1 Erlang Pytania

1.1

Jakim poleceniem skompilować moduł o nazwie: moj_mod?

A $c(moj\_mod)$.

B $f(moj\_mod)$.

C $i(moj\_mod)$.

D $cd(moj\_mod)$.

1.1.1 Odpowiedź to:

$c(moj\_mod)$.

1.2

Co robi polecenie $f()$. w shellu Erlanga?

A definiuje lambdę

B czyści WSZYSTKIE powiązania (ang. bindngs) w shellu

C czyści powiązanie (ang. bindngs) zmiennej X w shellu

D nic nie robi

1.2.1 Odpowiedź to:

czyści WSZYSTKIE powiązania (ang. bindngs) w shellu

1.2.2 Odpowiedź to:

czyści WSZYSTKIE powiązania (ang. bindngs) w shellu

1.2.3 Odpowiedź tą proszę dać za 2/3 punkta:

nie wiem, ale wiem ze mogę to sprawdzic poleceniem $help()$.

1.3

Co robi polecenie $f(X)$. w shellu Erlanga?

A definiuje lambdę

B czyści WSZYSTKIE powiązania(ang. bindngs)

C czyści powiązanie zmiennej X

D nic nie robi

1.3.1 Odpowiedź to:

czyści powiązanie zmiennej X

1.4

Co sie wypiesz po wpisaniu 1 + + 2 - - 3 + + 4 - - 5. (spacje miedzy plusami i minusami) w shellu Erlanga?

A 0

B error

C -1

D 15

1.4.1 Odpowiedź to:

15

1.5

Co sie wypiesz po wpisaniu 1 ++ 2 – 3 ++ 4 – 5. (bez spacji miedzy plusami i minusami) w shellu Erlanga?

A 0

B error (jakiegos typu błąd)

C -1

D 15

1.5.1 Odpowiedź to:

error (jakiegos typu błąd)

1.6

Co robi polecenie $b()$. w shellu Erlanga?

A wypisuje WSZYSTKIE powiązania (ang. bindngs)

B czyści WSZYSTKIE powiązania (ang. bindngs)

C wypisuje historię

D rekompiluje ostatni moduł

1.6.1 Odpowiedź to:

wypisuje WSZYSTKIE powiązania (ang. bindngs)

1.7

Co robi polecenie $h()$. w shellu Erlanga?

A wypisuje WSZYSTKIE powiązania (ang. bindngs)

B czyści WSZYSTKIE powiązania (ang. bindngs)

C wypisuje historię poleceń w shellu

D rekompiluje ostatni moduł

1.7.1 Odpowiedź to:

wypisuje historię poleceń w shellu

1.8

Co zwróci polecenie $lists : any(fun(X) -> X > 3 \quad end, [1, 2, 3, 4])$.?

A ok

B true

C false

D error

1.8.1 Odpowiedź to:

true

1.9

Co zwróci polecenie $lists : all(fun(X) -> X > 3 \quad end, [1, 2, 3, 4])$.?

A ok

B true

C false

D error

1.9.1 Odpowiedź to:

false

1.10

Co zwróci polecenie $lists : append([[1,2],[3,4],[a,b,c]])$.?

A ok

B $[1, 2, 3, 4, a, b, c]$

C $[[1, 2, 3, 4, [a, b, c]]$

D error

1.10.1 Odpowiedź to:

$[1, 2, 3, 4, a, b, c]$

1.11

Co zwróci polecenie $lists : append([[1,2],[3,4][a,b,c]])$.?

A ok

B $[1, 2, 3, 4, a, b, c]$

C $[[1, 2, 3, 4, [a, b, c]]$

D error

1.11.1 Odpowiedź to:

error

1.11.2 Komnetarz

Brak przecinka między 2 a 3 listą

1.12

Który z poniższych generuje listę $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$?

A $lists : seq(1, 10)$.

B $lists : seq(1, 10)$;

C $lists : ukeymerge(1, 10)$.

D $lists : ukeymerge(1, 10)$;

1.12.1 Odpowiedź to:

$lists : seq(1, 10)$.

1.13

co będzie rezultatem $lists : map(fun(X) -> 2 * X \quad end, [1, 2, 3, 4])$.?

A $[2, 3, 4, 5]$

B $[2, 4, 6, 8]$

C 10

D 20

1.13.1 Odpowiedź to:

$[2, 4, 6, 8]$

1.14

Co robi polecenie $lists : foldl(fun(X, Prod) -> X * -Prod \quad end, 1, [1, 2, 3])$.?

A ok

B false

C 6

D -6

1.14.1 Odpowiedź to:

-6

1.15

Co sie wypisze po wpisaniu

$$a < fun(X) -> Xend.$$

?

A ok

B false

C true

D error

1.15.1 Odpowiedź to:

true `http://erlang.org/doc/reference_manual/expressions.html`

1.16

Czy polecnie:

$$lists : foldl(fun(X, Prod) -> X * -Prod \quad end, 1, [1, 2, 3]).$$

i polecenie

$$lists : foldr(fun(X, Prod) -> X * -Prod \quad end, 1, [1, 2, 3]).$$

zwrócą różne wyniki?

A tak

B nie

C π

1.16.1 Odpowiedź to:

tak

1.17

Jaką operaję możesz wykonać na nieskończonej liście?

A foldl

B foldr

C foldl i foldr

D żadne z powyższych

1.17.1 Odpowiedź to:

foldr

1.17.2 Źródło:

`http://lambda.jstolarek.com/2012/09/why-foldr-works-for-infinite-lists-and-foldl-doesnt/`

1.18

rand:uniform() Returns a random float uniformly distributed in the value range $0.0 =< X < 1.0$ and updates the state in the process dictionary.

Napisz funkcję o nazwie my_fun która zwraca liczbę losową typu float z przedziału 2 do 4.

1.18.1 Przykładowe rozwiązanie:

$$my\_fun() -> (rand : uniform() + 1) * 2.$$

1.19

rand:uniform()

Returns a random float uniformly distributed in the value range 0.0 =< X < 1.0 and updates the state in the process dictionary.

Napisz funkcję o nazwie my_fun lub kod do konsoli który zwraca listę 10 losowych liczb losową typu float z przedziału 0 do 1.

1.19.1 Przykładowe rozwiązanie:

$$my\_fun() -> [rand:uniform()||_< - lists:seq(1,10)].$$

$$[rand:uniform()||_< - lists:seq(1,10)].$$

1.20

Napisz funkcję $mySort$ przyjmująca jako argument listę i sortującą ją.

1.20.1 Przykładowe rozwiązanie (dla chytrych):

$$mySort(List) -> lists:sort(List).$$

1.20.2 Przykładowe rozwiązanie (dla ambitnych):

$$mySort([H|T]) -> mySort([X||X < -T, X < H])$$
$$++ [H] ++$$
$$mySort([X||X < -T, X >= H]);$$
$$mySort([]) -> [].$$

1.20.3 Przykładowe rozwiązanie (dla przekoksów):

z uzyciem drzewa

1.21

Napisz funkcję o naziwe map_add która przyjmuje Mapę, Klucz i Wartość i zwraca mapę z dodana wartoscią i kluczem

1.21.1 Przykładowe rozwiązanie 1:

$$map\_add(Map, Key, Value) -> Map\#Key => Value.$$

1.21.2 Przykładowe rozwiązanie 2:

$$map\_add(Map, Key, Value) -> maps:put(Key, Value, Map).$$

1.22

Niech dany jest $PID =< 1.2.3 >$ oraz wiadomość w postaci atomu o treści *message*. które z poniższych wysyła ww wiadomość do procesu o ww pidzie?

A PID ! message.

B message ! PID.

C message.send(PID).

D PID.send(message).

1.22.1 Odpowiedź to:

PID ! message.

1.23

Niech dany jest $PID =< 1.2.3 >$ oraz wiadomość w postaci atomu o treści *message*. które z poniższych wysyła ww wiadomość do procesu o ww pidzie?

A erlang:send(<1.2.3>,message).

B message ! PID.

C message.send(PID).

D PID.send(message).

1.23.1 Odpowiedź to:

erlang:send(<1.2.3>,message).

1.24

Dany jest moduł moj_mod, który zawiera funkcję $moja\_fun$ która przyjmuje jako argument listę liczb całkowitych. Napisz kod który uruchomi ww funkcję jako NOWY porcesz listą liczb 1,2,3,4,5,6,7 jako argument.

1.24.1 Przykładowe rozwiązanie (za 1/1 pkt):

$$spawn(moj\_mod, moja\_fun, [1, 2, 3, 4, 5, 6, 7]).$$

1.24.2 Przykładowe rozwiązanie (za 2/1 pkt):

$$spawn(moj\_mod, moja\_fun, lists : seq(1, 7)).$$

1.25

Napisz funkcję o nazwie solution która bedzie oczekiwała na wiadomość *message* i zwróci ok gdy ją dostanie.

1.25.1 Przykładowe rozwiązanie:

$$solution() \quad -> receive \quad message \quad -> ok \quad end.$$

1.26

Napisz funkcję o nazwie solution która Otrymuje jako argument listę intów i zwraca tylko nieparzyste z nich:

1.26.1 Przykładowe rozwiązanie:

$$solution(List) \quad -> [X||X < -List, Xdiv2 =:= 1].$$

1.27

Napisz funkcję o nazwie solution która Otrymuje jako argument listę intów i zwraca tylko parzyste z nich:

1.27.1 Przykładowe rozwiązanie:

$$solution(List) \quad -> [X||X < -List, Xdiv2 =:= 0].$$

1.28

posumuj liczby od 1 do 100 w wybranym języku z: ADA(funkcja)/ERLANG(funkcja/shell).

1.28.1 Przykładowe rozwiązanie:

$$lists : sum(lists : seq(1, 100)).$$

1.29

Napisz funkcję o nazwie my_map która implementuje map wielowątkowo.

1.30

Dana jest lista

$$L1 = [17, 2, 3, 4, 5, 4, 32].$$

Napisz kod który zwróci najmniejszy element tej listy.

1.30.1 Rozwiązanie

$$lists : min(L1).$$

1.31

Stwórz mapę $a => 1, b => 2, c => 3$

1.32

Napisz polecenie w konsoli/ funkcję która wypisze liczby od 1 do 10 oddzielone przecinkami (ostatni znak moze być cyfrą (za 1 pkt) lub przecinkiem (za 3/4 pkt))

1.33

ile maxymalnie procesów można uruchomić (domyślnie) w wirtualnej maszynie Erlanga

A nieskonczenie wiele

B 2^{15}

C 2^{16}

D nie zależy to od wirtualnej maszyny Erlanga

1.33.1 Rozwiązanie

2^{15}

1.33.2 Źródło:

`http://erlang.org/doc/efficiency_guide/advanced.html`

1.34

napisz kod eportujący funkcję f/2 danego zdanego modułu

1.34.1 Odpowiedz

$$-export([f/2]).$$

1.34.2 Źródło

`http://erlang.org/doc/reference_manual/modules.html`

1.35

napisz kod importujący funkcję f/2 z modułu moj_mod

$$-import(moj\_mod, [f/2]).$$

1.35.1 Źródło

`http://erlang.org/doc/reference_manual/modules.html`

1.36

Dana są listy

$$L1 = lists : seq(1, 9).$$

$$L2 = lists : seq(1, 5).$$

dopisz kod zwracający listę $[6, 7, 8, 9]$

1.36.1 Odpowiedź przykładowa 1:

$$L1 - -L2.$$

1.36.2 Odpowiedź przykładowa 2:

$$lists : subtract(L1, L2).$$

1.37

napisz funkcje f która przyjmuje 1 argumant i zwraca a jesli argument jest mniejszy lub równy od 10, b jelsi argument jest w przedziale od 10 do 20,w kazdym inny przypadku zwrac c

1.37.1 Odpowiedź przykładowa 1:

f(X) when X =< 10 -> a; f(X) when X < 20 -> b; f(X) -> c.

1.38

Dana są listy

$$L1 = lists : seq(1, 5).$$

$$L2 = [a, b, c, d, e].$$

dopisz kod zwracający listę $[1, a, 2, b, 3, c, 4, d, 5, e]$

1.38.1 Odpowiedź przykładowa 1:

$$lists : zip(L1, L2).$$

1.39

napisz funkcje my_receive która nie przyjmuje argumantu ale odbiera wioadomosc i jesli jest ot atom pom to zwraca pom, jesli otrzyma cololwiek innego zwraca ok , jesli nie otrzyma nic przez 200 milisekund od uruchomienia zwraca timeout

1.39.1 Odpowiedź przykładowa 1:

$$my_receive() -> receivepom -> pom;\_ -> okafter200 -> timeoutend.$$

1.40

jak zakomentowac CAŁĄ 1 linie w Erlangu?

A %

B /*

C //

D /%

1.40.1 Rozwiązanie

%

1.41

czy da się podmienic kod programu w trakcie działania programu napisanego w Erlangu?

A tak

B nie

1.41.1 Rozwiązanie

tak

1.42

napisz kod zwracający 2 elementowa krotkę składajacą sie z atomu a i liczy 5.

1.42.1 Odpowiedź przykładowa 1:

$$\{a, 5\}$$

1.43

podaj przykład zachowania (behaviour) OTP

1.43.1 Odpowiedź przykładowa 1:

gen_server

1.43.2 Odpowiedź przykładowa 2:

gen_event

1.43.3 Odpowiedź przykładowa 3:

supervisor

1.44

Długość listy Napisz program liczący długość listy (len/1).

1.44.1 Odpowiedź przykładowa 1:

len(L) -> length(L).

1.44.2 Odpowiedź przykładowa 2:

len([]) -> 0; len([H|T]) -> 1 + len(T).

1.44.3 Odpowiedź przykładowa 3:

len(L) -> my_len(L,0). my_len([],X) -> X; my_len([H|T],X) -> my_len(T,X+1).

1.45

Min max Napisz program podający najmniejszy element listy (amin/1).

1.46

Napisz program podający największy element listy (amax/1).

1.47

Napisz program zwracający krotkę 2-elementową z najmniejszym i największym elementem listy (tmin_max/1).

1.48

Napisz program zwracający listę 2-elementową z najmniejszym i największym elementem listy (lmin_max/1).

1.49

Lista malejąca Napisz program, który dla danego N zwróci listę formatu [N,N-1,. . . ,2,1]

1.50

Napisz program generujący listę jedynek o zadanej długości.

1.51

Napisz program generujący listę o podanej długości składającą się z podanego elementu.

1.52

Zdefiniuj stałą dla całego modułu TIMOUT na wartosc 200
-define(TIMEOUT, 200).

2 Uwagi końcowe:

2.1 Fragmenty doukumentacji w pytaniach

Fragmenty doukumentacji w pytaniach są integralną częścią pytnia, nie spodziewam się biegłej pamieciowej znajomości dokumentacji a raczej zdolności logicznego myślenia i intuicji.

3 Źródła

1. `http://erlang.org/doc/man/shell.html`
2. `http://erlang.org/doc/man/lists.html`
3. `http://lambda.jstolarek.com/2012/09/why-foldr-works-for-infinite-lists-and-foldl-doesn`