

Chainblock

Description

Your good old friend Pesho (the big time capitalist) has invited you to join his team in the creation of a one of a kind database. You are ready to revolutionize the fintech industry beyond blockchains and payment solutions. You have the honor of taking the first step in the creation process. You need to implement a really fast data structure which will hold Transactions.

Transaction will hold:

- **int Id** – unique transaction id
- **TransactionStatus Status** – enumeration of work positions for the employee
- **string From** – the sender of the transaction
- **string To** – the receiver of the transaction
- **double Amount** – the salary of the employee

You need to support the following operations (and they should be **fast**):

- **Add()** – Add a transaction to the record. You will need to implement the **Contains()** methods as well.
- **Contains(Transaction)** – checks if a given transaction is present in the record. Keep in mind that **transaction Id is the only unique identifier**.
- **Contains(id)** – checks if a transaction with the given **id** exists in the record
- **Count** – returns the number of transactions in the record
- **ChangeTransactionStatus(id, status)** – changes the status of the transaction with the given id or throws **ArgumentException** if no such transaction exists.
- **RemoveTransactionById(id)** – remove the transaction from the record if the id exists, otherwise throws **InvalidOperationException**
- **GetById(id)** – return the transaction with the given **id**. If such transaction doesn't exist, throw **InvalidOperationException**.
- **GetByTransactionStatus(status)** – return the transactions with the given **status ordered by amount descending**. If there are no transactions with the given status, throw **InvalidOperationException**
- **GetAllSendersWithTransactionStatus(status)** – returns all senders which have transactions with the given status ordered by transactions amount (if there are multiple transactions with the same sender, return them all). If no transactions exist, throw **InvalidOperationException**

Example:

"svetlio" has 3 sent transactions -> 2 of them successful (5 leva and 6 leva sent) and 1 aborted transaction.

"mihail" has 1 sent transaction and It is successful (2leva sent).

The result of the call should be "svetlio", "svetlio", "mihail"

- **GetAllReceiversWithTransactionStatus(status)** – returns all receivers which have transactions with the given status in the same way as "**GetAllSendersWithTransactionStatus**". Throw **InvalidOperationException** if no such transactions are present in the record
- **GetAllOrderedByAmountDescendingThenById()** – returns all transactions ordered by amount descending and by id

- **GetBySenderOrderedByAmountDescending(sender)** – search for all transactions with a specific sender and return them ordered by amount descending. If there are no such transactions throw **InvalidOperationException**
- **GetByReceiverOrderedByAmountThenById(receiver)** – returns all transactions with particular receiver ordered by amount descending, then by id ascending. If there are no such transactions throw **InvalidOperationException**
- **GetByTransactionStatusAndMaximumAmount(status, amount)** – returns all transactions with given status and amount less or equal to a maximum allowed amount ordered by amount descending. Returns an empty collection if no such transactions were found
- **GetBySenderAndMinimumAmountDescending(sender, amount)** – returns all transactions with given sender and amount bigger than given amount ordered by amount descending. If there are no such transactions throw **InvalidOperationException**
- **GetByReceiverAndAmountRange(receiver, lo, hi)** – returns all transactions with given receiver and amount between lo (inclusive) and hi (exclusive) ordered by amount descending then by id. If there are no such transactions throw **InvalidOperationException**
- **GetAllInAmountRange(lo, hi)** – returns all transactions within a range by insertion order (the range is inclusive). Returns an empty collection if no such transactions were found.