**Difference between Protocol and Subclassing**

Protocols:

A class can conform to many protocols simultaneously.

Subclass:

A class can inherit only from one class directly, however a class can be a parent to many child classes.

Protocols:

Protocols are more flexible than a normal class interface, since they let you reuse a single API declaration in completely unrelated classes. This makes it possible to represent horizontal relationships.

Subclass:

A child class inheriting from a parent class has an "is-a" relationship. They are strictly meant to be used only in the classes that have related members and methods.

Protocols:

Protocols are used to declare methods which are supposed to be used for a specific situation. When the method is defined, it may have different definitions with respect to each class. A class conforms to a protocol only when the class has implemented all the required methods of the protocol. Thus, it ensures that a certain form of method template is implemented.

Subclass:

Subclassing is used when a class needs to inherit certain properties and behaviour from its another class, often termed as parent class. The child class also has a provision to either define its own additional behavior or override the behaviour it inherited from the parent class.

Protocols:

A protocol is used to declare methods and properties that are not specific to a class. It can be used by any class.

Subclass:

In contrast, classes are used to declare methods and properties associated with only that class.

Protocols:

Protocols do not have any variables associated with them and hence a class conforming to it will not have access to any additional variables.

Subclass:

A child class associates the variables with them which were used in a parent class thereby allowing access to the parent's variables as well.