

Компилятор упрощенной версии JavaScript под самописную виртуальную машину

Поддерживаемые операции:

- Объявление переменных с помощью *var*;
- Оператор присваивания: *=*
- Операторы сравнения: *<, >, <=, >=, ==, !=*
- Арифметические операторы: *+, -, *, /, %, **, ++, --*
- Логические операторы: *&&, ||, !*
- Условные операторы: *if ... else, while, do ... while*
- Цикл *for*, в том числе особые варианты реализации в JavaScript, например, *for (var i = 0, j = 8; (i <= 5) && (j > 2); i ++, j--)*
- Объявление функций (в том числе с параметрами);
- Возврат значения (*return*);
- Вызов функций.

Структура AST:

Parser – класс, который используется для парсинга кода.

TreeNode – базовый класс для всех остальных классов. Используется для вывода дерева в консоль.

EvalNode – класс, от которого наследуются классы, чьи значения можно привести к конкретному цифровому или буквенному значению.

ValueNode – класс, от которого наследуются классы *LiteralNode* и *IdentNode*, то есть те, которые могут содержать какие-либо значения.

ExprNode – класс, который является родительским для классов, описывающих выражения.

LiteralNode – класс, содержащий в себе какое-либо значение: число или строку.

IdentNode – класс, описывающий какой-либо идентификатор, то есть название переменной или функции.

BinExprNode – класс, описывающий бинарное выражение, то есть выражение, имеющее левую и правую часть, и оператор.

UnaryExprNode – класс, описывающий унарное выражение, то есть выражение, имеющее левую часть и оператор.

DeclaratorNode – класс, описывающий объявление переменной.

VarDeclarationNode – класс, описывающий объявления переменных. Содержит переменную *declarations*, в которой хранится множество с экземплярами класса *DeclaratorNode*.

BlockStatementNode – класс, содержащий в себе переменную *nodes*, описывающую множество всех узлов в этом блоке. В нашей реализации вся

программа является блоком. Функции, if, for, while и do while также содержат в себе блоки.

ArgsNode – класс, описывающий множество аргументов функции.

FuncDeclarationNode – класс, описывающий объявление функции. Содержит переменные *ident*, *params*, *block*. *ident* – название функции, *params* – аргументы функции, *block* – тело функции.

IfNode – класс, описывающий условный оператор if. Содержит переменные *test*, *consequent*, *alternate*. *test* – выражение, определяющее дальнейшее поведение, *consequent* – выполняется, если *test* принял истинное значение, *alternate* – выполняется, если *test* принял отрицательное значение.

ForNode – класс, описывающий цикл for. Содержит переменные *init*, *test*, *update*, *block*. *init* – объявление переменной-счётчика, *test* – выражение, определяющее дальнейшее поведение, т.е. будет ли исполняться код в *block*, *update* – обновление значений переменной-счётчика. *block* – выполняется, если *test* принял истинное значение.

WhileNode – класс, описывающий цикл while. Содержит переменные *test*, *block*. *test* – выражение, определяющее дальнейшее поведение, т.е. будет ли исполняться код в *block*, *block* – выполняется, если *test* принял истинное значение.

DoWhileNode – класс, описывающий цикл do while. Содержит переменные *test*, *block*. *test* – выражение, определяющее дальнейшее поведение, т.е. будет ли исполняться код в *block*, *block* – выполняется, если *test* принял истинное значение. Отличие от while в том, что код в *block* выполнится как минимум однократно.

CallNode – класс, описывающий вызов функции. *ident* – название функции, *args* – аргументы.

ReturnNode – класс, описывающий оператор return. Переменная *argument* – возвращаемое значение.