http://0861.ru

Парадигмы программирования

Лекция 6
Функциональное программирование

ст. препод. каф. ПОВТиАС
Голубничий Артем Александрович
artem@golubnichij.ru

Абакан, 2019

Структура занятия

- основные понятия;
- модель вычислений;
- функции высших порядков;
- чистота функций;
- рекурсия;
- подход к вычислению аргументов;
- языки функционального программирования.

Основные понятия

Функциональное программирование – парадигма программирования, в которой процесс вычисления трактуется как вычисление значений функций в математическом понимании последних (в отличие от функций как подпрограмм в процедурном программировании).

- предполагает обходиться вычислением результатов функций от исходных данных (аргументы) и результатов других функций
- не предполагает явного хранения состояния программы.
- не предполагает изменяемость состояния.

Сравнение моделей вычислений в ЯВУ

Критерий	Императивные	Функциональные
Программа	Последовательность инструкций	Выражение
Вычисление	Преобразование памяти	Редуцирование
Результат	Сохраняется в ячейку. Именованная ячейка памяти - переменная	Отредуцированное выражение

Модель вычисления в функциональных ЯВУ

```
(5 + 4 * 3) ^ 2
~> (5 + 12) ^ 2
~> 17 ^ 2
~> 289
```

Рисунок 1 – Пример реализация

Функции высших порядков

Функции высших порядков – функции, которые могут принимать в качестве аргументов и возвращать другие функции.

- функции высших порядков позволяют использовать карринг.

Карринг – преобразование функции от пары аргументов в функцию, берущую свои аргументы по одному.

Хаскелл Брукс Карри

Чистые функции

Чистые функции (ЧФ) – функции, которые не имеют побочных эффектов ввода-вывода и памяти (они зависят только от своих параметров и возвращают только свой результат).

ЧФ обладают свойствами:

- если результат ЧФ не используется, ее вызов может быть удален без вреда для других выражений;
- результат вызова ЧФ может быть мемоизирован, (сохранен в таблице значений вместе с аргументами вызова) и при повторном вызове взят без вычислений;
- если нет зависимости по данным между двумя ЧФ, то порядок их вычисления можно поменять или распараллелить
- если ЯВУ не допускает побочных эффектов, то можно использовать любую политику вычисления.

Рекурсия

Рекурсия – определение, описание, объекта или процесса внутри самого этого объекта или процесса, то есть ситуация, когда объект является частью самого себя.

- в функциональных языках вместо цикла используется рекурсия;
- рекурсивные функции вызывают сами себя, позволяя операции выполняться снова и снова;
- рекурсивные функции можно обобщить с помощью функций высших порядков, используя катаморфизм и анаморфизм.

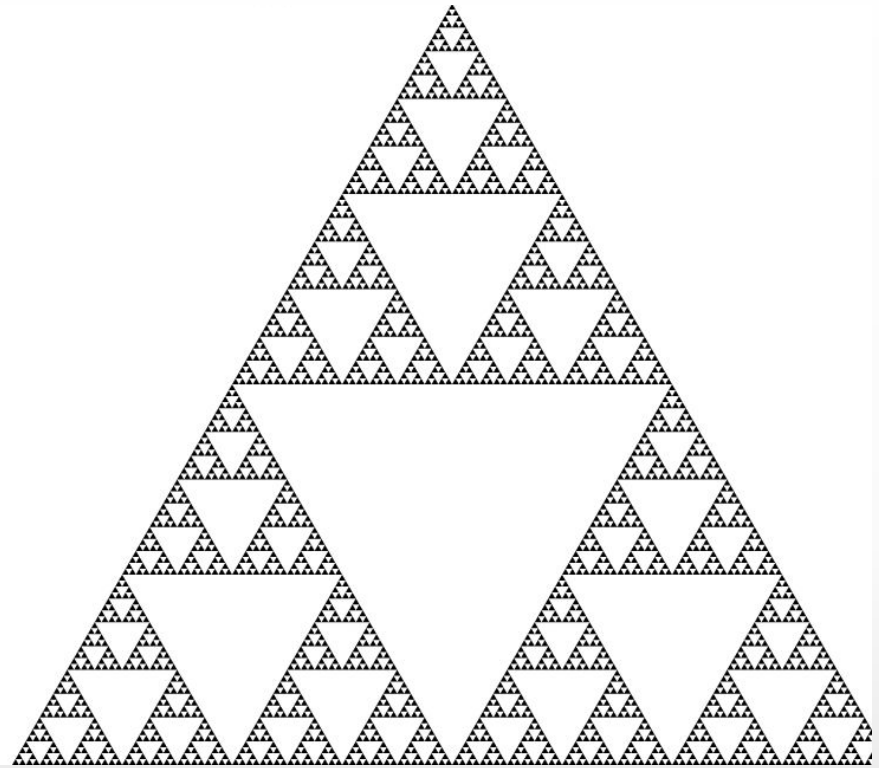

Рисунок 2 – Треугольник Серпинского

Требования к рекурсии

1. Функции в правой части должны применяться на значение отличное от исходного.
2. Рекурсивные вызовы должны прерываться (терминирующее условие)

- вычисление функции осуществляется заменой (подстановкой) формального параметра на фактической.

Расчет факториала рекурсивно (Haskell)

```
factorial n = if n == 0 then 1 else n * factorial (n - 1)
{-
factorial 2
  ~> if 2 == 0 then 1 else 2 * factorial 1
  ~> 2 * factorial 1
  ~> 2 * (if 1 == 0 then 1 else 1 * factorial 0)
  ~> 2 * 1 * factorial 0
  ~> 2 * factorial 0
  ~> 2 * (if 0 == 0 then 1 else 0 * factorial (-1))
  ~> 2 * 1
  ~> 2
}
```

Подход к вычислению аргументов

Строгое (аппликативное) вычисление предполагает расчет значений аргументов перед вычислением функции.
Нестрогое вычисление предполагает вычисление аргументов только в том случае если их значение понадобится.

```
print(len([2+1, 3*2, 1/0, 5-4]))
```

- при строгом вычислении – ошибка;
- при нестрогом вычислении – 4.

Языки функционального программирования

Лисп – (Джон Маккарти, 1958);
Erlang – (Joe Armstrong, 1986) функциональный язык с поддержкой процессов;
APL – предшественник современных научных вычислительных сред, таких как MATLAB;
F# – функциональный язык семейства ML для платформы .NET;
Scala;
Miranda – (Дэвид Тёрнер, 1985);
Nemerle – гибридный функционально/императивный язык;
Haskell – чистый функциональный. Назван в честь Хаскелла Карри.