DA16

Jave

success

max mise win
minimise loss

{ automatic Parallel }

Java Stream API

{ cache memory not main memory }

تطلع من الـ Collection
streams

operations
+ terminal operation

{ Stream do not have their own storage }

on the fly not on array or whatever

$\boxed{\text{عايز functional interface}}$
+ lambda implementation

طلب الـ host

Streampipeline

.stream() — method

.

.map

.sum();

doesn't run till got Terminal

lazy operation

eager operation

ياخد Stream
بس يطلع أي حاجة غير كده

intermediate

الأوامر ← **operation**

chunks
هكون تقطع مش مترتبة

filter → the elements satisfy a condition
distinct → only unique Element
عايز أول 20 limit →
إذا كان عايز أول فرد هيرجع ٢ س
map → mapped to new values
sorted → same number →

Terminal Operation ← آخر حاجة

مش هتطلع
stream
هتكون هي الطالع
حاجة

Loop
↳ forEach

Reduction operations
↳ return single value

Mutable Reduction operation
↳ Create a container for stream

Search Operations
↳ different search

إيه الفرق بين
collect
& toArray

average - count - max - min - reduce

collect - to Array - findFirst - findAny
لو مش لازم أول واحد

any Match - all Match

~~arry~~ ~~not~~ الـ array
methods بيتعامل بـ
stream. تختلف عن

lambda expression functional interface implementation

Stream

int Stream.of(values).forEach(→)

stream object مش الـ

ده بيتضارد

.min().getAsInt()

BigDecimal بيرجع

null بيقبل

بيتعامل الـ null بيطلع 0

Sorted
مش
parallel
شكلك

.reduce(0, (x,y) → x+y*y));

initial value

reduce

x = 0   y = 0+y*y

x =

class method

مش بيبدأ من العلامة

IntStream.range(1, 10)

.rangeClosed(,)

Stream.iterate

default behavior is Sequential

*** import static < method name >;

to use it directly

is Vegetarian
(lambda expression)

filter(Dish::isVegetarian);
.collect(toList());

إذا عايزها parallel
⊕ هم بيتعامل بالـ ...
بالكامل

print (object.toString)

override لها

physically concurrent بيحصل في نفس الوقت ← Parallel x sequential

fork & join

أنت مش بتعمل حاجة

best performance ... انت مسئول عن التقسيم والتركيب كله

stream parallel ... الجافا هي بتعمل كده بس انت مش شايفه parrallel

Stream .iterate (1L, i -> i+1)

لو تطلع بلا توقف

.limit (1000)

.reduce (0L, Long::sum)

stream to ... .parallel()

stream تطلع

أو

collection .parallelStream()

Big O تحليل

معالج الخوارزم

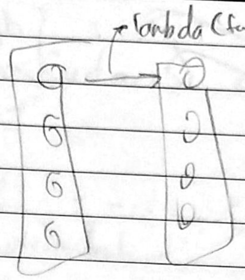
forEach (collection)

for sysout

flatmap

list

declarative programming



listed Countries

Countries. Stream. map (c → c.get City)

filter → predicate (cond)

صفر یک