УДК 004.43

Разработка языка программирования

С.Д. Мищенко, научный руководитель П.И. Ахтямов

Московский физико-технический институт

(национальный исследовательский университет)

Разработана и частично осуществлена реализация концепции языка программирования, позволяющего эффективно разрабатывать программное обеспечение средней сложности.

Целью работы является разработка языка программирования, максимально удовлетворяющего требованиям при создании програмного обеспечения средней сложности.

Для этого предпологалось спроектировать язык:

- с простым для понимания, консистентным синтаксисом
- с сильной статической типизацией значений
- компилируемый
- с возможностью использования высокоуровневых абстракций
- простой для разбора с помощью грамматик
- работающий напрямую в системе, без виртуальной машины

В ходе работы был проведён анализ различных языков программирования, в результате которого было решено создать в основном императивный язык программирования, использующий синтаксис, основанный на синтаксисе языков семейства ML (Standard ML, OCaml) и, в меньшей степени, языков семейства C.

Дизайн языка основывается на решениях C++, Haskell, Rust, OCaml. Так, в языке возможно использование алгебраических типов и классов типов [1].

На данный момент в большей части стабилизирован синтаксис языка и частично реализован интерпретатор:

- реализована трансформация кода в IR
- реализован поиск типов
- реализовано связывание типов
- частично реализована проверка типов
- частично реализовано исполнение

Примеры кода

1. Объявление типа-суммы

```
struct Optional 'A =
  | Some & 'A
  | None
```

2. Объявление типа-произведения

```
struct (Complex : #Value) =
  & Float
  & Float
```

3. Объявление и определение функции поиска подстроки в строке

```
decl find_substring : String -> String -> (Array Index)
def find_substring : str substr = {
  var result = (Array Index).default:

  const str_hashes = find_prefix_hashes Hash: str
  const substr_hash = Hash.of substr

  for i in 0..(str_hashes.size: - substr.size:) do {
    const part_hash = Hash.diff: str_hashes`(i + substr->size:) str_hashes`i
    if part_hash == substr_hash then {
      ; result.push: i
    }
  }
  return result
}
```

Литература

1. *Cordelia V. Hall, Kevin Hammond, Simon Loftus Peyton, Jones Philip Wadler* Type Classes in Haskell // Programming languages and systems - ESOP ’94. 5th European symposium on Programming, Edinburgh, GB, April 11–13, 1994. Proceedings