

---

MODULE *TransactionV2*

---

EXTENDS *Integers, Sequences*

CONSTANT *accounts, initialBalances*

VARIABLE *balances, msgs*

*Init*  $\triangleq$  *balances* = *initialBalances*  
 $\wedge$  *msgs* = {}

*TransferMoney*(*from, to, amount*)  $\triangleq$  *balances*[*from*]  $\geq$  *amount*  
Account needs to have enough balance, from property testing  
 $\wedge$  *msgs'* = *msgs*  $\cup$  {[*from*  $\mapsto$  *from, to*  $\mapsto$  *to, amount*  $\mapsto$  *amount*]}  
 $\wedge$  UNCHANGED (*balances*)

*DbUpdate*  $\triangleq$  *msgs*  $\neq$  {}  
 $\wedge$  LET *msg*  $\triangleq$  CHOOSE *msg*  $\in$  *msgs* : TRUE  
IN *msgs'* = *msgs*  $\setminus$  {*msg*}  
 $\wedge$  *balances'* = [[*balances* EXCEPT ![*msg.from*] = *balances*[*msg.from*] - *msg.amount*]  
EXCEPT ![*msg.to*] = *balances*[*msg.to*] + *msg.amount*]

*Next*  $\triangleq$  *DbUpdate*  
 $\vee$   $\wedge \exists$  *from, to*  $\in$  *accounts* :  
*from*  $\neq$  *to*  $\wedge \exists$  *amount*  $\in$  1 .. *balances*[*from*] : Send only positive integers, from property testing  
*TransferMoney*(*from, to, amount*)

#### HELPERS

RECURSIVE *SumBalance*(-, -, -)

*SumBalance*(*accs, bal, total*)  $\triangleq$  IF *accs* = {}  
THEN *total*  
ELSE LET *acc*  $\triangleq$  CHOOSE *acc*  $\in$  *accs* : TRUE  
IN *SumBalance*(*accs*  $\setminus$  {*acc*}, *bal*, *total* + *bal*[*acc*])

#### INVARIANTS

*TypeOK*  $\triangleq$  *msgs*  $\subseteq$  [*from* : *accounts, to* : *accounts, amount* : *Int*]

*BalancesAlwaysPositive*  $\triangleq$   $\forall$  *acc*  $\in$  *accounts* : *balances*[*acc*]  $\geq$  0

*TotalMoneyStable*  $\triangleq$  *SumBalance*(*accounts, initialBalances, 0*) = *SumBalance*(*accounts, balances, 0*)

---

\ \* Modification History  
\ \* Last modified *Fri Aug 13 13:10:53 CEST 2021* by *rchaves*  
\ \* Created *Sun Aug 08 21:06:07 CEST 2021* by *rchaves*