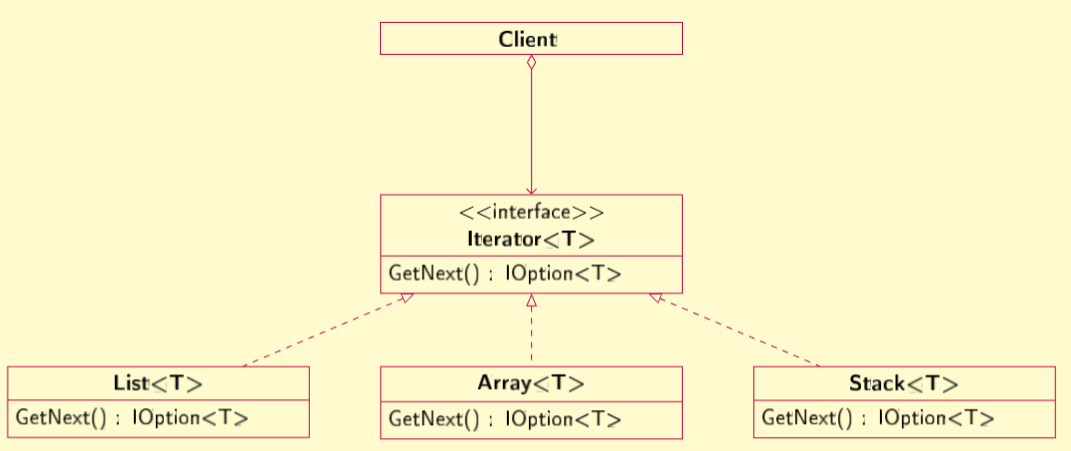
**The Iterator design pattern**

**Formal representation of a generic Iterator design pattern. (source: slides Dev 4)**

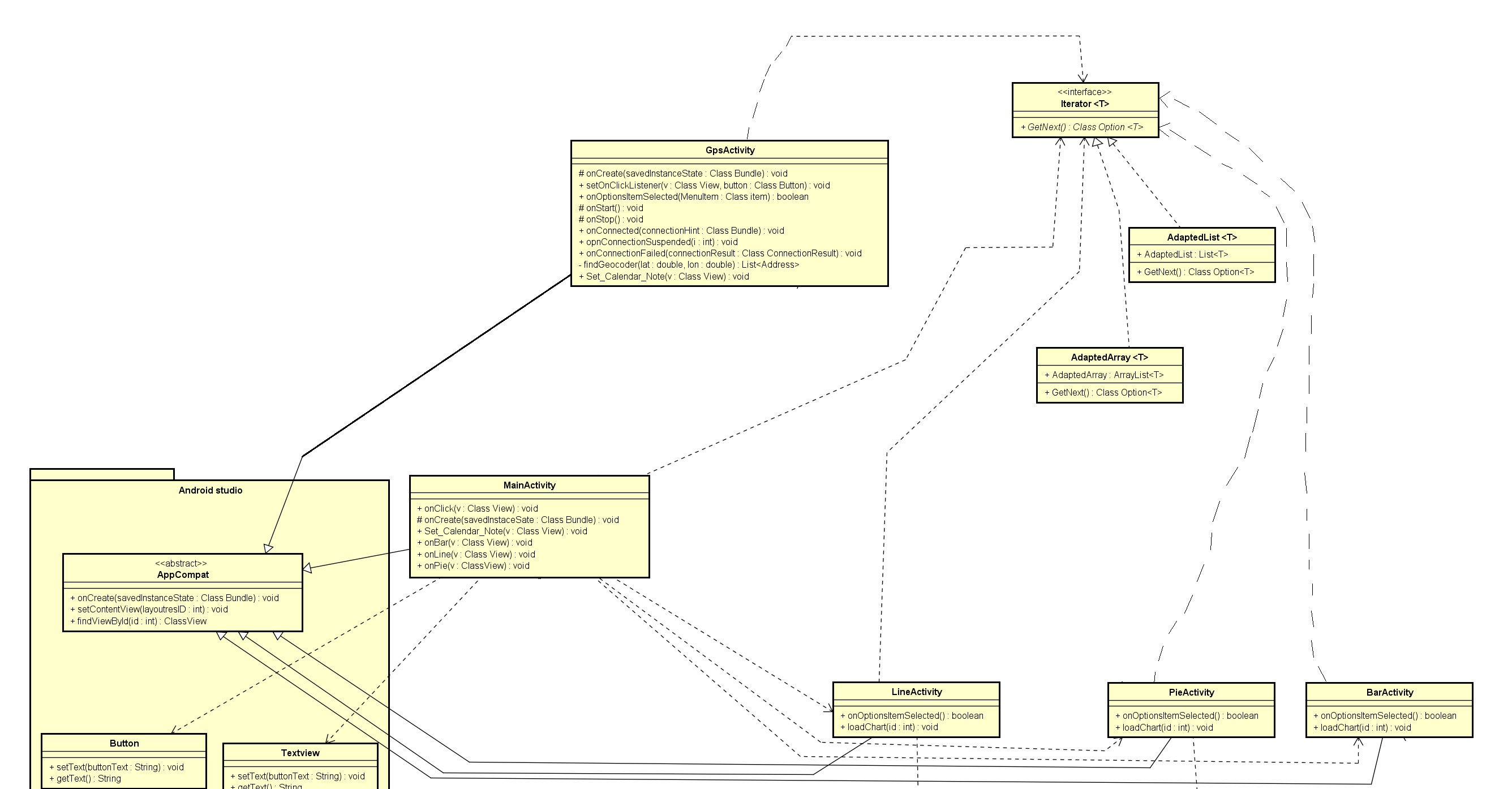


All the classes that have Activity in its name, are represents of the clients of the iterator interface.

Interface iterator, what is also an adapter (this part will be explained in the adapter chapter), represents obviously the iterator interface in the formal generic iterator design pattern.

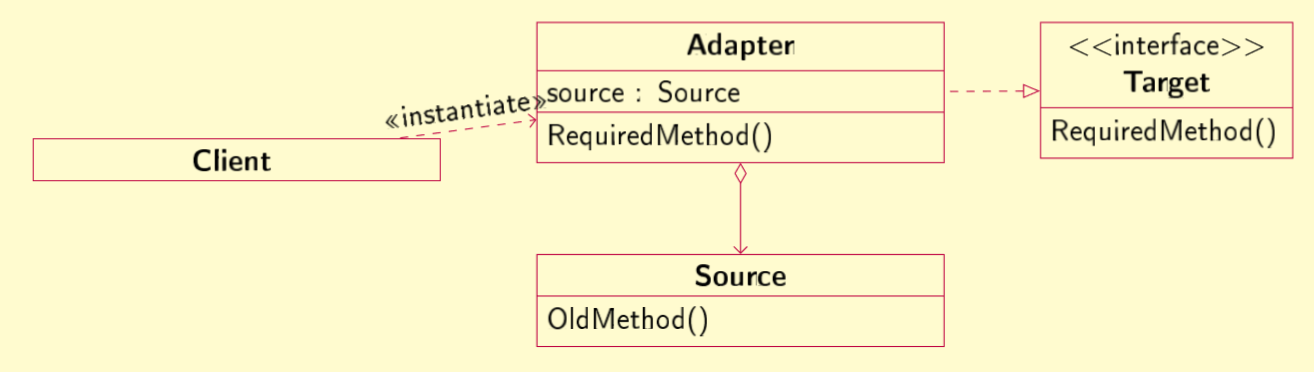
The AdaptedArray and the AdaptedList are the realization classes of iterator.

**Concrete implementation of an Iterator design pattern.**



**The Adapter design pattern**

**Formal representation of a generic Iterator design pattern. (source: slides Dev 4)**

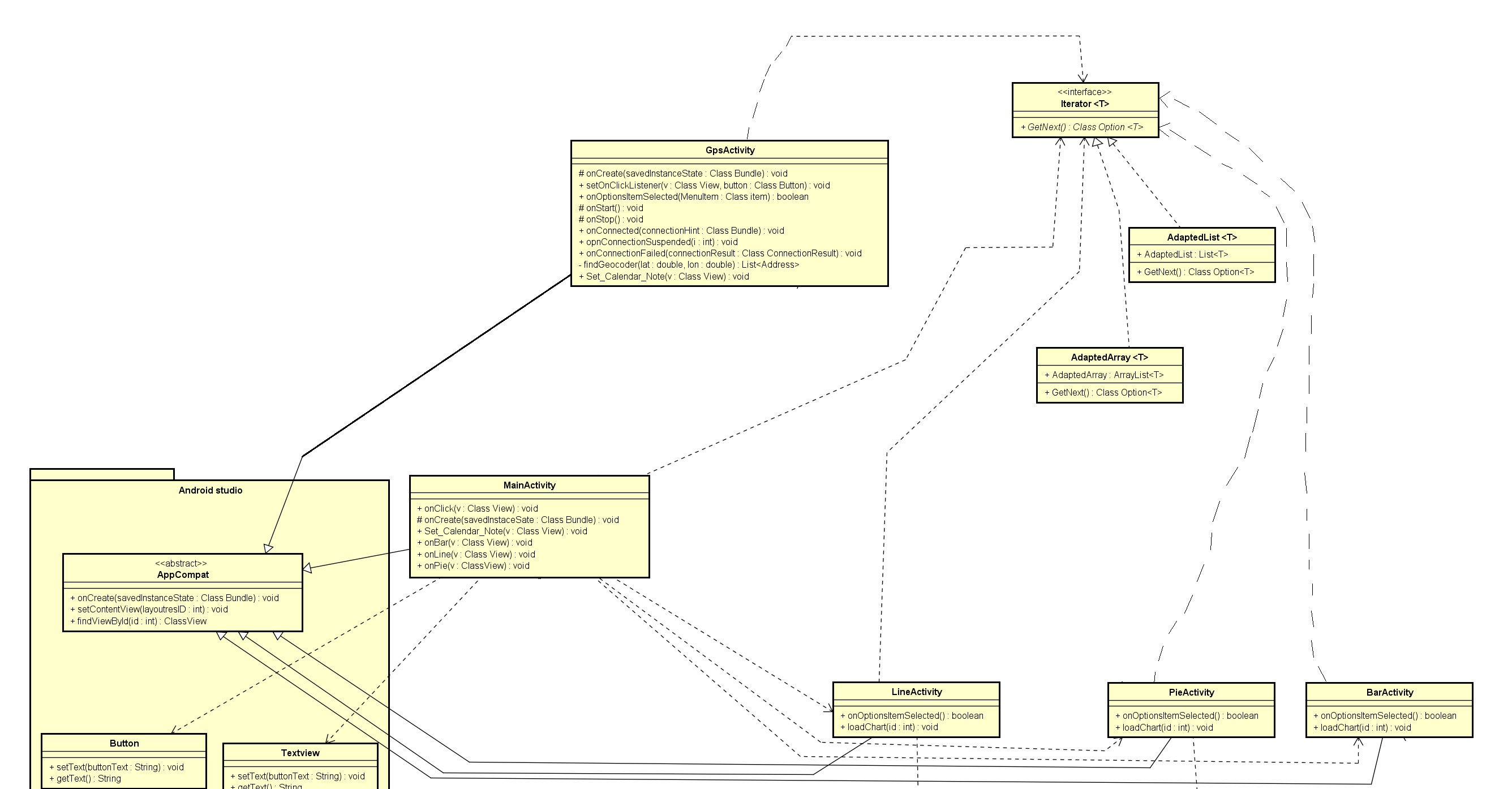
****

The source attribute here is in the adapted array and the adapted list, and the attribute is from a (unchangeable) source list and array class from java. This means the adapter is the AdaptedArray for the array java class, and the AdaptedList for the list java class. The added RequiredMethod() on the two adapters is the GetNext() method.

The clients using the adapter classes are all the activity classes, but they communicate with the adapters via the iterator interface (of course after instantiating, like the arrow above). If you require an extra method, you can just add it in the adapter and you can automatically use it on the java list or array (like an adapter is supposed to be able to)

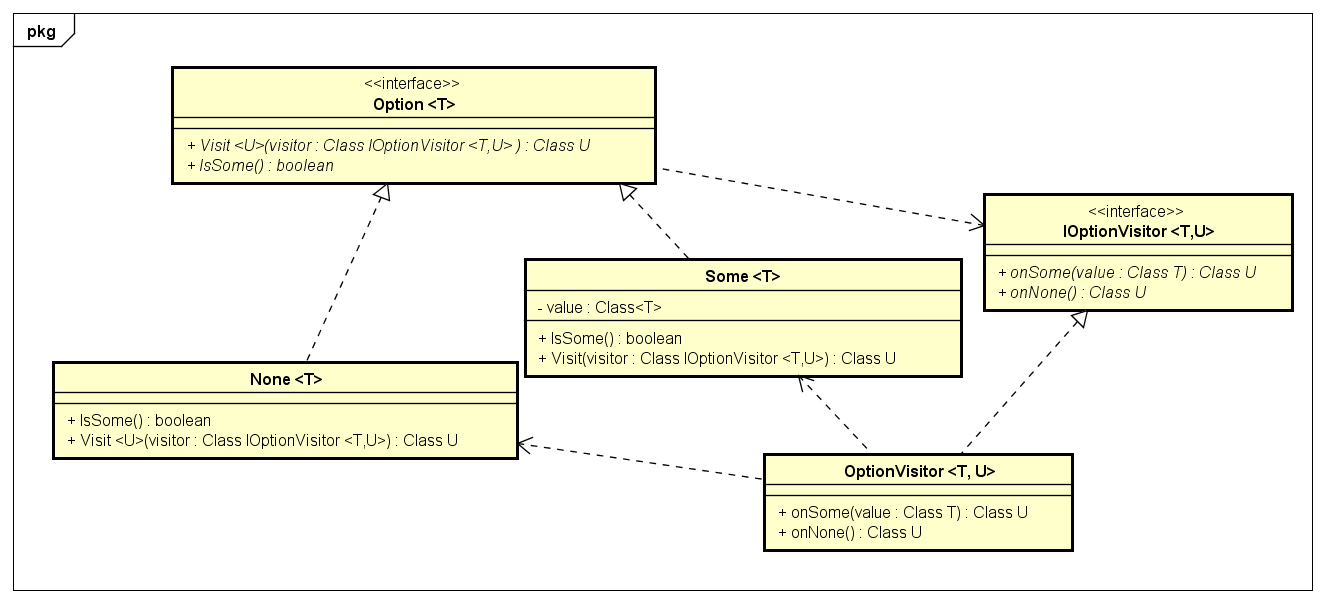
In short: AdapterArray and AdaptedList are adapters, the source is for the AdaptedArray the java array and for the AdaptedList the java list, the target interface is the iterator interface, and all the activity classes are the clients

**Concrete implementation of an Iterator design pattern.**



**The visitor design pattern**

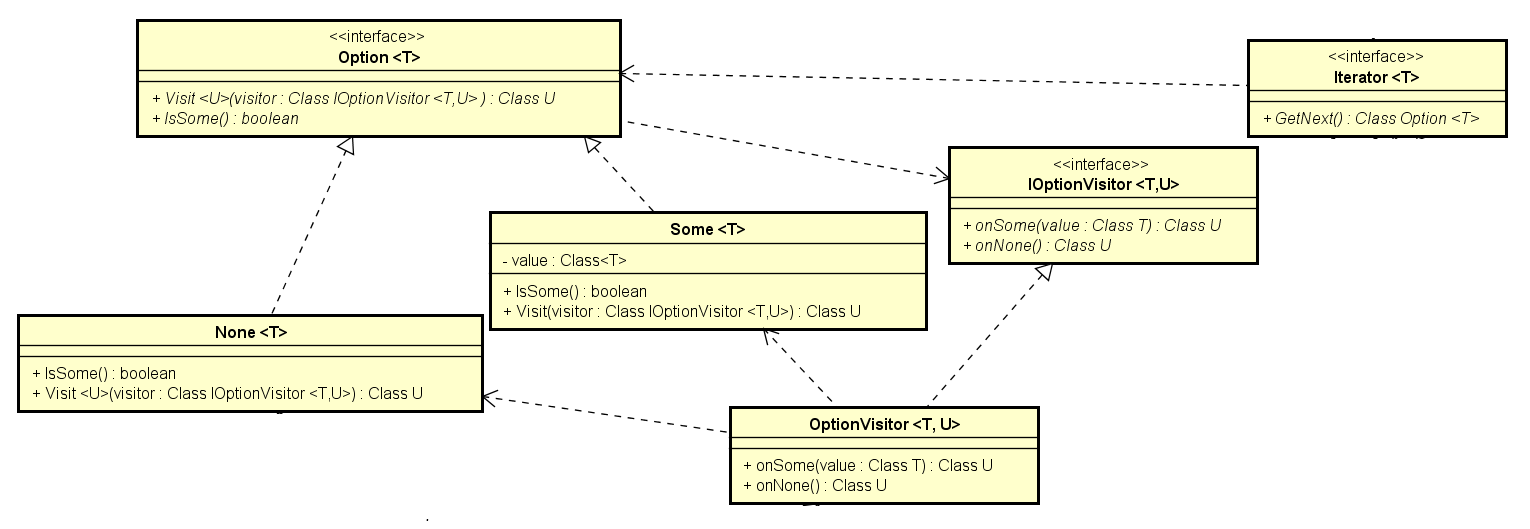
**Formal representation of a generic Iterator design pattern. (Abbadi’s approval for making the formal generic design pattern by ourselfs)**



Client

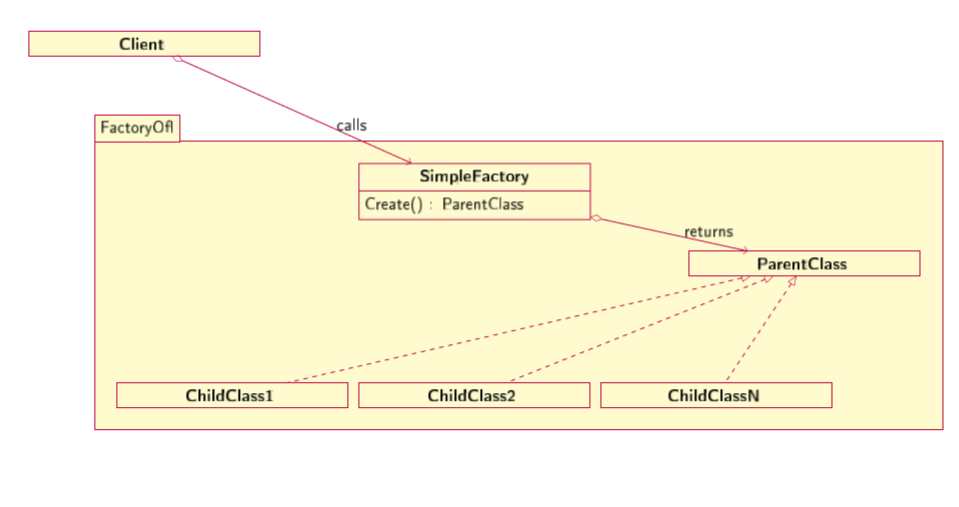
Because the concrete implementation is supposed to be very generic, not that much has changed. The only difference here is that a client is specified. The client is in this case the iterator that passes the information to a lot of other classes in the application.

**Concrete implementation of an Iterator design pattern.**



**The factory design pattern**

**Formal representation of a generic Iterator design pattern. (source: slides Dev 4)**



Asdasdasda

**Concrete implementation of an Iterator design pattern.**