

Serie 4

Content:

– Design Patterns : Composite - Strategy

(17) Design Pattern : Composite - Arithmetic Expressions

Here is an abstract syntax for arithmetic expressions :

```
ArithmeticExpression ::= Expression BinOp Expression
Expression           ::= Constant | ArithmeticExpression
BinOp                ::= + | - | * | :
Constant             ::= INT
```

Examples of arithmetic expression would be :

- (5 + 7)
- (5 + 7) * (7 - 4)
- (10 : ((3 + 2) - (6 : 3))) * (4 + 3)

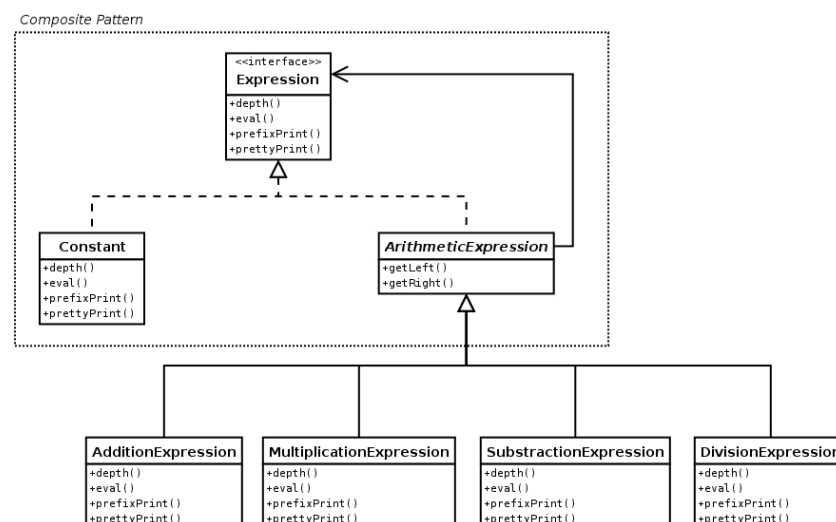


Figure 1 – Class diagram for arithmetic expressions, package composite.syntax.

1. Implement the 6 classes of the class diagram of Figure 1, using the Composite design pattern.
2. Provide appropriate implementations for the methods of the interface.
For the expression $12 : ((5 + 2) - (2 * 2))$:
 – prettyPrint() displays (12 : ((5 + 2) - (2 * 2))) ;
 – prefixPrint() displays <: 12 <- <+ 5 2> <* 2 2> > >;
 – eval() returns 4;
 – depth() returns 4.
3. Test your code with the provided main test class composite.TestCompositeExpressions.

On [1] you can find the code for :

- the class composite.TestCompositeExpressions;
- and the interface composite.syntax.Expression.

(15) Design Pattern : Strategy - Screen layout of clocks

In the preceding exercise concerning the Observer pattern we implemented a little clock application supporting different kind of displays.

Now we want to take advantage of the Strategy pattern to be able to have different layouts of the clock observers on the screen and to change it dynamically at run-time.

On [1] you can find an improved version of the clock package, where the PositionManager Singleton is the context delegating the layout tasks to a given layout strategy.

1. Study the new version of the application (see the class diagram on Figure 2).
2. Provide at least one new original layout strategy (e.g. RandomLayoutStrategy) for the various clocks.
Note : The Layout menu of the application (ClockApp) will adapt automatically !

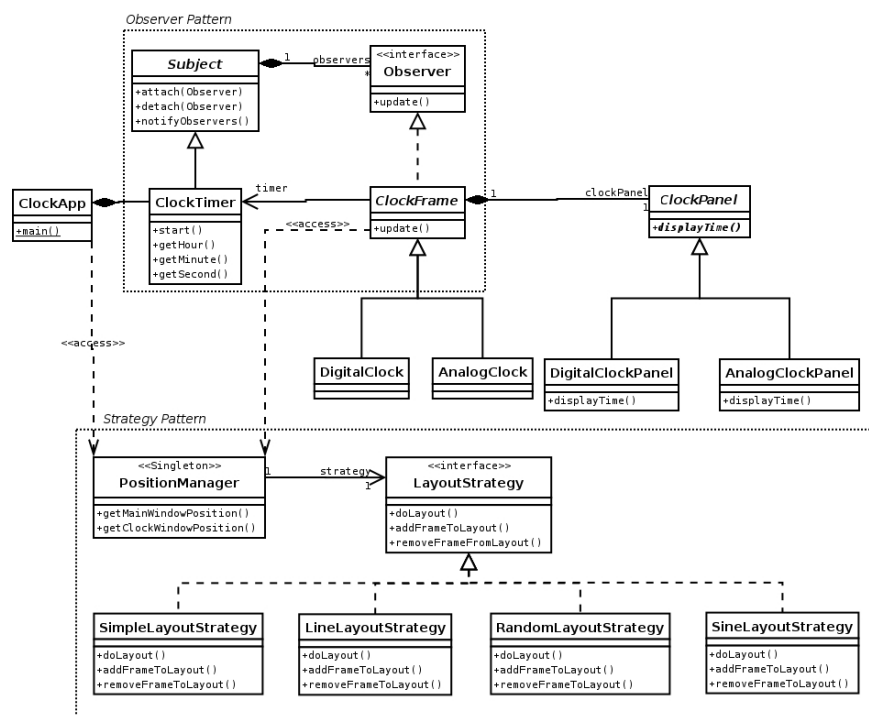


Figure 2 – Class diagram for the “strategy enhanced” clock application.

Références

- [1] Jacques Pasquier. Génie logiciel I, 2013. <http://moodle2.unifr.ch/course/view.php?id=1252> (accessed Apr 08, 2013).