Core Java

Trainer: Nilesh Ghule

Human Readable Text Files

↓

character Streams

```
              Writer                        Reader
              (Sink)                        (Source)
```

Writer (Sink):
① void write (int b);
② void write (char [] arr);
③ void flush ();
④ void close ();

Reader (Source):
① int read ();
② int read (char [] arr);
③ void close ();

Character encoding 'A' ↬ 65 ↬ @100 0001

① ASCII ———— 1 byte ————→ @100 0001

② UTF-16 → 2 bytes → Low Byte: 65, High Byte: 0

  ↳ UTF-16 LE | @100 0001 | 0000 0000 |
                    100   101

  ↳ UTF-16 BE | 0000 0000 | @100 0001 |
                    100   101

③ UTF-32

④ UTF-8

⑤ ISO-8859-1

---

character → Writer → byte sequence

character ← Reader ← byte sequence

Writer hierarchy

Output Stream Writer : Bridge betn
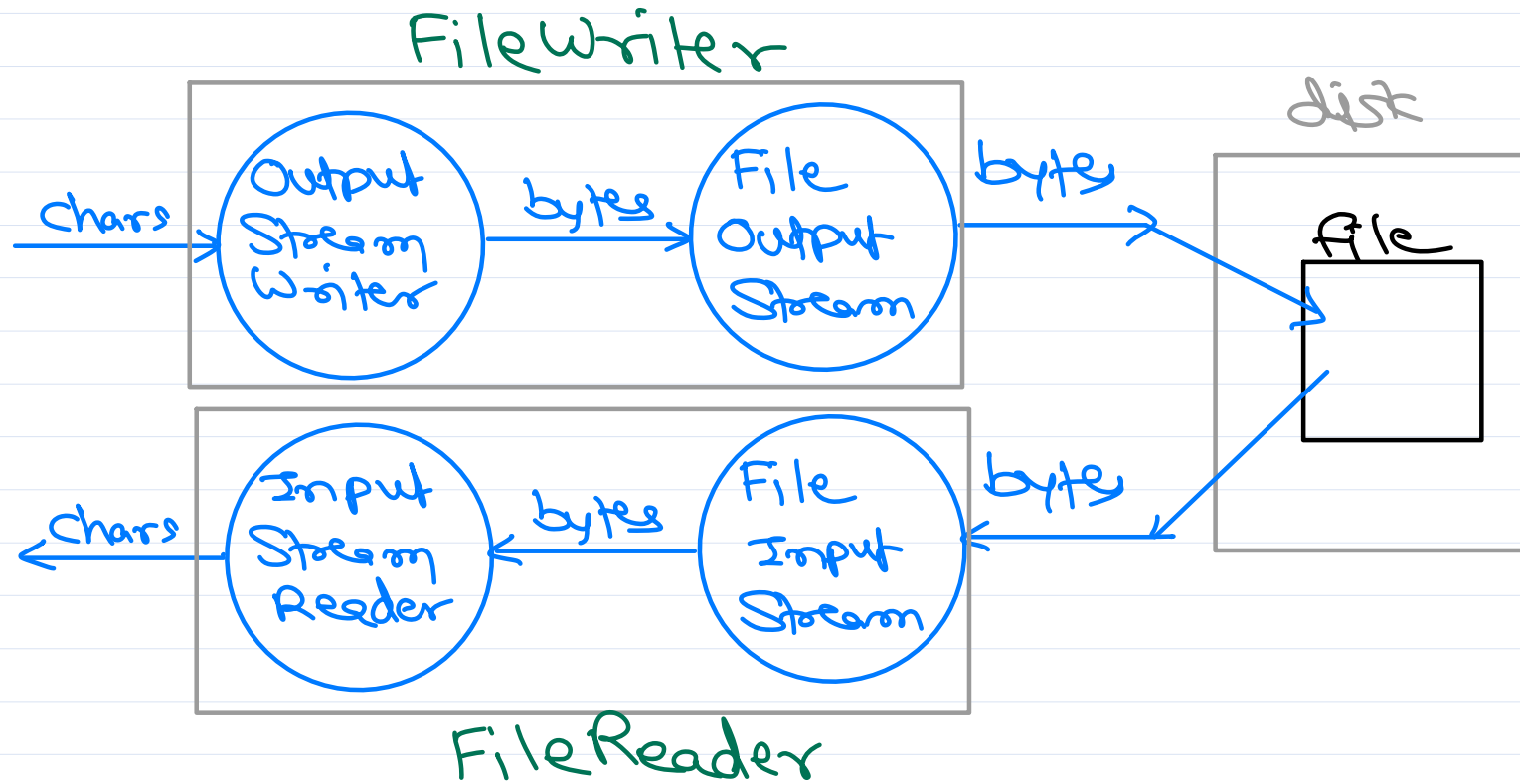byte Streams & Char Streams.
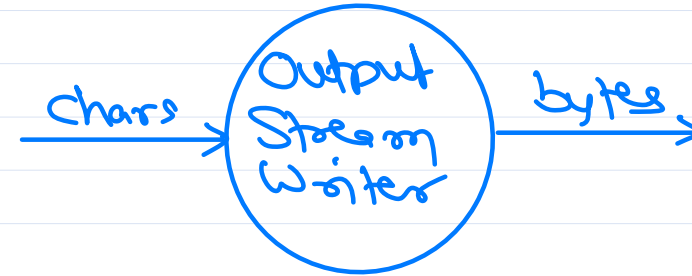
Closeable

Writer

Buffered Writer
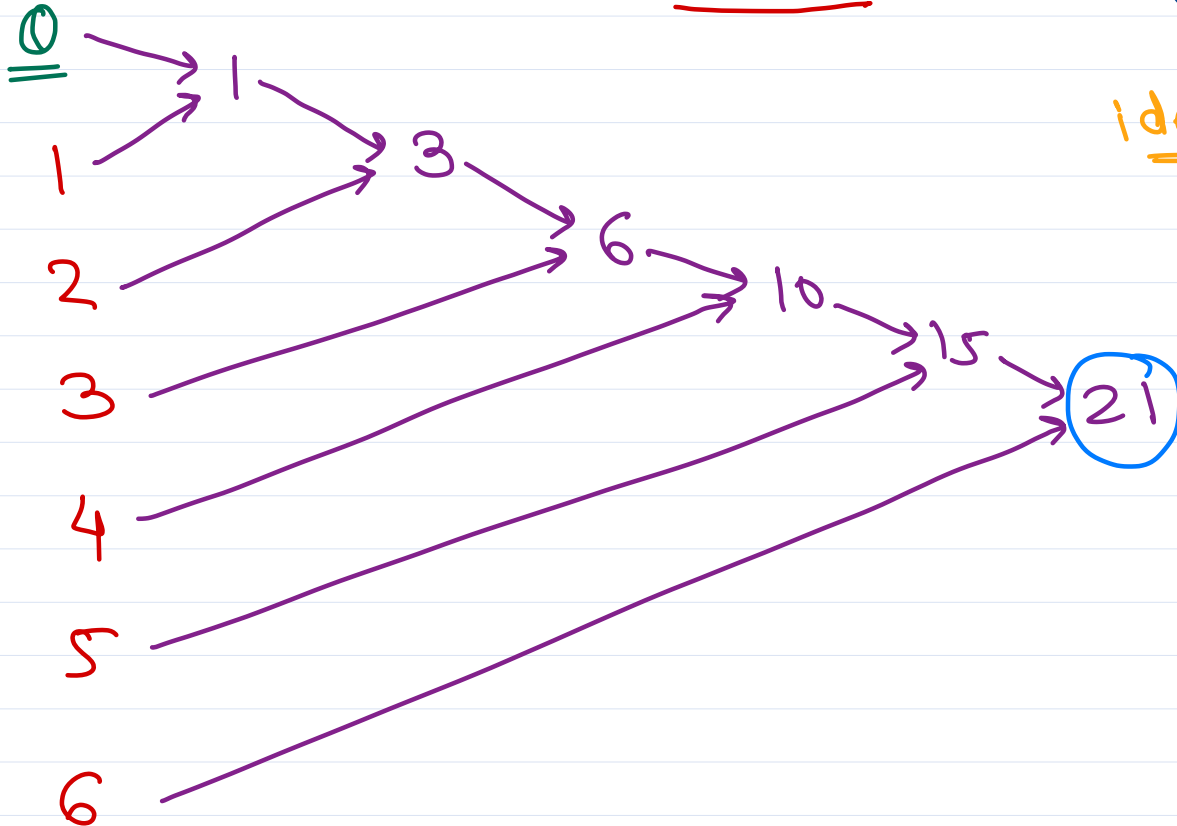
Output Stream Writer

Print Writer

File Writer

Chars → Output Stream Writer → bytes

FileWriter and FileReader

FileWriter

chars → Output Stream Writer → bytes → File Output Stream → bytes → disk / file

FileReader

chars ← Input Stream Reader ← bytes ← File Input Stream ← bytes ← file

reduce() operation

$$strm = Stream.iterate(1, x \rightarrow x+1).limit(6);$$

$$res = strm.reduce(0, (a, e) \rightarrow a+e);$$

identity   BinaryOperator

0

1
1
2   3
3   6
4   10
5   15
6   (21)

collect()

toArray() → { ele1, ele2, ele3, ele4, ele5 }

Stream

ele1
|
ele2
|
ele3
|
ele4
|
ele5

Collect() → [ele1, ele2, ele3, ele4, ele5]

Collector

Collectors. toList()

Reflection

JVM

.class
meta data
byte code

Classloader

byte code → meta-data (object of java.lang.Class)

method area   heap area

· · ·

JIT   Garbage Collector

Native Code

CPU
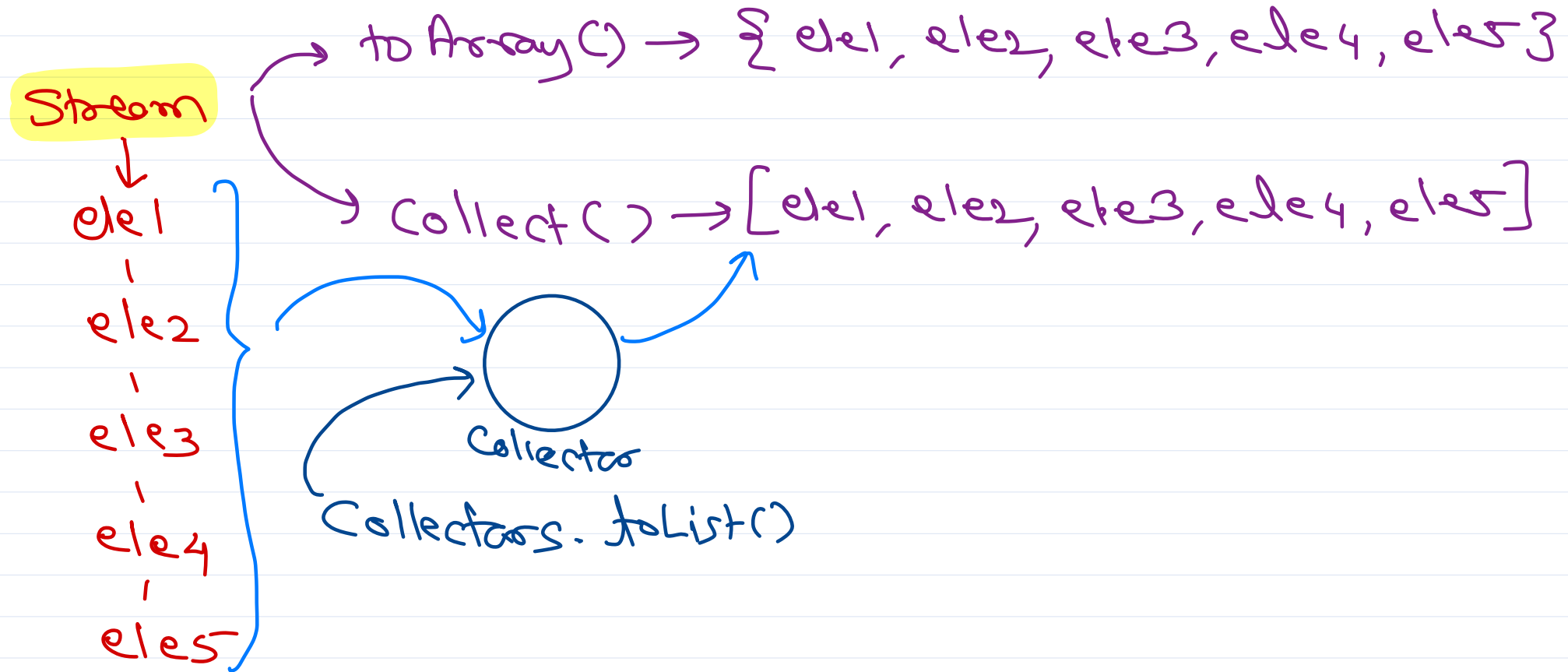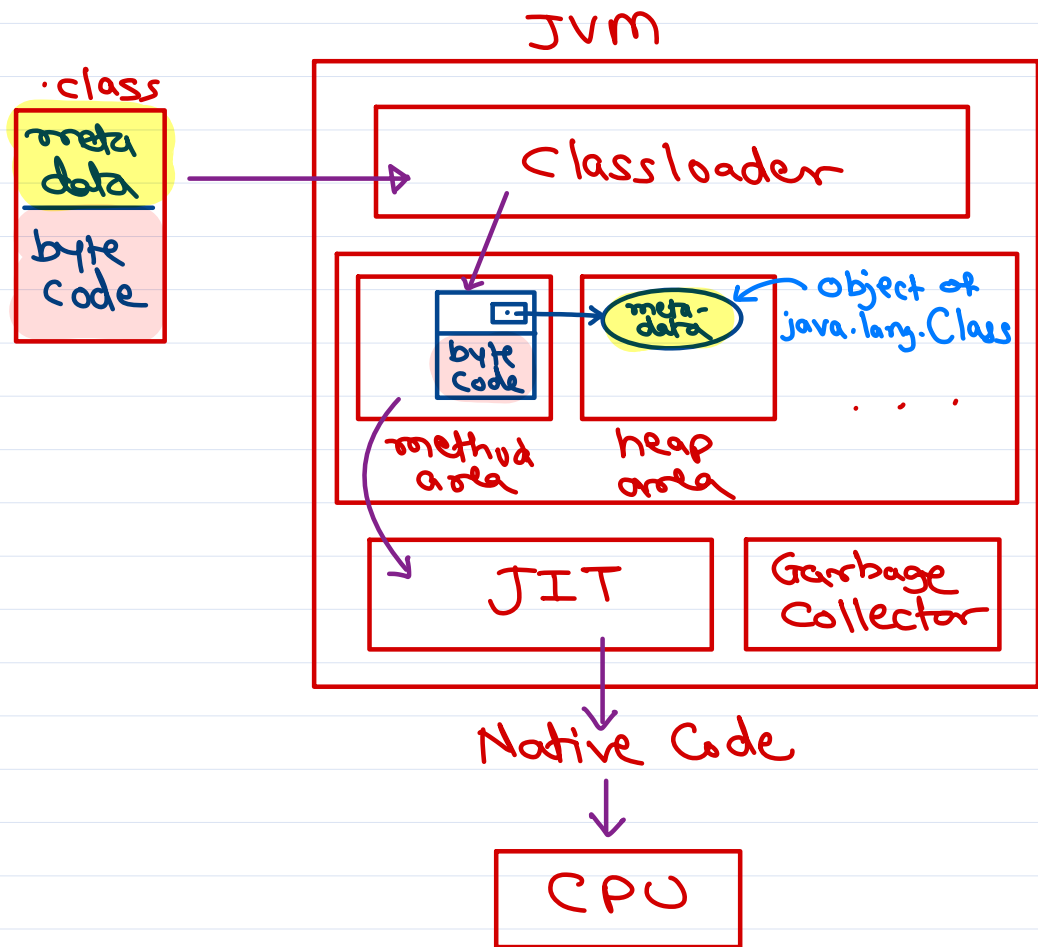
To get java.lang.Class obj associated with any class — 3 ways:

① c = Class.forName("pkg.ClassName");

② c = ClassName.class;

③ c = obj.getClass();

To access metadata of a type — methods in java.lang.Class.

① getName()

② getSuperClass()

③ getInterfaces()

④ getFields()   or   getDeclaredFields()

⑤ getMethods()   or   getDeclaredMethods()

⑥ getConstructors() ...

⑦ getAnnotations() ...

⑧ isAbstract(), isFinal(), isEnum(), isInterface(),...

Annotation

annotation → additional meta data associated with

- type (class, interface, enum)
- fields
- methods & constructors
- arguments
- local variable
- annotation

meta-annotations

@ Target ({ —,—})

@ Retention (~~~)

@Retention (RUNTIME)
@Target ({ TYPE, METHOD })
@interface Developer {
  String value();
  String designation();
  String company();
3

@ Override
@ Deprecated
@ SuppressWarning
@ Functional Interface

annotation retention levels

① SOURCE   @Override
② CLASS   @KeepName
③ RUNTIME   @Table

.java

.class

JVM / RT

Java Compiler

Java Classloader

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>