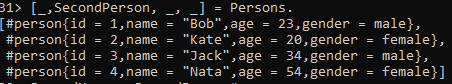
**Задание 1**

1. *[\_, SecondPerson, \_, \_] = Persons.*

Проверьте и объясните присвоенное значение *SecondPerson*.

При выполнении данной команды, в переменную SecondPerson запишется второе значение списка Persons, так как первое значение списка Persons игнорируется при помощи \_.





1. *SecondName = SecondPerson#person.name.*

*SecondAge = SecondPerson#person.age.*

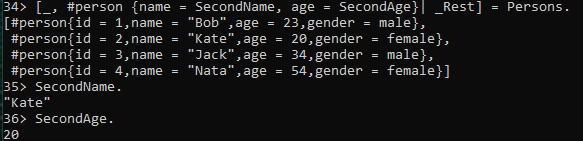


1. *[\_, #person{name = SecondName, age = SecondAge} | \_Rest] = Persons.*

*SecondName.*

*SecondAge.*

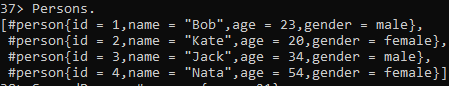
Проверьте присвоенные *на 2 шаге* значения *SecondName*, *SecondAge*  
и объясните почему на 3 шаге сопоставление с образцом (pattern matching) прошло успешно.



Так как присвоенные элементы являются значениями SecondPerson, следовательно значения второго элемента списка Persons. Указанная команда сравнивает вторые элементы списков. И так как они значения атрибутов равны (Отсутствующие данные gender и id не учитываются), pattern matching прошёл успешно.

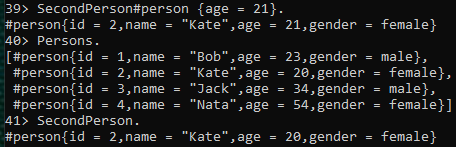
1. *Persons.*

Проверьте, что список Persons не изменился.



1. *SecondPerson#person{age = 21}.*

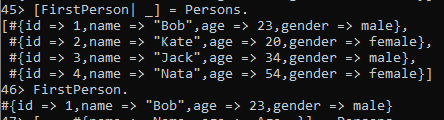
Проверьте, что список Persons и *SecondPerson* не изменились.



**Задание 2**

1. *[FirstPerson | \_] = Persons.*

Проверьте и объясните присвоенное значение *FirstPerson*.



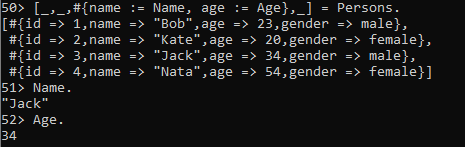
При выполнении данной команды, в переменную FirstPerson запишется голова списка Persons, т.е. оно будет хранить в себе первую запись списка Persons.

1. *[\_, \_, #{name := Name, age := Age},  \_] = Persons.*

Проверьте и объясните присвоенное значение *Name, Age*.

*Name.*

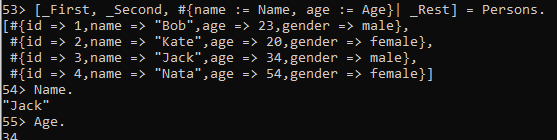
*Age.*



Name = “Jack”, Age = 34 – данные переменные получили значения ключей 3-ей записи, из-за расположения в левой части выражения.

1. *[\_First, \_Second, #{name := Name, age := Age} | \_Rest] = Persons.*

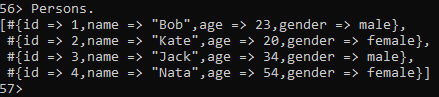
Проверьте и объясните присвоенные значения Name, Age в пункте 3 и 4. Почему команда 4 завершилась успешно (Name, Age уже связаны)?



Name и Age уже связанны с значениями “Jack” и 34 соответственно. Новые значения к переменным не были присвоены, произошёл pattern matching, при котором сравнивание значений ключей произошло успешно.

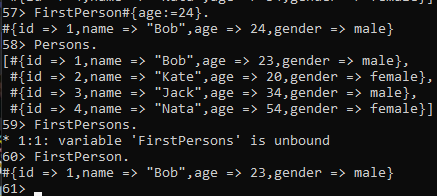
1. *Persons.*

Проверьте, что список Persons не изменился.



1. *FirstPerson#{age := 24}.*

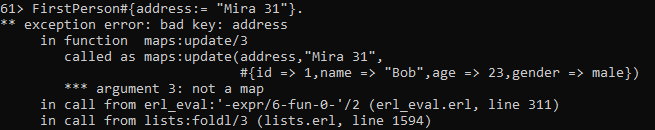
Проверьте, что список Persons и *FirstPerson* не изменились. Почему?



Значение переменных не заменилось, так как это невозможно, если они уже связанны с какими-либо значениями и *:=* обновляет данные для вывода текущего обращения, но не перевязывает значение ключа.

1. *FirstPerson#{address := “Mira 31”}.*

Проверьте и объясните результат*.*



Произошло обряжение к несуществующему ключу map FirstPerson. Из-за этого произошёл exception.

**Задание 3**

Создайте модуль *converter.erl* с функцией to\_rub/1 на вход которой поступает тип валюты и сумма и возвращается результат конвертации в рубли {ok, Result} или {error, badarg}.  
Пример:  
*to\_rub({usd, Amount}) ->*

*io:format(“Convert ~p to rub, amount ~p~n”, [usd, Amount]),*

*{ok, Amount \* 75.5};*

*to\_rub({euro, Amount}) ->*

*io:format(“Convert ~p to rub, amount ~p~n”, [euro, Amount]),*

*{ok, Amount \* 80};*

*…*

5 валют и курс конвертации в рубли:  
Usd - 75.5; Euro - 80; Lari - 29; Peso - 3; Krone - 10  
Для всех остальных валют верните ошибку {error, badarg} и выведите ошибку на экран с помощью функции io:format/2.

Используйте проверку что Amount целое число и больше нуля.

Запустите EShell из папки где у вас находится файл *converter.erl*  
В консоли скомпилируйте файл и вызовите функции:  
*c(“converter.erl”).*

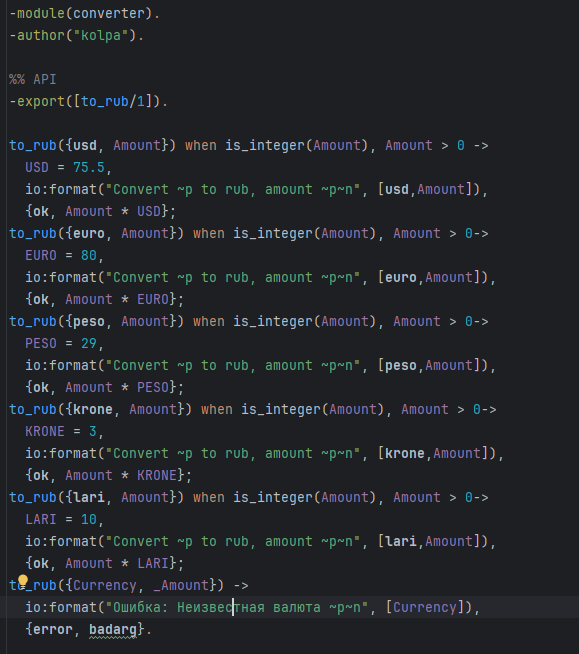
*converter:to\_rub({usd, 100}).*

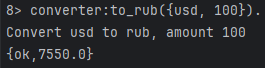
*converter:to\_rub({peso, 12}).*

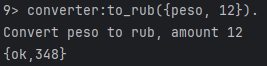
*converter:to\_rub({yene, 30}).*

*converter:to\_rub({euro, -15}).*

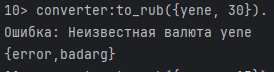
Прокомментируйте строки в которых вызов функции завершился ошибкой*.*







Ошибка объявилась, потому что эта информация не обрабатывается по условию, следовательно выводится ошибка.



Ошибка объявилась, потому что эта указанное количество валюты меньше 0 и не обрабатывается по условию, следовательно выводится ошибка.

