Contract設計

Contract名 : DeviceArea

|  |  |  |  |
| --- | --- | --- | --- |
| 関数名 | 引数 | 返り値 | 修飾詞 |
| areaRegister | String \_areaName | 無し | Public |
| getAreaName | Address \_userId | String | View, internal |

areaRegister : mappingを用いて引数の地域名と送信者のIDと紐付け。

getAreaName : IDに紐づけられた地域名を参照。IDに何も紐づけられていない場合はエラー処理。

Contract名 : Temperatures(DeviceAreaを継承)

|  |  |  |  |
| --- | --- | --- | --- |
| 関数名 | 引数 | 返り値 | 修飾詞 |
| temperaturesRegister | Int16 \_temperature  string \_time  Address \_id | 無し | public |
| getTemperaturesById | Uint \_id | String, string, int16[] | View |
| getTemperaturesByArea | String \_areName | String, string, int16[] | view |
| getTemperaturesByEOA | Address \_userId | String, string, int16[] | view |

Temperatures構造体 :

temperature\_data{string areaName, string time , int16 temperature}

temperaturesRegister : 可変長のTemperatures構造体の配列を作成しておき、追加していく形でデータを保存する。インデックスをIDとし、送信者のアドレスと紐づけておく。

getTemperaturesById : データのID（配列のインデックス）で検索する。その結果を返す。

getTemperaturesByArea : 地域名を配列の中から線形探索で探す。結果は複数あると思われるので、配列として返す。

getTemperaturesByEOA : そのアカウントが登録したデータを線形探索で検索。これも配列で返す。