

# Презентационный материал

по курсу лекций «Проектирование информационных систем»

С.П. Строев

Орловский государственный университет имени И. С. Тургенева

ОГУ им. И.С. Тургенева, 2020 г.

Дисциплина «Проектирование информационных систем»:

- 24 лекции; 10 лабораторных занятий;
- экзамен в форме тестирования;
- курсовая работа.

Необходимые информационные технологии:

- язык программирования Delphi (возможно Python) ;
- CASE-технологии: Ramus Education, umbrello, [app.creately.com](https://app.creately.com);
- курсовая работа.

## Жизненный цикл ИС

Совокупность взаимосвязанных процессов создания и последовательного изменения состояния ИС, от формирования исходных требований к ней до окончания эксплуатации и утилизации ИС.

## Нормативная база

- ГОСТ 19.102-77 Стадии разработки;
- ГОСТ 34.601-90 Автоматизированные системы. Стадии создания;
- ГОСТ Р ИСО/МЭК 15271-02 Процессы жизненного цикла программных средств;
- ГОСТ Р ИСО/МЭК 12207-2010 Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств

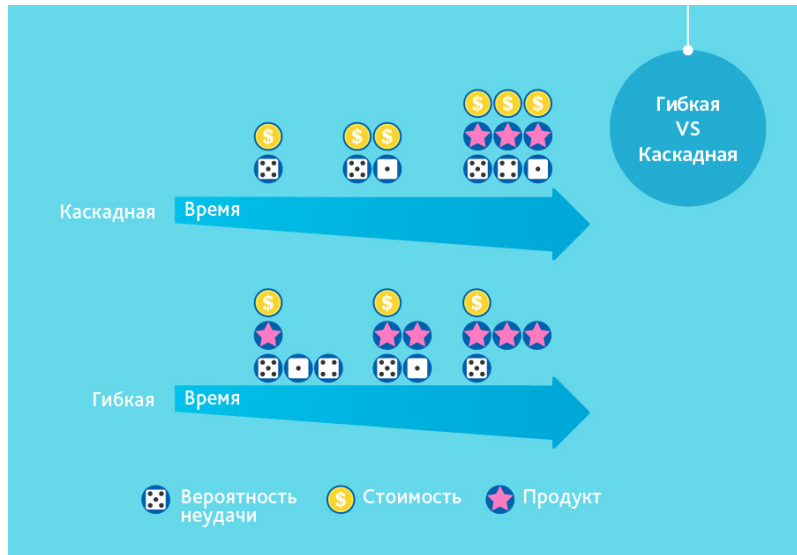
## Модели жизненного цикла по ГОСТ Р ИСО/МЭК 15271-02

- каскадная;
- инкрементная;
- эволюционная

## Другие модели жизненного цикла

- V-модель;
- RAD Model (rapid application development model или быстрая разработка приложений);
- Agile Model (гибкая методология разработки);
- Iterative Model (итеративная или итерационная модель)

# Основы::Жизненный цикл ИС::Модели



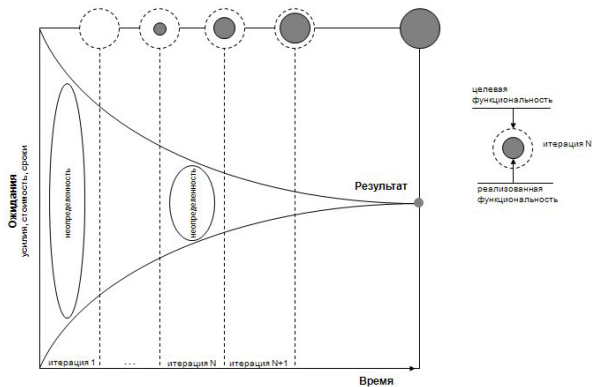
## Каскадная модель (модель «водопада»)

Каскадная модель жизненного цикла — модель, реализующая принцип однократного выполнения каждого из видов деятельности в их естественных границах.



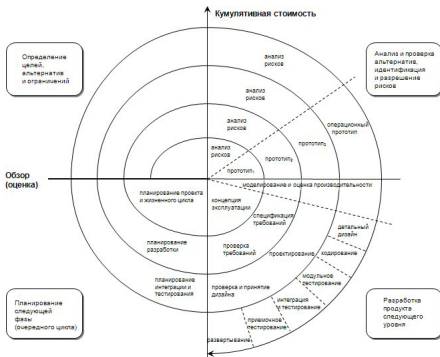
## Инкрементная модель

Инкрементная модель жизненного цикла — модель, предполагающая разделение полных требований к системе на различные сборки



## Эволюционная модель

Эволюционная модель жизненного цикла — модель, представляющая разработку системы в виде отдельных конструкций, но в **отличие от инкрементной модели** требования изначально не могут быть полностью осознаны и установлены.





## Содержание ЖЦ ИС по ГОСТ–12207

Группа процессов	Процессы	Работы	Задачи
Основная	5	35	135
Вспомогательная	8	25	70
Организационная	4	14	27
Всего	17	74	232

## Основной процесс::Заказ::Работы

подготовка; подготовка заявки на подряд; подготовка и корректировка договора; надзор за поставщиком; приемка и закрытие договора

## Основной процесс::Поставка::Работы

подготовка; подготовка ответа; подготовка договора; планирование; выполнение и контроль; проверка и оценка; поставка и закрытие договора

## Основной процесс::Разработка::Работы

подготовка процесса; анализ требований к системе; проектирование системной архитектуры; анализ требований к программным средствам; проектирование программной архитектуры; техническое проектирование программных средств; программирование и тестирование программных средств; сборка программных средств; квалификационные испытания программных средств; сборка системы; квалификационные испытания системы; ввод в действие программных средств; обеспечение приемки программных средств

## Основной процесс::Эксплуатация::Работы

подготовка процесса; эксплуатационные испытания; эксплуатация системы; поддержка пользователя

## Основной процесс::Сопровождение::Работы

подготовка процесса; анализ проблем и изменений; внесение изменений; 4) проверка и приемка при сопровождении; перенос; снятие с эксплуатации

## Типы отношений:

- ассоциация между действующим лицом и вариантом использования;
- обобщение между действующими лицами;
- обобщение между вариантами использования;
- зависимости между вариантами использования.

# UML::Диаграмма вариантов использования::Отношения

Типы отношений:

- ассоциация между действующим лицом и вариантом использования;
- обобщение между действующими лицами;
- обобщение между вариантами использования;
- зависимости между вариантами использования.

# UML::Диаграмма вариантов использования::Отношения

## Ассоциация

Ассоциация между действующим лицом и вариантом использования показывает, что действующее лицо тем или иным способом взаимодействует с вариантом использования<sup>а</sup>.

<sup>а</sup>Единственное обязательное отношение

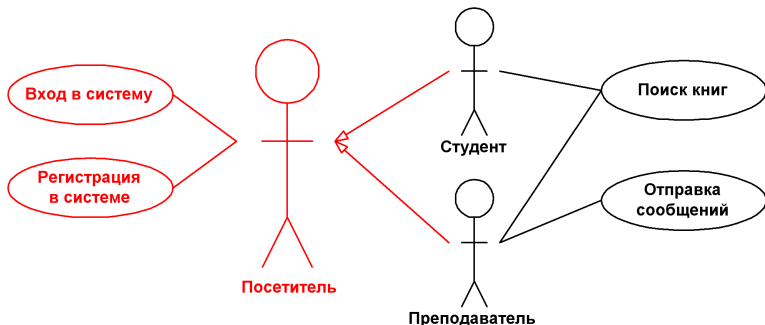


# UML::Диаграмма вариантов использования

## использования::Отношения

### Обобщение

Обобщение между действующими лицами показывает, что одно действующее лицо наследует все свойства (в частности, участие в ассоциациях) другого действующего лица.



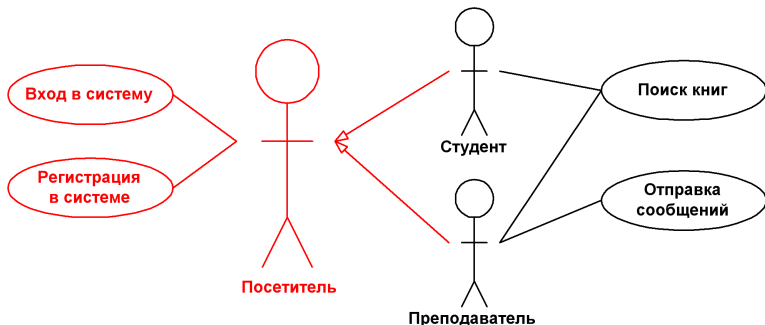


# UML::Диаграмма вариантов использования

## использования::Отношения

### Включение

Обобщение между действующими лицами показывает, что одно действующее лицо наследует все свойства (в частности, участие в ассоциациях) другого действующего лица.



## Определение 0.1

Класс — это множество объектов, обладающих общей структурой, поведением и семантикой.

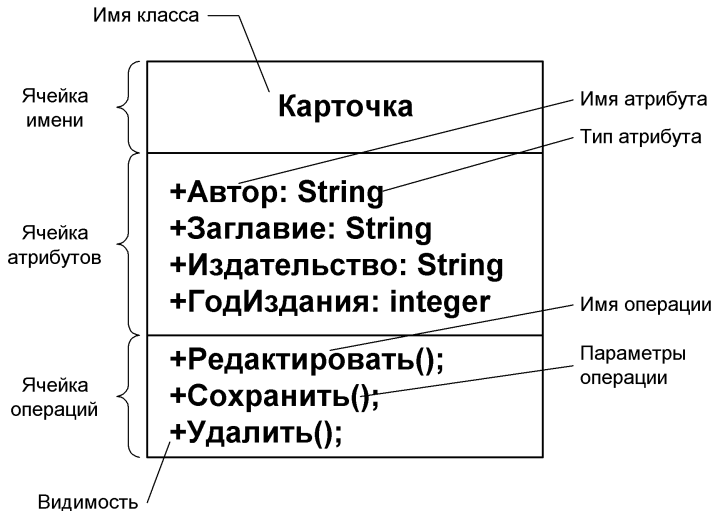
## Определение 0.2

Диаграмма классов показывает классы и их отношения в логической схеме проекта.

Основные элементы нотации диаграммы классов:

- класс;
- отношение.

# UML::Диаграмма классов::Нотация класса



## Синтаксис имени класса

«стереотип» ИмяКласса кратность

## Синтаксис атрибута класса

видимость ИмяАтрибута кратность : тип = начальное значение

## Синтаксис операции класса

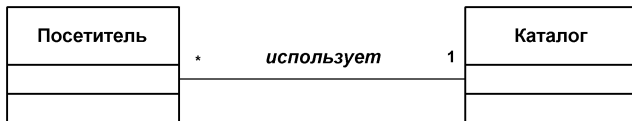
видимость ИмяОперации(направление Параметр : тип = значение): тип

## Отношения между классами

- ассоциация;
- обобщение;
- агрегация;
- композиция (объединение).

## Ассоциация

Ассоциация соединяет два класса и означает семантическую связь между ними.



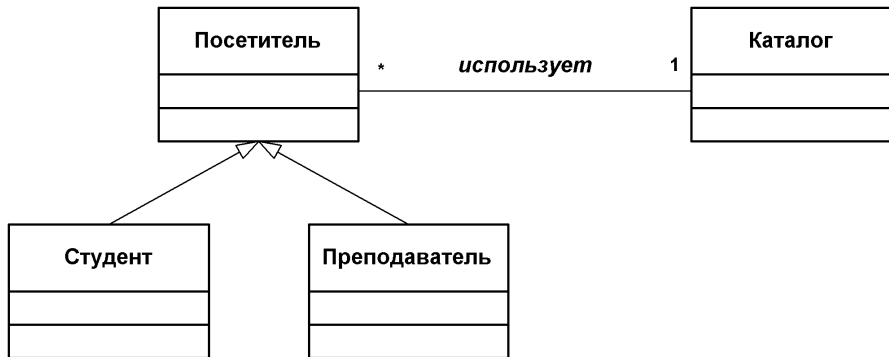
## Кратность

Кратность — указывает количество связей между каждым экземпляром класса с экземплярами целевого класса:

- 1 одна связь;
- \* неограниченное количество;
- $a...b$  конкретный диапазон.

## Обобщение

Обобщение — это отношение между более общей сущностью и более специальной сущностью (отношение «является»).



## Агрегация

Агрегация является слабоопределенным типом ассоциации и отражает самую общую зависимость типа «часть-целое»



## Композиция

Композиция между классами означает, что объекты одного класса входят состав другого класса и принадлежат только ему.

## Свойства композиции

- экземпляр части в каждый момент времени принадлежит только одному целому;
- целое ответственно за создание и удаление своих частей;
- при уничтожении целого его части либо уничтожаются, либо присоединяются к другому.

## Класс анализа

Классы анализа — это классы, которые:

- представляют четкую абстракцию предметной области;
- должны проецироваться на реальные бизнес-понятия.

## Нотация класса анализа

- Имя класса — обязательно;
- Атрибуты класса — указываются наиболее значимые;
- Операции класса — могут указываться наиболее значимые.

## Признаки хорошего класса анализа

- класс представляют четкую абстракцию предметной области;
- классы должны проецироваться на реальные бизнес-понятия;
- имя класса отражает его назначение;
- класс является четкой абстракцией, моделирующей один конкретный элемент предметной области;
- класс проецируется на четко определяемую возможность предметной области;
- класс имеет небольшой четко определенный набор обязанностей;
- высокая внутренняя связность;
- низкая связанность с другими классами.

## Рекомендации по созданию классов анализа

- в каждом классе должно быть три-пять обязанностей;
- ни один класс не является изолированным;
- остерегайтесь создания множества очень мелких классов;
- не создавайте «всемогущие» классы;
- необходимо избегать «глубоких» деревьев наследования.

## Рекомендации по выявлению классов анализа

- выявление классов с помощью анализа существительное/глагол;
- выявление классов с помощью CRC-анализа (карточки);
- выявление классов путем применения стереотипов RUP (граница, управление, сущность);
- базовые шаблоны.

## Реализация прецедентов

Реализации прецедентов показывают, как взаимодействуют классы, чтобы реализовать функциональность системы.

## Задачи реализации прецедентов

- взаимодействие каких классов анализа обеспечивает поведение, определенное прецедентом (могут быть новые классы);
- какими сообщениями должны обмениваться экземпляры этих классов для реализации заданного поведения (атрибуты, операции, отношения);
- скорректировать модель прецедентов, классов.

## Диаграммы взаимодействия

Диаграммы взаимодействий показывают, как взаимодействуют экземпляры классификаторов для реализации поведения системы.

## Типы диаграмм взаимодействия

- диаграмма последовательностей (sequence diagram);
- коммуникационная диаграмма (communication diagram);
- диаграмма обзора взаимодействия;
- временная диаграмма.

## Элементы диаграмм взаимодействия

- линия жизни (lifeline);
- сообщение (message).

## Линия жизни

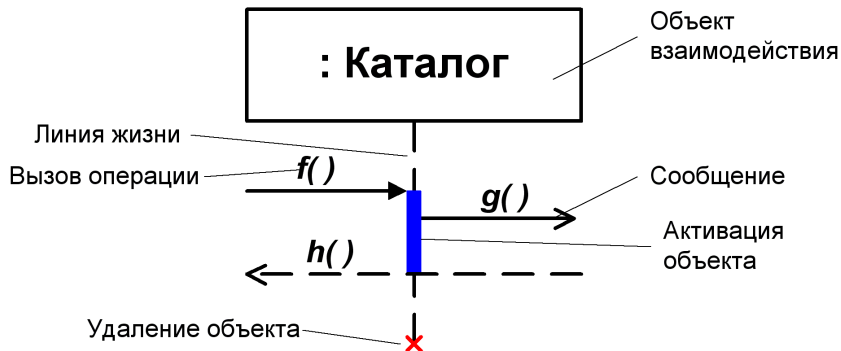
Линия жизни представляет объект во взаимодействии.



# UML::Реализация прецедентов::Моделирование взаимодействия

## Линия жизни

Линия жизни представляет объект во взаимодействии.



# UML::Реализация прецедентов::Моделирование взаимодействия

## Сообщение

Сообщение — это особый вид коммуникации между линиями жизни.

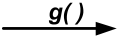
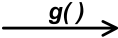
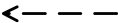
## Назначение сообщений

- вызов операции;
- создание или уничтожение объекта;
- отправка сигнала.

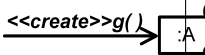
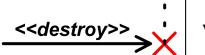
## Сообщение

- сообщение вызова, получаемое линией жизни, является запросом на вызов операции, имеющей аналогичную сообщению сигнатуру.

# UML::Реализация прецедентов::Моделирование взаимодействия

Синтаксис	Имя	Семантика
	Синхронное сообщение	Отправитель ожидает завершения выполнения сообщения получателем
	Асинхронное сообщение	Отправитель посылает сообщение и продолжает исполнение — он не ожидает возврата от получателя
	Возврат	Получатель сообщения возвращает фокус управления отправителю этого сообщения

# UML::Реализация прецедентов::Моделирование взаимодействия

Синтаксис	Имя	Семантика
	Создание объекта	Отправитель создает экземпляр классификатора, определенного получателем
	Уничтожение объекта	Отправитель уничтожает получателя

# UML::Моделирование поведения::Диаграмма деятельности

## Диаграмма деятельности

Диаграмма деятельности (activity diagram) — средство описания поведения в UML.


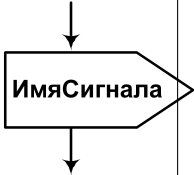
## Применение

- на этапе анализа:
  - для графического моделирования потока прецедента;
  - для моделирования потока между прецедентами;
- на этапе проектирования:
  - для моделирования деталей операции;
  - для моделирования деталей алгоритма;
- при моделировании деловой активности:
  - для моделирования бизнес-процесса.

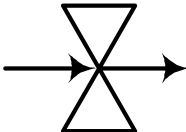
## Элементы нотации диаграммы деятельности

- узел действий (action node);
- узлы управления (control nodes):
  - начальные и конечные;
  - принятия решений и слияния (decision and merge);
  - ветвления и объединения (fork and join);
- узлы объектов (object nodes).

# UML::Реализация прецедентов::Моделирование взаимодействия


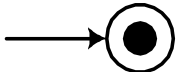
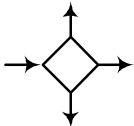
Синтаксис	Имя	Семантика
	Узел вызова действия	Иницирует деятельность, поведение или операцию
	Посылка сигнала	Действие посылки сигнала — посылает сигнал асинхронно (отправитель не ожидает подтверждения получения сигнала)

# UML::Реализация прецедентов::Моделирование взаимодействия

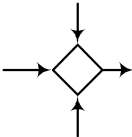
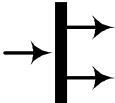
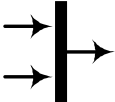
Синтаксис	Имя	Семантика
 <p>ПринятьСобытие</p>	Узел действия, принимающий событие	Принимает событие — ожидает события, установленного объектом-владельцем, и выдает событие на выходе
	Узел действия, принимающий событие времени	Принимает событие времени — отвечает на определенное значение времени



# UML::Реализация прецедентов::Моделирование взаимодействия

Синтаксис	Имя	Семантика
	Начальный узел	Указывает, где начинается поток при вызове деятельности
	Конечный узел деятельности	Завершает деятельность
	Узел решения	Поток проходит по исходящему ребру, сторожевое условие которого истинно

# UML::Реализация прецедентов::Моделирование взаимодействия

Синтаксис	Имя	Семантика
	Узел слияния	Копирует входные маркеры в единственное выходное ребро
	Узел ветвления	Разделяет поток на несколько параллельных потоков
	Узел объединения	Синхронизирует несколько параллельных потоков