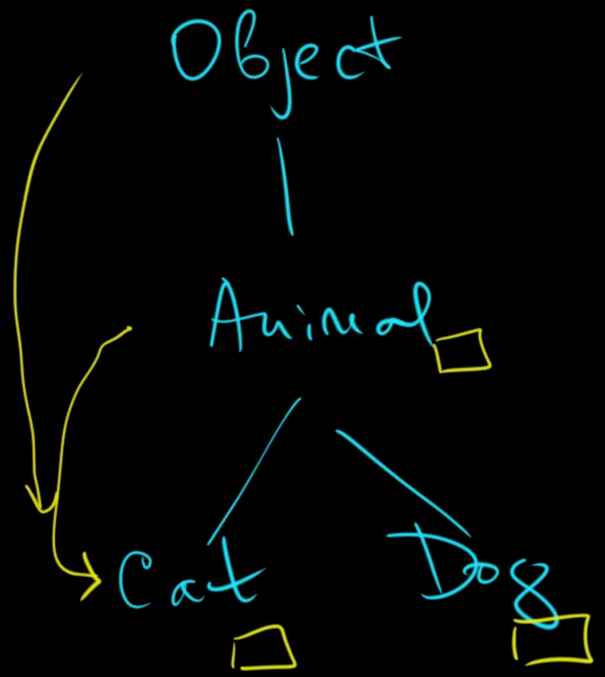

Object
Animal
Cat
Dog

```
public class Pizza
```

public

all packages

full access

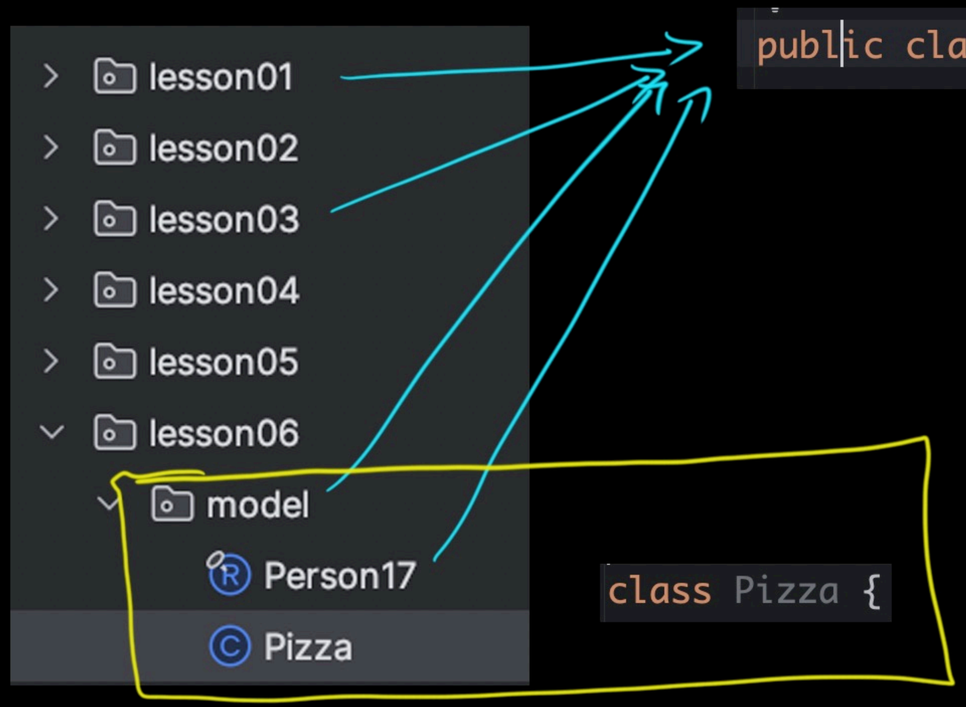
```
class Pizza {
```

current package ONLY

protected

private

method Access from the same class



extends

Access

private
public
protected

final class

```
final public class Cat
```

```
public class MegaCat extends Cat
```

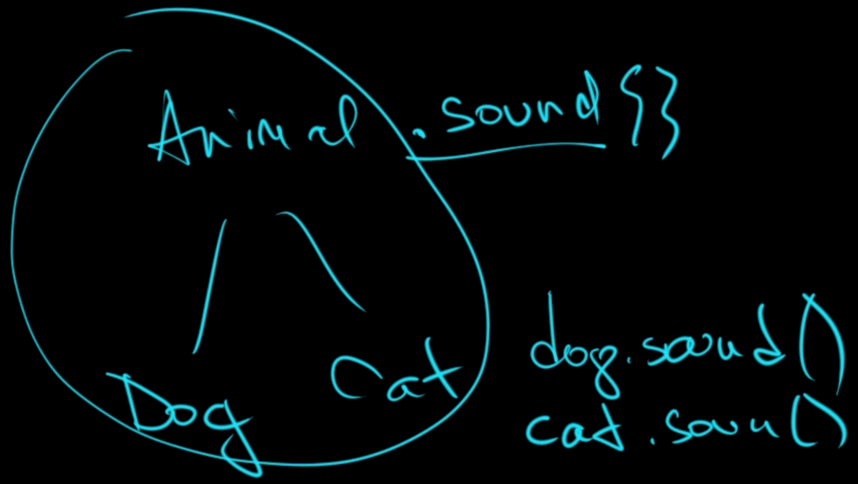
new Cat.

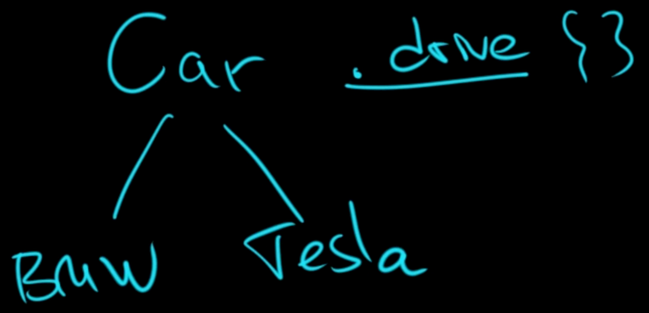
Free (cat, Done)

constructor chain

sound —
eat — Animal ?
go —

Dog Cat



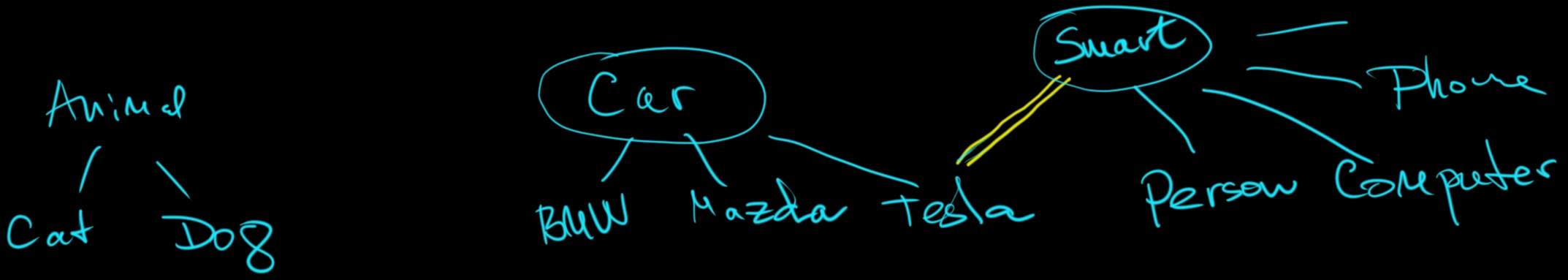



```
static class Animal{
  void sound() {
    //????
  }
}
static class Cat extends Animal{
  @Override
  void sound() {
    System.out.println("meow");
  }
}
static class Dog extends Animal{
  @Override
  void sound() {
    System.out.println("gav");
  }
}
```

```
static abstract class Animal{
  abstract void sound();
}
static class Cat extends Animal{
  @Override
  void sound() {
    System.out.println("meow");
  }
}
static class Dog extends Animal{
  @Override
  void sound() {
    System.out.println("gav");
  }
}
```

```
Animal a = new Animal();
```

```
Animal a = new Animal() {
  @Override
  void sound() {
    System.out.println("a-ha");
  }
};
```



~~Tesla extends Car + Smart~~

only ONE class can be extended

interface

[Abstract] class
has fields
extend ONLY ONE
(is) ∈

Interface
has no fields
implement MANY
(has) ∈

①

Generics

```
static class Box<A> {
  A x;

  public Box(A x) {
    this.x = x;
  }
}
```

compiler only

Integer

2

A=Integer

1

```
Box<Integer> box1 = new Box<Integer>(x:1);
Box<Pizza> box2 = new Box<Pizza>(new Pizza(name:"Margarita"));
```

```
new Box<Integer>(x:"1");
```

```
static class BoxInt {
  int x;

  public BoxInt(int x) {
    this.x = x;
  }
}

static class BoxDouble {
  double x;

  public BoxDouble(double x) { this.x = x; }
}
```

```
static class Box<A> {
  A x;

  public Box(A x) {
    this.x = x;
  }
}
```

```
Box<Integer> box1 = new Box<Integer>(x:1);
```

```
Box<Integer> box1b = new Box<>(x:1);
```

```
new Box<>(x:1);
```

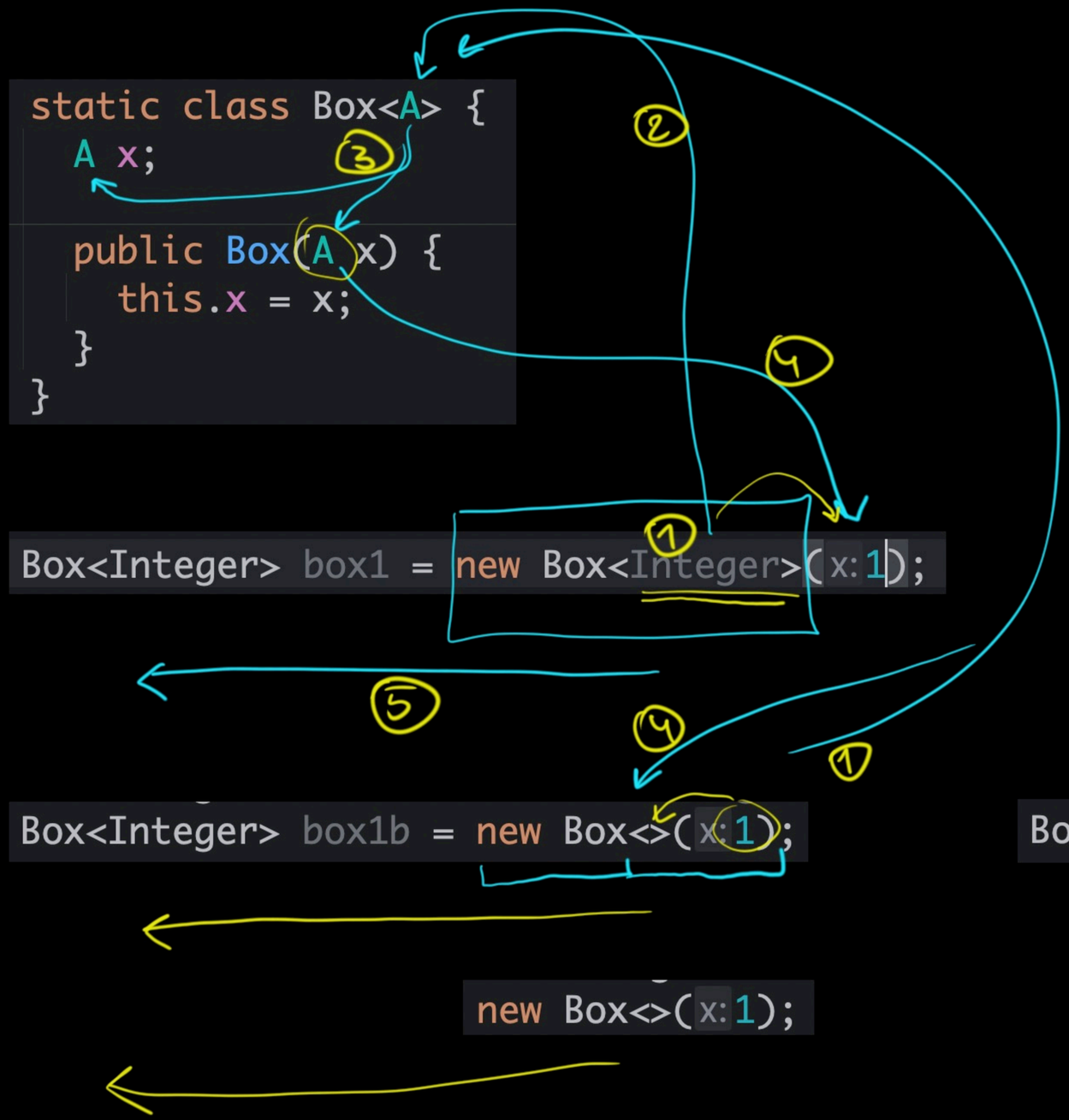
```
Box<Integer> box1b = new Box<>(x:1);
```

type

Box<A>

f (A) -> Box A

Box (A, x)



- int
- short
- Byte
- long
- char
- false bool
- 0.0 float
- 0.0 double

Integer
Short
Byte
long
Character
boolean
float
Double

default value

Object ← final ← Heap

int x = 5

class Car
int x; → Heap

x = null

Integer x = 10

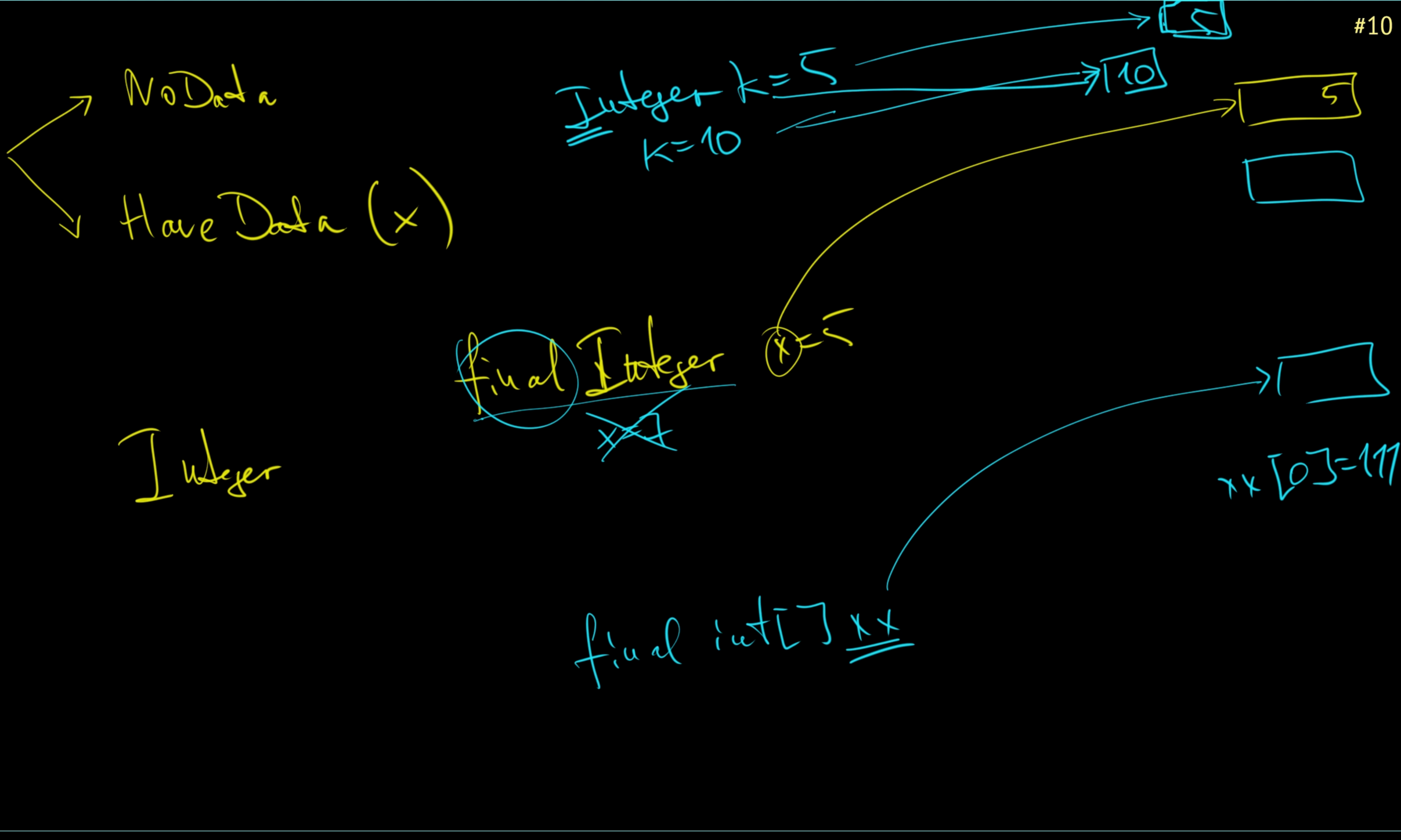
10

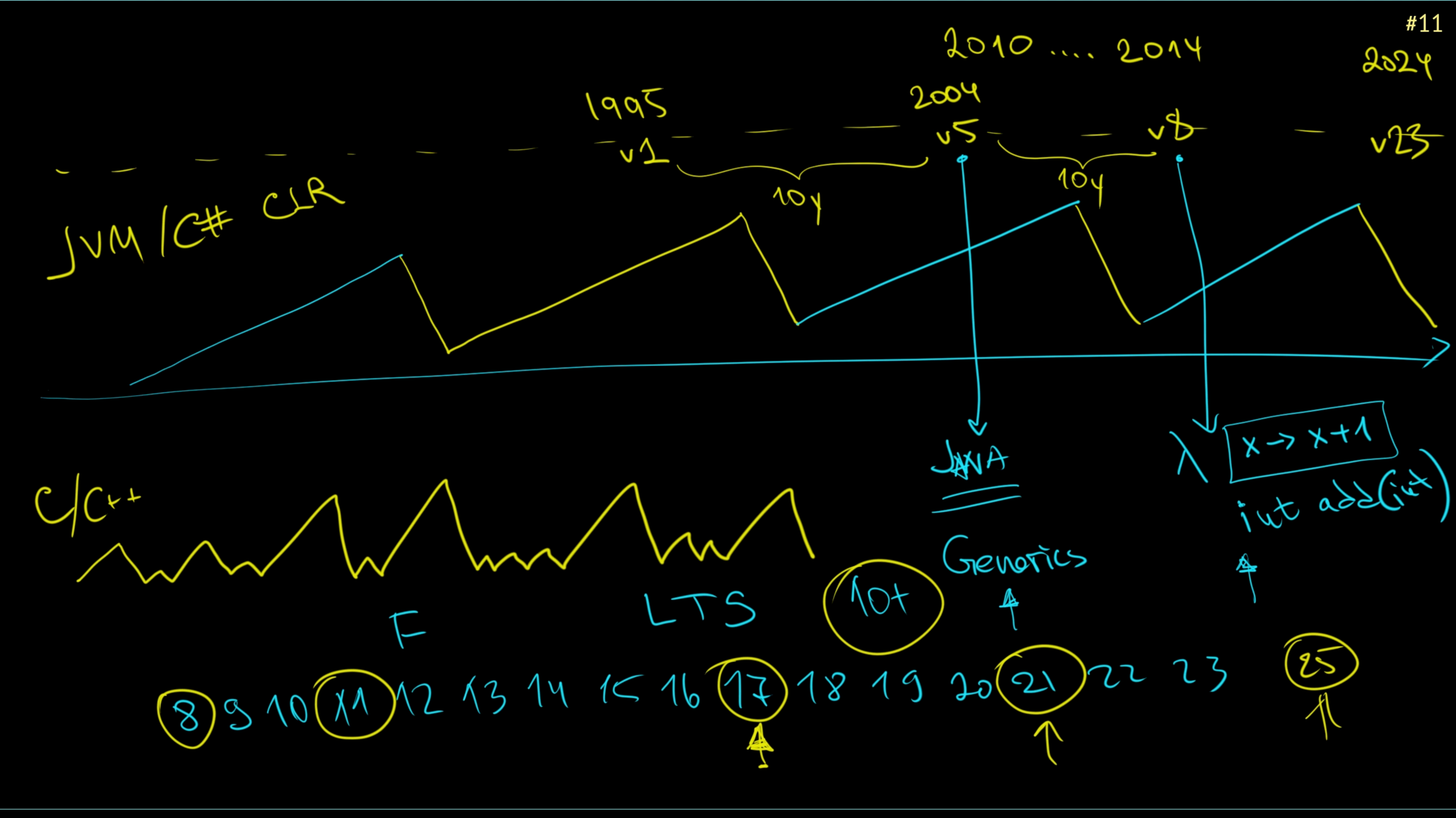
x

Integer x = 5 → 5

Integer k = x → 10

x = 10


NoData
HaveData (x)
Integer k=5
k=10
5
10
5
final Integer x=5
x=7
Integer
xx[0]=111
final int[] xx


2010 2014
2024
2004
1995
v5
v8
v23
v1
10y
10y
JVM / C# CLR
JAVA
λ
x -> x+1
int add(int)
C/C++
Generics
F
LTS
10+
8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 25

```
s = switch (x) {
    case 1: -> "one"
    -
    -
    break;
}
```

(1)

---

(2) record Person (String name, int age) {}

---

(3) String s = """
{
"x": 1,
"y": "hello"
}
"""