

FUNBOX

Квалификационные задания
для **Erlang** разработчиков



О заданиях

Если конкретные ответы на какие-то вопросы дать не получается, то напишите свои соображения по ним.

Результат присылайте нам вместе с рассказом о себе на почтовый ящик wanted@fun-box.ru с пометкой «Вакансия разработчика Erlang». После проверки задания мы обязательно сообщим вам о нашем решении.

В общем, это всё. Перейдя на следующую страницу, вы увидите наши вопросы и задания.



Level I

Здесь несколько теоретических вопросов, с помощью которых вы сможете в свободной форме рассказать нам о своем личном опыте в том или ином аспекте.

Q1

Чем отличается `==` от `:=` в Erlang?

Q2

Зачем нужен OTP?

Q3

Что такое `behaviour`? Зачем он нужен?

Q4

Придумайте пример использования `gen_statem`.

Q5

Что такое `gen_server:cast/2`? Назовите три причины его не использовать.

Q6

Erlang-нода, обрабатывая данные, съела 15GB памяти и вот-вот будет убита системой. Как понять, куда делась память? Как расследовать ситуацию?



Level II

На этом уровне находится практическое задание.

Task Description

Необходимо реализовать приложение, генерирующее равномерно распределенные случайные числа от 2 до N (конфигурируемый параметр) и проверяющее их на простоту.

Приложение должно состоять из генератора и фильтра (способ их реализации — на ваше усмотрение).

При запуске приложения генератор начинает генерировать случайные числа с частотой ровно 3000 чисел в секунду и складывать их в очередь в Redis (в структуру типа `list`). Чем точнее соблюдается скорость 3000 чисел в секунду и чем равномернее по времени они генерируются, тем лучше.

Фильтратор при запуске приложения сразу начинает принимать числа из очереди, проверять их на простоту и добавлять оказавшиеся простыми числа в Redis в структуру типа `set`.

Task Parameters

Параметры приложения (способ их передачи в приложение — на ваше усмотрение):

- N — см. выше;
- `RedisHost`, `RedisPort`, `RedisDB` — параметры подключения к Redis;
- `QueueKey` — имя ключа в Redis, по которому находится очередь, используемая генератором;
- `ResultSetKey` — имя ключа в Redis, по которому находится множество, в которое фильтр складывает результат.

Task Requirements

- Возможность запуска на последней мажорной версии Erlang/OTP;
- Использование OTP;
- Наличие тестов.

Спасибо за время, потраченное на выполнение заданий!