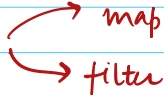


AGENDA:

Lambda and Streams

1> Functional Interfaces

2> Lambda

3> streams 

INTERFACE

containers for abstract methods.

```
public interface fun {  
    void a();  
    void b();  
}
```

Q1: Can interfaces only hold abstract methods?

Before Java 8

↓
YES

default methods
are now
allowed

Functional Interface:

Interface with SINGLE ABSTRACT METHOD

{ It can have default methods }

```
interface {  
    void fun1();  
    default fun2();  
    default fun3();  
}
```

← Only single abstract method.

Can I add more methods to this interface?

Yes.

@ Functional Interface ←

→ not mandatory but good to have
→ compile time check to make sure
interface only has a single abstract method

Have you ever worked with functional interfaces ?

- 1) Runnable \leftarrow `run()`
- 2) Callable \leftarrow `call()`
- 3) Comparator \leftarrow
- 4) Comparable
- 5) Iterable \rightarrow `iterator()`

Consumer: $\langle T \rangle$

\downarrow

`void accept (T t)`

\downarrow

[This method can do some operation on a parameter and return nothing]

BiConsumer $\langle T, U \rangle$

\downarrow

`void accept (T, U)`

Predicate $\langle T \rangle$

\downarrow

`boolean test (T t)`

\swarrow

Similarly \rightarrow biPredicate

Callable

Function $\langle T, U \rangle$

U apply $(T \ t)$
 \downarrow \uparrow

10:00

BiFunction $\langle T, U, V \rangle$

\forall apply $(T \ t, U \ u)$