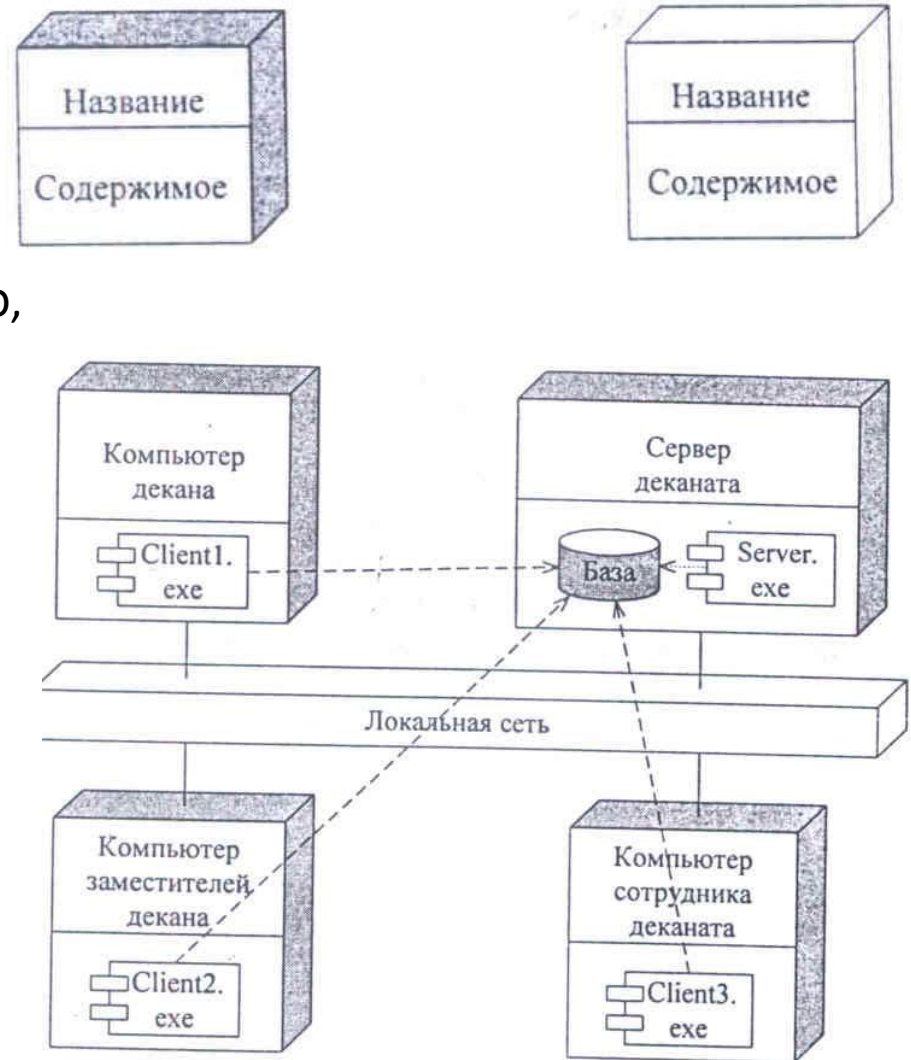


Компоновка программных



Проектирование размещения программных компонентов для распределенных программных систем

Диаграмма размещения отражает физические взаимосвязи между программными и аппаратными компонентами системы. Каждой части аппаратных средств системы, например, компьютеру или датчику, на диаграмме размещения соответствует узел. Соединения узлов означают наличие в системе соответствующих коммуникационных каналов. Внутри узлов указывают размещенные на данном оборудовании программные компоненты разрабатываемой программной системы, сохраняя указанные на диаграмме компонентов отношения зависимости.



Спасибо за внимание!

