ДАДАТАК В

Зыходны код класа Recalculation

```
import logging
import threading
import time
from router import Router

class Recalculation(threading.Thread):
    TIMEOUT = 5

    def __init__(self, router, coefficient, threshold):
        super().__init__()
        self.router = router
        self.coefficient = coefficient
        self.threshold = threshold

    def run(self):
        while True:
            try:
                interfaces = self.router.get_interfaces()
                for interface in interfaces:
                    name = interface.get_name()
                    old_txload = interface.get_txload()
                    current_txload = self.router.get_interface_txload(name)

                    new_txload = round(
                        self.coefficient * old_txload +
                        (1 - self.coefficient) * current_txload)
```

Працяг дадатку В

```
if abs(new_txload - old_txload) > self.threshold:
                    interface.set_txload(current_txload)
                    self.router.restart_eigrp()
                    logging.critical('Metric is recalculated for router: {router}'
                        .format(router=self.router.get_host()))
                    logging.critical('Old Load: {old_load}; New Load: {new_load}'
                        .format(old_load=old_txload, new_load=current_txload))
                    break
                else:
                    print('Load change on {interface_name} is less than threshold'
                        .format(interface_name=name))
                    print('New Load: {new_load}; Old Load: {old_load}'
                        .format(new_load=new_txload, old_load=old_txload))
            time.sleep(Recalculation.TIMEOUT)
        except:
            logging.error('Router ({host}) is not found'.format(host=self.router.get_host()))
            break
```