

ООП. Лекция *3

1. Отношения между классами

Известные типы отношений между классами

1. От частного к общему
... is a ...
Обобщение.



Mac

Ноутбук



2. От целого к части
... has a ...
Класс является частью другого класса.

- Процессор
 - Опер.память
 - Видеокарта
- = ...



Часть
Mac



3. Ассоциация



ton-baron.com.ua

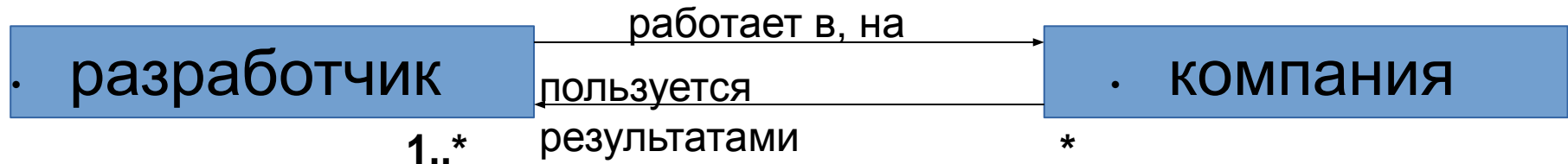


МКОК.рф

- Языки программирования (ЯП) выработали несколько общих подходов к выражению *отношений этих 3 типов*.
- Большинство объектно-ориентированных ЯП поддерживают следующие отношения между классами:
 - 1. **ассоциация** → объекты классов вступают во взаимоотношения между собой
 - 2. **наследование** (обобщение) → объект дочернего класса наследует все свойства родительского класса
 - 3. **агрегация** → объекты одного класса входят в объекты другого класса
 - 4. **композиция** (строгая агрегация) → объекты одного класса входят в объекты другого класса и зависят друг от друга по времени жизни
 - 5. **класс – метакласс** → отношение, при котором объектами одного класса являются другие классы.

Ассоциация

- Это тип отношений, который связан с передачей сообщения путем делегирования.
- Обычно аналитика констатирует наличие ассоциации и, постепенно уточняя, превращает ее в какую-то связь.



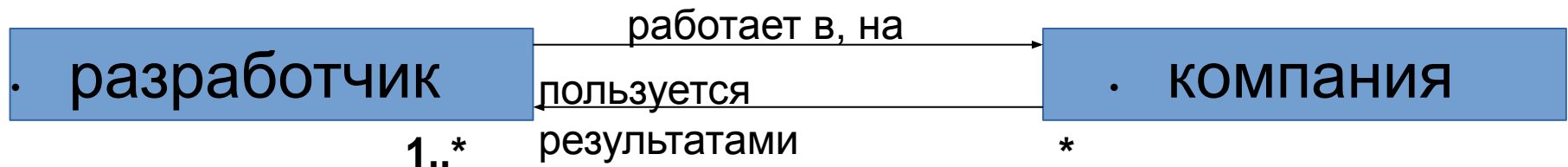
- Ассоциация — это пример связи между классами, по которой можно перемещаться между этими классами.
- Ассоциация может иметь имя, показывать природу отношений между классами.

Бинарная ассоциация
— связь между 2
классами

Связь между большим
количеством классов —
N-арная ассоциация

Ассоциация: кратность

- Часто будет необходимо указать, сколько объектов может быть связано посредством одной экземпляра ассоциации, т. е. одной связи.
- Это число называется кратностью роли ассоциации и записывается в виде значения, которое представляет собой диапазон. 0..1 1..* 1..10



-
- Указанная кратность на одном конце ассоциации говорит, что столько объектов (1..*) должно соответствовать каждому объекту на другом конце (*).

Ассоциация

- Таким образом ассоциация — просто смысловая связь, которая не указывает направление, не указывает, как классы взаимодействуют друг с другом.
- Ассоциация лишь указывает роли, которые классы играют друг для друга.
- Именно ассоциация нужна для анализа на ранней стадии в области исследований.

Наследование

Используя наследование (иерархичность), мы решаем следующие задачи:

- избавляемся от избыточности кода
- защищаем данные в каждом классе.

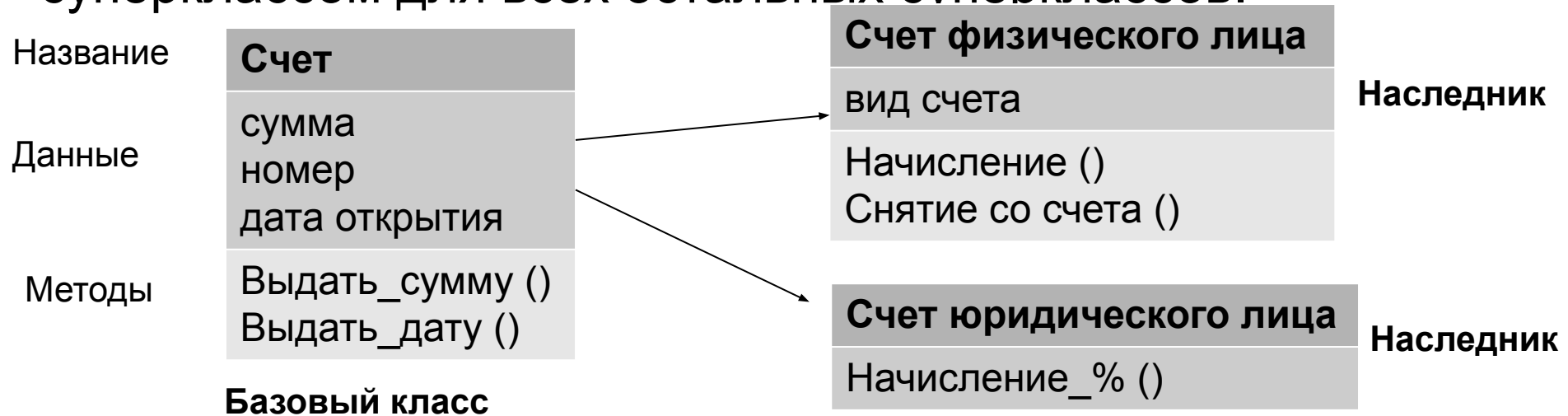
Одиночное наследование — один класс повторяет структуру другого класса.

Супер класс — класс, структура которого наследуется.

Подкласс — производный класс от суперкласса. Подкласс обычно расширяет либо ограничивает структуру суперкласса.

Абстрактный класс — класс, объекты которого не создаются.

В C# определен один базовый класс Object, который является суперклассом для всех остальных суперклассов.



Использование

- Отношение использования — наиболее изменчивый тип отношений между классами.
- Один класс использует другой, если при выполнении действий он опирается на свойства объектов другого класса.
- Использование между классами может быть клиент-сервер — когда при ассоциации один из классов, называемый *клиентом*, пользуется услугами другого класса — *сервера*.
- Строгое использование иногда ограничивается, т.е. клиент имеет доступ только к конкретной части интерфейса сервера.
- Вызов методов другого класса является использованием.

Пример: При отрисовке фигуры задается объект, использующий экран монитора, т.е. класс «Фигура» использует класс «Монитор».

Инстанцирование

- Параметризованный класс называют шаблоном.
- Например, если класс *очередь* — шаблон, то может создать очередь целых чисел, очередь экранных объектов, очередь задач, выполняемых программой. Объекты этих классов различны, поэтому классы не могут иметь общий супер класс.