



WS2018/19 – Design Patterns and Frameworks

Extensibility Patterns

Professor: Prof. Dr. Uwe Aßmann
Lectuer: Dr.-Ing. Sebastian Götz
Tutor: Dr. rer. nat. Marvin Triebel

Task 1 The Degree of Polynomials

Consider the set of polynomials over one variable (x) and their degree¹. Examples are:

$$2x^2 - 5 \text{ with degree } 2 \quad (1)$$

$$x(177x - 15x) \text{ with degree } 2 \quad (2)$$

$$(x - 2x^2)(2 + 4x) \text{ with degree } 3 \quad (3)$$

- Which design pattern can be used for representing polynomials? Draw the class diagram!
- What is the smallest yet reasonable amount of classes in the diagram?
- The function `int countSigns()` shall count the number of minus signs in a given polynomial. Which design patterns are suitable? Which patterns have which (dis)advantages?
- Which design pattern can be used to compute the degree of a polynomial?
- Implement the function `int degree()` in the created class of a polynomial.

Task 2 Secant Method of polynomials

The secant method² is a simple way to find a zero of a polynomial numerically. A pseudocode version can be found on the Wikipedia article. The secant method evaluates the polynomial for each iteration at a new x .

- What is the `Interpreter` pattern? What is its structure?

¹https://en.wikipedia.org/wiki/Degree_of_a_polynomial

²https://en.wikipedia.org/wiki/Secant_method

- b) Implement the method `evaluate(double x)` in the class `Polynomial` of the previous task. Use the `Interpreter` pattern.
- c) *optional*: Implement the second method and test the `Polynomial` class and evaluation function.
- d) You want to extend your class `Polynomial` to geometric functions (sinus, cosinus, tangens) of polynomials. Which design pattern can you use for the extension?

Task 3 Chained Observer

Consider the chained variant of the *observer* design pattern with three agents *A*, *B*, and *C*. Consider the following case: *A* observes *B* and *B* observes *C*.

- a) Draw a sequence diagram of the given scenario, where *C* notifies its observers.
- b) Now assume that also *C* observes *A*. Which problem occurs? How can we fix that problem?
- c) Draw a sequence diagram of your solution.
- d) In your solution, did you apply the *Mediator* design pattern?

References

- [1] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design patterns: Elements of Reusable Object-Oriented Software*. Pearson Education, 1994.