A `Subject` is a sort of bridge or proxy that acts both as an `Subscriber` and as an `Observable`. Because it is a Subscriber, it can subscribe to one or more Observables, and because it is an Observable, it can pass through the items it observes by reemitting them, and it can also emit new items.

For more information about the varieties of Subject and how to use them, see the ReactiveX `Subject` documentation.

Serializing

When you use a Subject as a Subscriber, take care not to call its `onNext()` method (or its other `on` methods) from multiple threads, as this could lead to non-serialized calls, which violates the Observable contract and creates an ambiguity in the resulting Subject.

To protect a Subject from this danger, you can convert it into a `SerializedSubject` with code like the following:

```
mySafeSubject = new SerializedSubject( myUnsafeSubject );
```