

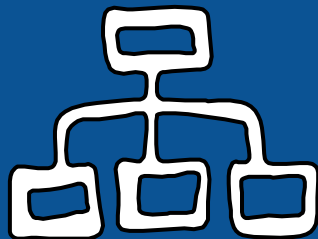
Алгоритмы и Алгоритмические Языки

Семинар #24:

1. Синтаксис и семантика языка арифметических выражений.
2. Иерархия классов для арифметического выражения.
3. Передача аргументов командной строки в программу.
4. Обсуждение ДЗ №2.

18.02.2025

Синтаксис и семантика языка арифметических выражений



Какие выражения будем моделировать?

Выражение – это:

- **Число с плавающей точкой.**
- **Именованная переменная.**
- **Сумма двух других выражений: $E_1 + E_2$.**
- **Умножение двух других выражений: $E_1 * E_2$.**

Примеры выражений:

- 1, 123, 123456.002.
- x, y, length1, sum_05.
- $x + 0.5$.
- $x * (2 + y)$.
- $((a + b) + (c + d)) + (e + f) + g$.

Как более формально задать язык выражений?

Синтаксис языка

Синтаксис языка – это набор слов языка и правила построения предложений из слов этого языка.

Набор слов языка арифметических выражений:

- `Value(val)` - это число с плавающей точкой.
- `Variable(name)` - это строковый идентификатор.

Правила построения предложений языка арифметических выражений:

- `OperationAdd(E_1 , E_2)` – это операция сложения выражений.
- `OperationMul(E_1 , E_2)` – это операция умножения выражений.

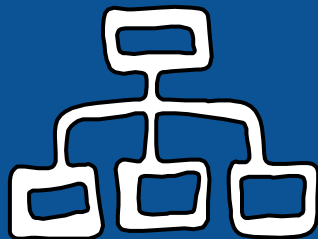
Семантика языка

Семантика языка – функция, задающая для каждого слова и предложения языка его смысл.

Смысл арифметического выражения – результат его вычисления.
Смысл выражения $(x + 2)$ зависит от контекста – значения переменной.
Для нас контекст – это словарь значений всех переменных.

```
eval(Value(val), context)      = val
eval(Variable(name), context) = context[name]
eval(OperationAdd(E1, E2), context) = eval(E1, context) + eval(E2, context)
eval(OperationMul(E1, E2), context) = eval(E1, context) * eval(E2, context)
```

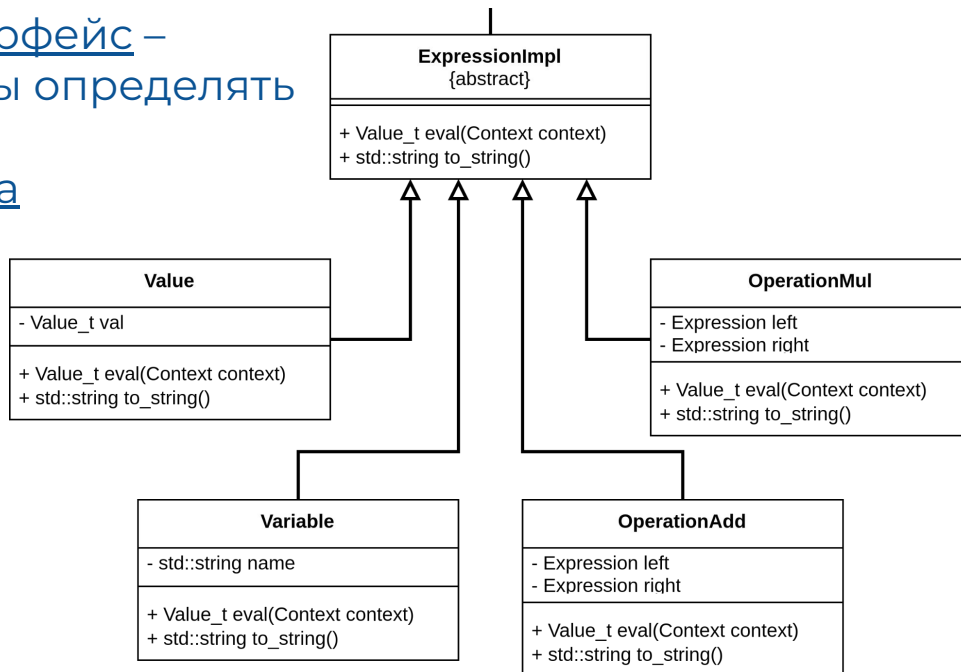
Иерархия классов для арифметического выражения



Паттерн «Абстрактный класс»

Паттерн «Абстрактный класс»:

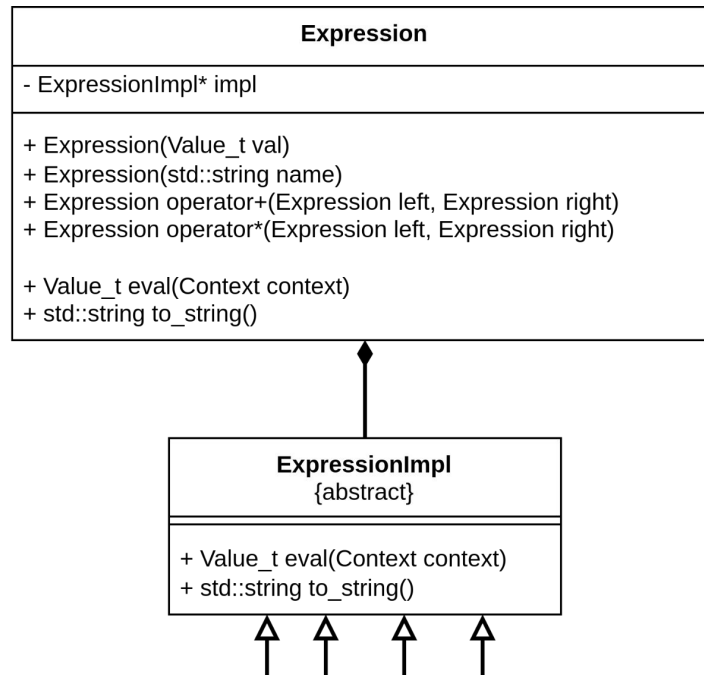
- Абстрактный класс задаёт интерфейс – набор методов, которые должны определять все реализации.
- Каждая реализация интерфейса наследуется от абстрактного класса и определяет свои реализации функций интерфейса.



Паттерн «Указатель на реализацию»

Паттерн «Соккрытие реализации»:

- Пользователю не важно, как устроена реализация.
- Обновление реализации не требует пересборки кода.



Передача аргументов командной строки в программу



Обсуждение ДЗ №2



Вопросы?



Красивые иконки взяты с сайта handdrawngoods.com