



Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Лабораторна робота №4-5

Програмування смарт-контрактів

Тема: Смарт-контракт на Near мовою Rust

Виконав

студент групи ІП-11:

Панченко С. В.

Перевірив:

Яланецький В.А.

ЗМІСТ

1 Мета лабораторної роботи.....	6
2 Завдання.....	7
3 Виконання.....	8
ДОДАТОК А ТЕКСТИ ПРОГРАМНОГО КОДУ.....	12

1 МЕТА ЛАБОРАТОРНОЇ РОБОТИ

Ознайомитися з блокчейном Near Protocol та основами мови програмування Rust. Ознайомитися із Near SDKта шалонами доступу.

2 ЗАВДАННЯ

Реалізувати смарт-контракт на Near мовою Rust згідно індивідуального варіанту. Застосувати Near SDK та шаблонів доступу до розробленого смарт-контракту у попередній роботі, протестувати його роботу.

3 ВИКОНАННЯ

Створимо акаунт у мережі testnet у NEAR на рисунку 3.1.

```
near create-account sideshowbobgot.testnet --useFaucet
```

```
(base) sideshowbobgot@localhost:~/university/prometheus_blockchain_test/lab_4$ near create-account sideshowbobgot.testnet --useFaucet --network-id testnet
INFO Your transaction:
| signer id: testnet
| actions:
|   -- create account: sideshowbobgot.testnet
|   -- add access key:
|       public key: ed25519:7kdqF8wxCYMvKD38fpjQivSngeQWYdoqYLJrWbBsPGB
|       permission: FullAccess
|
INFO
| New account <sideshowbobgot.testnet> created successfully.
| The data for the access key is saved in the keychain
|
INFO
| Transaction ID: BPD6pvKZF99sa9g2erhUVvbrNLUEE6ha2A9UfbXh7Qdo
| To see the transaction in the transaction explorer, please open this url in your browser:
| https://explorer.testnet.near.org/transactions/BPD6pvKZF99sa9g2erhUVvbrNLUEE6ha2A9UfbXh7Qdo
|
Here is your console command if you need to script it or re-run:
| near account create-account sponsor-by-faucet-service sideshowbobgot.testnet autogenerate-new-keypair save-to-keychain network-config testnet create
(base) sideshowbobgot@localhost:~/university/prometheus_blockchain_test/lab_4$
```

Рисунок 3.1 — створення тестового акаунта

Збілдимо контракт та задеплоїмо його на рисунках 3.2, 3.3.

```
(base) sideshowbobgot@localhost:~/university/prometheus_blockchain_test/lab_4$ cargo near build
> non-reproducible-wasm - Fast and simple (recommended for use during local development)
• Checking the host environment...
• done
• Collecting cargo project metadata...
• done
• Generating ABI
  Compiling proc-macro2 v1.0.94
  Compiling unicode-ident v1.0.18
  Compiling hashbrown v0.15.2
  Compiling serde_json v1.0.140
  Compiling equivalent v1.0.2
  Compiling ryu v1.0.20
  Compiling itoa v1.0.15
  Compiling serde v1.0.219
  Compiling memchr v2.7.4
```

Рисунок 3.2 — білдинг контракта

```
(base) sideshowbobgot@localhost:~/university/prometheus_blockchain_test/lab_4$ near deploy sideshowbobgot.testnet ./target/near/lab_4.wasm --network-id testnet
INFO Unsigned transaction:
| signer id: sideshowbobgot.testnet
| receiver id: sideshowbobgot.testnet
| actions:
|   -- deploy code <CZmx4y621ZjjsWdNsKFU6RLAdmYLSHoFjAK1BDmCN3MP> to a account <sideshowbobgot.testnet>
|
INFO Your transaction was signed successfully.
| Public key: ed25519:7kdqF8wxCYMvKD38fpjQivSngeQWYdoqYLJrWbBsPGB
| Signature: ed25519:5pxmRG4TjyvoeKZ7fuuMjhsFnCXPUIHdgjDmUk1jhF3MhGBYSL295jHbmomVZ3oBNw9Gvhhin6v61Z5JDXoAMcEy
|
INFO
| Contract code has been successfully deployed.
|
| Gas burned: 9.0 Tgas
| Transaction fee: 0.000898991756939 NEAR
| Transaction ID: E2AreibQdf1FJfdGXsr58WNfakZwa1GZAsWpgJnU1Gc5
| To see the transaction in the transaction explorer, please open this url in your browser:
| https://explorer.testnet.near.org/transactions/E2AreibQdf1FJfdGXsr58WNfakZwa1GZAsWpgJnU1Gc5
|
Here is your console command if you need to script it or re-run:
| near contract deploy sideshowbobgot.testnet use-file ./target/near/lab_4.wasm without-init-call network-config testnet sign-with-keychain send
(base) sideshowbobgot@localhost:~/university/prometheus_blockchain_test/lab_4$
```

Рисунок 3.3 — деплой контракта

Бачимо, що на початку у тестового акаунта 10 NEAR на рисунку .

```
(base) sideshowbobgot@localhost:~/university/prometeus_blockchain_test/lab_4$ near state sideshowbobgot.testnet --network-id testnet
>>>> Receiving an inquiry about your account ...

-----
sideshowbobgot.testnet      At block #190928717
                             (5nZxeARC5aRyFqZsiPrXo4ajGywCfSVkoRmW3XHUVCRg)
-----
NEAR Social profile unavailable The profile can be edited at https://near.social
                             or using the cli command: bos social-db manage-profile
                             (https://github.com/bos-cli-rs/bos-cli-rs)
-----
Native account balance      10.00 NEAR
-----
Validator stake             0 NEAR
-----
Storage used by the account  117.9 KB
-----
Contract (SHA-256 checksum hex) abd5f772857a1c42b5713c3ebb28a1976f72e78b8d43a4768d0a9e867056c75e
-----
Access keys                 1 full access keys and 0 function-call-only access keys
-----

Here is your console command if you need to script it or re-run:
  near account view-account-summary sideshowbobgot.testnet network-config testnet now

(base) sideshowbobgot@localhost:~/university/prometeus_blockchain_test/lab_4$
```

Рисунок 3.4 — початковий стан тестового акаунта

Створимо ще один тестовий акаунт sideshowbobgot2 на рисунку 3.5.

```
(base) sideshowbobgot@localhost:~/university/prometeus_blockchain_test/lab_4$ near create-account sideshowbobgot2.testnet --useFaucet --network-id testnet
INFO Your transaction:
| signer id: testnet
| actions:
|   -- create account: sideshowbobgot2.testnet
|   -- add access key:
|       public key: ed25519:GW86yuDvVYLB6tV9iDtkqaFUjirRu9R62y69NZGYbCmZ
|       permission: FullAccess
|
INFO
| New account <sideshowbobgot2.testnet> created successfully.
| The data for the access key is saved in the keychain
|
INFO
| Transaction ID: HS7Wq5YRCfrNbyatUrvx12kSLQVfu9xFf5q48fzVTZZn
| To see the transaction in the transaction explorer, please open this url in your browser:
| https://explorer.testnet.near.org/transactions/HS7Wq5YRCfrNbyatUrvx12kSLQVfu9xFf5q48fzVTZZn
|
Here is your console command if you need to script it or re-run:
  near account create-account sponsor-by-faucet-service sideshowbobgot2.testnet autogenerate-new-keypair save-to-keychain network-config testnet create

(base) sideshowbobgot@localhost:~/university/prometeus_blockchain_test/lab_4$
```

Рисунок 3.5 — створення другого тестового акаунта

Викличемо метод у першого тестового акаунта з другого тестового. Маємо побачити, що у першого тестового акаунта стало близько 12 NEAR, а у другого близько 8 NEAR. Поточний обмін 1 NearToken — 1 Point. Побачити це можна на рисунках 3.6, 3.7.

```
(base) sideshowbobgot@localhost:~/university/prometheus_blockchain_test/lab_4$ near call sideshowbobgot.testnet award_points '{}' --deposit 2 --use-account sideshowbobgot2.testnet --network-id testnet
INFO Unsigned transaction:
| signer_id: sideshowbobgot2.testnet
| receiver_id: sideshowbobgot.testnet
| actions:
|   -- function call:
|     method name: award_points
|     args: {}
|     gas: 30.0 Tgas
|     deposit: 2 NEAR
INFO Your transaction was signed successfully.
| Public key: ed25519:GW86yduVvYLBgtV9IDtkqaFujirRu9R62y69NZGYbCmZ
| Signature: ed25519:z1KfS7mJux7szxSNuAPQCu5NGMsjnZpyhVoLwJyz1A2fRTHz7voN9YQZAGtD2u7w5Q5nqNwjZFckSfP7vCdL7A
INFO
| The "award_points" call to <sideshowbobgot.testnet> on behalf of <sideshowbobgot2.testnet> succeeded.
|
| Gas burned: 2.6 Tgas
| Transaction fee: 0.0002350995440496 NEAR
| Transaction ID: 98jNB8K8VyA1D8yUVofDsmHXKKAjYmYvSTxbpBe9fjnC
| To see the transaction in the transaction explorer, please open this url in your browser:
| https://explorer.testnet.near.org/transactions/98jNB8K8VyA1D8yUVofDsmHXKKAjYmYvSTxbpBe9fjnC
Here is your console command if you need to script it or re-run:
near contract call-function as-transaction sideshowbobgot.testnet award_points json-args {} prepaid-gas '30.0 Tgas' attached-deposit '2 NEAR' sign-as sideshowbobgot2.testnet network-id testnet
```

Рисунок 3.6 — виклик методу у першого контракта

```
(base) sideshowbobgot@localhost:~/university/prometheus_blockchain_test/lab_4$ near state sideshowbobgot.testnet --network-id testnet
>>>> Receiving an inquiry about your account ...
-----
sideshowbobgot.testnet      At block #190929672
                           (H8DyZSg4bfzhozRm25tyr8tioaFvMGBN4S5RYeFqqj2e)
-----
NEAR Social profile unavailable The profile can be edited at https://near.social
or using the cli command: bos social-db manage-profile
(https://github.com/bos-cli-rs/bos-cli-rs)
-----
Native account balance      12.00 NEAR
-----
Validator stake             0 NEAR
-----
Storage used by the account 118.1 KB
-----
Contract (SHA-256 checksum hex) 42d9b088caa72ec487b578a54a1e0c0d854f67d26d9262c63b850de1d7837448
-----
Access keys                 1 full access keys and 0 function-call-only access keys
-----
Here is your console command if you need to script it or re-run:
near account view-account-summary sideshowbobgot.testnet network-config testnet now
(base) sideshowbobgot@localhost:~/university/prometheus_blockchain_test/lab_4$ near state sideshowbobgot2.testnet --network-id testnet
>>>> Receiving an inquiry about your account ...
-----
sideshowbobgot2.testnet     At block #190929680
                           (2GppXCV7rJbuEsyAaQJ9wJhsVwhetkkyabBEBnZrJvT5)
-----
NEAR Social profile unavailable The profile can be edited at https://near.social
or using the cli command: bos social-db manage-profile
(https://github.com/bos-cli-rs/bos-cli-rs)
-----
Native account balance      8.00 NEAR
-----
Validator stake             0 NEAR
-----
Storage used by the account 182 B
-----
Contract (SHA-256 checksum hex) No contract code
-----
Access keys                 1 full access keys and 0 function-call-only access keys
-----
Here is your console command if you need to script it or re-run:
```

Рисунок 3.7 — баланси акаунтів після виклику метода

Тепер з другого акаута витратимо Points доки вони не закінчаться на рисунках 3.8, 3.9.

```

(base) sideshowbobgot@localhost:~/university/prometheus_blockchain_test/lab_4$ near call sideshowbobgot.testnet redeem_points '{"points_to_spend": 2}' --use-account sideshowbobgot2.testnet --network-id testnet
INFO Unsigned transaction:
| signer_id: sideshowbobgot2.testnet
| receiver_id: sideshowbobgot.testnet
| actions:
|   -- function call:
|     method name: redeem_points
|     args:
|       {
|         "points_to_spend": 2
|       }
|     gas: 30.0 Tgas
|     deposit: 0 NEAR
|
INFO Your transaction was signed successfully.
| Public key: ed25519:GW86yUDvVYLBgtV9iDtkqaFujirRu9R62y69NZGybCnZ
| Signature: ed25519:31MUNBuMf9g9zhUN2qGB0EEeXXKaRon2S2mzrHVKTWQvPVrWQRWm7xLdkUwP51LKR4sBuqXQYcmwaVAebH6YfXXPW
|
INFO
| The "redeem_points" call to <sideshowbobgot.testnet> on behalf of <sideshowbobgot2.testnet> succeeded.
|
| Gas burned: 2.1 Tgas
| Transaction fee: 0.0002003067956476 NEAR
| Transaction ID: CRhaQwaDesgyCTS5J5Aj2JKXQhnLMhFWpnziaRMvRQxS
| To see the transaction in the transaction explorer, please open this url in your browser:
| https://explorer.testnet.near.org/transactions/CRhaQwaDesgyCTS5J5Aj2JKXQhnLMhFWpnziaRMvRQxS
|
Here is your console command if you need to script it or re-run:
near contract call function as-transaction sideshowbobgot.testnet redeem_points json-args '{"points_to_spend": 2}' prepaid-gas '30.0 Tgas' attached-deposit '0 NEAR' sign-as sideshowbobgot2.testnet network-config testnet sign-with-keychain send

```

Рисунок 3.8 — виклик redeem_points

The screenshot shows the transaction details for the `redeem_points` method call on the `sideshowbobgot.testnet` contract. The transaction is successful and occurred at block 190,929,944. The sender and receiver are both `sideshowbobgot.testnet`. The transaction burned 1.69 Tgas and had a fee of 0.000169 NEAR. The action details show the method `redeem_points` being called with the argument `{ "points_to_spend": 2 }`. The value of the transaction is 0 NEAR, and the result is `"2"`. There are no logs for this transaction.

Field	Value
Receipt	G6rvrQ9Er6cbFvtIWpCqYDuaZo3ymX1BGqNOB1HYtLA9
Status	Success
Block	190,929,944
From	sideshowbobgot.testnet
To	sideshowbobgot.testnet
Burnt Gas & Tokens by Receipt	1.69 Tgas 0.000169 NEAR
Actions	Called method <code>redeem_points</code> in contract <code>sideshowbobgot.testnet</code> <pre>{ "points_to_spend": 2 }</pre>
Value	0 NEAR
Result	"2"
Logs	No Logs

Рисунок 3.9 — результат виклику методу у вигляді рядку з кількістю витрачених Point

ДОДАТОК А ТЕКСТИ ПРОГРАМНОГО КОДУ

Тексти програмного коду
(Найменування програми (документа))

Жорсткий диск
(Вид носія даних)

(Обсяг програми (документа), арк.)

Студента групи ІП-11 4 курсу
Панченка С. В

```
// Find all our documentation at https://docs.near.org
```

```
type LoyaltySystemPoint = u32;
```

```
#[near_sdk::near(serializers = [borsh])]
```

```
struct ParticipantInfo {
    points: LoyaltySystemPoint
}
```

```
enum LoyaltySystemStoragePrefix {
    Participants
}
```

```
// Define the contract structure
```

```
#[near_sdk::near(contract_state)]
```

```
pub struct LoyaltySystem {
```

```
    owner_id: near_sdk::AccountId,
    participants: near_sdk::store::LookupMap<near_sdk::AccountId,
ParticipantInfo>
}
```

```
// Define the default, which automatically initializes the
contract
```

```
impl Default for LoyaltySystem {
```

```
    fn default() -> Self {
```

```
        Self {
```

```
            owner_id: near_sdk::env::predecessor_account_id(),
```

```
            participants:
```

```
near_sdk::store::LookupMap::new(LoyaltySystemStoragePrefix::Partic
ipants as u8)
```

```
        }
```

```
    }
```

```
}
```

```
const YOCTO_NEAR_PER_TOKEN: u128 =
```

```
near_sdk::NearToken::from_near(1).as_yoctonear();
```

```

#[near_sdk::near]
impl LoyaltySystem {

    #[payable]
    pub fn award_points(&mut self) -> near_sdk::Promise {
        let deposit =
near_sdk::env::attached_deposit().as_yoctonear();
        near_sdk::require!(deposit >= YOCTO_NEAR_PER_TOKEN);

        let participant_id =
near_sdk::env::predecessor_account_id();
        let tokens_to_award = (deposit / YOCTO_NEAR_PER_TOKEN) as
LoyaltySystemPoint;

        if !self.participants.contains_key(&participant_id) {
            self.participants.set(participant_id.clone(), Some(
                ParticipantInfo {points: tokens_to_award}
            ));
        } else {
            let participant_info =
self.participants.get_mut(&participant_id).unwrap();
            let result_points =
participant_info.points.checked_add(tokens_to_award);
            near_sdk::require!(result_points.is_some());
            participant_info.points = result_points.unwrap();
        }

        let near_tokens_to_return = deposit - YOCTO_NEAR_PER_TOKEN
* (tokens_to_award as u128);

near_sdk::Promise::new(participant_id).transfer(near_sdk::NearToken::from_yoctonear(near_tokens_to_return))
    }

    pub fn redeem_points(&mut self, points_to_spend: u32) ->
String {

```

```
        let participant_id =
near_sdk::env::predecessor_account_id();
        let participant_info =
self.participants.get_mut(&participant_id);
        near_sdk::require!(participant_info.is_some());
        let participant_info = participant_info.unwrap();

        near_sdk::require!(participant_info.points >=
points_to_spend);
        participant_info.points -= points_to_spend;

        points_to_spend.to_string()
    }
}
```
