This section explains the `ConnectableObservable` subclass and its operators:

- **`ConnectableObservable.connect()`** — instructs a Connectable Observable to begin emitting items
- **`Observable.publish()`** — represents an Observable as a Connectable Observable
- **`Observable.replay()`** — ensures that all Subscribers see the same sequence of emitted items, even if they subscribe after the Observable begins emitting the items
- **`ConnectableObservable.refCount()`** — makes a Connectable Observable behave like an ordinary Observable

A Connectable Observable resembles an ordinary Observable, except that it does not begin emitting items when it is subscribed to, but only when its `connect()` method is called. In this way you can wait for all intended Subscribers to subscribe to the Observable before the Observable begins emitting items.

The following example code shows two Subscribers subscribing to the same Observable. In the first case, they subscribe to an ordinary Observable; in the second case, they subscribe to a Connectable Observable that only connects after both Subscribers subscribe. Note the difference in the output:

Example #1:

```
Observable firstMillion = Observable.range(1, 1000000).sample(7, java.util.concurrent.TimeUn

firstMillion.subscribe(next -> System.out.println("Subscriber #1: " + next), // onNext
        throwable -> System.out.println("Error: " + throwable), // onError
        () -> System.out.println("Sequence #1 complete") // onComplete
    );
firstMillion.subscribe(next -> System.out.println("Subscriber #2: " + next), // onNext
        throwable -> System.out.println("Error: " + throwable), // onError
        () -> System.out.println("Sequence #2 complete") // onComplete
    );
```

```
Subscriber #1:211128
Subscriber #1:411633
Subscriber #1:629605
Subscriber #1:841903
Sequence #1 complete
Subscriber #2:244776
Subscriber #2:431416
Subscriber #2:621647
Subscriber #2:826996
Sequence #2 complete
```

Example #2:

```
ConnectableObservable firstMillion = Observable.range(1, 1000000).sample(7, java.util.concu
firstMillion.subscribe(next -> System.out.println("Subscriber #1: " + next), // onNext
        throwable -> System.out.println("Error: " + throwable), // onError
        () -> System.out.println("Sequence #1 complete") // onComplete
    );

firstMillion.subscribe(next -> System.out.println("Subscriber #2: " + next), // onNext
        throwable -> System.out.println("Error: " + throwable), // onError
        () -> System.out.println("Sequence #2 complete") // onComplete
    );

firstMillion.connect();
```

```
Subscriber #2:208683
Subscriber #1:208683
Subscriber #2:432509
Subscriber #1:432509
Subscriber #2:644270
Subscriber #1:644270
Subscriber #2:887885
Subscriber #1:887885
Sequence #2 complete
Sequence #1 complete
```

see also:

- javadoc: `ConnectableObservable`
- Introduction to Rx: Publish and Connect