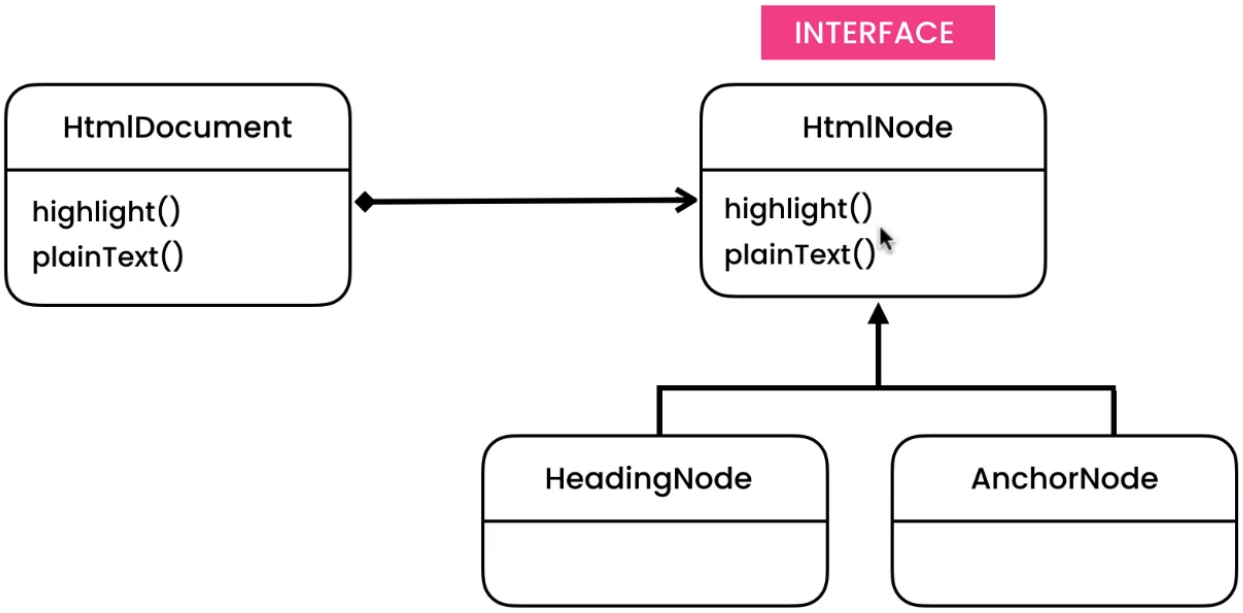
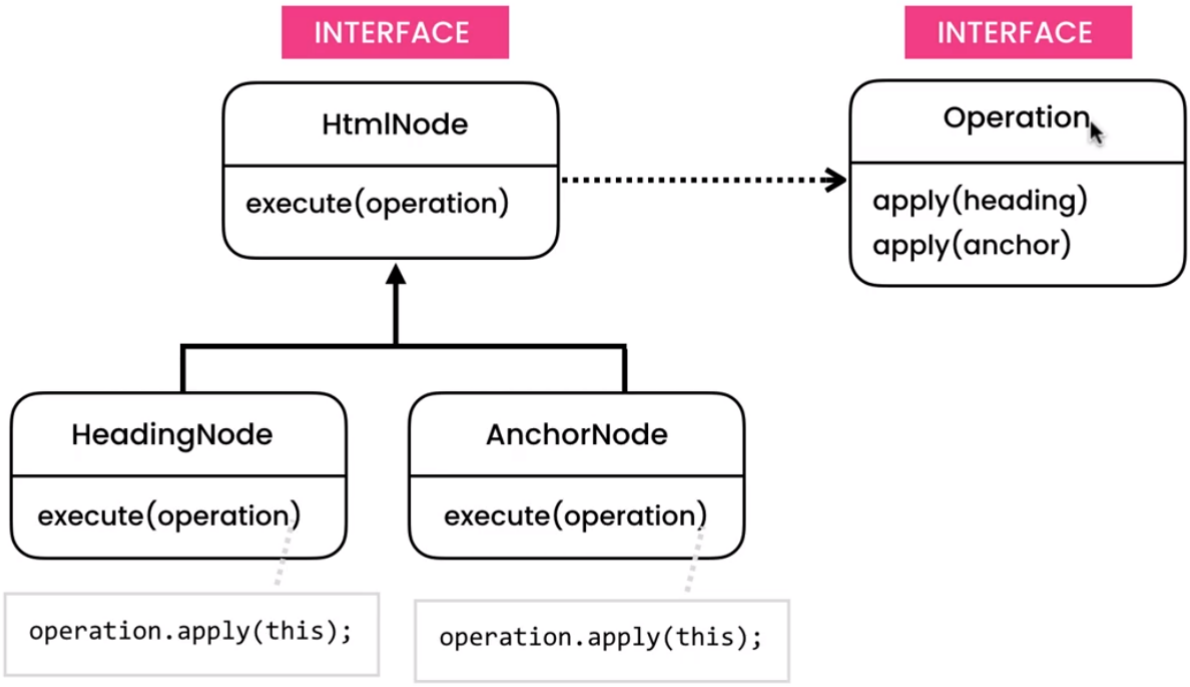
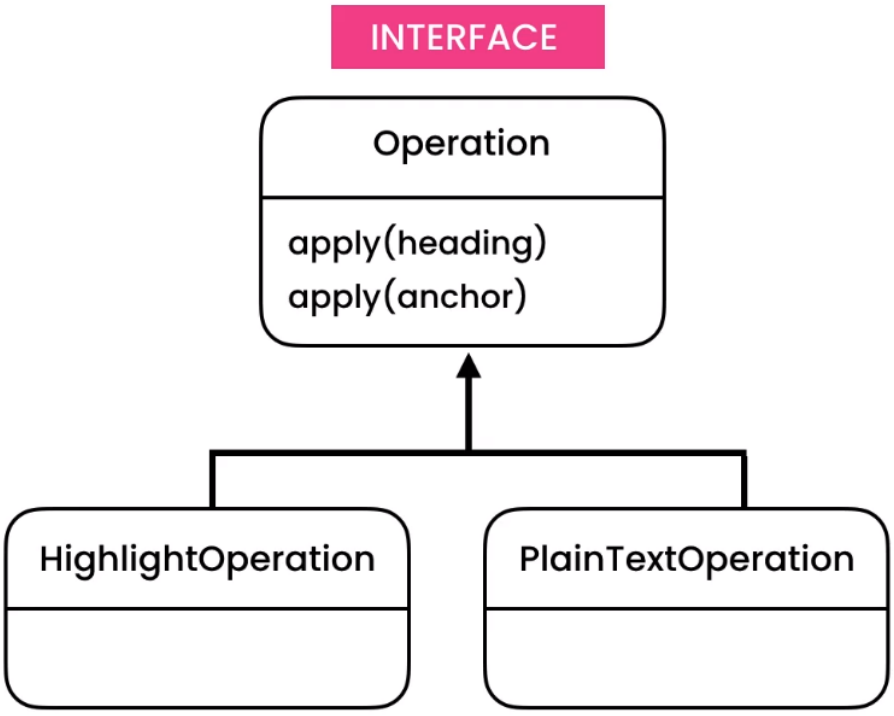
Visitor

Example Problem - Using an interface to ensure a given method is called in for eaxample a HTML text editor such as VS Code - currently the Heading HTML Node Class implements the highlight function as does the AnchorNode. But you want to intoduce a plainText method to grab the text from headings so you add it to your HtmlNode interface the result being that you now have to add it the AnchorNode class leading to code duplication and the logic for the plainText being spread over multiple classes



Solution - To solve this problem we can implement an Operation/Visitor interface which contains an overloaded apply method for each Object Type - this means that this solution is only useful when your Objects are likely to remain stable but the operations you wish to perform on them are likely to change. Such as with a HTML Docuemnet which has roughly 30 different objects / tag types but a HTML Editor's features will constantly update.





HtmlDocument	HtmlNode	AnchorNode	HeadingNode

```
public class HtmlDocument {
  private List<HtmlNode> nodes = new ArrayList<>();

 public void add(HtmlNode node) {
    nodes.add(node);
 }
//iterates over all the nodes in the document

//applying the operation which is passed to it
  public void execute(Operation operation) {
    for (var node : nodes)
      node.execute(operation);
 }
}
```

```
public interface HtmlNode {
  void execute(Operation operation);
}
```

```
public class AnchorNode implements HtmlNode{
  @Override
  public void execute(Operation operation){

    operation.apply(this);

//using this will cause the correct method to be selected

//depending ion the type of the operation class


 }
}
```

```
public class HeadingNode implements HtmlNode{
  @Override
  public void execute(Operation operation){
    operation.apply(this);
 }
}
```

Operation

```
public interface Operation {
//one for each of the Node Objects above

  void apply(HeadingNode heading);
 void apply(AnchorNode anchor);
}
```

PlainTextOperation

```
public class PlainTextOperation implements Operation {
  @Override
  public void apply(HeadingNode heading) {
    System.out.println("text-heading");
 }

  @Override
  public void apply(AnchorNode anchor) {
    System.out.println("text-anchor");
 }
}
```

HighlightOperation

```
public class HighlightOperation implements Operation {
  @Override
  public void apply(HeadingNode heading) {
    System.out.println("highlight-heading");
 }

  @Override
  public void apply(AnchorNode anchor) {
    System.out.println("highlight-anchor");
 }
}
```

Main

```
public class Main {

 public static void main(String[] args) {

var document = new HtmlDocument();

documentadd(new HeadingNode());

document.add(new AnchorNode());

//can be swapped out for a new operation without need for updating any of the classes above

document.execute(new PlainTextOperation());

}

}
```