

А. О. Вольштейн, Г. Л. Карасёва
(ГГУ им. Ф. Скорины, Гомель)

РАЗРАБОТКА ИНТЕРПРЕТАТОРА ВЫСОКОУРОВНЕГО ЯЗЫКА ОБЩЕГО НАЗНАЧЕНИЯ

Стояла задача в создании консольного приложения интерпретатора арифметических выражений с поддержкой переменных, чисел с плавающей запятой, литералов валюты (таких как 1\$, 2Br и т.д.) и операций вида «умножение числа на валюту» и «сложение различных валют». Умножение валют между собой и сложение числа с валютой должно выдавать ошибку. Также приложение должно содержать лек-

сер (токенизатор), парсер (построитель абстрактного синтаксического дерева из токенов) и виртуальную машину.

Приложение должно поддерживать два режима работы: интерпретация и вычисление результата выражения с помощью обхода абстрактного синтаксического дерева; компиляция из абстрактного синтаксического дерева байт-кода для внутренней виртуальной машины и вычисление результата выражения как результата выполнения байт-кода на виртуальной машине приложения.

Разработано консольное приложение интерпретатор. Разработка велась на языке программирования Go в Visual Studio Code с использованием плагина Golang. Первым этапом разработки интерпретатора арифметических выражений с валютными литералами стала разработка простого интерпретатора арифметических выражений.

Следующим шагом в разработке было добавление в лексификатор поддержки валютных литералов, определение типа «валюта» с модульными тестами для типа и последующая доработка парсера для поддержки типа валюта и конвертации между валютами по следующему правилу: первая валюта, которая указана в выражении, определяет тип результата интерпретируемой строки.

Далее была добавлена поддержка переменных в интерпретатор. Каждая переменная фактически представляет собой запись в символьной таблице, где имени соответствует некоторое значение в памяти. Затем в разработке было добавление виртуальной машины с собственным набором команд, байт-кодом. Компиляция абстрактного синтаксического дерева отличается от его обхода тем, что во время компиляции никаких вычислений не производится: результатом компиляции является байт-код.