```
std::vector<std::future<void>> tasks;

for(unsigned i = 0; i < 1; ++i)
{
 tasks.push_back(std::async(transfer,0,1,10));
 tasks.push_back(std::async(transfer,1,0,10));
}
```

```
void transfer(int from, int to, int sum)
{
    // deadlock
    auto& acc1 = _accounts[from];
    auto& acc2 = _accounts[to];

    lock_guard lk1(acc1.get_mutex());
    lock_guard lk2(acc2.get_mutex());

    std::cout << "Moving money from = " << from
   << " to = " << to << " sum = " << sum << "\n";

    if (acc1.balance() >= sum)
    {
        acc1.deposit(-sum);
        acc2.deposit(sum);
    }
}
```