

НАСЛЕДОВАНИЕ В КЛАССАХ

01001

00101

01100

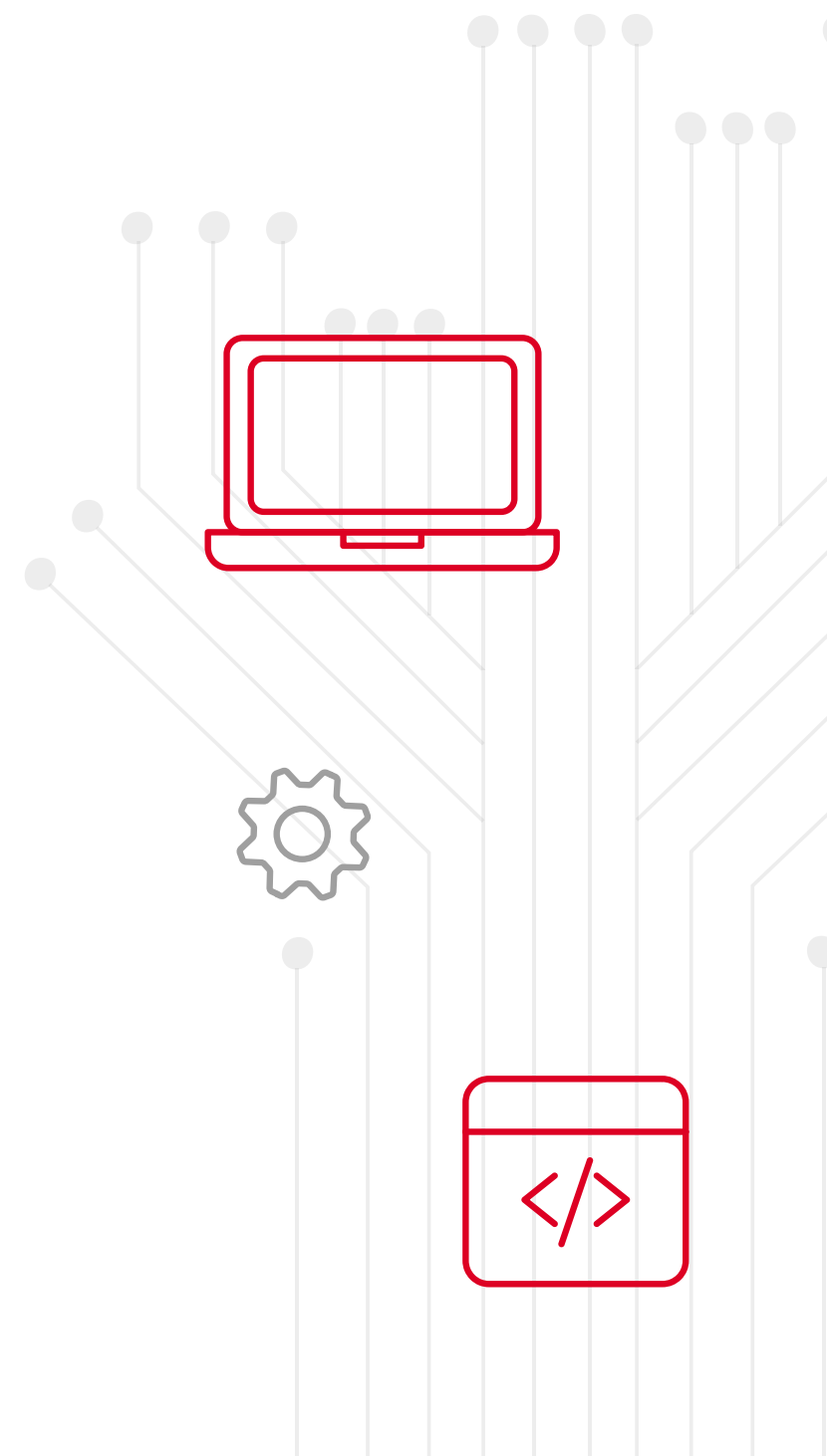
010

001

0110

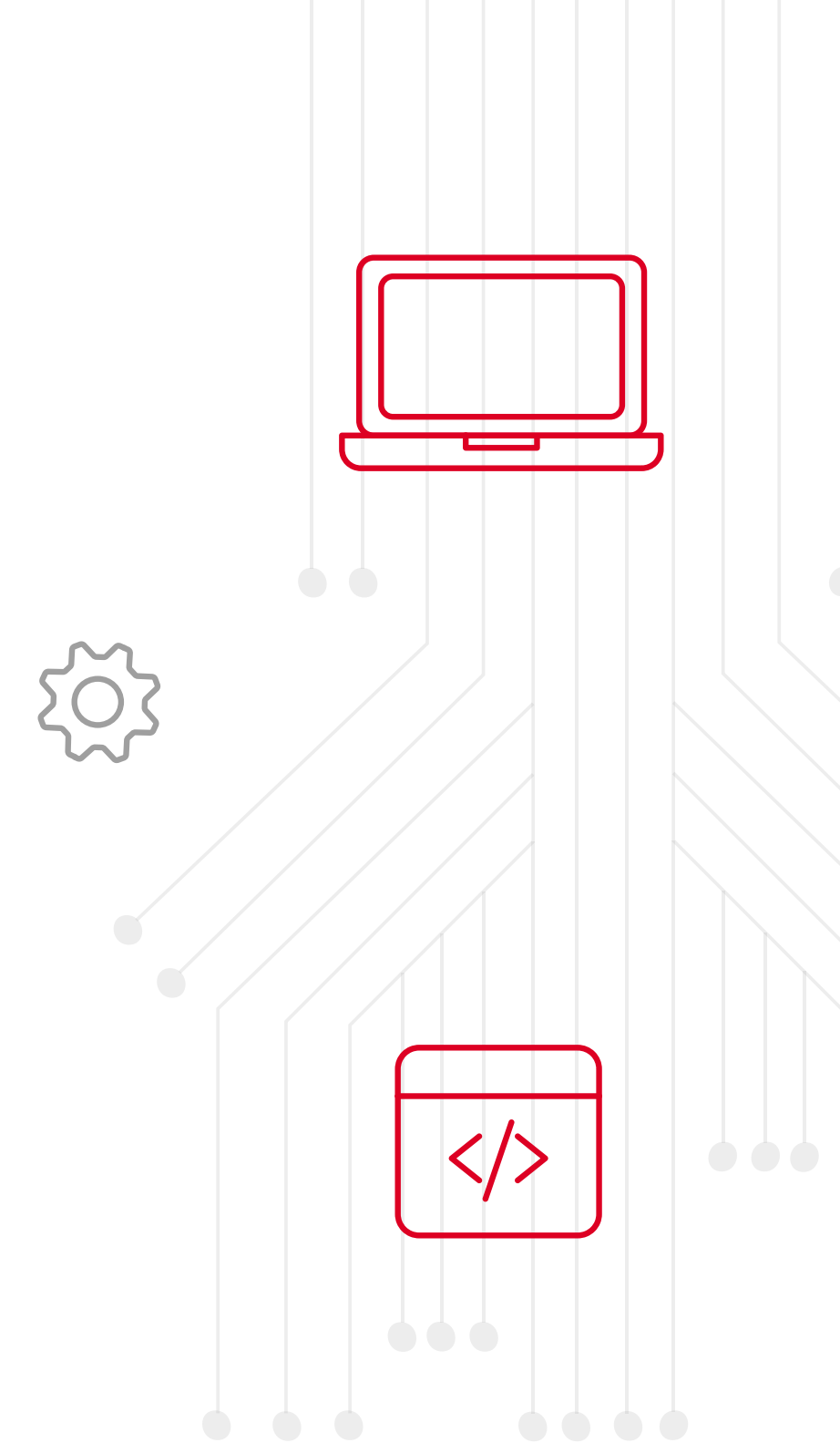
О чем поговорим сегодня

- Наследование
- Классы






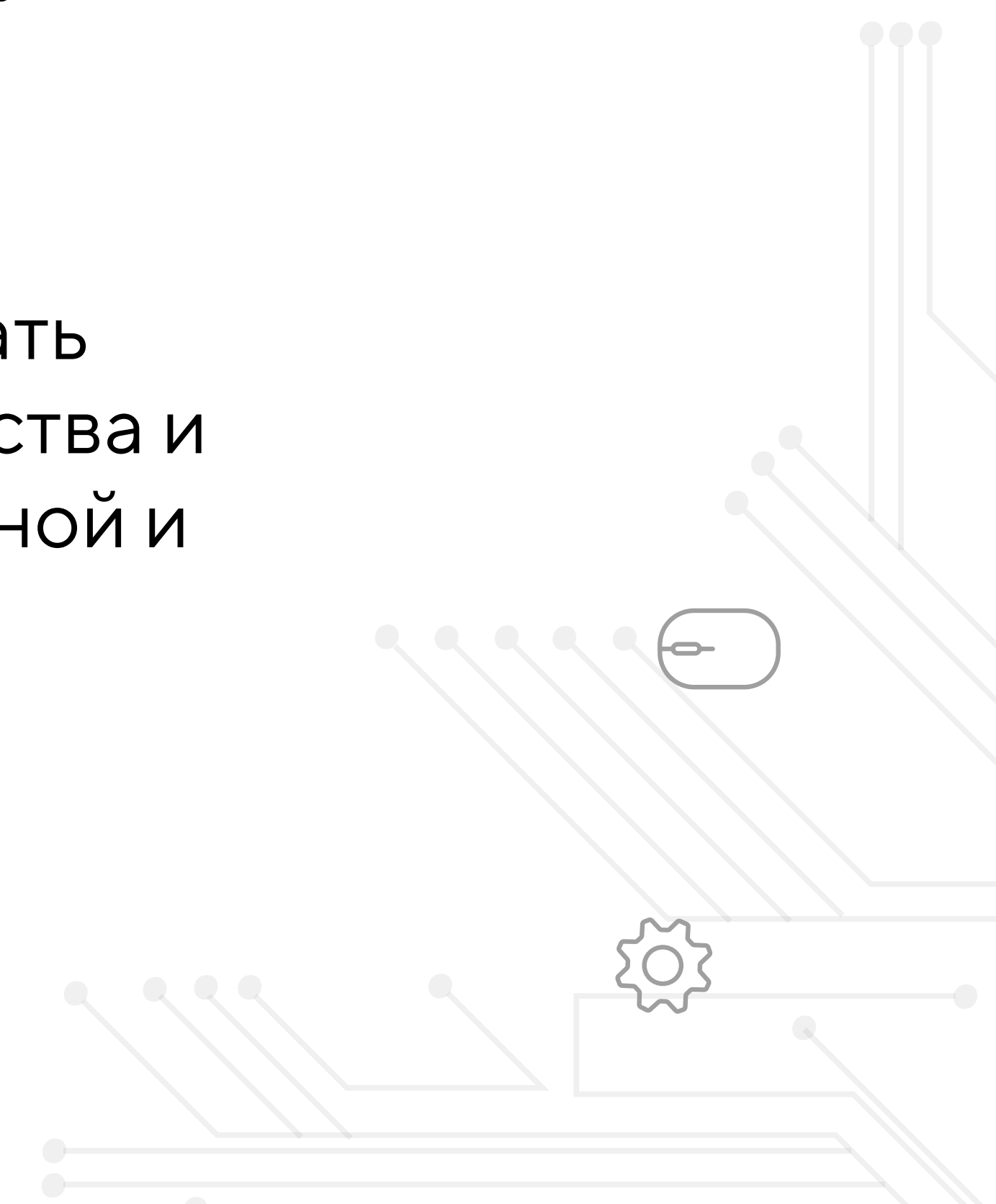
Наследование

концепция, согласно которой одни классы, называемые родительскими или базовыми классами, могут быть использованы в качестве основы для создания других классов, называемых дочерними или производными классами.



Преимущества наследования в ООП

-  **Повторное использование кода:** наследование позволяет создавать новые классы на основе уже существующих, что способствует повторному использованию кода и уменьшению дублирования.
-  **Расширяемость:** дочерние классы могут добавлять новые атрибуты и методы к уже существующим, что позволяет расширять функциональность базового класса.
-  **Упрощение иерархии классов:** наследование позволяет создавать иерархию классов, где дочерние классы наследуют общие свойства и методы от родительских классов, что способствует более логичной и структурированной организации кода.



Пример

Создаем базовый класс Animal

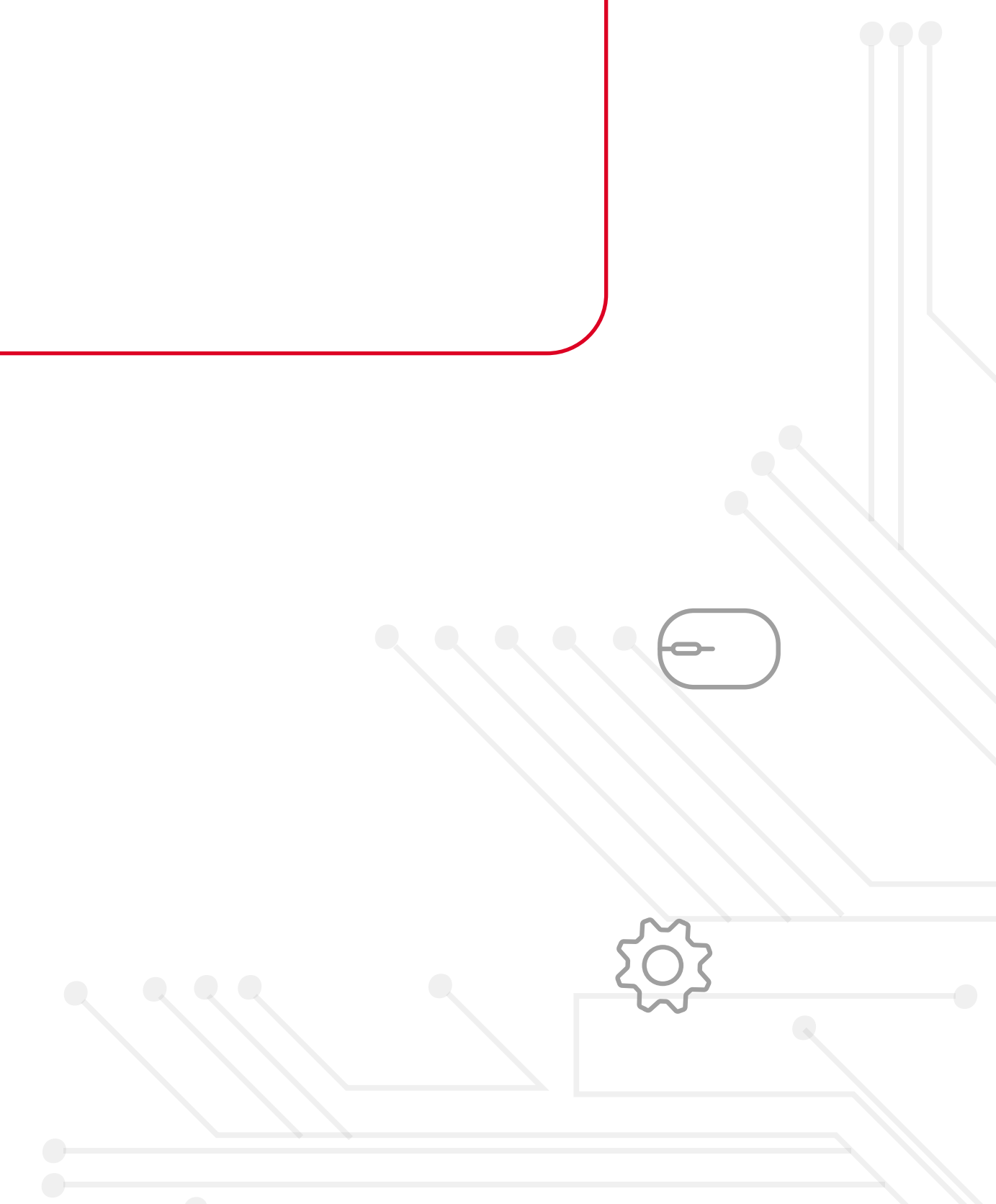
```
class Animal:
def __init__(self, name):
self.name = name

def speak(self):
print("Animal speaks")
```

Создаем дочерний класс Dog, который наследует атрибуты и методы класса Animal

```
class Dog(Animal):
def __init__(self, name, breed):
super().__init__(name)
self.breed = breed

def speak(self):
print("Dog barks")
```



Пример

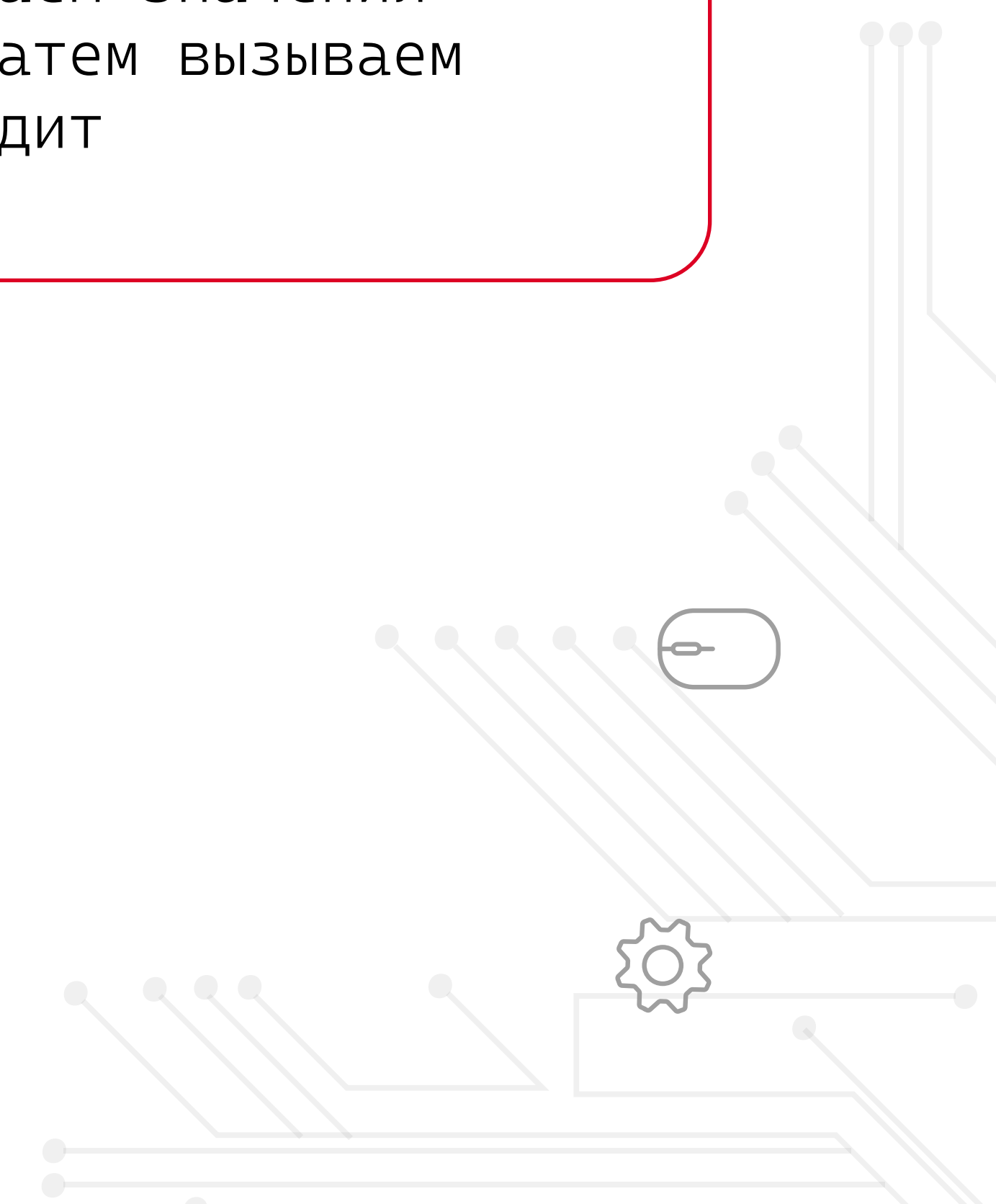
Создаем объекты классов Animal и Dog

```
animal = Animal("Animal")  
dog = Dog("Buddy", "Labrador")
```

Вызываем методы объектов

```
animal.speak() # Выводит "Animal  
speaks"  
dog.speak() # Выводит "Dog  
barks"
```

В данном примере класс Dog наследует атрибут name и метод speak() от базового класса Animal. Дочерний класс Dog также добавляет свой собственный атрибут breed и переопределяет метод speak(). При создании объекта класса Dog мы передаем значения атрибутов name и breed, а затем вызываем метод speak(), который выводит соответствующее сообщение.



Подведем итоги

- Классы могут иметь родительские и дочерние

