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

COS 214 Class Test 6 - L22 to L26

- This test takes place on **16th October 2020**.
- The maximum duration of this test is **40 minutes**.
- This test consists of **4 questions** for a total of **20 marks**.

Question 1 ....(5 marks)

Consider the following code for the Adaptee, Target and Adapter participants of the Adapter design pattern.

```
class InsaneWrapper {
public:
  InsaneWrapper(int const& _value);
  InsaneWrapper& operator = (int _value) ;
  InsaneWrapper& operator = (InsaneWrapper &_wrapper) ;
  operator int const () const;
  InsaneWrapper& operator ++ ();
  InsaneWrapper& operator ++ (int);
  InsaneWrapper& operator -- ();
  InsaneWrapper& operator -- (int);
  bool operator == (InsaneWrapper const& _wrapper);
private:
  int value;
};

class Wrapper {
public:
  virtual void print(ostream&) = 0;  // print the object
  virtual void increment() = 0;  // increment the wrapped object
  virtual void decrement() = 0; // decrement the wrapped object
  virtual void update(Wrapper*) = 0; // update the wrapped object
  virtual ~Wrapper() {};
};

class SaneWrapper : public Wrapper {
public:
  SaneWrapper();
  SaneWrapper(int);
  virtual void print(ostream&);
  virtual void increment();
  virtual void decrement();
  virtual void update(Wrapper*);
  virtual ~SaneWrapper();
protected:
  InsaneWrapper* object;
};
```

1.1 The Adapter participant in the given code is which of the following classes? (1)

A. `InsaneWrapper`

B. `Wrapper`

C. `SaneWrapper`

D. None of the above.

1.2 `InsaneWrapper` is which participant of the Adapter design pattern? (1)

A. `Adaptee`

B. `Adapter`

C. Client

D. `Target`

1.3 Which of the following class definitions will result in the given code being an implementation of a Class Adapter rather than an Object Adapter. (1)

A. The following line remains the same, **class** SaneWrapper : **public** Wrapper

B. **class** ClassSaneWrapper : **public** Wrapper

C. **class** SaneWrapper : **public** Wrapper, InsaneWrapper

D. **class** SaneWrapper : **private** Wrapper, InsaneWrapper

E. **class** SaneWrapper : **public** Wrapper, **private** InsaneWrapper

F. **class** SaneWrapper : **private** Wrapper, **public** InsaneWrapper

1.4 Which lines need to be removed from the given `SaneWrapper` implementation? (1)

A. Line 23

B. Line 25

C. Line 34

D. Line 35

E. Lines 23 and 25

F. Lines 34 and 35

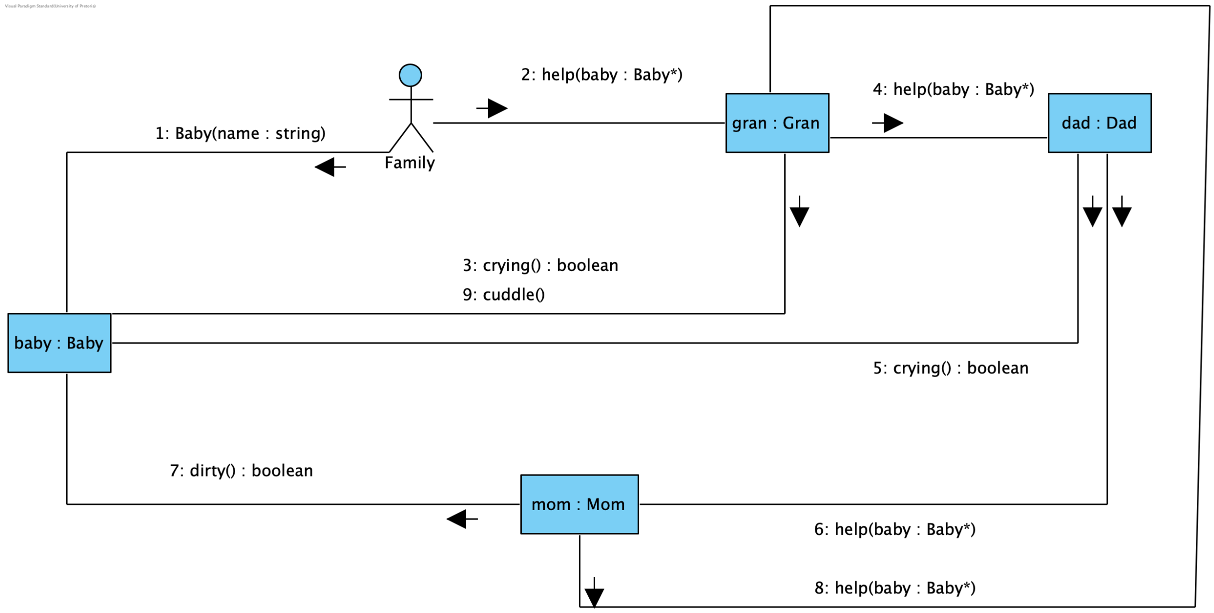
1.5 "An implementation of the destructor is necessary for the Class Adapter implementation of the `SaneWrapper`". (1)

A. True

B. False

Question 2 .... (6 marks)

Consider the following UML Communication diagram and answer the questions that follow.



Assume that all objects, other than `baby`, are instantiations of classes which inherit from the class `Adult`. Class `Adult` implements an association which results in the `Adult` hierarchy implementing *recursive composition*. The design pattern which gave rise to the communication diagram above is the *Chain of responsibility.*

2.1 Which of the following functions will be defined in class `Baby`? (3)

A. `crying`

B. `cuddle`

C. `dirty`

D. `help`

2.2 *Recursive composition* in terms of the `Adults` class means that: (1)

A. an aggregation exists between class `Adult` and class `Gran`

B. at least a composition association exists from class `Adult` to class `Adult`.

C. a chain of composition associations exist from the client to the `Gran` to `Dad` to `Mom` classes.

D. a chain of composition associations exist from the client to the `Baby` to `Gran` to `Dad` to `Mom` classes.

2.3 If you were to explain to a fellow student how the chain works you would in all probability draw a representation of the chain. Which representation of the chain would you use for your explanation? (1)

A. dad : Dad → mom : Mom → gran : Gran

B. dad : Dad → gran : Gran → mom : Mom

C. gran : Gran → dad : Dad → mom : Mom

D. gran : Gran → mom : Mom → dad : Dad

E. mom : Mom → gran : Gran → dad : Dad

F. mom : Mom → dad : Dad → gran : Gran

2.4 Which function represents the *handleRequest* function of the *Chain of responsibility* design pattern? (1)

A. `crying`

B. `cuddle`

C. `dirty`

D. `help`

Question 3 ....(4 marks)

You have been given the following list of classes and told that they are all participants of the Builder design pattern.

- `HousePlan` - an abstract class
- `House` - a specialisation of class `HousePlan`
- `Home` - another abstract class
- `Igloo` and `Tipi` - specialisations of `Home`, each hold a handle to a `House` object.
- `CivilEngineer` - has an aggregation association with Home.

For each of the following participants of the Builder design pattern, identify the corresponding classes.

3.1 Builder (1)

A. `HousePlan`

B. `Home`

C. `Igloo`

D. `CivilEngineer`

3.2 ConcreteBuilder (1)

A. `House`

B. `Home`

C. `Tipi`

D. `CivilEngineer`

3.3 Director (1)

A. `House`

B. `Home`

C. `Igloo`

D. `CivilEngineer`

3.4 Product (1)

A. `House`

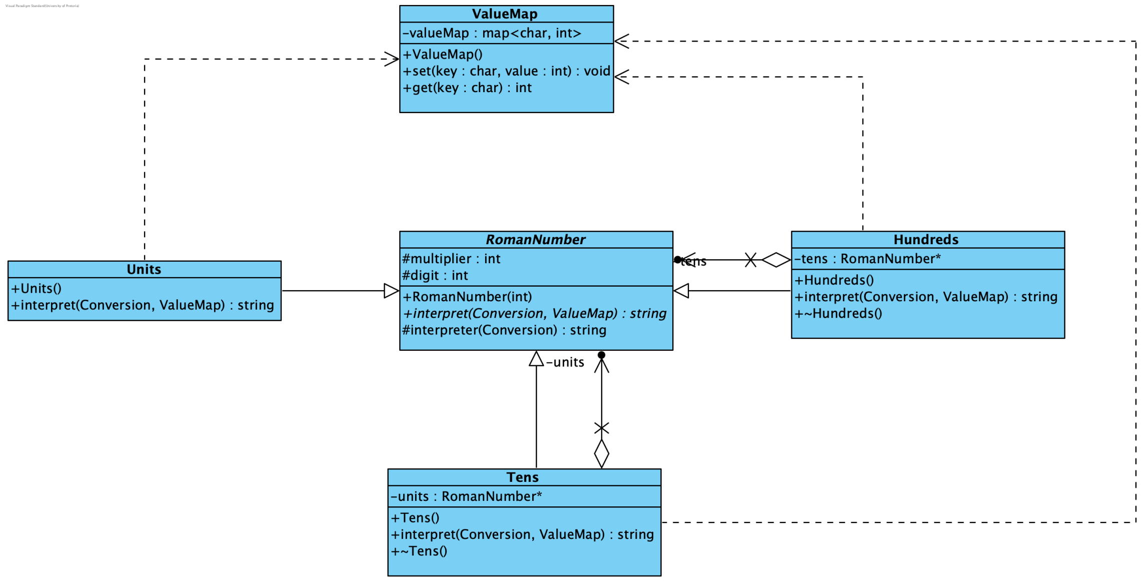
B. `Home`

C. `Tipi`

D. `CivilEngineer`

Question 4 ....(5 marks)

Consider the following UML class diagram and answer the questions that follow.


Visual Paradigm Standard(University of Pretoria)
ValueMap
-valueMap : map<char, int>
+ValueMap()
+set(key : char, value : int) : void
+get(key : char) : int
Units
+Units()
+interpret(Conversion, ValueMap) : string
RomanNumber
#multiplier : int
#digit : int
+RomanNumber(int)
+interpret(Conversion, ValueMap) : string
#interpreter(Conversion) : string
tens
Hundreds
-tens : RomanNumber*
+Hundreds()
+interpret(Conversion, ValueMap) : string
+~Hundreds()
-units
Tens
-units : RomanNumber*
+Tens()
+interpret(Conversion, ValueMap) : string
+~Tens()

4.1 Which pattern is shown in the given UML class diagram? (1)

A. Adapter

B. Builder

C. Composite

D. Decorator

E. Interpreter

4.2 The `Hundreds` class is referred to as the ________ participant of the pattern. (1)

A. Composite

B. Leaf

C. Nonterminal

D. Terminal

4.3 Which class in the diagram represents the `Context` participant of the pattern? (1)

A. Hundreds

B. RomanNumber

C. Tens

D. Units

E. ValueMap

4.4 What is the multiplicity of the aggregate relationship between the `Tens` and the `RomanNumbers` classes? (1)

A. 0

B. 1

C. 0..1

D. 0..n

E. 1..*

4.5 What is the largest integer value that can be converted to a Roman numeral when the design given in the above class diagram is implemented? The BNF for the conversion from integer to Roman numerals, on which the class diagram is based, is given by: (1)

```
RomanNumber ::= Hundreds Tens Units
Hundreds ::= LowHundreds | CD | D LowHundreds
LowHundreds ::= Empty | LowHundreds C
Tens ::= LowTens | XL | L LowTens | XC
LowTens ::= Empty | LowTens X
Units ::= LowUnits | IV | V LowUnits | IX
LowUnits ::= Empty | LowUnits I
```

A. 99

B. 199

C. 599

D. 899

E. 999

F. 1000

COS 214 Class Test 6 – L22 to L26

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

Student Number: u18050362
First Name: Werner
Last Name: Graaff

Cell: 0716407441
Email: u18050362@tuks.co.za

Question 1

1.1 C

1.2 A

1.3 E

1.4 F

1.5 B

Question 2

2.1 A

2.2 B

2.3 C

2.4 D

Question 3

3.1 B

3.2 C

3.3 D

3.4 A

Question 4

4.1 E

4.2 C

4.3 E

4.4 B

4.5 D