

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

КАТЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Інженерія програмного забезпечення

Лабораторна робота №7

«Шаблони поведінки. Шаблони memento, state,
command, interpreter»

Виконав:
студент групи ІО-32
Крадожон М.
Перевірів(-ла):
Васильєва М.

Київ – 2023

Лабораторна робота №7

Тема: «Шаблони поведінки. Шаблони memento, state, command, interpreter»

Мета: Вивчення шаблонів поведінки. Отримання базових навичок з застосування шаблонів memento, state, command, interpreter.

Варіант: $3215 \% 8 = 7$

Визначити специфікації класів для розбору алгебраїчних виразів з операціями +, -, *, /.

Код (TypeScript):

```
1  /**
2   * Інтерфейс для вузлів виразу. Забезпечує методи для оцінки значення та конвертації в
рядок.
3   */
4   interface ExpressionNode {
5   /**
6    * Оцінює значення вузла виразу.
7    * @returns Числове значення.
8    */
9    evaluate(): number
10
11   /**
12    * Перетворює вузол виразу в рядкове представлення.
13    * @returns Рядок, що представляє вираз.
14    */
15    toString(): string
16   }
17
18   /**
19    * Інтерфейс для зберігання операторів та їх логіки виконання.
20    */
21   interface OperatorsProps {
22     [key: string]: (a: number, b: number) => number
23   }
24
25   /**
26    * Клас для представлення числового вузла в дереві виразу.
27    */
28   class NumberNode implements ExpressionNode {
29     private value: number
30
31     /**
32      * Створює новий числовий вузол.
33      * @param value Числове значення.
34      */
35     constructor(value: number) {
36       this.value = value
```

```

37 }
38
39 /**
40  * Оцінює значення числового вузла.
41  * @returns Числове значення вузла.
42  */
43 evaluate(): number {
44     console.log(`Метод evaluate викликаний для NumberNode зі значенням ${this.value}`)
45     return this.value
46 }
47
48 /**
49  * Повертає рядкове представлення числового вузла.
50  * @returns Рядок з числовим значенням.
51  */
52 toString(): string {
53     return this.value.toString()
54 }
55 }
56
57 /**
58  * Клас для представлення вузла оператора в дереві виразу.
59  */
60 class OperatorNode implements ExpressionNode {
61     private operator: string
62     private left: ExpressionNode
63     private right: ExpressionNode
64
65     /**
66      * Статичне поле для зберігання доступних операторів і їх логіки.
67      */
68     private static operators: OperatorsProps = {
69         '+': (a, b) => a + b,
70         '-': (a, b) => a - b,
71         '*': (a, b) => a * b,
72         '/': (a, b) => {
73             if (b === 0) throw new Error('Division by zero!')
74             return a / b
75         }
76     }
77
78     /**
79      * Створює новий вузол оператора.
80      * @param operator Оператор ('+', '-', '*', '/').
81      * @param left Лівий дочірній вузол.
82      * @param right Правий дочірній вузол.
83      */
84     constructor(operator: string, left: ExpressionNode, right: ExpressionNode) {
85         this.operator = operator
86         this.left = left
87         this.right = right
88     }
89

```

```

90 /**
91  * Оцінює значення вузла оператора шляхом виконання операції.
92  * @returns Числове значення результату операції.
93  */
94 evaluate(): number {
95     const leftValue = this.left.evaluate()
96     const rightValue = this.right.evaluate()
97
98     console.log(`Метод evaluate викликаний для OperatorNode зі значенням ${this.operator}`)
99
100    return OperatorNode.operators[this.operator](leftValue, rightValue)
101 }
102
103 /**
104  * Повертає рядкове представлення вузла оператора.
105  * @returns Рядок, що представляє вираз з оператором.
106  */
107 toString(): string {
108     return `(${this.left.toString()} ${this.operator} ${this.right.toString()})`
109 }
110 }
111
112 const node1 = new NumberNode(3)
113 const node2 = new NumberNode(4)
114 const addition = new OperatorNode('+', node1, node2)
115
116 console.log(`Вираз: ${addition.toString()}`)
117 console.log(`Результат: ${addition.evaluate()}`)
118

```

Результат:

```
> yarn run start
```

Вираз: (3 + 4)

Метод evaluate викликаний для NumberNode зі значенням 3

Метод evaluate викликаний для NumberNode зі значенням 4

Метод evaluate викликаний для OperatorNode зі значенням +

Результат: 7

Документація (за допомогою TypeDoc):

lab3 / NumberNode /

Class NumberNode

Клас для представлення числового вузла в дереві виразу.

Implements

- [ExpressionNode](#)

Defined in index.ts:28

▼ Methods

evaluate

`evaluate(): number`

Оцінює значення числового вузла.

Returns *number*

Числове значення вузла.

Implementation of [ExpressionNode.evaluate](#)

Defined in index.ts:43

toString

`toString(): string`

Повертає рядкове представлення числового вузла.

Returns *string*

Рядок з числовим значенням.

Implementation of [ExpressionNode.toString](#)

Висновок:

Для виконання лабораторної роботи було прочитано та засвоєно необхідний теоретичний матеріал. Виконано 7 варіант, створений проект зі інтерфейсом та необхідними класами, для реалізації проекту було використано шаблон “Interpreter”. Також було створено коментарі у кожному файлі для TypeDoc. Мета лабораторної роботи виконана.