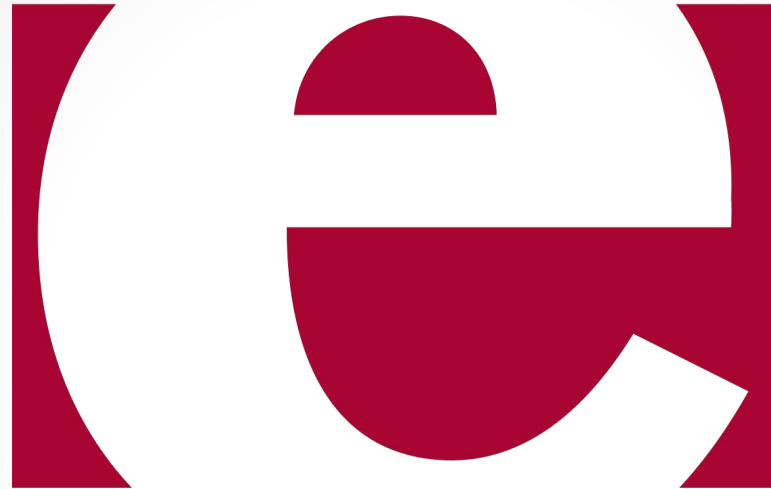


Erlang Academy



ERLANG

Вводная лекция

Автор курса: Костюшкин Сергей

План

- Виды вычислений
- Калькулятор против компьютера
- Кодирование
- История развития языков программирования
- Парадигмы программирования
- Об Erlang
- Предпосылки появления
- Синтаксические достоинства
- Достоинства виртуальной машины
- Краткий экскурс в историю языка
- Сравнение Erlang и C++ по производительности

Чем компьютер отличается
от калькулятора?

Calculator - Вычислитель

Computer - Вычислитель

Calculator - Арифметический вычислитель

Computer - Логический вычислитель

Два вида логики

Дедуктивная логика	Индуктивная логика
Абсолютная истина	Истина с высокой вероятностью
Абсолютная ложь	Ложь с высокой вероятностью

Примеры дедуктивной логики

Утверждения

Все А являются В

Все В являются С

Вывод

Все А являются С

Примеры дедуктивной логики

Утверждения

Аристотель - человек

Все люди смертны

Вывод

Аристотель смертный

Кодирование информации

Обычно кодирование информации осуществляется для преобразования из формы удобной для использования, в форму, удобную для передачи, хранения или обработки.

Средством кодирования служит таблица соответствия знаковых систем, которая устанавливает взаимно однозначное соответствие между знаками или группами знаков двух различных знаковых систем.

Кодирование данных

Двоичный код	Число	Текст (UTF-8)
00001010	10	Перевод каретки (\n)
00100000	32	Пробел
01000001	65	А (латиница)
01011010	90	Z (латиница)
01100001	97	а (латиница)
01111010	122	z (латиница)
00000100 00000111	1031	ї (кирилица)
00000100 00101111	1071	Я (кирилица)
00000100 01001111	1103	я (кирилица)
00000100 01010111	1111	й (кирилица)

Кодирование машинных команд

Машинный код	Асемблер
10110100 00000100	mov ah, 4
10110000 00000010	mov al, 2
00000000 11000100	add ah, al

Уровни абстрактности языков

Уровень	Решаемая проблема	Язык
7	Механизмы обмена данными между процессами, узлами виртуальной машины, приложениями на других языках Проблемы связанные с отказоустойчивостью	Erlang
6	Механизмы обмена данными между процессами	Go
5	Повышение скорости создания прототипов	Perl / PHP / Ruby / Python / JavaScript
4	Автоматизация работы с памятью	Java / C# / Rust
3	Обеспечить переносимость между разными архитектурами процессоров	C / C++
2	Чтение кода в понятном человеку виде	Ассемблер
1	Автоматическое выполнения операций работы с памятью	Машинный код

Парадигмы программирования

Удобно машине	Удобно человеку
<ul style="list-style-type: none">● Императивное программирование● Процедурное программирование	<ul style="list-style-type: none">● Декларативное программирование● Функциональное программирование

Императивная парадигма

- Императивное программирование — это парадигма, в которой решение задачи описывается через последовательность команд меняющих состояние.
- Императивность подразумевает изменяемость переменных.
- Разработчик во время решения задачи вынуждено сдвигает свое внимание с результата на процесс его получения, т.е. решает задачу от частных к общему.
- Данная парадигма ставит эффективность решения выше его корректности.

Пример императивного кода

```
mov ah, 4  
mov al, 2  
add ah, al
```

Процедурная парадигма

Процедурное программирование — это парадигма, в которой программирование происходит на императивном языке, и последовательно выполняемые операторы собираются в подпрограммы, то есть более крупные целостные единицы кода.

Описание процедуры включает в себя информацию о имени процедуры, а информация о вводных данных и результате в явном виде не задается.

Пример процедурного кода

```
void set_num (int *x)
{
    *x = 17;
}
int main ()
{
    int z = 5;
    set_num (&z); /* теперь z = 17 */
}
```

Декларативная парадигма

- Декларативное программирование — это парадигма, в которой задается спецификация решения задачи, то есть описывается, что представляет собой проблема, и ожидаемый результат.
- Декларативность подразумевает неизменяемость переменных.
- Разработчик во время решения задачи фокусирует свое внимание на результате, а не на процессе его получения, и решает задачу от общего к частностям.
- Данная парадигма ставит корректность решения выше эффективности его получения.

Функциональная парадигма

Функциональное программирование — это парадигма, в которой программирование происходит на декларативном языке, а процесс вычисления происходит через вычисление значений функций.

Описание функции включает в себя информацию о имени функции, вводных данных и результате.

История Erlang

1985 — исследование существующих языков

1987 — начало разработки Erlang

1995 — релиз Erlang

1996 — релиз OTP

1998 — Erlang/OTP выпущены в open source

2002 — Начало работы над ejabberd

2006 — тренд в сторону многоядерных процессоров

2009 — Взлет популярности Erlang благодаря интересу FaceBook к ejabberd

2009 — Выход первой книги об Erlang

Об Erlang

Разработчик: [Ericsson](#)

Среда выполнения: [виртуальная машина BEAM](#)

Парадигмы: [декларативная, функциональная](#)

Типизация: [динамическая, строгая](#)

Стратегия вычислений: [жадная](#)

Модель конкурентности: [модель акторов](#)

Модель обмена данными: [передача сообщений](#)

Синтаксические достоинства

- Компактность кода
- Сопоставление по образцу
- Декларативность (иммутабельность)
- Динамическая типизация

Достоинства BEAM

- Легкие процессы
- Встроенные механизмы передачи сообщений
- Собственный планировщик
- Встроенное хранилище
- Множественные способы взаимодействия с кодом написанным на других языках
- Очень экономный расход памяти
- Распределенность
- Продвинутое методы горячего обновления кода
- Сборка мусора индивидуально для каждого процесса

Языки построенные на BEAM

- Erlang
- Elixir (like a Ruby)
- Efene (like a Python)
- LFE (like a Lisp)
- Joxa (like a Lisp)
- Erlog (like a Prolog)
- Alpaca (like a ML)
- Gleam (like an ELM)

Computer Science Laboratory

Бьёрн Дэкер
(Bjarne Däcker)



Джо Армстронг
(Joe Armstrong)



Роберт Вирдинг
(Robert Virding)



Майк Вильямс
(Mike Williams)

Сравнение Erlang и C++ по производительности (2006 год)

Исследование Heriot-Watt University показало, что для телекоммуникационных приложений код на Erlang был на 70-85 % короче, чем на C++, а производительность системы при переписывании кода с C++ на Erlang возрастала почти на 100 %

Источник: Phil Trinder, Professor at Heriot-Watt University

<http://www.macs.hw.ac.uk/~trinder/papers/CPE2006.pdf>

<http://www.dcs.gla.ac.uk/~trinder/papers/Erlang06Abstract.pdf>

50 инженеров WhatsApp обслуживают 900М пользователей

CADE METZ BUSINESS 09.15.15 07:00 AM

WHY WHATSAPP ONLY NEEDS 50 ENGINEERS FOR ITS 900M USERS

EARLIER THIS MONTH, in a post to his Facebook page, WhatsApp CEO Jan Koum announced that his company's instant messaging service is now used by more than 900 million people. And then Facebook CEO Mark Zuckerberg promptly responded with two posts of his own. One said "congrats," and the other included a cheeky photo Zuckerberg had taken of Koum as the WhatsApp CEO keyed his 900-million-user post into a smartphone. "Here's an action shot of you writing this update," Zuckerberg wrote.

Источник: <https://www.wired.com/2015/09/whatsapp-serves-900-million-users-50-engineers/>

Домашнее задание

- [Joe Armstrong - Programming Erlang](#)
- [Francesco Cesarini - Erlang Programming](#)
- [Fred Hebert - Learn You Some Erlang for great good!](#)