

Лекция №2

Объектно-ориентированное программирование.

Вопросы

- Что такое ООП?
- Связи между классами/объектами;
- Абстракция.

Программа ВЫГЛЯДИТ как-ТО ТАК

```
0000000: cffa edfe 0700 0001 0300 0080 0200 0000 .....
0000010: 1000 0000 f006 0000 8500 2000 0000 0000 .....
0000020: 1900 0000 4800 0000 5f5f 5041 4745 5a45 ....H...__PAGEZE
0000030: 524f 0000 0000 0000 0000 0000 0000 0000 RO.....
0000040: 0000 0000 0100 0000 0000 0000 0000 0000 .....
0000050: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000060: 0000 0000 0000 0000 1900 0000 7802 0000 .....x...
0000070: 5f5f 5445 5854 0000 0000 0000 0000 0000 __TEXT.....
0000080: 0000 0000 0100 0000 0070 0000 0000 0000 .....p.....
0000090: 0000 0000 0000 0000 0070 0000 0000 0000 .....p.....
00000a0: 0700 0000 0500 0000 0700 0000 0000 0000 .....
00000b0: 5f5f 7465 7874 0000 0000 0000 0000 0000 __text.....
00000c0: 5f5f 5445 5854 0000 0000 0000 0000 0000 __TEXT.....
00000d0: 1810 0000 0100 0000 0849 0000 0000 0000 .....I.....
```

Задача

Спроектировать программу для анализа котировок акций.

Необходимо ответить на вопрос:

Что делает программа?

Одно из возможных решений

- Получить данные о цене акций;
- Проанализировать акции;
- Сортировать акции согласно выбранным метрикам;
- Вывести список рекомендаций.

-
- `retrieveQuotes();`
 - `analyzeQuotes();`
 - `sortQuotes();`
 - `outputRecommendations();`

Процедурный способ

1. Процедурный способ проектирования программы хорошо работает, когда есть линейная последовательность действий;
2. Процедурный способ не предполагает ничего о способе представления данных в программе.

Данные и функции их обработки разделены в коде программы.

Объектно-ориентированный способ

~~Что делает программа?~~

Какие объекты реального мира мы моделируем?

Объектно-ориентированный способ

- Классы;
- Компоненты;
- Свойства;
- Поведение.

ООП: Классы

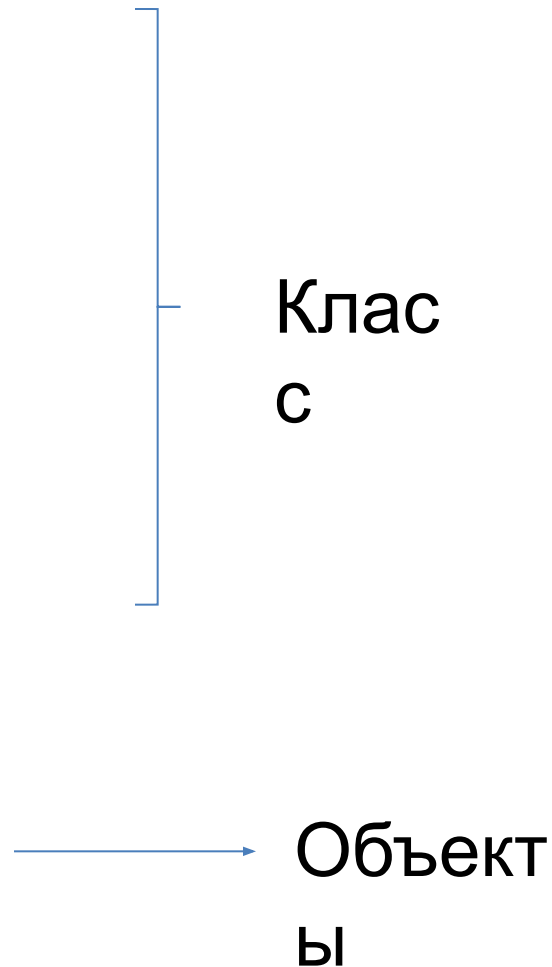
- Классы – абстрактный тип данных (АТД) в ООП, определяющий строение/поведение всех объектов данного класса;
- Объект – конкретный представитель класса.

ООП: Классы

Что такое апельсин?
Чем его можно охарактеризовать?

ООП: Классы

- фрукт;
- вкус;
- размер;
- вес;
- цвет;
- форма.
- ...



ООП: Классы

```
class Orange {  
  //свойства  
  int flavor;  
  int size;  
  double weight;  
  int color;  
  string shape;  
};
```

```
Orange o;  
o.size = 1; o.color += 1;
```

ООП: Компоненты

- Сложные объекты реального мира состоят из множества *компонентов*.

Самолет:

фюзеляж;

двигатель;

посадочный механизм;

...

ООП: Компоненты

Самолет:

фюзеляж;

двигатель;

посадочный механизм:

посадочное колесо;

электромотор;

...

...

ООП: Компоненты

```
class Airplane {  
  //свойства  
  Engine engine;  
  Fuselage f;  
};
```

ООП: Поведение

Что объект делает?

Что можно сделать с объектом?

ООП: Поведение

Апельсин

- съесть;
- купить/продать;
- почистить;
- катать;
- ...

ООП: Поведение

```
class Orange {  
    // поведение  
    void eat();  
    void sell();  
    void buy();  
    void roll();  
};
```

ООП

Спроектировать программу для анализа котировок акций.

- Получить данные о цене акций;
- Проанализировать акции
- Сортировать акции согласно выбранным метрикам.

ООП

КотировкаАкций: `class StockQuote {`

•название;	<code>string name;</code>
•СИМВОЛ;	<code>char sym;</code>
•цена;	<code>double price;</code>
•анализировать;	<code>void analyze();</code>
•купить;	<code>void buy();</code>
•продать;	<code>void sell();</code>
	<code>};</code>

ООП

КоллекцияКотировокАкции:
список котировок;

получить список котировок;
предоставить сортированный
список котировок по критерию;

```
class StockQuoteCollection  
{  
    StockQuote stocks[10];  
  
    void GetCollection();  
    list sortBy();  
};
```

ООП: «Overobjectication»

Объекты игры «Крестики-Нолики»

Поле 3x3 массив

Ячейки

X char

O char

ООП: Слишком общие объекты

Объекты каталога магазина цифровой информации (фильмы, игры, книги, ..):

Цифровая информация:
данные

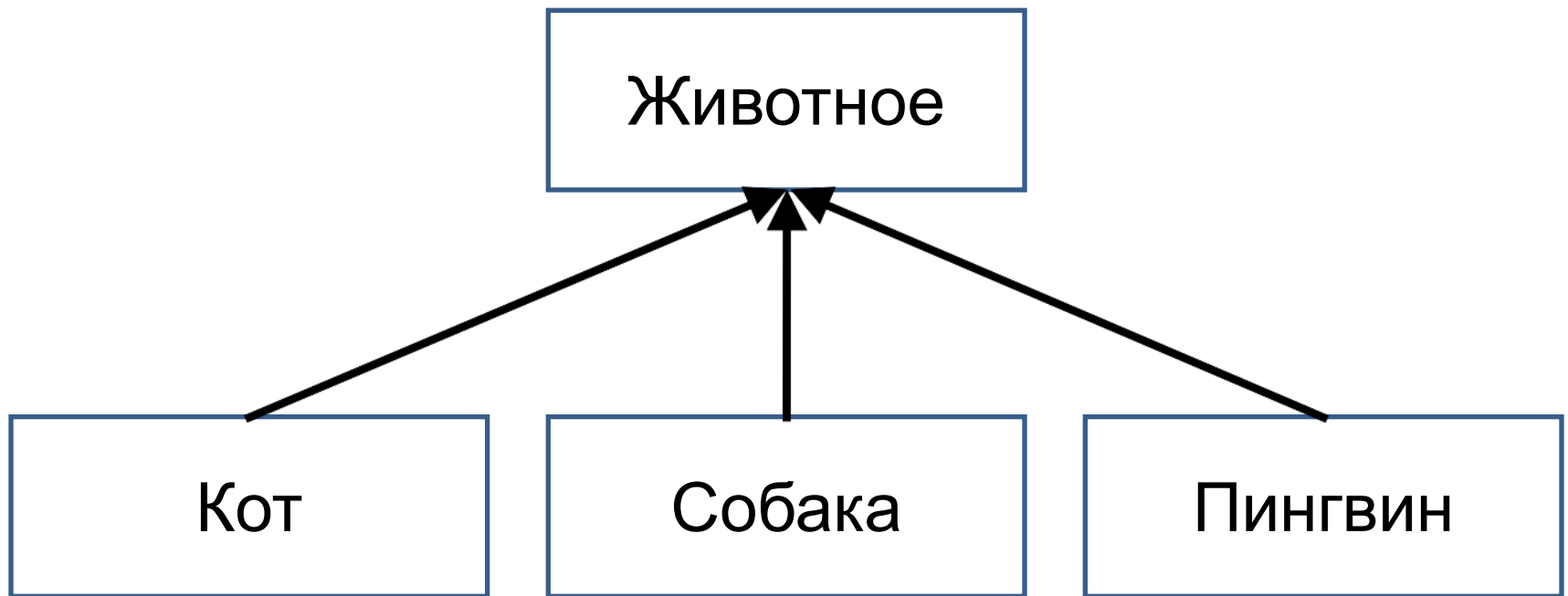
делать/выполнить
получить

ООП: Отношения между объектами

Агрегация (has-a) – отношение между объектами, в котором объект В является частью объект А.

ООП: Отношения между объектами

Наследование (is-a) – отношение между объектами, в котором объект А является объектом В или А частично похож на В.



ООП. Наследование

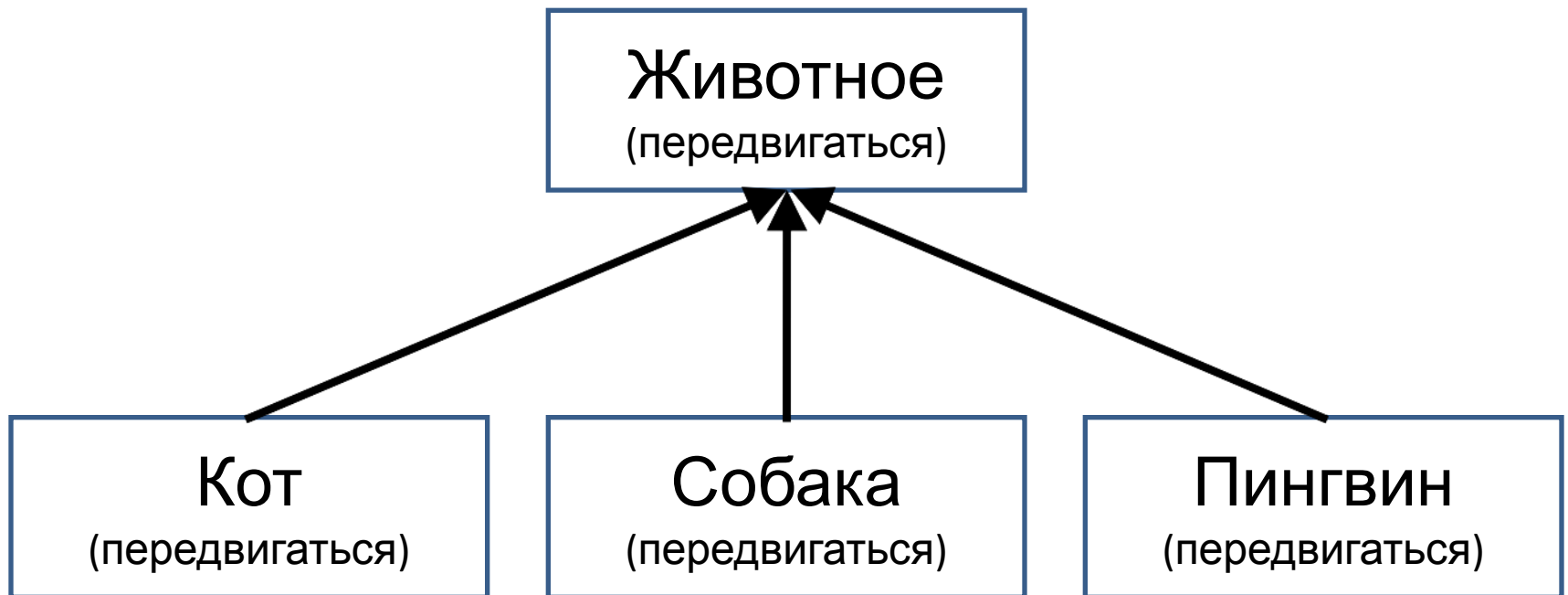
1. Дочерний класс имеет доступ к свойствам и поведению родительского класса (но не наоборот(!));
2. У одного родительского класса может быть сколько угодно классов-наследников;
3. Классы могут составлять иерархию наследования.

Для чего можно использовать наследование

- Добавление функциональности (поведения);
- Замена функциональности (поведения);
- Добавление свойств;
- Замена свойств.

Полиморфизм

Целью *полиморфизма*, применительно к ООП, является использование одного имени для задания общих для класса действий.



Разница между агрегацией (has-a) и наследованием (is-a)

```
// key -> value
```

```
Hashtable hashtable;
```

```
hashtable.insert("key", "value"); // key -> value
```

```
hashtable.get("key"); // value
```

```
// key -> [value, value1, ....];
```

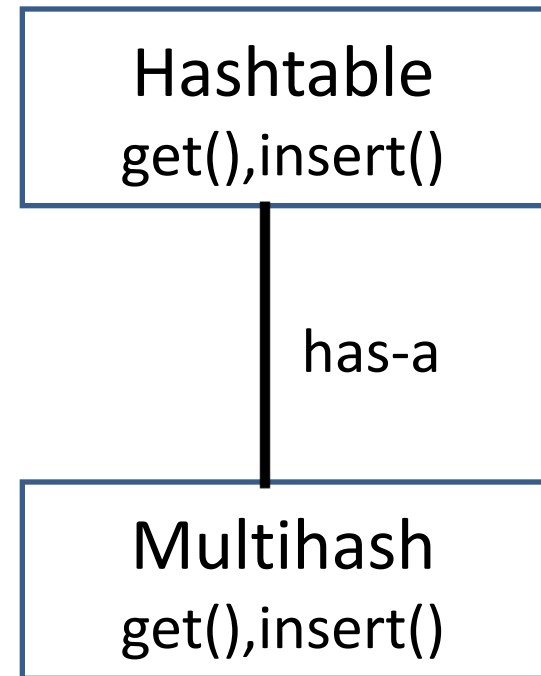
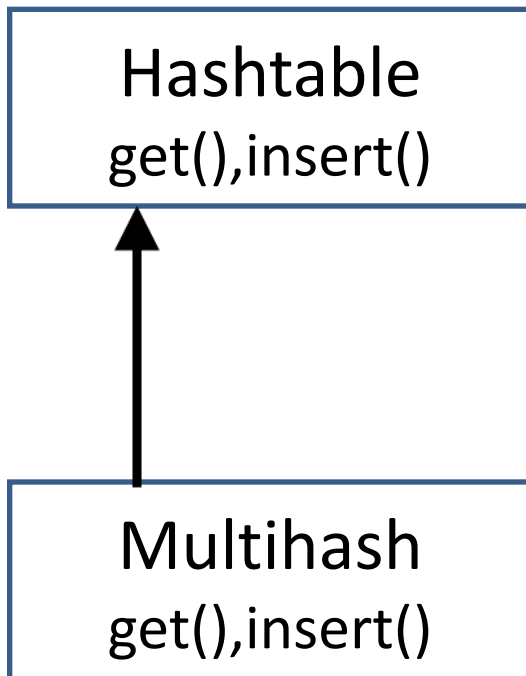
```
MultiHash multihash;
```

```
multihash.insert("key", "value"); // [value]
```

```
multihash.insert("key", "value1"); // [value, value1]
```

```
multihash.get("key") // [value, value1]
```

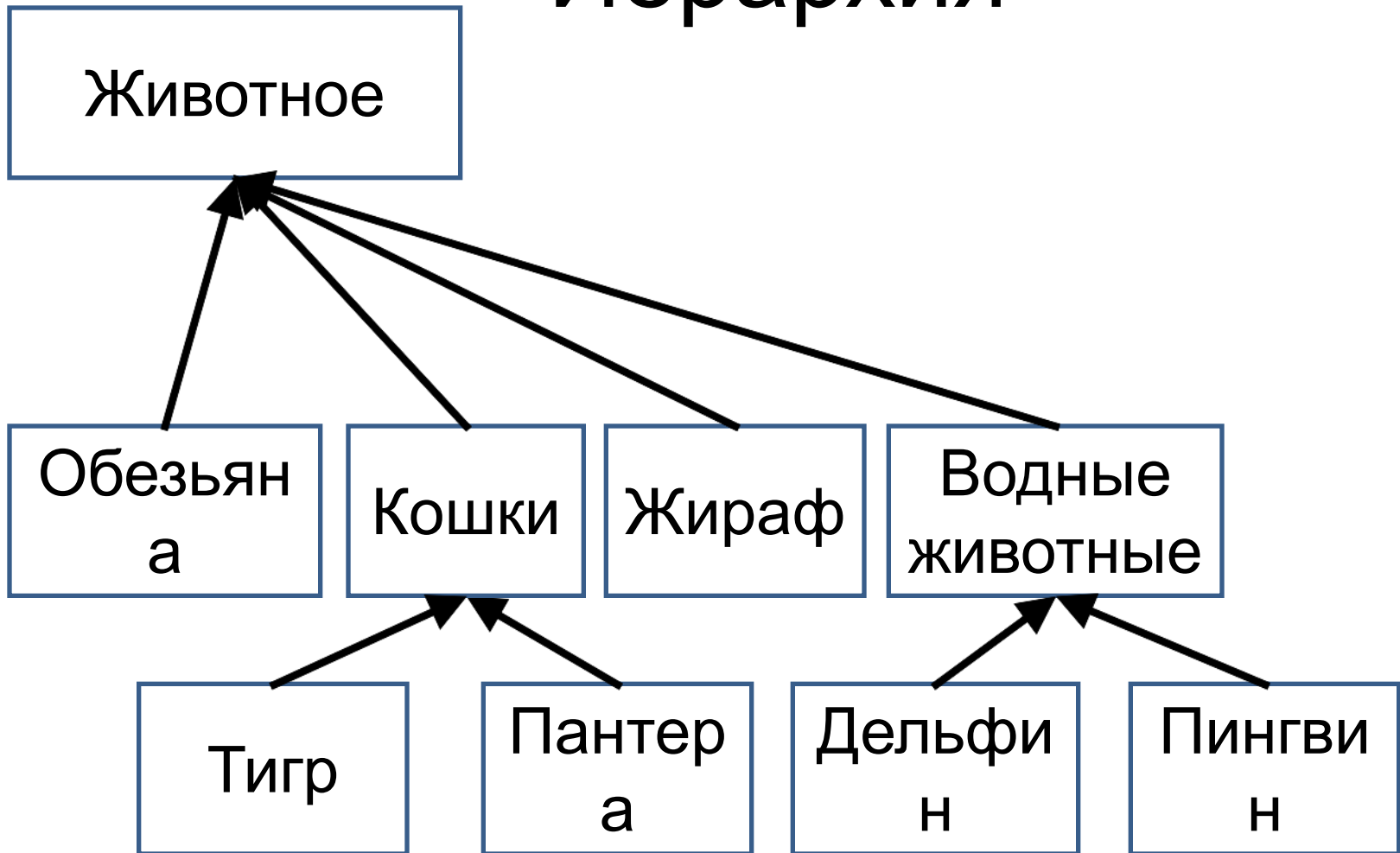
Разница между агрегацией (has-a) и наследованием (is-a)



Разница между агрегацией (has-a) и наследованием (is-a)

	Наследование	Агрегация
за	<ul style="list-style-type: none">• Объекты являются одной сущностью с разными характеристиками;• Дочерний объект реализует такое же поведение, что и базовый.	<ul style="list-style-type: none">• Дочерний объект может иметь любое поведение вне зависимости от базового объекта.
проти в	<ul style="list-style-type: none">• Объекты разные по сущности (Объект переопределяет все поведение базового объекта);• Возможность «просочившейся	<ul style="list-style-type: none">• Отсутствие переиспользования кода между объектами;• Некоторые свойства базового объекта могут быть скрыты от дочернего.

Иерархия



Иерархия

- + Организация классов в иерархию согласно логическим связям;
- + Переиспользование общей функциональности в базовых классах;
- Переопределение в дочерних классах слишком большого числа функциональности;

ООП. Инкапсуляция

Инкапсуляция – механизм ЯП, с помощью которого выполняется ограничение области видимости для переменных и функций классов.

- Ограничение области видимости;

ООП. Инкапсуляция

```
class Pot {  
    private:  
        double temperature;  
    public:  
        void turnOn();  
        double getTemperature();  
        void setTemperature(double t);  
};
```

ООП. Абстрагирование

Абстрагирование – техника программирования, основная идея которой разделить данные на те, которые важны для решения задачи и те, которые несущественны.

ООП. Абстрагирование

Пример: объект электрический чайник

Функции:

1. Нагревание воды;
2. Показывать температуру;
3. Настроить температуру

ООП. Абстрагирование

```
class Pot {  
    double temperature;  
public:  
    void turnOn();  
    double getTemperature();  
    void setTemperature(double t);  
};
```