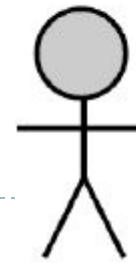


Диаграмма прецедентов

Эктор



- Любые (в том числе и программные) системы проектируются с учетом того, что в процессе своей работы они будут использоваться людьми и/или **взаимодействовать** с другими системами.
- Сущности, с которыми взаимодействует система в процессе своей работы, называются **экторами**.
- **Эктор (actor)** - это множество логически связанных ролей, исполняемых при взаимодействии с прецедентами или сущностями (система, подсистема или класс).
- Эктором может быть человек или другая система, подсистема или класс, которые представляют нечто вне сущности.
- эктор образовано от слова **action**, что в переводе означает *действие*. Дословный же перевод слова "эктор" - *действующее лицо*



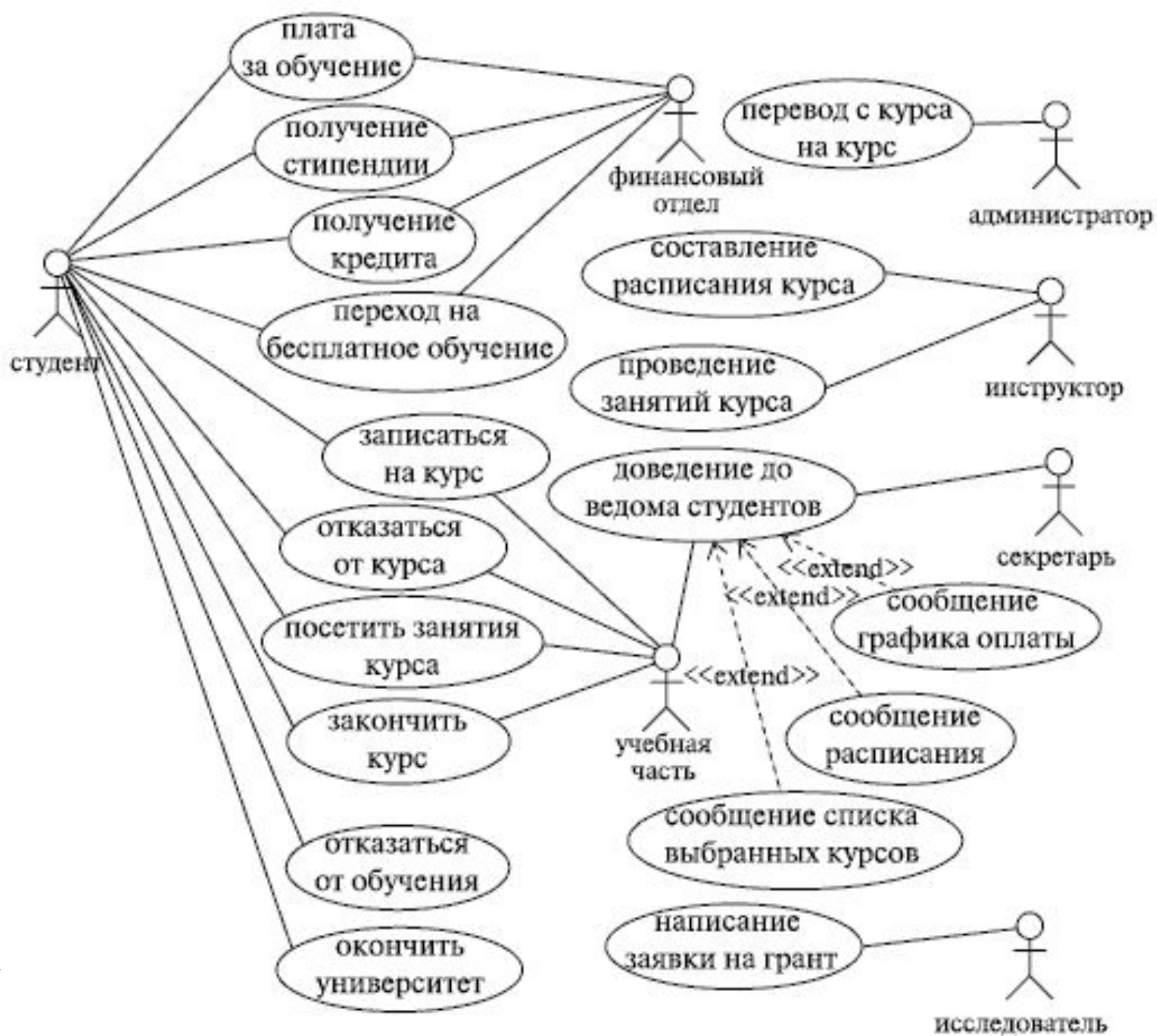
Прецедент

- ▣ **Прецедент (use-case)** - описание отдельного аспекта поведения системы с точки зрения пользователя (Буч).
 - ▣ **Прецедент (use case)** - описание множества последовательных событий (включая варианты), выполняемых системой, которые приводят к наблюдаемому эктором результату. Прецедент представляет поведение сущности, описывая взаимодействие между экторами и системой. Прецедент не показывает, "как" достигается некоторый результат, а только "что" именно выполняется.
 - ▣ *Прецеденты и экторы соединяются с помощью линий.*
-



Примеры диаграммы прецедентов





Цели создания диаграмм прецедентов:

- определение границы и контекста моделируемой предметной области на ранних этапах проектирования;
- формирование общих требований к поведению проектируемой системы;
- разработка концептуальной модели системы для ее последующей детализации;
- подготовка документации для взаимодействия с заказчиками и пользователями системы.

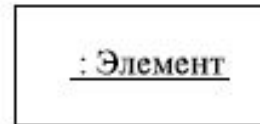
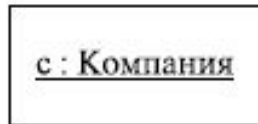
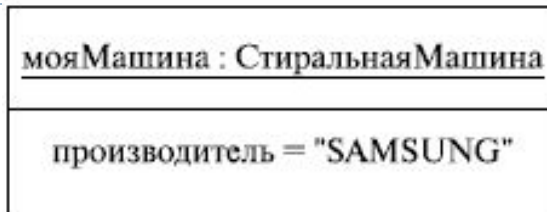




Диаграмма объектов



Объект

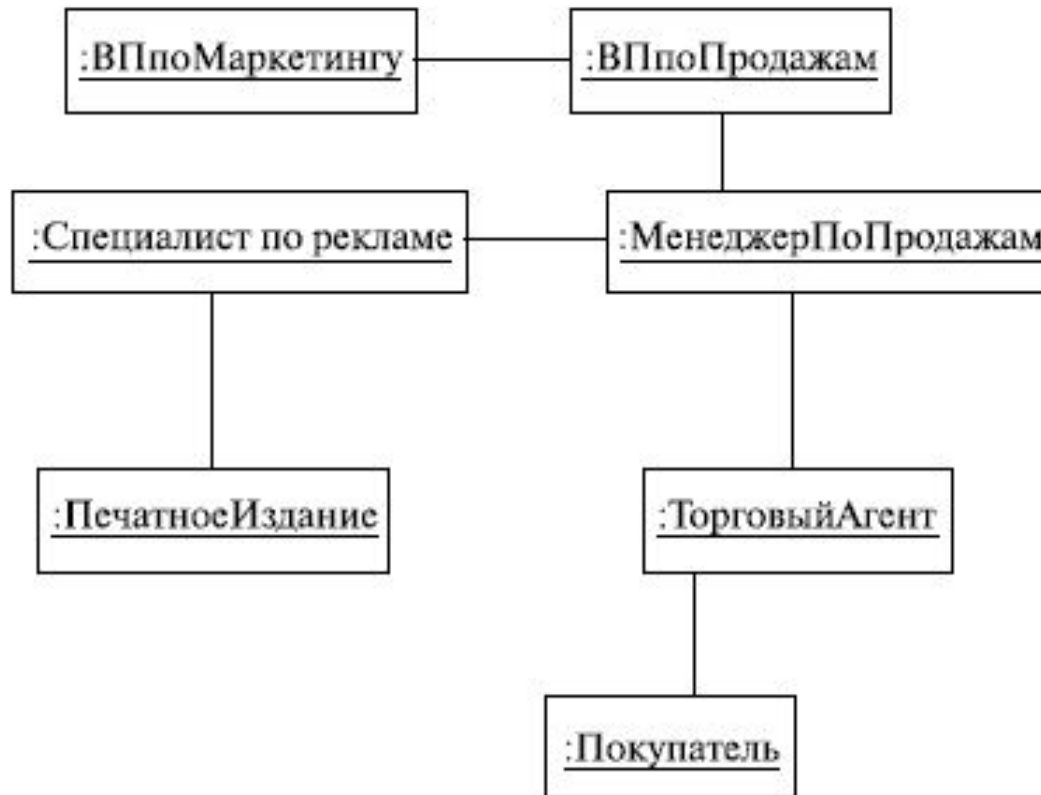


- **Объект (object)** - экземпляр класса.
- **Объект уникально идентифицируется значениями атрибутов, определяющими его состояние в данный момент времени.**
- Например, все мы являемся объектами класса "человек" и различимы между собой по таким признакам (значениям атрибутов), как имя, цвет волос, глаз, рост, вес, возраст и т. д.
- Объект может быть анонимным: это нужно в том случае, если в данный момент не важно, какой именно объект данного класса принимает участие во взаимодействии



Диаграмма объектов

- *Диаграммы объектов* представляют статический вид системы с точки зрения проектирования и процессов, являясь основой для сценариев, описываемых диаграммами взаимодействия



Пример диаграммы объектов

