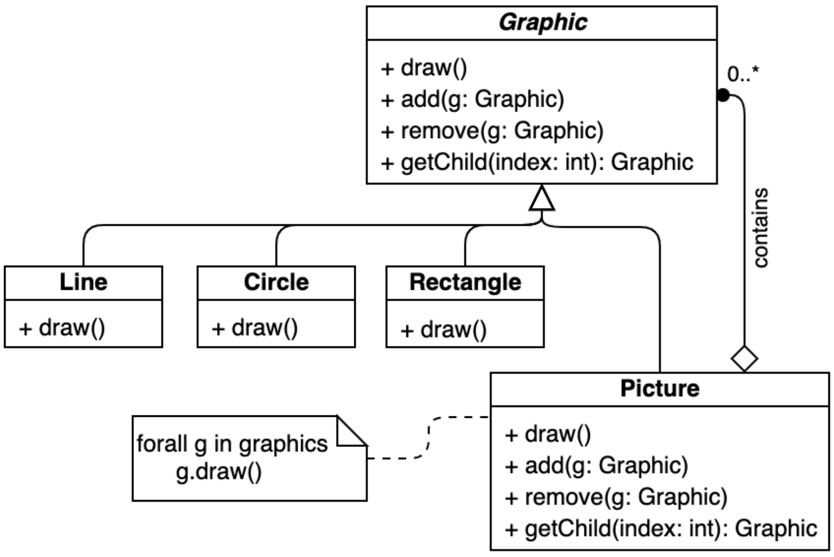
CSF202 – Lab Class 8 – Wednesday 11/11/2020

This lab class involves getting some hands-on experience with the Composite design pattern in Java.

☐ Task 8.1



For this task we will implement the Composite pattern in Java by following the "Graphic" example above.

Implement the above class diagram, creating a composition hierarchy in which we have a *Graphic* class which represents both the primitive shapes *Rectangle, Line and Circle* and the container class *Picture.*

The methods in Graphic can either be made abstract and implementations created in the subclasses or they could be ordinary methods that throw an UnsupportedOperationException.

In the Picture class **getChild()** can be handled by the **get()** method of an ArrayList.

Draw methods can simply be a print statement, i.e. `myRectangle.draw()` will print out `"A pretty rectangle"`. It may be useful for the `draw()` method of the composite class to indicate we are inside a container of some form. Perhaps:

```
myPicture.draw();
>>    "A pretty Picture, containing[ "
>>    "A pretty Rectangle"
>>    "]"
```

What you should do:

- Implement the above class diagram of the Composite pattern in Java.
- In a main class create a Picture. Add a few primitive shapes to it and draw the contents.
- Add a new Picture inside our outer picture and add some shapes to the nested picture. Draw the whole tree.
- What happens when we remove the nested picture and draw again?