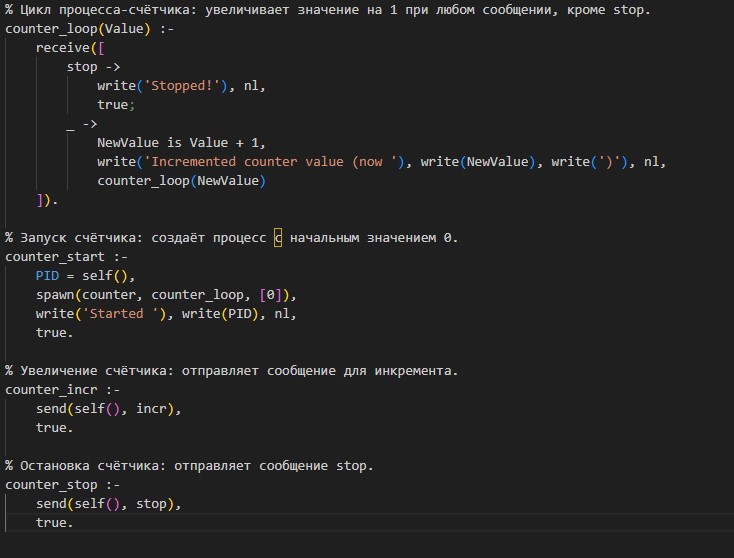
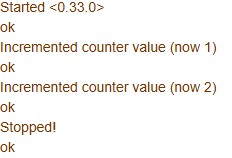
Лабораторная работа 4

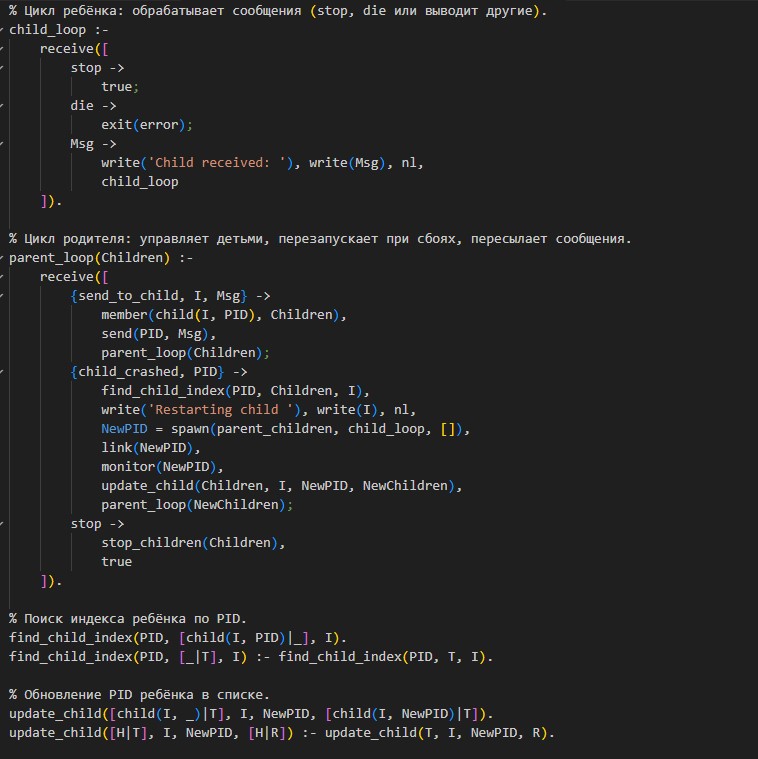
№1

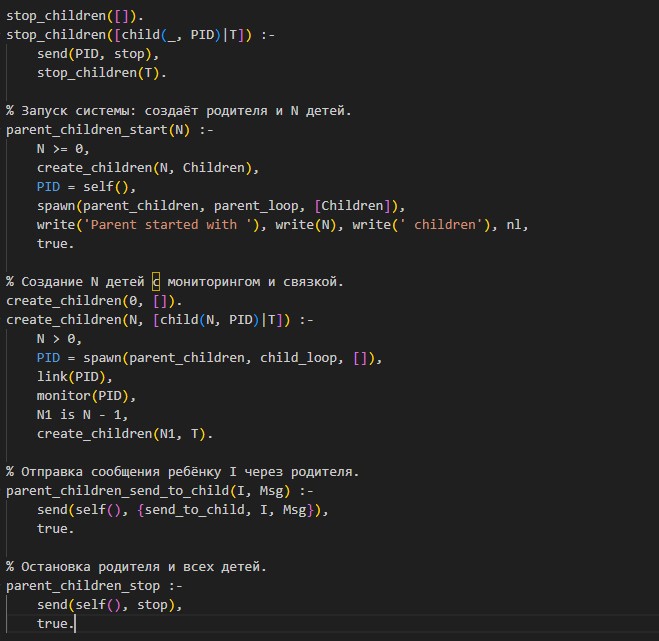
1. Реализуйте процесс-"счётчик", который запускается со значением 0 и 1) если получен атом stop, то он выводит в оболочке текущее значение и заканчивает работу; 2) если получено любое другое сообщение, то значение увеличивается на 1 и выводится сообщение об этом. Для удобства использования модуль должен предоставлять интерфейс counter:start() => ok counter:incr() => ok counter:stop() => ok Пример работы: > counter:start(). Started <0.33.0> ok > counter:incr(). Incremented counter value (now 1) ok > counter:incr(). Incremented counter value (now 2) ok > counter:stop(). Stopped! ok

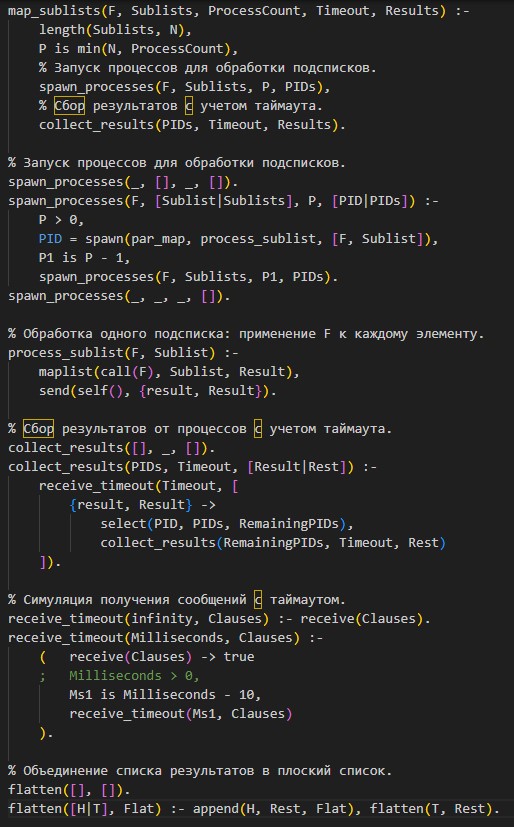




№2 (общая для всех вариантов, делать после задачи 1). Реализуйте модуль parent\_children: \* start(N::integer()) запускает N+1 процесс: "родитель" и N "детей". Каждый из детей ждёт сообщений. Если получено сообщение stop, процесс останавливается без ошибки; если получено сообщение die, процесс падает с ошибкой; любое другое сообщение печатается в оболочке. В случае, если один из детей умрёт с ошибкой, родитель его перезапускает и печатает сообщение об этом. Если родитель умирает, все дети тоже должны умереть. \* send\_to\_child(I::integer(), Msg::any()) посылает родителю сообщение, после которого он пересылает Msg ребёнку номер I. \* stop() останавливает родителя.







№3

3. Реализуйте функцию par\_map(F, List), которая возвращает список с теми же элементами, что lists:map(F, List) (но не обязательно в том же порядке).

