

Язык программирования Тривиль

Алексей Недоря

12.02.2023

Содержание

1	Назначение	3
2	Обзор языка	4
3	Лексика	5
4	Идентификаторы	5

1. Назначение

Язык программирования Тривиль разработан в рамках работы на семейством языков программирования "Языки выходного дня"(ЯВД) [link](#) проекта Интенсивное программирования [link](#).

Тривиль является нулевым языком семейства ЯВД, предназначенным для реализации компиляторов и экосистемы других языков семейства. В рамках классификации языков, принятом в проекте Интенсивное программирования, это язык L2.

Основными требованиями к языку при разработке были поставлены

- Язык должен быть минимально достаточным для удобной разработки компиляторов. Требование это во многом субъективно, так как компиляторы можно писать существенно по разному.
- Язык должен быть русскоязычным и с синтаксисом минимизирующим переключение на латиницу в процессе разработки программ.

Название языка происходит от слова "тривиальный что означает, что при разработке языка практически везде использовались решения, проверенные в других современных языках программирования, в первую очередь "донорами" являются Go, Swift, Kotlin и Oberon.

Несмотря на узкую направленность на разработку компиляторов, Тривиль является языком программирования общего назначения, пригодным для решения широкого круга задач.

Язык (и экосистема) обладает существенными предпосылками для использование его в качестве учебного языка для обучения студентов разработке компиляторов, библиотек, средств разработки, алгоритмов оптимизации и так далее, в первую очередь это:

- Простота языка
- Современный вид и набор конструкций языка
- Простота компилятора
- Открытая лицензия.

2. Обзор языка

Тривиль - это модульный язык с явным экспортом и импортом, автоматическим управлением памятью (сборка мусора), с поддержкой ООП.

Программа на языке Тривиль состоит из модулей (единиц компиляции), исходный текст каждого модуля расположен в одном или нескольких исходных файлах.

Пример программы:

```
1  модуль x
2
3  импорт "стд/вывод"
4
5  вход {
6      вывод.ф("Привет! \n")
7  }
```

Для описания языка используется EBNF в формате ANTLR4 [link](#).

```
Программа
    : Модуль+
    ;
```

3. Лексика

Исходный текст есть последовательность лексем: идентификаторов и ключевых слов, литералов, знаков операций, разделителей и комментариев. Каждая лексема состоит из последовательности Unicode символов (unicode code point) в кодировке UTF-8.

4. Идентификаторы

Идентификатор - это последовательность *слов*, разделенных пробелами или символами дефис '-' с опционально завершающим знаком препинания:

Каждое слово состоит из *Букв* и цифр, и начинается с Буквы. Буквой считается любой Unicode символ, имеющий признак 'Letter', и дополнительно символы '№' и '_'.

```
Идентификатор
    : Слово ((' ' | '-') Слово)* Знак-препинания?
    ;
Слово
    : Буква (Буква | Цифра)*
    ;
Буква
    : Unicode-letter
    | '_'
    | '№'
    ;
Знак-препинания
    : '?'
    | '!'
    ;
Цифра
    : '0' .. '9'
    ;
```

Примеры идентификаторов:

- | | |
|---|------------------|
| 1 | буква |
| 2 | буква-или-цифра |
| 3 | №-символа |
| 4 | Цифра? |
| 5 | Пора паниковать! |