

Ćwiczenia 11 - Interpreter, Iterator (kursor), Łańcuch zobowiązań, Mediator, Metoda szablonowa

Interpreter

<http://devman.pl/pl/techniki/wzorze-projektowe-interpreterinterpreter/>

<https://www.dotnettricks.com/learn/designpatterns/interpreter-design-pattern-c-sharp>

<https://www.geeksforgeeks.org/interpreter-design-pattern/>

<https://www.c-sharpcorner.com/UploadFile/20c06b/learn-design-patterns-interpreter/>

J1. Stwórz projekt implementujący wzorec Interpretera do zamiany liczby z systemu szesnastkowego na dziesiętny.

Iterator (kursor)

<http://devman.pl/pl/techniki/wzorze/wzr-opr/wzorze-projektowe-iteratoriterator/>

<https://refactoring.guru/design-patterns/iterator/csharp/example>

<https://www.c-sharpcorner.com/article/iterator-design-pattern/>

J2. Zmodyfikuj kod ze wzorca Kompozytu z poprzednich ćwiczeń o strukturę Iteratora.

Łańcuch zobowiązań

<http://lukaszkosiorowski.pl/programowanie/lancuch-zobowiazan-chain-of-responsibility/>

<https://refactoring.guru/design-patterns/chain-of-responsibility/csharp/example>

<https://www.codeproject.com/Articles/743783/Reusable-Chain-of-responsibility-in-Csharp>

<https://www.dofactory.com/net/chain-of-responsibility-design-pattern>

J3. Stwórz projekt zamieniający liczby z systemu dziesiętnego (cyry arabskie) na system rzymski z wykorzystaniem Łańcucha Zobowiązań.

Mediator

<https://refactoring.guru/design-patterns/mediator/csharp/example>

<https://www.dofactory.com/net/mediator-design-pattern>

<http://devman.pl/pl/techniki/wzorze-projektowe-mediatormediator/>

J4. Stwórz dwie testowe klasy i mediatora między nimi.

Metoda szablonowa

<https://refactoring.guru/design-patterns/template-method/csharp/example>

<https://www.dotnettricks.com/learn/designpatterns/template-method-design-pattern-c-sharp>

J5. Stwórz projekt zgodnie ze wzorcem metody szablonowej do przechowywania liczb w różnych systemach pozycyjnych.