```
1
     #include <iostream>
 2
     #include "Page.cpp"
     #include "Paragraph.cpp"
 3
     #include "Span.cpp"
 4
     #include "Image.cpp"
 5
 6
 7
     int main() {
 8
         std::shared ptr<Page> page = std::make shared<Page>();
9
         std::shared ptr<Paragraph> p1 = std::make shared<Paragraph>();
10
         p1->addElement(std::make shared<Span>());
         p1->addElement(std::make shared<Span>());
11
12
         std::shared ptr<Paragraph> p2 = std::make shared<Paragraph>();
13
         p2->addElement(std::make shared<Image>());
14
         page->addElement(p1);
15
         page->addElement(p2);
16
         std::cout << page->render();
17
     }
18
19
20
     /* Image.cpp */
     #include "PageElement.cpp"
21
22
23
     class Image : public PageElement {
24
     public:
25
         std::string render() {
             return "<img src=\"IMG\"/>";
26
27
28
     };
29
30
     /* Page.cpp */
31
     #include <forward list>
32
     #include <list>
     #include "PageElement.cpp"
33
34
35
     class Page : public PageElement {
36
     private:
37
         std::list< std::shared ptr<PageElement> > mylist;
38
     public:
39
         void addElement(std::shared ptr<PageElement> e){
40
            mylist.push back(std::move(e));
41
42
         std::string render() {
43
44
             std::string rendered = "<page>\n";
             for (std::shared ptr<PageElement> e : mylist)
45
46
                  rendered += e->render() + "\n";
47
             return rendered + "</page>";
48
         }
49
50
     /* PageElement.cpp */
51
     #pragma once
52
53
     class PageElement {
54
     public:
55
         virtual std::string render() = 0;
56
     };
57
58
59
60
61
62
63
```

64

/afs/.ist.utl.pt/users/6/5/ist178865/ist/P01516/9_aula/9_aula.cpp Page 2 of 2 Wed 18 Nov 2015 08:49:28 AM WET

```
65
     /* Paragraph.cpp */
66
     #include <forward list>
     #include <list>
67
     #include "PageElement.cpp"
68
69
     class Paragraph : public PageElement {
70
71
     private:
         std::list< std::shared ptr<PageElement> > mylist;
72
73
     public:
74
         void addElement(std::shared ptr<PageElement> e){
75
            mylist.push back(std::move(e));
76
         }
77
78
         std::string render() {
79
             std::string rendered = "\n";
80
             for (std::shared ptr<PageElement> e : mylist)
81
                 rendered += e->render() + "\n";
             return rendered + "";
82
83
         }
84
     };
85
86
     /* Span.cpp */
87
     #include "PageElement.cpp"
88
89
     class Span : public PageElement {
90
     public:
91
         std::string render() {
92
             return "<span>TEXT</span>";
93
         }
94
     };
95
```