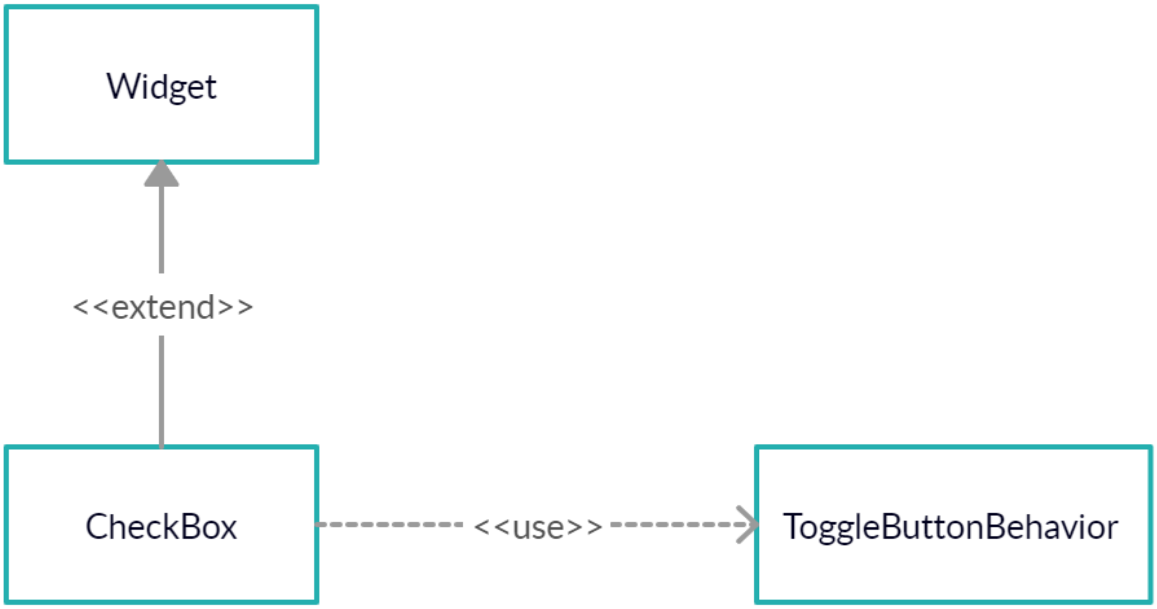
Patterns in Kivy Framework

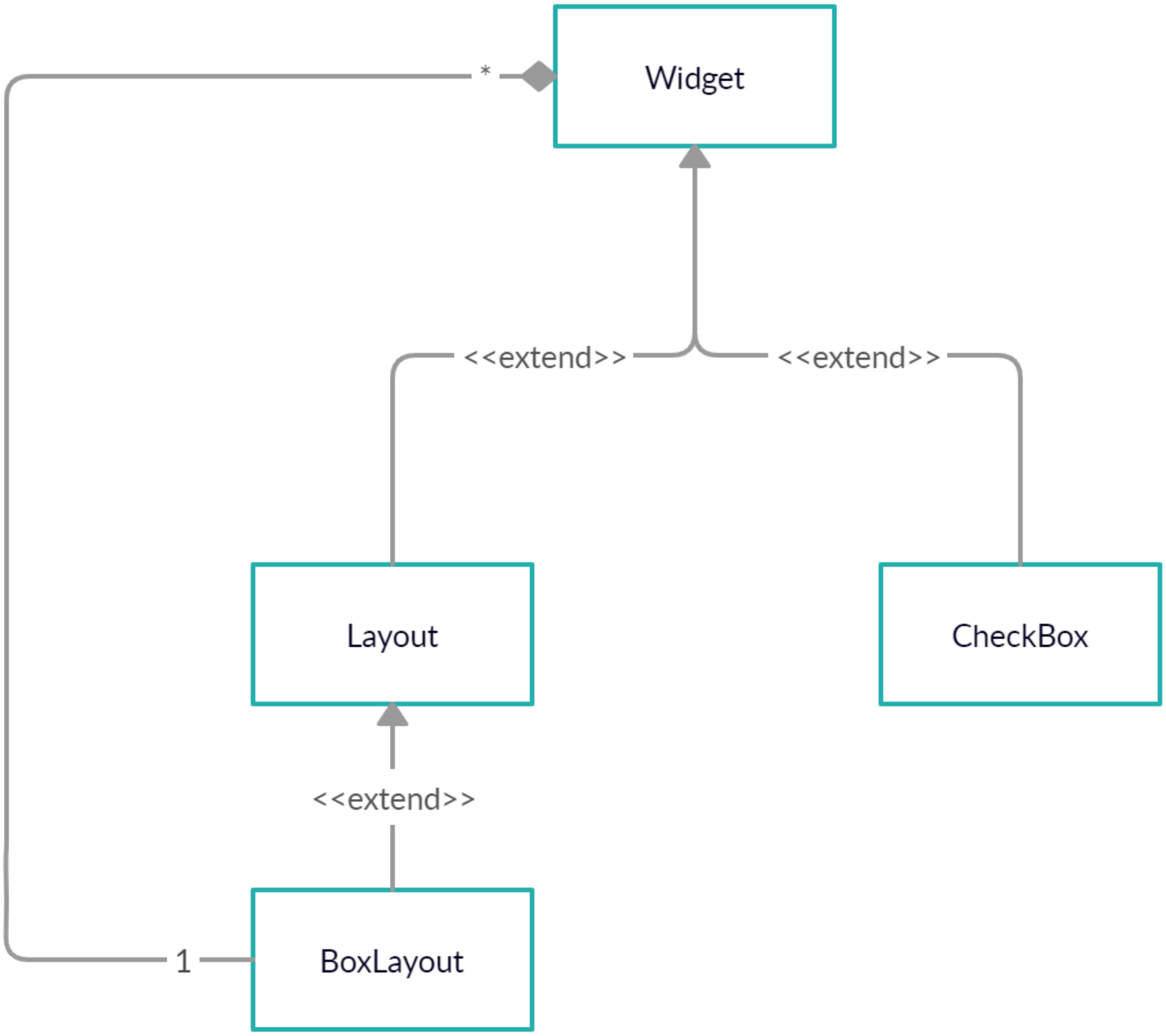
Group17 Members:

MSCS18002
MSEE19001
MSCS20001

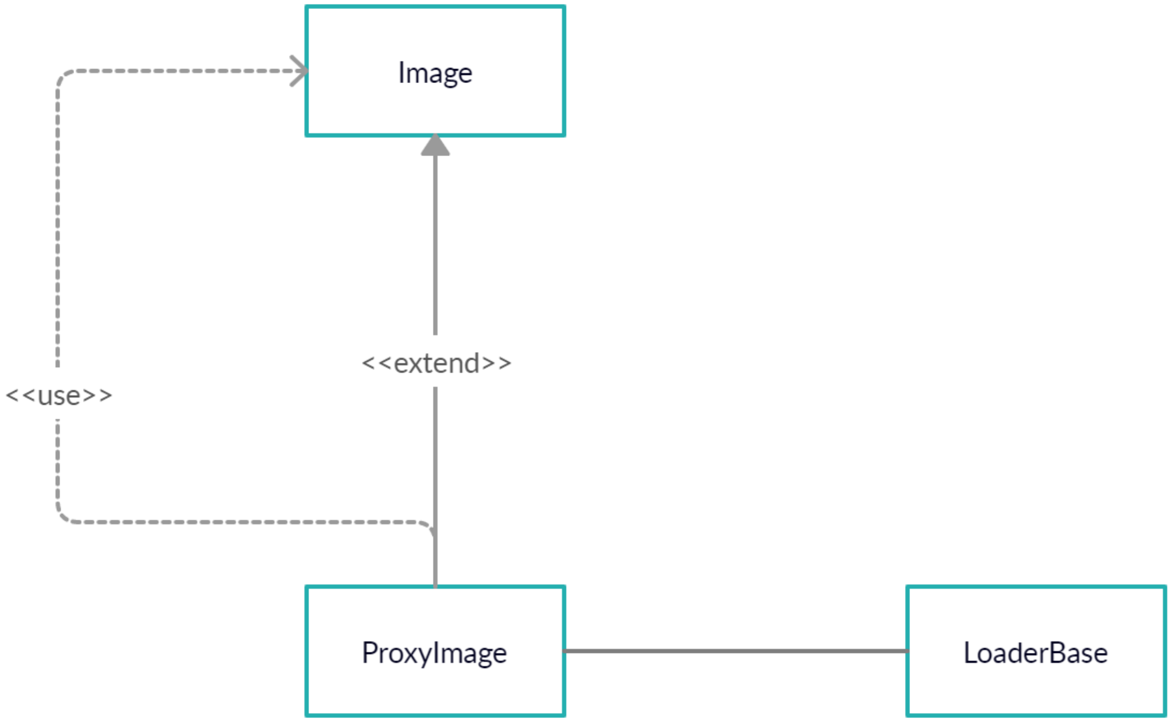
Strategy Pattern



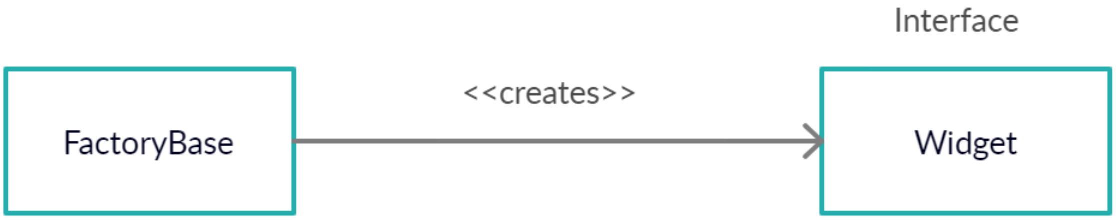
Composite Pattern



Proxy Pattern (Not Exactly)



Abstract Factory



```
>>> widget = Factory.Widget(pos=(456,456))
```

Besides, Observer Pattern and MVC Patterns also seem to exist but code base is very complex and we cannot make a class diagram.

Observer Pattern

Observer pattern is used in Kivy's properties:

```
class MyClass(EventDispatcher):
    numeric_var = NumericProperty(1)
```

To use them, **you have to declare them at class level**. That is, directly in the class, not in any method of the class. A property is a class attribute that will automatically create instance attributes. Each property by default provides an ``on_<propertyname>`` event that is called whenever the property's state/value changes.

Kivy provides the following properties:

- :mod:`~kivy.properties.NumericProperty`,
- :mod:`~kivy.properties.StringProperty`,
- :mod:`~kivy.properties.ListProperty`,
- :mod:`~kivy.properties.ObjectProperty`,
- :mod:`~kivy.properties.BooleanProperty`,
- :mod:`~kivy.properties.BoundedNumericProperty`,
- :mod:`~kivy.properties.OptionProperty`,
- :mod:`~kivy.properties.ReferenceListProperty`,
- :mod:`~kivy.properties.AliasProperty`,
- :mod:`~kivy.properties.DictProperty`,

MVC Pattern

The adapter part of the RecycleView which together with the layout is the view part of the MVC pattern.

The view module handles converting the data to a view using the adapter class which is then displayed by the layout. A view can be any Widget based class.

However, inheriting from RecycleDataViewBehavior adds methods for converting the data to a view.