МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
імені ІГОРЯ СІКОРСЬКОГО
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ ЗАХИСТУ ІНФОРМАЦІЇ

ЗВІТ
з лабораторної роботи № 1

Виконав:
студент ФБ-41мн
Дрик Владислав
Олександрович
«20» Березня 2025

КИЇВ 2024

Мета роботи: «Отримання навичок налаштування платформ виконання смарт-контрактів та криптовалют».

ЧАСТИНА 1

Провести порівняльний аналіз особливостей розгортання систем криптовалют у порівнянні із системою Etherium. Зробити висновок про можливість чи неможливість взаємозаміни модулів різних систем та пояснити причини.

Буду порівнювати Solana з Etherum.

 vs 

https://ethereum.org/en/
https://docs.soliditylang.org/en/v0.8.29/
https://solana.com/
https://docs.solana.com/

Мови програмування для смарт-контрактів

Ethereum використовує мови високого рівня, такі як Solidity та Vyper, причому Solidity є найпоширенішою. Вона доволі проста у вивченні, має хорошу документацію, численні бібліотеки та приклади, тому підходить навіть для новачків. Контракти розгортаються у віртуальній машині Ethereum (EVM).

Solana, натомість, орієнтована на мови Rust, C та C++. Основна мова — Rust, яка є потужною, але складнішою у використанні, особливо для розробників без досвіду роботи з системними мовами. Вона дозволяє отримати більше контролю над пам’яттю та продуктивністю, проте вимагає більш глибоких знань.

Інфраструктура та інструменти для розгортання

Для Ethereum існує багато зручних інструментів: Truffle, Hardhat, Remix IDE, Brownie. Вони мають хорошу документацію, дозволяють зручно тестувати, налагоджувати і розгортати контракти локально або в тестових мережах. Існує велика кількість готових шаблонів та відкритих бібліотек.

У випадку з Solana основним інструментом є Anchor Framework — надбудова над Rust, яка полегшує розробку смарт-контрактів. Також використовується Solana CLI для взаємодії з мережею. Проте загальна інфраструктура ще розвивається, документації менше, а процес розгортання трохи складніший.

Швидкість транзакцій та масштабованість

Ethereum має обмежену швидкість транзакцій — приблизно 15–30 транзакцій на секунду в основній мережі. Через це під час високого навантаження зростає час підтвердження та вартість газу. Для покращення масштабованості активно впроваджуються рішення другого рівня, як-от Arbitrum, Optimism тощо.

Solana може обробляти до 65 000 транзакцій на секунду (теоретично), завдяки інноваційному механізму Proof of History (PoH) у поєднанні з Proof of Stake. Це забезпечує високу пропускну здатність, швидке підтвердження транзакцій і стабільно низькі комісії.

Розгортання та хостинг вузлів

Для запуску повноцінного вузла Ethereum достатньо стандартного VPS або сервера із середніми характеристиками. Це робить мережу доступною для більшої кількості учасників.

Solana ж потребує високопродуктивного обладнання — сучасного багатоядерного процесора, великої кількості оперативної пам’яті (від 128 ГБ) та швидкого SSD. Через це стати валідатором у мережі складніше і дорожче, ніж в Ethereum.

Вартість розгортання

Комісії в Ethereum можуть бути дуже високими, особливо у пікові періоди. Це накладає обмеження на проєкти з великою кількістю транзакцій. Навіть розгортання контракту може коштувати сотні доларів, тому оптимізація коду — критично важлива.

Solana виграє у цьому плані: вартість однієї транзакції — частки цента, а розгортання контрактів — в десятки або сотні разів дешевше, ніж на Ethereum. Це дозволяє тестувати й запускати dApp-и без великих фінансових витрат.

Висновок

Solana підходить для застосунків, де важлива швидкість, масштабованість та низька вартість транзакцій, але вимагає глибших знань у програмуванні (особливо в Rust) та сильнішої інфраструктури.

Ethereum є стандартом де-факто у світі смарт-контрактів, з величезною екосистемою і чудовою підтримкою. Хоча його продуктивність нижча, він значно простіший для розробників і має більше готових інструментів.

Модулі Solana та Ethereum не є взаємозамінними напряму.
Причини цього — технічні відмінності на рівні архітектури, мови, виконання коду та середовища.

Причини

1) Ethereum використовує Solidity, а Solana використовує Rust або C/C++
2) Ethereum працює на віртуальній машині (EVM), яка однакова на всіх вузлах мережі, Solana — має свою власну модель обробки транзакцій, засновану на PoH і BPF, що не має нічого спільного з EVM.
3) Ethereum контракти живуть і працюють у EVM, Solana контракти працюють як окремі програми на рівні системного коду.

Буду порівнювати Poligon Matic з Etherum.

vs

Мова програмування

І Ethereum, і Polygon використовують Solidity як основну мову для написання смарт-контрактів. Це означає, що один і той самий код можна практично без змін розгорнути в обох мережах.

Розробники можуть використовувати ті ж самі інструменти: Hardhat, Truffle, Remix тощо. Завдяки EVM-сумісності Polygon, розробка контрактів не потребує додаткового вивчення нових мов або структур.

Інструменти розгортання

І в Ethereum, і в Polygon використовуються однакові фреймворки для розгортання: Hardhat, Truffle, Remix IDE. Різниця — тільки у вказаній мережі під час налаштування (наприклад, Goerli або Polygon Mumbai).

У Polygon доступні також тестові мережі, як-от Mumbai, які дуже зручні для перевірки dApp без витрат. Тобто з точки зору розгортання — все виглядає майже однаково, але в Polygon усе працює швидше і дешевше.

Комісії за транзакції

Ethereum має високі комісії за газ, особливо в години пік, що може зробити розгортання та взаємодію з контрактами досить дорогими.

Polygon же, як Layer 2 або sidechain, має надзвичайно низькі комісії — буквально копійки за транзакцію. Це дає перевагу в розгортанні складних dApp або роботи з великими обсягами даних.

Швидкість транзакцій

Ethereum обробляє 15–30 транзакцій на секунду, що є обмеженням при великому навантаженні.

Polygon, натомість, здатен обробляти до 7 000 транзакцій на секунду, що робить розгортання в реальному часі значно ефективнішим — особливо для інтерактивних застосунків або ігор.

Інфраструктура і підтримка

Ethereum має найбільше ком’юніті та найбільш усталену інфраструктуру — сотні бібліотек, шаблонів, гайдлайнів, DAO, NFT-маркетплейсів і DeFi-протоколів.

Polygon має сумісність з Ethereum, але також розвиває власні рішення, включно з zkEVM, PoS-мережею, і підтримує тисячі проєктів. Багато сервісів вже підтримують Polygon "із коробки" — це значно спрощує інтеграцію після розгортання.

Безпека

Ethereum вважається однією з найбезпечніших блокчейн-платформ, бо його захищають десятки тисяч валідаторів і нод.

Polygon працює на мережі з меншим числом валідаторів, але вона все ще досить надійна. До того ж Polygon часто перенаправляє частину безпеки через Ethereum, як основний ланцюг (особливо в рішенні zkEVM).

Розгортання на практиці

На практиці, розгортаючи смарт-контракт, розробник:

- пише Solidity-код;
- використовує Hardhat/Truffle для компіляції та деплою;
- змінює лише параметри RPC-мережі (Ethereum або Polygon);
- отримує адресу контракту та може одразу взаємодіяти.

Тобто сам процес розгортання майже ідентичний, але Polygon дає економію часу та грошей.

Висновок:

Ethereum — це надійний, усталений і максимально децентралізований варіант, проте дорогий і повільніший.Polygon — це швидкий, дешевий і повністю сумісний з Ethereum варіант для розгортання dApp, ідеальний для проєктів із високим трафіком або обмеженим бюджетом.

Завдяки EVM-сумісності, модулі та контракти з Ethereum можна легко перенести в Polygon, що робить його чудовим кандидатом для масштабування проєктів без втрати функціоналу.

ЧАСТИНА 2

Провести налаштування обраної системи та виконати тестові операції в системі.

Запуск Bitcoin Core в Docker (у тестовій мережі)

```
prosto_acc@Vladyslavs-MacBook-Pro labs % docker pull ruimarinho/bitcoin-core

Using default tag: latest
latest: Pulling from ruimarinho/bitcoin-core
4f4fb700ef54: Already exists
39e9e0c792fb: Download complete
66dbba0fb1b5: Download complete
5b3c02233c89: Download complete
369e78ef44b1: Download complete
Digest: sha256:79dd32455cf8c268c63e5d0114cc9882a8857e942b1d17a6b8ec40a6d44e3981
Status: Downloaded newer image for ruimarinho/bitcoin-core:latest
docker.io/ruimarinho/bitcoin-core:latest
prosto_acc@Vladyslavs-MacBook-Pro labs %
```

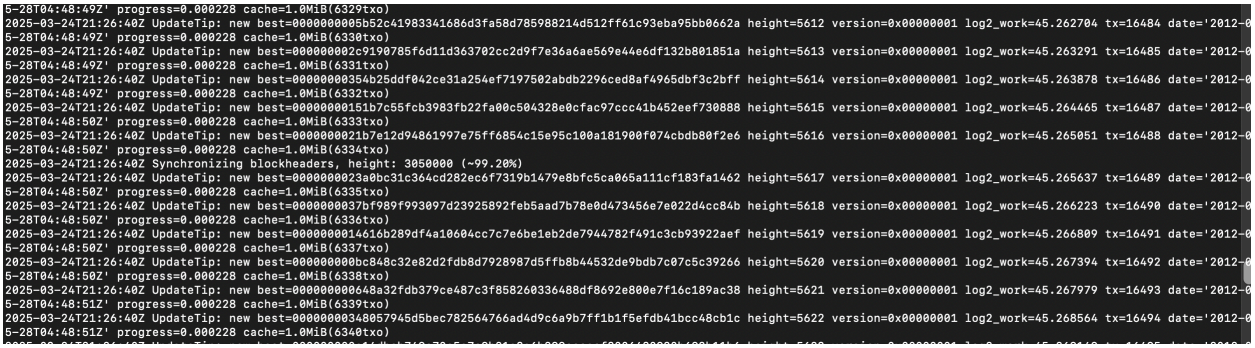
```
prosto_acc@Vladyslavs-MacBook-Pro labs % docker run -d \
  --name=bitcoin-node \
  -v $HOME/bitcoin-data:/bitcoin/.bitcoin \
  -p 8332:8332 -p 18332:18332 \
  ruimarinho/bitcoin-core \
  -testnet=1 -printtoconsole -rpcallowip=0.0.0.0/0 -rpcbind=0.0.0.0 \
  -rpcuser=bitcoin -rpcpassword=securepassword

4e41bdb075dc2937b54e093337c670f27c7148d6dbdfb869af15ba22e897d4c1
prosto_acc@Vladyslavs-MacBook-Pro labs %
```

-testnet=1 — включає тестову мережу

rpcuser / rpcpassword — для авторизації

-v — зберігає дані між перезапусками

Все запустилось та працює



Спробуємо деякі запити до системи

1) створюємо гаманець

```
prosto_acc@Vladyslavs-MacBook-Pro ~ % curl --user bitcoin:securepassword --data-binary \
'{"jsonrpc": "1.0", "id":"curltest", "method": "createwallet", "params": ["TestWallet"] }' \
-H 'content-type: text/plain;' http://localhost:18332/

{"result":{"name":"TestWallet","warning":""},"error":null,"id":"curltest"}
prosto_acc@Vladyslavs-MacBook-Pro ~ %
```

2) Отримуємо адресу гаманця

```
prosto_acc@Vladyslavs-MacBook-Pro ~ % curl --user bitcoin:securepassword --data-binary \
'{"jsonrpc": "1.0", "id":"curltest", "method": "getnewaddress", "params": [] }' \
-H 'content-type: text/plain;' http://localhost:18332/wallet/TestWallet

{"result":"tb1qxgj4fhwv7wwyklnl4kyqpxy7wzqscqcyfye50e","error":null,"id":"curltest"}
prosto_acc@Vladyslavs-MacBook-Pro ~ %
```

3) Дивимось баланс(getbalance)

```
prosto_acc@Vladyslavs-MacBook-Pro ~ % curl --user bitcoin:securepassword --data-binary \
'{"jsonrpc": "1.0", "id":"curltest", "method": "getbalance", "params": [] }' \
-H 'content-type: text/plain;' http://localhost:18332/

{"result":0.00000000,"error":null,"id":"curltest"}
prosto_acc@Vladyslavs-MacBook-Pro ~ %
```

4) Висота останього блоку блокчейн (getblockcount)

```
prosto_acc@Vladyslavs-MacBook-Pro ~ % curl --user bitcoin:securepassword --data-binary \
'{"jsonrpc": "1.0", "id":"test", "method": "getblockcount", "params": []}' \
-H 'content-type: text/plain;' http://localhost:18332/
{"result":128430,"error":null,"id":"test"}
prosto_acc@Vladyslavs-MacBook-Pro ~ %
```

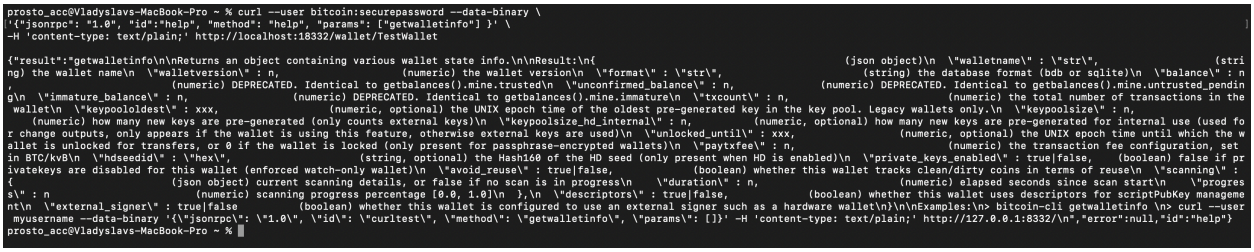
5) Або наприклад інформацію про мережу (getnetworkinfo)

```
prosto_acc@Vladyslavs-MacBook-Pro ~ % curl --user bitcoin:securepassword --data-binary \
'{"jsonrpc": "1.0", "id":"test", "method": "getnetworkinfo", "params": []}' \
-H 'content-type: text/plain;' http://localhost:18332/
{"result":{"version":240001,"subversion":"/Satoshi:24.0.1/","protocolversion":70016,"localservices":"0000000000000409","localservicesnames":["NETWORK","WITNESS","NETWORK
_LIMITED"],"localrelay":true,"timeoffset":0,"networkactive":true,"connections":10,"connections_in":0,"connections_out":10,"networks":[{"name":"ipv4","limited":false,"rea
chable":true,"proxy":"","proxy_randomize_credentials":false},{"name":"ipv6","limited":false,"reachable":true,"proxy":"","proxy_randomize_credentials":false},{"name":"oni
on","limited":true,"reachable":false,"proxy":"","proxy_randomize_credentials":false},{"name":"i2p","limited":true,"reachable":false,"proxy":"","proxy_randomize_credentia
ls":false},{"name":"cjdns","limited":true,"reachable":false,"proxy":"","proxy_randomize_credentials":false}],"relayfee":0.00001000,"incrementalfee":0.00001000,"localaddr
esses":[],"warnings":""},"error":null,"id":"test"}
prosto_acc@Vladyslavs-MacBook-Pro ~ %
```

6) Команда щоб вивести всі команди(help)

```
prosto_acc@Vladyslavs-MacBook-Pro ~ % curl --user bitcoin:securepassword --data-binary \
'{"jsonrpc": "1.0", "id":"help", "method": "help", "params": [] }' \
-H 'content-type: text/plain;' http://localhost:18332/wallet/TestWallet

{"result":"== Blockchain ==\ngetbestblockhash\ngetblock \"blockhash\" ( verbosity )\ngetblockchaininfo\ngetblockcount\ngetblockfilter \"blockhash\" ( \"filtertype\" )\ngetblockfrompeer \"blockhash\" peer_
id\ngetblockhash height\ngetblockheader \"blockhash\" ( verbose )\ngetblockstats hash_or_height ( stats )\ngetchaintips\ngetchaintxstats ( nblocks \"blockhash\" )\ngetdeploymentinfo ( \"blockhash\" )\nget
difficulty\ngetmempoolancestors \"txid\" ( verbose )\ngetmempooldescendants \"txid\" ( verbose )\ngetmempoolentry \"txid\"\ngetmempoolinfo\ngetrawmempool ( verbose mempool_sequence )\ngettxout \"txid\" n
( include_mempool )\ngettxoutproof [\"txid\",...] ( \"blockhash\" )\ngettxoutsetinfo ( \"hash_type\" hash_or_height use_index )\ngettxspendingprevout [{\"txid\":\"hex\",\"vout\":n},...]\npreciousblock \"b
lockhash\"\npruneblockchain height\nsavemempool\nscantxoutset \"action\" ( [scanobjects,...] )\nverifychain ( checklevel nblocks )\nverifytxoutproof \"proof\"\n\n== Control ==\ngetmemoryinfo ( \"mode\" )\
ngetrpcinfo\nhelp ( \"command\" )\nlogging ( [\"include_category\",...] [\"exclude_category\",...] )\nstop\nuptime\n\n== Mining ==\ngetblocktemplate ( \"template_request\" )\ngetmininginfo\ngetnetworkhash
ps ( nblocks height )\nprioritisetransaction \"txid\" ( dummy ) fee_delta\nsubmitblock \"hexdata\" ( \"dummy\" )\nsubmitheader \"hexdata\"\n\n== Network ==\naddnode \"node\" \"command\"\nclearbanned\ndisc
onnectnode ( \"address\" nodeid )\ngetaddednodeinfo ( \"node\" )\ngetconnectioncount\ngetnettotals\ngetnetworkinfo\ngetnodeaddresses ( count \"network\" )\ngetpeerinfo\nlistbanned\nping\nsetban \"subnet\"
 \"command\" ( bantime absolute )\nsetnetworkactive state\n\n== Rawtransactions ==\nanalyzepsbt \"psbt\"\ncombinepsbt [\"psbt\",...]\ncombinerawtransaction [\"hexstring\",...]\nconverttopsbt \"hexstring\"
 ( permitsigdata iswitness )\ncreatepsbt [{\"txid\":\"hex\",\"vout\":n,\"sequence\":n},...] [{\"address\":amount,...},{\"data\":\"hex\"},...] ( locktime replaceable )\ncreaterawtransaction [{\"txid\":\"he
x\",\"vout\":n,\"sequence\":n},...] [{\"address\":amount,...},{\"data\":\"hex\"},...] ( locktime replaceable )\ndecodepsbt \"psbt\"\ndecoderawtransaction \"hexstring\" ( iswitness )\ndecodescript \"hexstr
ing\"\nfinalizepsbt \"psbt\" ( extract )\nfundrawtransaction \"hexstring\" ( options iswitness )\ngetrawtransaction \"txid\" ( verbose \"blockhash\" )\njoinpsbts [\"psbt\",...]\nsendrawtransaction \"hexst
ring\" ( maxfeerate )\nsignrawtransactionwithkey \"hexstring\" [\"privatekey\",...] ( [{\"txid\":\"hex\",\"vout\":n,\"scriptPubKey\":\"hex\",\"redeemScript\":\"hex\",\"witnessScript\":\"hex\",\"amount\":a
mount},...] \"sighashtype\" )\ntestmempoolaccept [\"rawtx\",...] ( maxfeerate )\nutxoupdatepsbt \"psbt\" ( [\"\",{\"desc\":\"str\",\"range\":n or [n,n]},...] )\n\n== Signer ==\nenumeratesigners\n\n== Util
 ==\ncreatemultisig nrequired [\"key\",...] ( \"address_type\" )\nderiveaddresses \"descriptor\" ( range )\nestimatesmartfee conf_target ( \"estimate_mode\" )\ngetdescriptorinfo \"descriptor\"\ngetindexin
fo ( \"index_name\" )\nsignmessagewithprivkey \"privkey\" \"message\"\nvalidateaddress \"address\"\nverifymessage \"address\" \"signature\" \"message\"\n\n== Wallet ==\nabandontransaction \"txid\"\nabortr
escan\naddmultisigaddress nrequired [\"key\",...] ( \"label\" \"address_type\" )\nbackupwallet \"destination\"\nbumpfee \"txid\" ( options )\ncreatewallet \"wallet_name\" ( disable_private_keys blank \"pa
ssphrase\" avoid_reuse descriptors load_on_startup external_signer )\ndumpprivkey \"address\"\ndumpwallet \"filename\"\nencryptwallet \"passphrase\"\ngetaddressesbylabel \"label\"\ngetaddressinfo \"addres
s\"\ngetbalance ( \"dummy\" minconf include_watchonly avoid_reuse )\ngetbalances\ngetnewaddress ( \"label\" \"address_type\" )\ngetrawchangeaddress ( \"address_type\" )\ngetreceivedbyaddress \"address\" (
 minconf include_immature_coinbase )\ngetreceivedbylabel \"label\" ( minconf include_immature_coinbase )\ngettransaction \"txid\" ( include_watchonly verbose )\ngetunconfirmedbalance\ngetwalletinfo\nimpor
taddress \"address\" ( \"label\" rescan p2sh )\nimportdescriptors \"requests\"\nimportmulti \"requests\" ( \"options\" )\nimportprivkey \"privkey\" ( \"label\" rescan )\nimportprunedfunds \"rawtransaction
\" \"txoutproof\"\nimportpubkey \"pubkey\" ( \"label\" rescan )\nimportwallet \"filename\"\nkeypoolrefill ( newsize )\nlistaddressgroupings\nlistdescriptors ( private )\nlistlabels ( \"purpose\" )\nlistlo
ckunspent\nlistreceivedbyaddress ( minconf include_empty include_watchonly \"address_filter\" include_immature_coinbase )\nlistreceivedbylabel ( minconf include_empty include_watchonly include_immature_co
inbase )\nlistsinceblock ( \"blockhash\" target_confirmations include_watchonly include_removed include_change )\nlisttransactions ( \"label\" count skip include_watchonly )\nlistunspent ( minconf maxconf
 [\"address\",...] include_unsafe query_options )\nlistwalletdir\nlistwallets\nloadwallet \"filename\" ( load_on_startup )\nlockunspent unlock ( [{\"txid\":\"hex\",\"vout\":n},...] persistent )\nmigratewa
llet\nnewkeypool\npsbtbumpfee \"txid\" ( options )\nremoveprunedfunds \"txid\"\nrescanblockchain ( start_height stop_height )\nrestorewallet \"wallet_name\" \"backup_file\" ( load_on_startup )\nsend [{\"a
ddress\":amount,...},{\"data\":\"hex\"},...] ( conf_target \"estimate_mode\" fee_rate options )\nsendall [\"address\",{\"address\":amount,...},...] ( conf_target \"estimate_mode\" fee_rate options )\nsend
many \"\" {\"address\":amount,...} ( minconf \"comment\" [\"address\",...] replaceable conf_target \"estimate_mode\" fee_rate verbose )\nsendtoaddress \"address\" amount ( \"comment\" \"comment_to\" subtr
actfeefromamount replaceable conf_target \"estimate_mode\" avoid_reuse fee_rate verbose )\nsethdseed ( newkeypool \"seed\" )\nsetlabel \"address\" \"label\"\nsettxfee amount\nsetwalletflag \"flag\" ( valu
e )\nsignmessage \"address\" \"message\"\nsignrawtransactionwithwallet \"hexstring\" ( [{\"txid\":\"hex\",\"vout\":n,\"scriptPubKey\":\"hex\",\"redeemScript\":\"hex\",\"witnessScript\":\"hex\",\"amount\":
amount},...] \"sighashtype\" )\nsimulaterawtransaction ( [\"rawtx\",...] {\"include_watchonly\":bool,...} )\nunloadwallet ( \"wallet_name\" load_on_startup )\nupgradewallet ( version )\nwalletcreatefunded
psbt ( [{\"txid\":\"hex\",\"vout\":n,\"sequence\":n,\"weight\":n},...] ) [{\"address\":amount,...},{\"data\":\"hex\"},...] ( locktime options bip32derivs )\nwalletdisplayaddress \"address\"\nwalletlock\nw
alletpassphrase \"passphrase\" timeout\nwalletpassphrasechange \"oldpassphrase\" \"newpassphrase\"\nwalletprocesspsbt \"psbt\" ( sign \"sighashtype\" bip32derivs finalize )\n\n== Zmq ==\ngetzmqnotificatio
ns","error":null,"id":"help"}
```

7) Або допомога з специфічною командою наприклад getwalletinfo



Висновок

Виконавши лабораторну роботу, я провів порівняльний аналіз особливостей розгортання систем криптовалют у порівнянні із системою Etherium, а саме Solana та більше схожої до Etherum – Polygon Matic. Та зробив висновок про можливість чи неможливість взаємозаміни модулів різних систем та пояснити причини. Також провів налаштування обраної системи, а саме Bitcoin та виконав тестові операції в системі.