

# Лекция 2

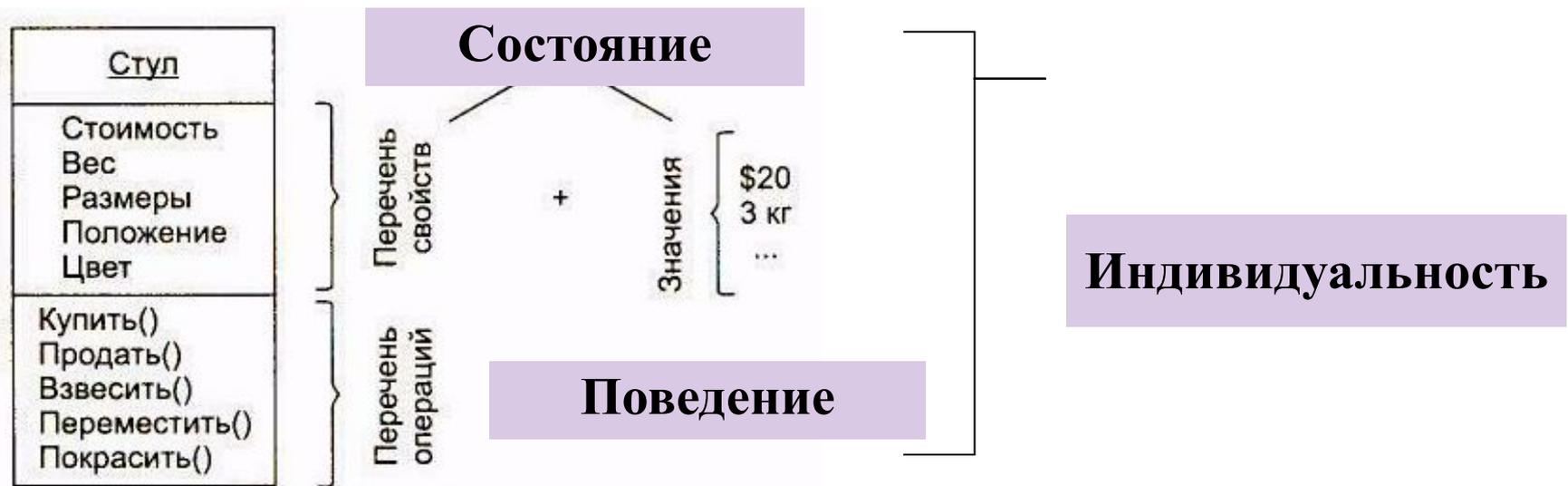
## Тема: «Объектно-ориентированное проектирование ИС»

### Вопросы:

1. Основы объектно-ориентированного проектирования ИС.
2. Принципы объектно-ориентированного представления ИС.
3. Представление проектных решений в виде UML-диаграмм.
4. Инструментальные средства объектно-ориентированного проектирования ИС.

# Общая характеристика объектов

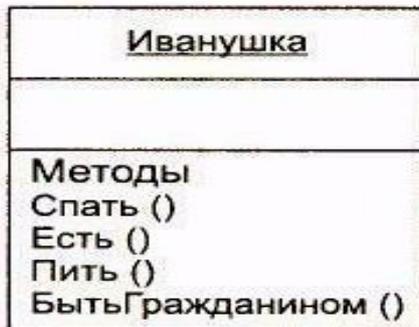
**Объект** – это конкретное представление абстракции, экземпляр класса.



# Виды операций клиента над объектом

Вид операции	Значение операции	Пример операции
Модификатор	Изменяет состояние объекта	Пополнить (кг)
Селектор	Дает доступ к состоянию, но не изменяет его	КакойВес () : integer
Итератор	Дает доступ к содержанию объекта по частям, в строго определенном порядке	ПоказатьКаталогТоваров () : string
Конструктор	Создает объект и инициализирует его состояние	СоздатьРобот (параметры)
Деструктор	Разрушает объект и освобождает занимаемую им память	УничтожитьРобот ()

# Характеристика объектов



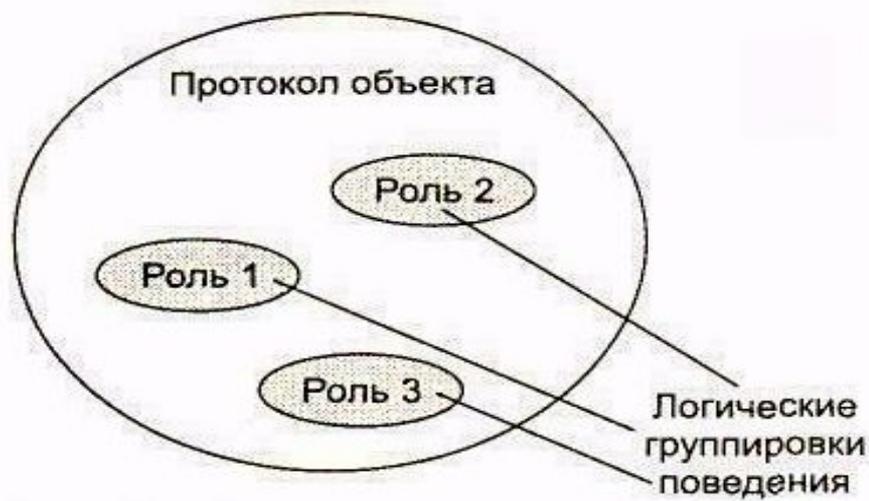
**Методы объекта**

Свободные подпрограммы

Голосовать (имя)

ВыражатьМнение (имя)

**Свободные подпрограммы**



**Протокол и роли**

# Виды отношений между объектами

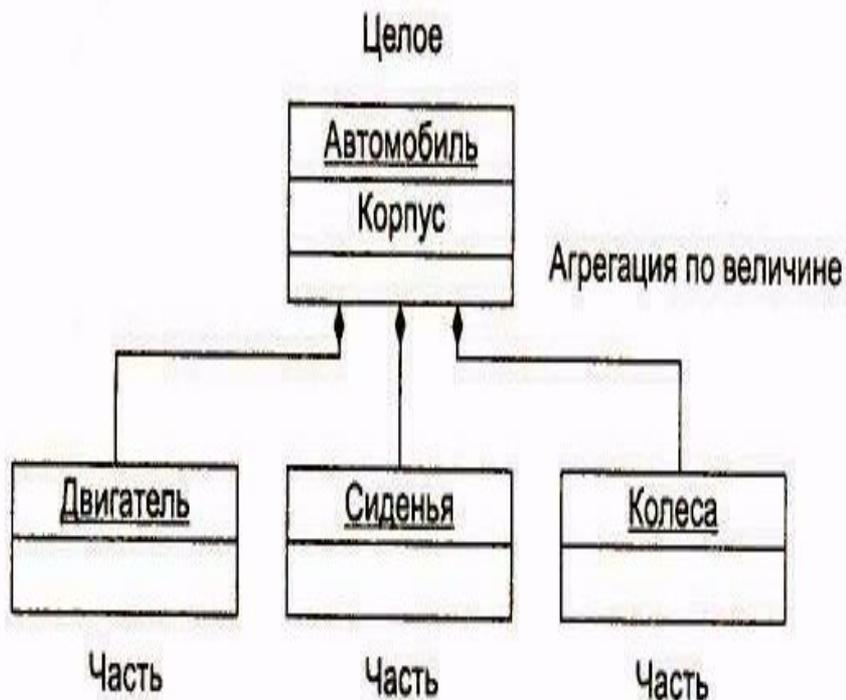
**Связь** – это физическое или понятийное соединение между объектами.

**Как участник связи объект может играть одну из ролей:**

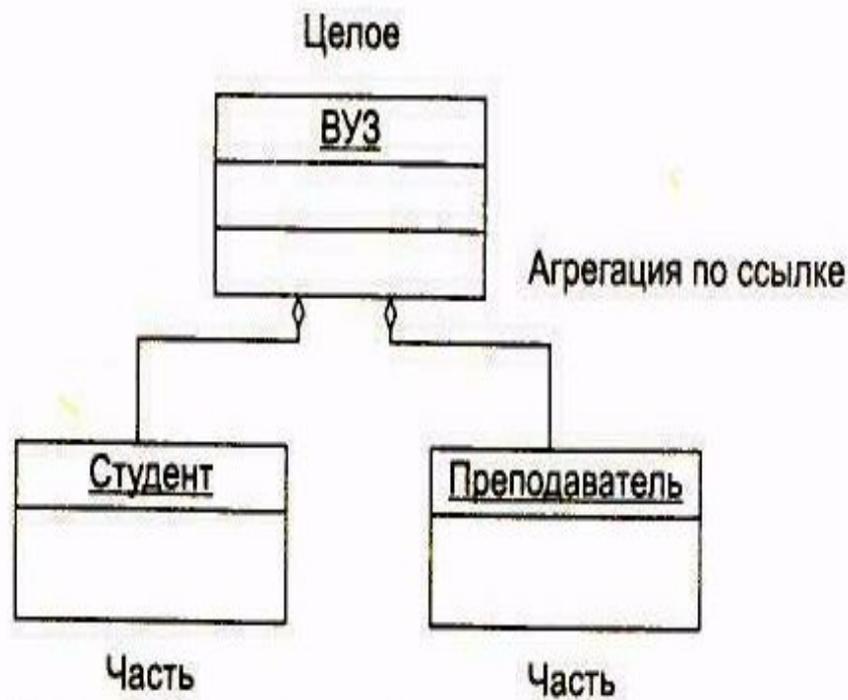
- **актер** – объект, который может воздействовать на другие объекты, но никогда не подвержен воздействию других объектов;
- **сервер** – объект, который никогда не воздействует на другие объекты, он только используется другими объектами;
- **агент** – объект, который может как воздействовать на другие объекты, так и использоваться ими. Агент создается для выполнения работы от имени актера или другого агента.

# Виды отношений между объектами

**Агрегация** – это отношение объектов в иерархии «целое/часть».



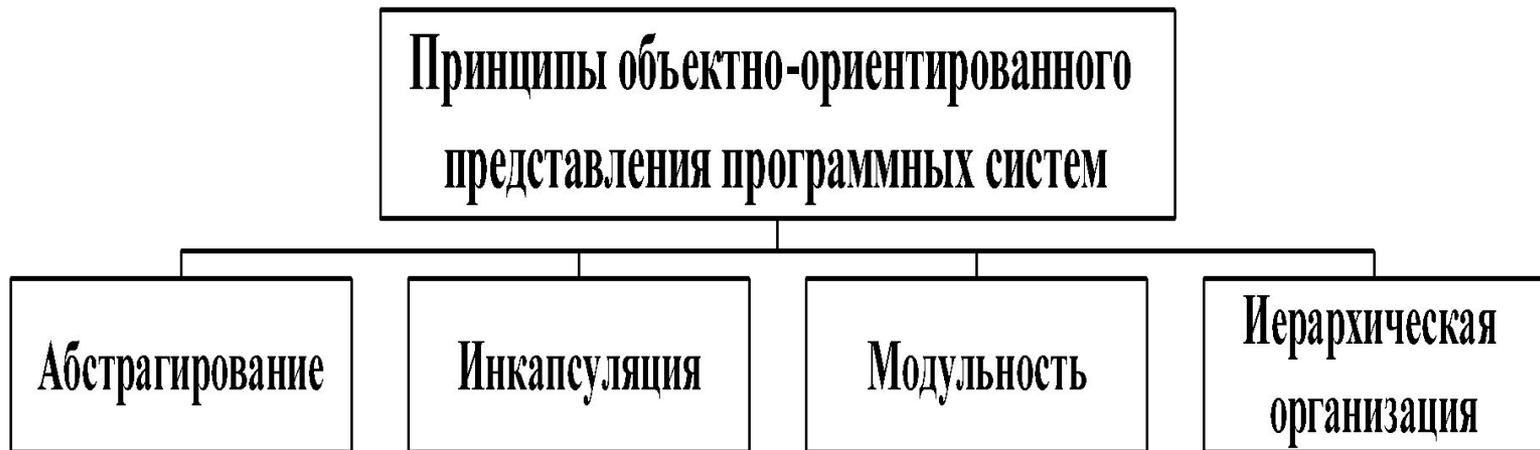
**Физическое включение  
частей в агрегат**



**Нефизическое включение  
частей в агрегат**

# Принципы ООП ПС

**Объектно-ориентированное представление ПС (ООП ПС)** основывается на принципе объектной декомпозиции, при которой структура системы описывается в терминах объектов и связей между ними, а ее поведение описывается в терминах обмена сообщениями между объектами.



# Абстрагирование и инкапсуляция

**Инкапсуляция и абстракция** – взаимодополняющие понятия: **абстракция** выделяет внешнее поведение объекта, а **инкапсуляция** содержит и скрывает реализацию, которая обеспечивает это поведение.

## Модульность

**Модульность** – это свойство системы, которая может подвергаться декомпозиции на ряд внутренне связанных и слабо зависящих друг от друга модулей.

# Свойства модулей

**Информационная  
закрытость**

**Связность модуля**

**Сцепление модулей**

Содержание модулей должно быть скрыто друг от друга.

Все модули независимы, обмениваются только информацией, необходимой для работы.

# Свойства модулей

**Информационная  
закрытость**

**Связность модуля**

**Сцепление модулей**

*Связность модуля* –  
это мера зависимости его  
частей модуля.

# Виды связности модуля

1.	<b>Связность по совпадению (СС=0)</b>	<b>Части модуля вообще не имеют никаких отношений друг с другом.</b>
2.	<b>Логическая связность (СС=1)</b>	<b>Части модуля объединены по принципу функционального подобия, принадлежат к действиям одной категории.</b>
3.	<b>Временная связность (СС=3)</b>	<b>Части модуля не связаны, но необходимы в один и тот же период работы системы.</b>
4.	<b>Процедурная связность (СС=5)</b>	<b>Части модуля связаны порядком выполняемых ими действий, реализующих некоторый сценарий поведения.</b>
5.	<b>Коммуникативная связность (СС=7)</b>	<b>Части модуля связаны по данным – используют одни и те же данные.</b>
6.	<b>Информационная (последовательная) связность (СС=9)</b>	<b>Выходные данные одной части используются как входные данные в другой части модуля.</b>
7.	<b>Функциональная связность (СС=10)</b>	<b>Части модуля вместе реализуют одну функцию.</b>

# Свойства модулей

**Информационная  
закрытость**

**Связность модуля**

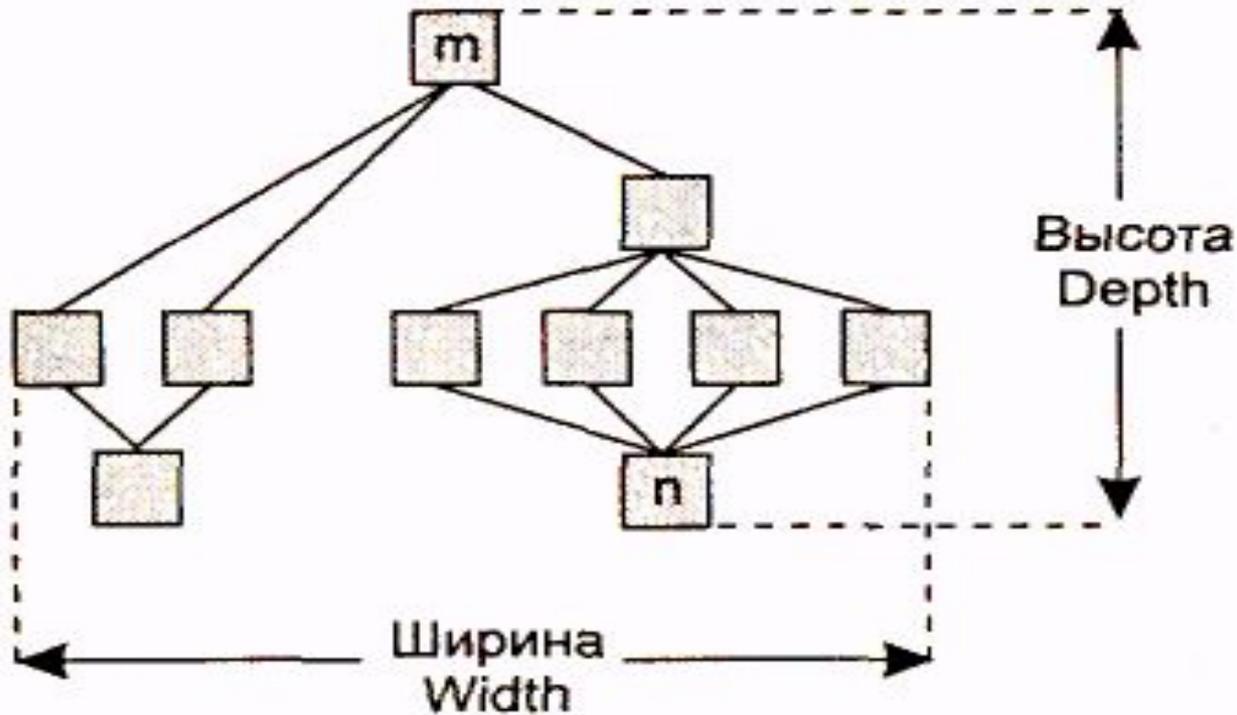
**Сцепление модулей**

*Сцепление* – это мера  
взаимозависимости  
модулей по данным.

# Виды сцепления модулей

1.	Сцепление по данным (СЦ=1)	Модуль А вызывает модуль В. Все входные и выходные параметры вызываемого модуля – простые элементы данных.
2.	Сцепление по образцу (СЦ=3)	В качестве входных параметров используются структуры данных.
3.	Сцепление по управлению (СЦ=4)	Модуль А явно управляет функционированием модуля В, посылая ему управляющие данные.
4.	Сцепление по внешним ссылкам (СЦ=5)	Модули А и В ссылаются на один и тот же глобальный элемент данных.
5.	Сцепление по общей области (СЦ=7)	Модули разделяют одну и ту же глобальную структуру данных.
6.	Сцепление по содержанию (СЦ=9)	Один модуль прямо ссылается на содержание другого модуля. Например, коды их команд перемежаются друг с другом.

# Иерархическая организация



Первичные характеристики – *количество вершин* (модулей) и *количество ребер* (связей между модулями).

Глобальные характеристики – *высота* и *ширина*.

Локальные характеристики – *коэффициент объединения по входу*  $Fan\_in(i)$  и *коэффициент разветвления по выходу*  $Fan\_out(i)$ .

# Типовая последовательность построения основных диаграмм UML

