# Homework: Smart Contracts Optimizing

## Optimizing

* Optimize this contract for publishing/execution: <https://pastebin.com/zzuqneF0>
* Try to lower the gas cost as much as you can
* Remove any anti-patterns, restructuring the contract
* You can change anything in the contract, as long as it provides the desired functionality:
* To allow the owner to add members. The members have the right to donate money to the contract
* Other than optimizing the contract, create a text file where you explain what were the anti-patterns in the contract that you corrected
* Example:
  + “**The contract initialized structures line-by-line, where they could be initialized on a single line for the same result**”
* The assignment is considered completed when the gas cost of adding a member and of donating to the contract is lower than that of the given contract and a text file is available, explaining (most of) the changes that lead to the optimization

## Fund distributing

* Write a contract, that can receive funds from any address
* The contract has n members (list of addresses by the owner)
* n >= 3
* Each member has an importance (an integer, the higher value the more important is this member’s opinion)
* Only the owner can make a proposal for certain funds to be sent to an address
* Members vote on the proposal
* The owner cannot be a member
* Each member votes with his importance points
* For it to be successful, half of the total importance of the members has to vote for it.
* Use a Library for voting-related actions
* Use a withdraw pattern to allow the proposed address to get the voted funds
* Have a minimal constructor and separate init()
* You can’t avoid loops, but you can make their use minimal :)
* Use all other techniques in the lecture to avoid anti-patterns!