

Учебный курс (на основе материалов УГАТУ)

Объектно-ориентированный анализ и программирование

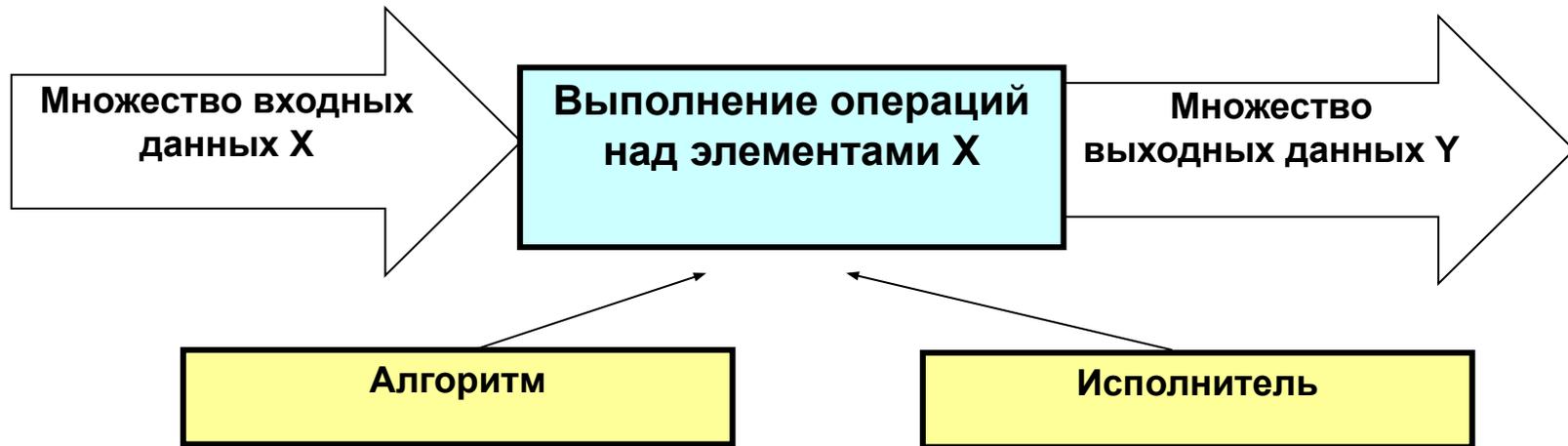
Лекция 1

Программа как модель предметной области

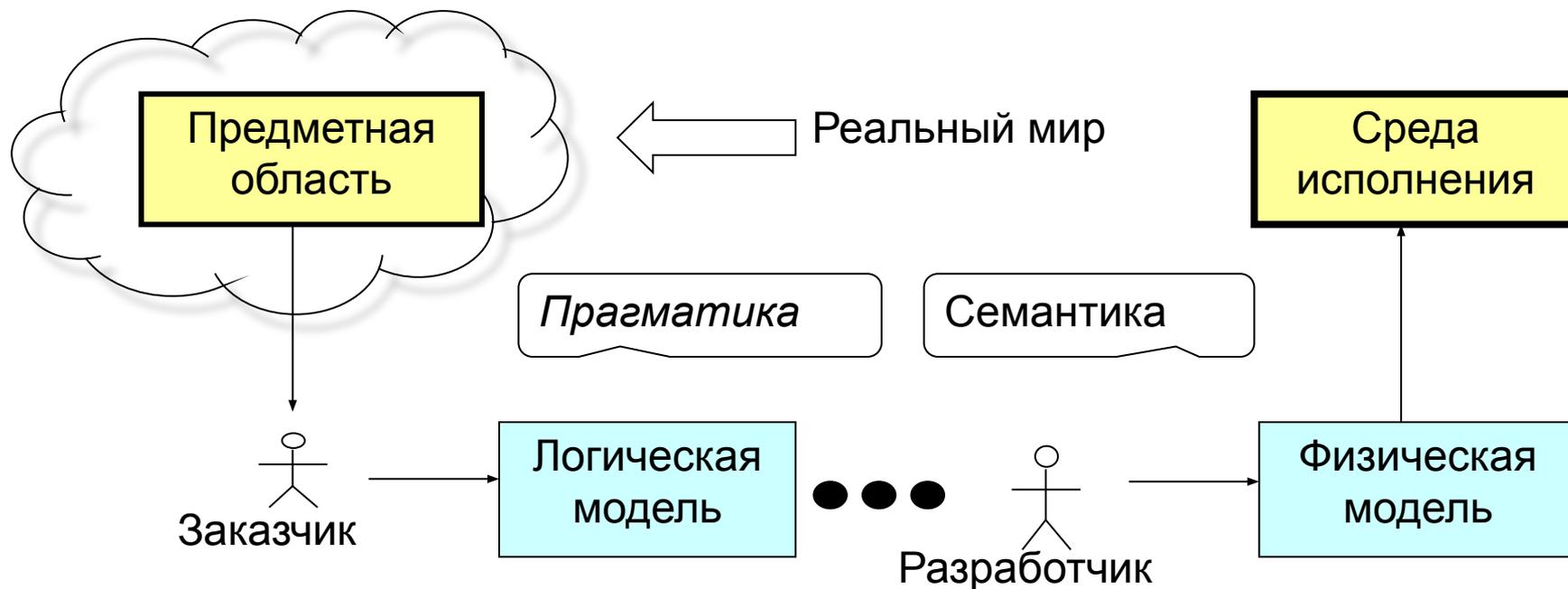
Информационные процессы

Информационный процесс - процесс получения, хранения, обработки и передачи информации с помощью компьютерных и других технических средств

Данные - информация любой природы, зафиксированная тем или иным способом



Алгоритм – это точное описание последовательности действий над входными данными, выполнение которых исполнителем приводит к получению выходных данных.



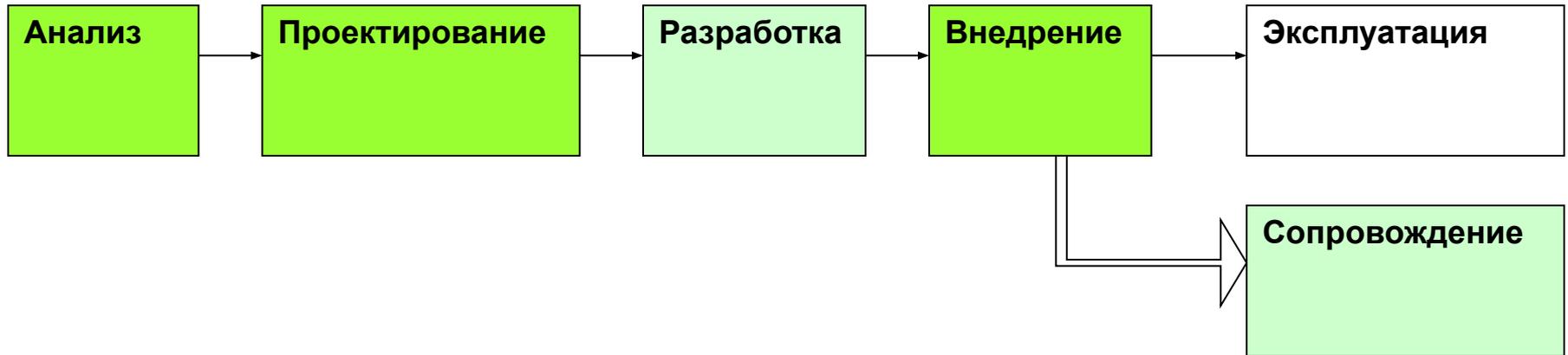
Модель – формализованное описание предметной области.

Логическая модель отражает взгляд на предметную область со стороны заказчика

Физическая модель отражает взгляд на предметную область со стороны разработчика

Жизненный цикл программы - совокупность взаимосвязанных и следующих во времени этапов, начиная от разработки требований к программе и заканчивая полным отказом от ее использования

Жизненный цикл программы формально можно рассматривать как переход от логической модели предметной области к физической модели предметной области через промежуточные модели. Каждая из моделей отражает точку зрения на разрабатываемую программу определенного участника процесса разработки.



Анализ – разработка логической модели и определение спецификации.

Проектирование – определение структуры программы и распределение функциональности по элементам этой структуры.

Разработка – определение способа реализации требуемой функциональности в каждом из элементов структуры и разработка физической модели предметной области на некотором языке программирования.

Внедрение – конфигурирование программы под конкретную среду исполнения и проведение испытаний на соответствие программы требованиям заказчика.

Эксплуатация – применение программы по назначению.

Сопровождение – модификация программы с целью исправления ошибок, выявленных при эксплуатации.



Так объяснил заказчик



Так понял лидер проекта



Так спроектировал аналитик



Так реализовал программист



Так описал бизнес-консультант



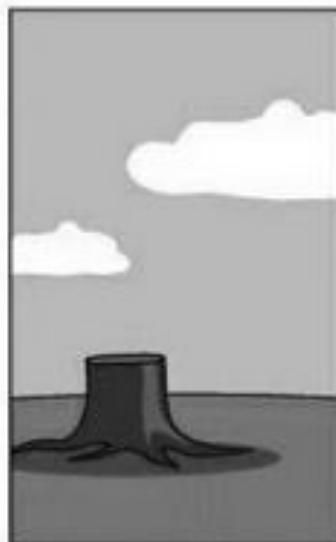
Так проект был документирован



Так продукт был проинсталлирован



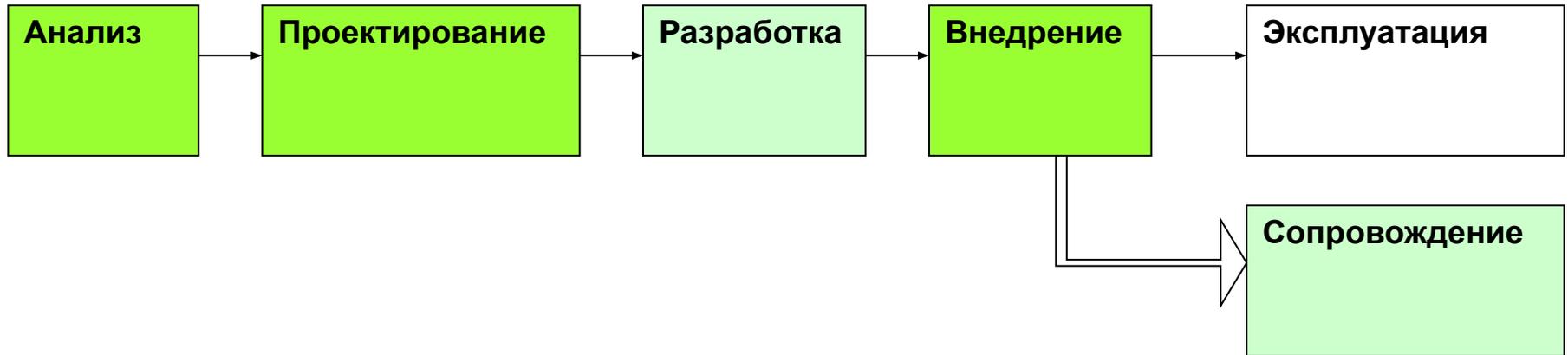
Такой счет был выставлен заказчику



Так осуществлялась техническая поддержка



А вот чего на самом деле хотелось заказчику



Анализ – разработка логической модели и определение спецификации.

Проектирование – определение структуры программы и распределение функциональности по элементам этой структуры.

Разработка – определение способа реализации требуемой функциональности в каждом из элементов структуры и разработка физической модели предметной области на некотором языке программирования.

Внедрение – конфигурирование программы под конкретную среду исполнения и проведение испытаний на соответствие программы требованиям заказчика.

Эксплуатация – применение программы по назначению.

Сопровождение – модификация программы с целью исправления ошибок, выявленных при эксплуатации.

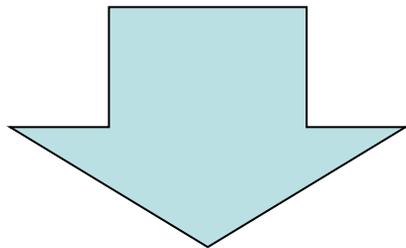
Итерационная модель жизненного цикла



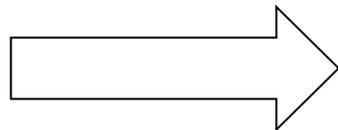
Фазы

- Инициализация
- Усовершенствование
- Создание
- Переход

- программа
- программа
- программа
- программа

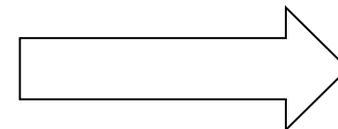


Время



Начало фазы

Анализ и определение требований

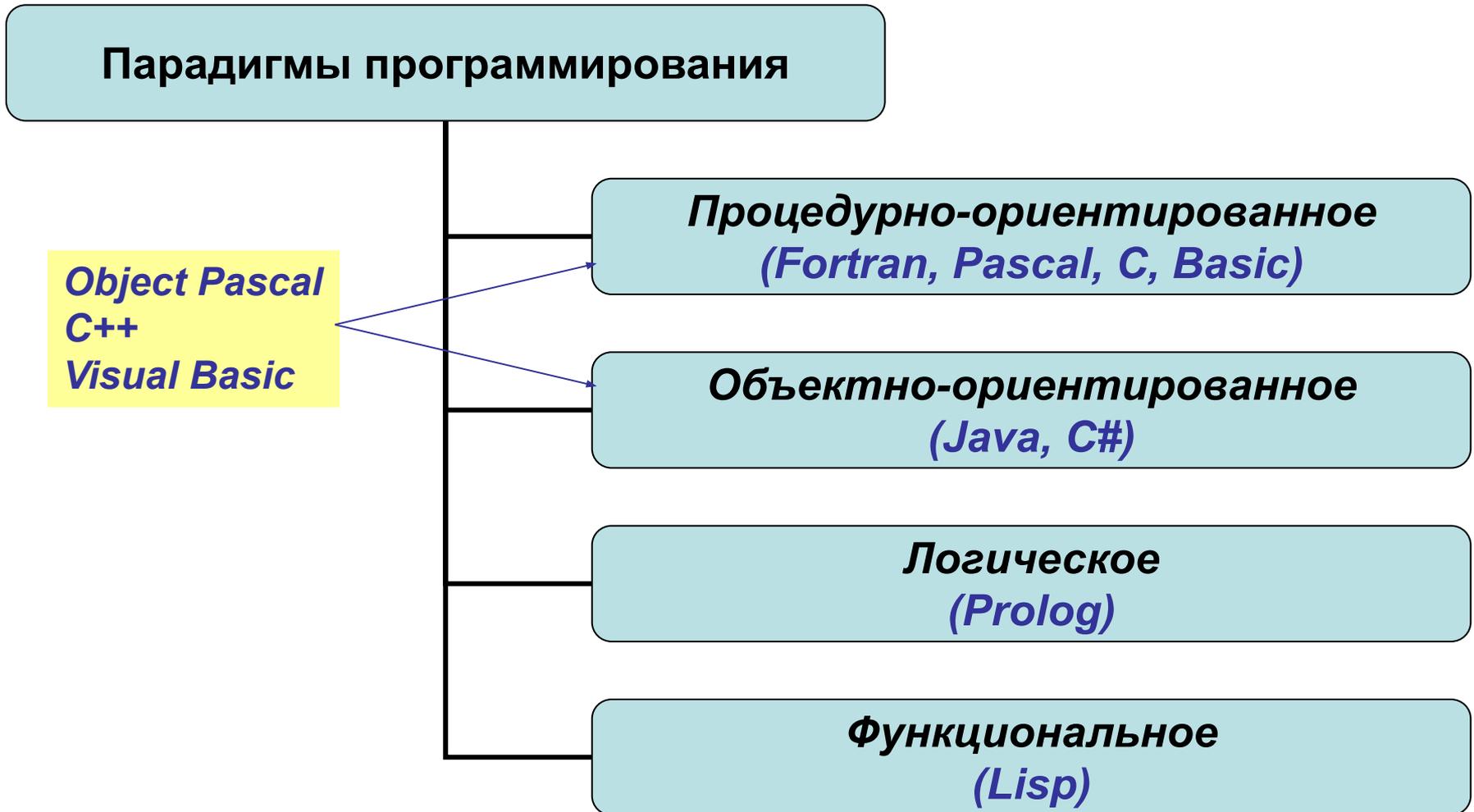


Завершение фазы

Rational Unified Process (RUP)



Парадигма программирования - понятийный аппарат, используемый для разработки модели предметной области



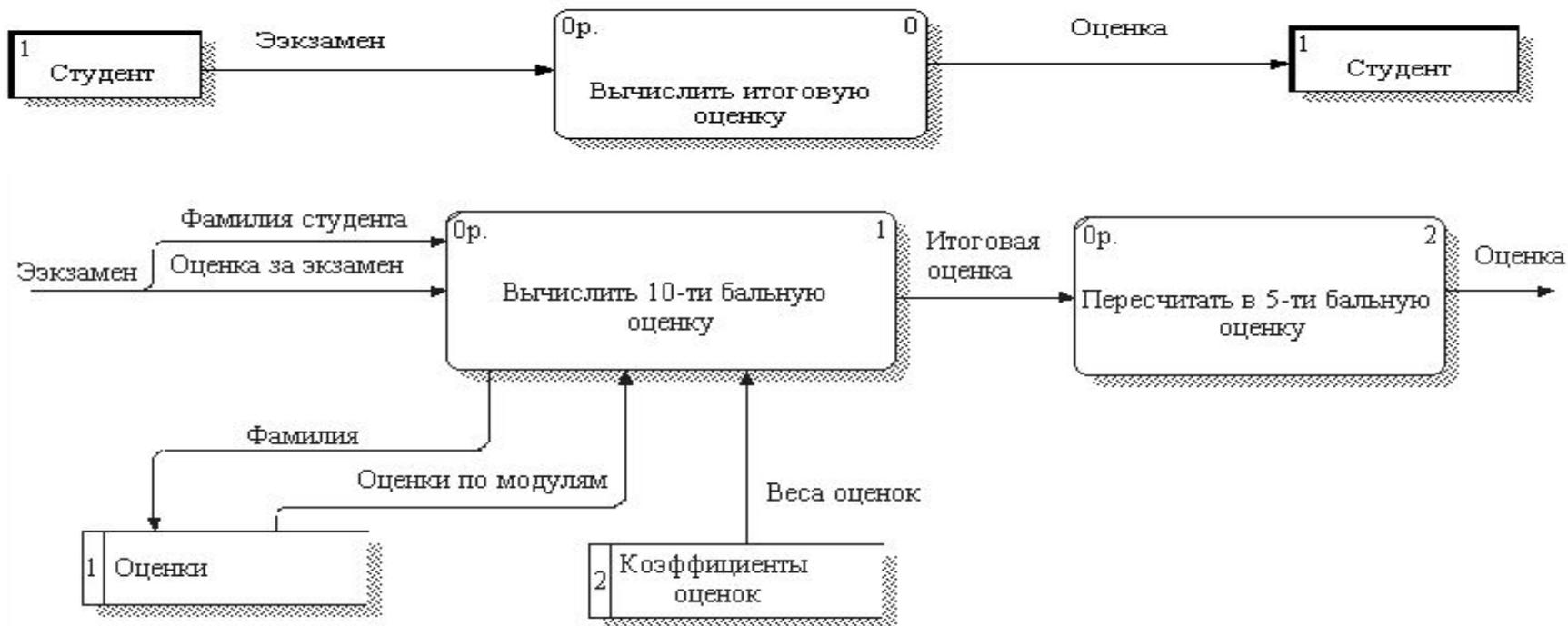
Процедурно-ориентированное программирование



Предметная область – множество процессов, воздействующих на входные данные с целью их преобразования в выходные данные

Логическая модель (диаграмма потоков данных) определяют:

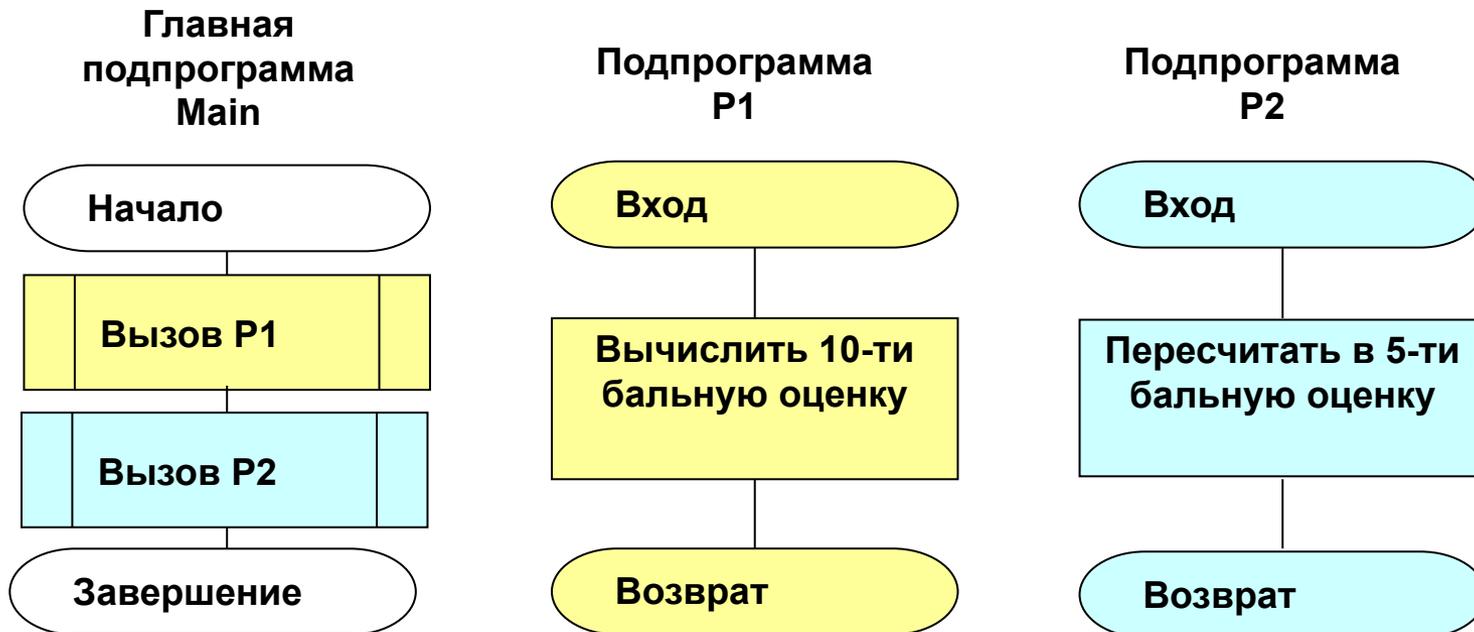
- Входные данные
- Источники входных данных
- Выходные данные
- Потребители выходных данных
- Данные, подлежащие долговременному хранению (накопители данных)
- Процессы преобразования входных данных в выходные данные



Структура процедурно-ориентированной программы



ГОСТ 19.701-90



1. Запуск Main
2. Вызов P1
3. Обработка данных в P1 и возврат в Main
4. Вызов P2
5. Обработка данных в P2 и возврат в Main
6. Завершение программы



Предметная - множество взаимодействующих объектов.

Цель взаимодействия – достижение заданной функциональности

Объект - мыслимая или реальная сущность, обладающая характерным поведением и характеристикой и являющаяся важной для данной предметной области.

Поведение – набор операций, которые может выполнять объект

Характеристика – атрибуты, отличающие данный объект от других объектов

Класс объектов - множество объектов, которые имеют одинаковый набор операций и одинаковый набор атрибутов

Модель системы – совокупность логически взаимосвязанных диаграмм.

Диаграммы разрабатываются в нотации языка UML
(Unified Modeling Language)

Интегрированная модель системы

Интегрированная модель

Функциональность

Диаграмма вариантов использования
use case diagram

Структура

Диаграмма классов
class diagram

Поведение

Диаграмма состояний
statechart diagram

Диаграмма кооперации
collaboration diagram

Диаграмма последовательности
sequence diagram

Диаграмма деятельности
activity diagram

Реализация

Диаграмма компонентов
component diagram

Диаграмма развертывания
deployment diagram

Диаграмма вариантов использования



Определяет функциональность программы в терминах актеров и вариантов использования.

Актер - инициатор выполнения варианта использования

Вариант использования - заданная функциональность

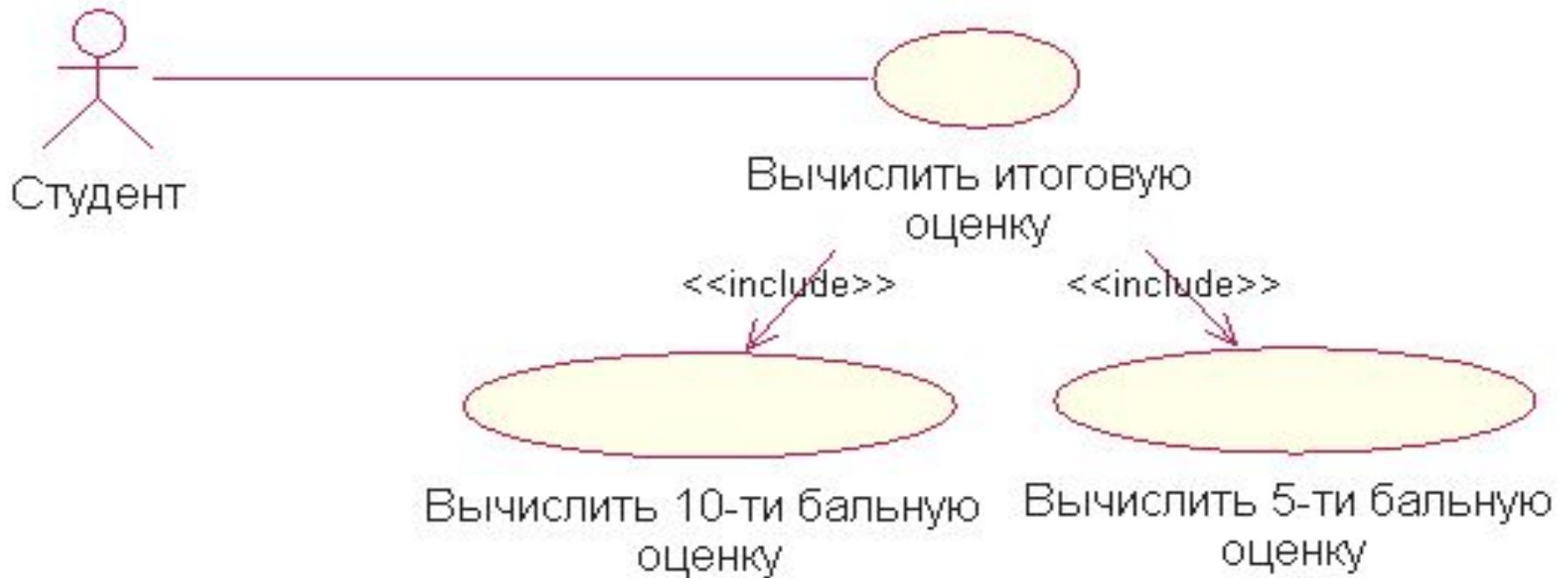
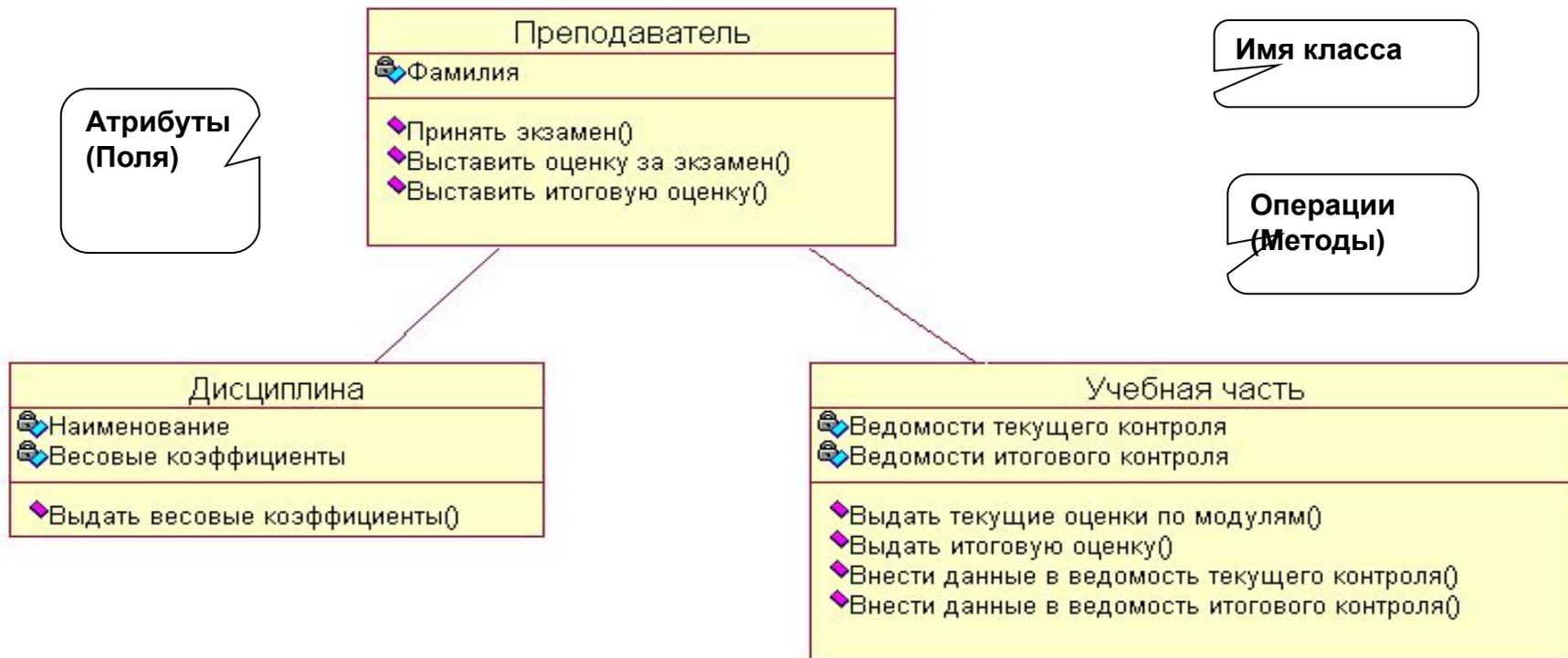


Диаграмма классов



Определяет структуру предметной области (программы) в виде набора классов, находящихся между собой в определенных отношениях

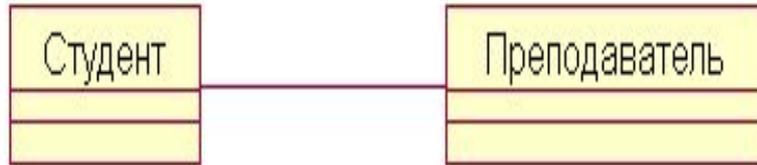


Только в составе
объекта или класса
(**ИНКАПСУЛЯЦИЯ**)

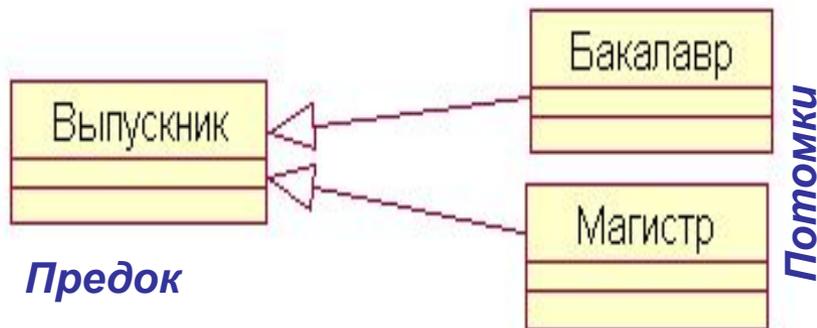
Объектные языки	UML	Процедурные языки
Поле	Атрибут	Данные
Метод	Операция	Подпрограмма

Самостоятельные
сущности

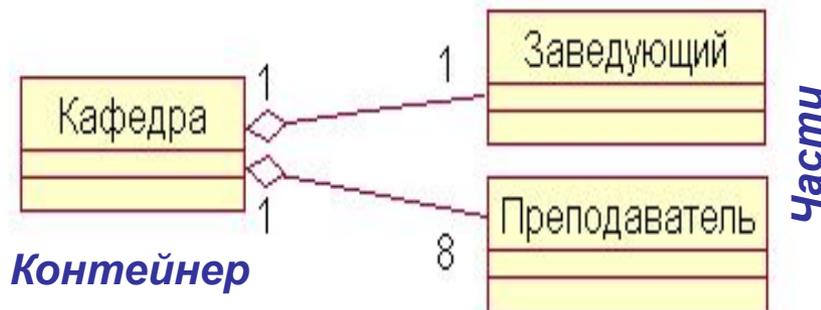
Отношения между классами



Ассоциация
Классы относятся к одной предметной области



Наследование
Классы-потомки являются разновидностями класса-предка



Агрегация
Класс-контейнер включает классы-части

Интегрированная модель системы

Интегрированная модель

Функциональность

Диаграмма вариантов использования
use case diagram

Структура

Диаграмма классов
class diagram

Поведение

Диаграмма состояний
statechart diagram

Диаграмма кооперации
collaboration diagram

Диаграмма последовательности
sequence diagram

Диаграмма деятельности
activity diagram

Реализация

Диаграмма компонентов
component diagram

Диаграмма развертывания
deployment diagram

Диаграмма кооперации

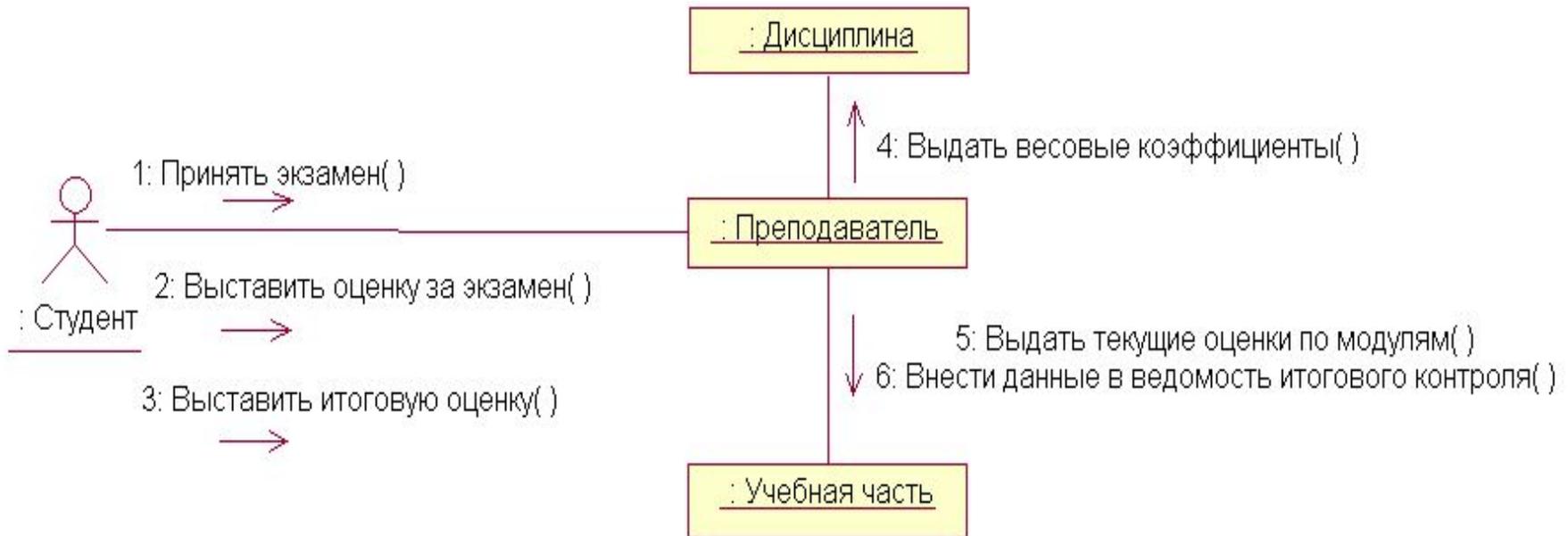


Определяет динамические свойства предметной области в виде взаимодействия между объектами в конкретном варианте использования

Взаимодействие - передача сообщений между объектами. В ответ на полученное сообщение объект выполняет операцию и может изменить свое состояние.

Состояние объекта – текущее значение его атрибутов.

Изменение состояния объекта может привести к передаче сообщения другому объекту



UML	Сообщение
Объектные языки	Вызов метода

Интегрированная модель системы

Интегрированная модель

Функциональность

Диаграмма вариантов использования
use case diagram

Структура

Диаграмма классов
class diagram

Поведение

Диаграмма состояний
statechart diagram

Диаграмма кооперации
collaboration diagram

Диаграмма последовательности
sequence diagram

Диаграмма деятельности
activity diagram

Реализация

Диаграмма компонентов
component diagram

Диаграмма развертывания
deployment diagram

Постановка задачи

Приложение предназначено для автоматизации процессов продажи билетов для проезда на поездах частной транспортной компании “Рога и копыта”.

Обслуживание клиентов производится в офисе частной компании, занимающем одно помещение. Клиент может приобретать билет на нужный ему поезд и дату выезда по предъявлению паспорта при наличии свободных мест. Приобретенный билет может быть сдан при предъявлении паспорта с удержанием части стоимости билета в соответствии с действующим тарифом – 1 процент от стоимости билета. Для выполнения покупки или сдачи билета клиент обращается к диспетчеру, который направляет его к кассиру по возврату билетов или к одному из двух кассиров по продаже билетов.

Любая из касс по указанию администрации может закрываться на технологический перерыв длительностью 15 минут. Перед закрытием завершается обслуживание клиента, заявка которого уже принята. Очередь заявок при закрытии кассы не сохраняется.

Одновременное закрытие касс по продаже билетов не допускается. Для приобретения билета запрос клиента направляется в открытую кассу по продаже билетов с меньшей очередью.

По окончании смены по указанию администрации кассы закрываются, при этом кассиры по продаже билетов представляют отчет о проданных билетах. Закрытие касс производится только после завершения обслуживания уже принятой заявки. Очередь заявок при закрытии кассы по окончании смены не сохраняется.

Интегрированная модель системы

Интегрированная модель

Функциональность

Диаграмма вариантов использования
use case diagram

Структура

Диаграмма классов
class diagram

Поведение

Диаграмма состояний
statechart diagram

Диаграмма кооперации
collaboration diagram

Диаграмма последовательности
sequence diagram

Диаграмма деятельности
activity diagram

Реализация

Диаграмма компонентов
component diagram

Диаграмма развертывания
deployment diagram