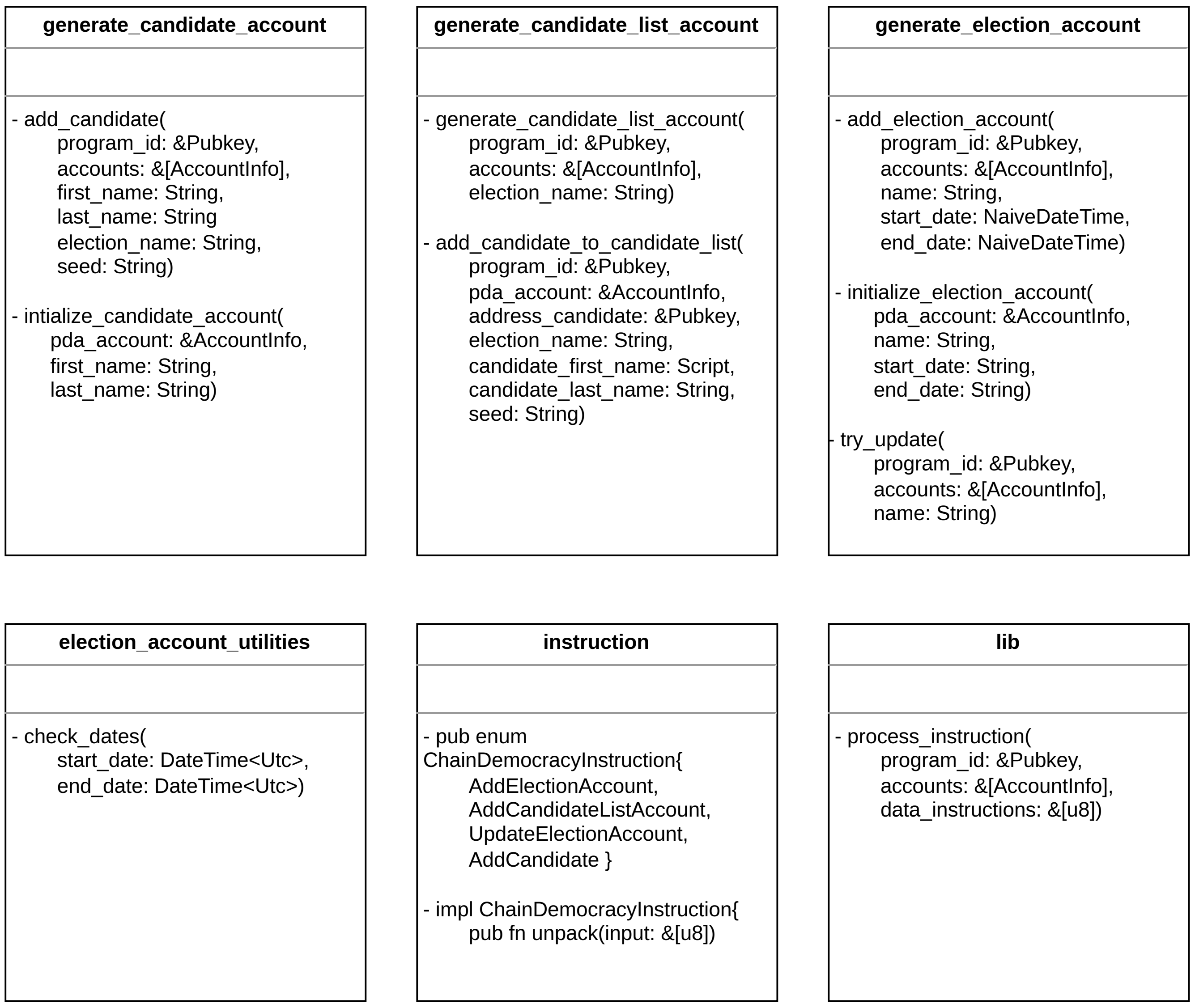

generate_candidate_account

```
- add_candidate(
        program_id: &Pubkey,
        accounts: &[AccountInfo],
        first_name: String,
        last_name: String
        election_name: String,
        seed: String)

- intialize_candidate_account(
       pda_account: &AccountInfo,
       first_name: String,
       last_name: String)
```

generate_candidate_list_account

```
- generate_candidate_list_account(
        program_id: &Pubkey,
        accounts: &[AccountInfo],
        election_name: String)

- add_candidate_to_candidate_list(
        program_id: &Pubkey,
        pda_account: &AccountInfo,
        address_candidate: &Pubkey,
        election_name: String,
        candidate_first_name: Script,
        candidate_last_name: String,
        seed: String)
```

generate_election_account

```
- add_election_account(
        program_id: &Pubkey,
        accounts: &[AccountInfo],
        name: String,
        start_date: NaiveDateTime,
        end_date: NaiveDateTime)

- initialize_election_account(
       pda_account: &AccountInfo,
       name: String,
       start_date: String,
       end_date: String)

- try_update(
       program_id: &Pubkey,
       accounts: &[AccountInfo],
       name: String)
```

election_account_utilities

```
- check_dates(
        start_date: DateTime<Utc>,
        end_date: DateTime<Utc>)
```

instruction

```
- pub enum
ChainDemocracyInstruction{
        AddElectionAccount,
        AddCandidateListAccount,
        UpdateElectionAccount,
        AddCandidate }

- impl ChainDemocracyInstruction{
        pub fn unpack(input: &[u8])
```

lib

```
- process_instruction(
        program_id: &Pubkey,
        accounts: &[AccountInfo],
        data_instructions: &[u8])
```