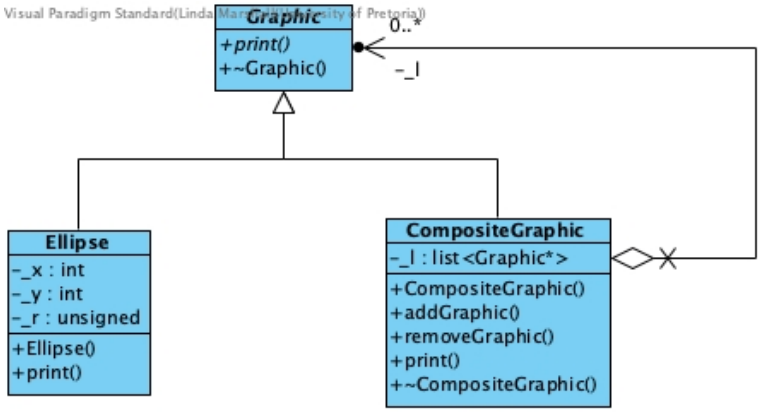
UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

COS 214 Tutorial 7

- This tutorial takes place on **18 October 2021**.
- This tutorial consists of **5 questions**.
- The tutorial does not contribute towards your final marks.

Consider the following UML Class diagram and answer all the questions that follow.



Question 1: Activity Diagram ..... (18 marks)

The following is the code of the main function of a client that was written to demonstrate the system. The `initiate()` function of the `Client` generates a random composite figure.

```
1  int main()
2  {
3    bool HORIZONTAL = true;
4    bool VERTICAL = false;
5    unsigned int g1Size, g2Size;
6    CompositeGraphic* graph1 = initiate();
7    CompositeGraphic* graph2 = initiate();
8    g1Size = graph1->getNumOfLeaves();
9    g2Size = graph2->getNumOfLeaves();
10   while (g1Size == g2Size)
11   {
12     delete graph1;
13     graph1 = initiate();
14     g1Size = graph1->getNumOfLeaves();
15   }
16   if (g1Size < g2Size)
17   {
18     graph1->rotate(90);
19   }
20   else
21   {
22     delete graph1;
23     graph1 = initiate();
```

```
24      g1Size = graph1->getNumOfLeaves();
25      if (g1Size == g2Size){
26        graph2->rotate(180);
27        graph1->flip(HORIZONTAL);
28      }
29      else if (g1Size < g2Size)
30      {
31        graph2->flip(VERTICAL);
32      }
33      else
34      {
35        graph1->flip(VERTICAL);
36      }
37    }
38    graph1->display();
39    graph2->display();
40    delete g1;
41    delete g2;
42    return 0;
43  }
```

Draw a UML Activity diagram of the program flow of this code. Your diagram should have swimlanes for `Client` and `Graphic`.

Question 2: Command and Mediator ..... (21 marks)

You have a TV and its remote. The remote only has two buttons, one for toggling between on and off, and the other to change between the 5 programmed channels on the TV. A testing code snippet from the client perspective is given by:

```
1   TVRemote* tvr = new TVRemote;
2   tvr->onPushed();
3   tvr->channelChanged();
4   tvr->channelChanged();
5   tvr->channelChanged();
6   tvr->onPushed();
7   tvr->channelChanged();
8   tvr->channelChanged();
```

2.1 Which participants of the Command pattern do the TV and TV remote represent? (2)

2.2 Given that the *Command* participant is defined as follows:

```
1  class Command {
2    public:
3      virtual void execute() = 0;
4    protected:
5      virtual char* getStatus() = 0;
6  };
```

a) Draw the UML class diagram showing all participants in the Command hierarchy. (8)

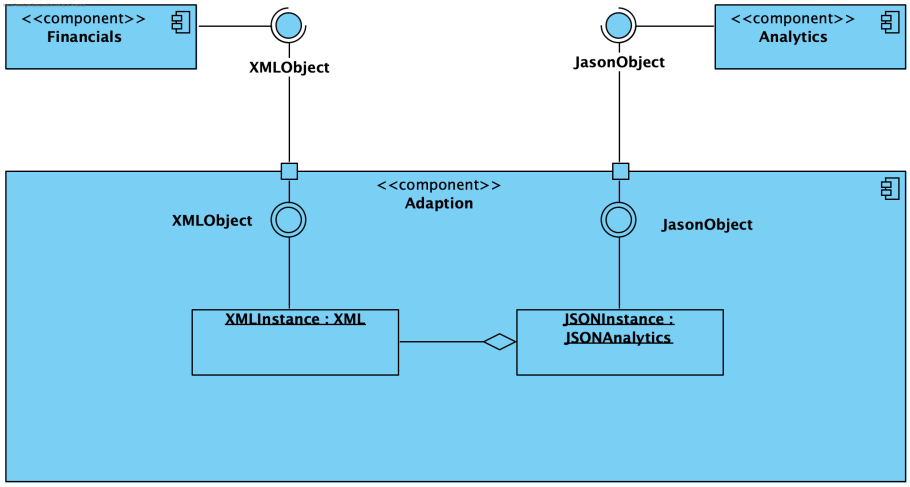
b) Provide a class definition and implementation of the functions for the TV remote. (6)

2.3 Assume this remote has a touch screen interface and can be used on multiple TV's by choosing the TV to be changed on the list presented on the touch screen interface. How would you apply the Mediator Design pattern to provide this kind of functionality? (5)

Your client is currently using a financial system that makes use of XML to exchange data between components of their system. They have invested large quantities of money into this system and have a vast amount of data linked to this system which is stored in a data lake. Due to this investment, in the existing system, they are not ready to move over to a new system just yet. They have however

purchased a system that will move them to the cutting edge again in terms of data analytics on the data they have in their data lake. This new system uses JSON to exchange data between its components.

Below is an architectural view of the systems and how they are to be adapted so that they can work together.



This is a UML Component diagram. The components (Financials, Analytics and Adaption) comprise of multiple classes. The lollipops (blue circles), represent the provided interface between components and the surrounding circle (socket), the required interface. `XML` *and* `JSONAnalytics` *are classes defined within the Adaption component.*

Effectively, communication can take place between the Finances component and the Analytics component and vice versa. We will only be considering communication from the Finances to the Analytics for the majority of the questions.

Question 3: 3 ..... (13 marks)

You decide to apply the Object Adapter design pattern to adapt the XML data format to JSON.

3.1 The classes that exist in the Adaption component are `XMLInterface`, `XML`, `JSONInterface` and `JSONAnalytics`. `XMLInterface` and `JSONInterface` are both abstract classes. (5)

Draw the UML Class diagram for the Object Adapter for this scenario. Show only the classes, whether they are abstract and the relationships between the classes. *Do NOT include any features (functions and attributes)*

3.2 Identify the corresponding classes for the participants of the Adapter design pattern. (3)

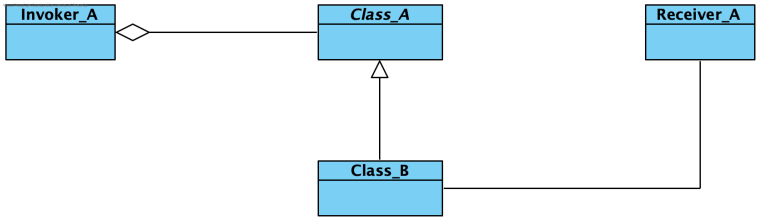
3.3 The `JSONInterface` class is given by: (5)

```
1    class JSONInterface {
2    public:
3    virtual string convert(string) = 0;
4    };
5
```

Provide a definition for the `JSONAnalytics` class.

Question 4: 4 ..... (11 marks)

Rather than sending XML data from the Financials system to the Adaption system as text strings, you decide to send the data '*encapsulated*' as a command. The following is an incomplete representation of the Command design pattern.



4.1 In which components will the classes representing the Invoker and Receiver participants of the command pattern be defined? (2)

4.2 Provide apt names, taking the scenario into account, for the classes labelled `Class_A` and `Class_B`. (2)

4.3 Which class, defined in Question 3, should have a relationship with `Class_B` and what type of relationship should this be? (2)

4.4 If commands were to be sent from the `Analytics` component to the `Adaption` component, how would the Command UML Class diagram given above change? (2)

4.5 Propose a definition for `Class_A`. (3)

Question 5: 5 ..... (12 marks)

5.1 What is the main purpose of each of the following? (2)

a) Doxygen

b) GitHub

5.2 The following section of a web page was generated using Doxygen. The code that was used to generate the page is the JSONInterface class definition.

Detailed Description

The JSON Interface

Member Function Documentation

◆ convert()

virtual string JSONInterface::convert (string) pure virtual

Pure virtual function to convert a string from XML to JSON

Parameters

xmlString is a string

a) How are multi-line comments specified in Doxygen? (1)

b) Where do you think the text used to populate the *Detailed Description* section is placed in the code file? (1)

c) What text on the web page was produced using a tag and what tag was it that was used to produce the text? (2)

d) The return type of `convert()` is not shown in the documentation. How would you include it? (2)

5.3 What advantages does using a Git repository hold over not using one? (4)