**<Java로 구현한 Blockchain> 황보성훈**

[**https://medium.com/programmers-blockchain/create-simple-blockchain-java-tutorial-from-scratch-6eeed3cb03fa**](https://medium.com/programmers-blockchain/create-simple-blockchain-java-tutorial-from-scratch-6eeed3cb03fa)

**[NoobChain.java]** (main() 함수 위치)

**[Wallet.java]**

public Wallet()

public void generateKeyPair(): private key, public key 생성(한 쌍)

public float getBalance(): UTXO의 총합을 구해서 지갑에 얼마 있는지 확인

public Transaction sendFunds(PublicKey \_recipient, float value): \_recipient 은 받는 사람, value 는 금액

**[Transaction.java]**

public Transaction(PublicKey from, PublicKey to, float value, ArrayList<TransactionInput> inputs)

private String calculateHash(): from, to, value, sequence를 통해 해시를 만듦 🡪 transactionID로 사용

public void generateSignature(PrivateKey privateKey): privateKey를 이용해서 signature 만듦

public boolean verifySignature(): 보내는 사람의 publickey를 이용해서 signature가 정당한 것인지 확인

public boolean processTransaction(): transaction의 생성 유무를 boolean으로 return

public float getInputsValue(): inputs(UTXOs)의 금액의 합을 return

public float getOutputsValue(): outputs의 합을 return

**[Block.java]**

public Block(String previousHash): hash, previousHash, merkleRoot, transactions(ArrayList형태), timestamp, nonce가 저장됨

public String calculateHash(): block의 contents들을 이용해서 해시 만듦

public void mineBlock(int difficulty): nonce를 하나 씩 올리면서 difficulty에 만족하는 해시를 만들 때까지 실행

public boolean addTransaction(Transaction transaction): 이 block에 transaction 더해줌

**[TransactionInput.java]**

public TransactionInput(String transactionOutputId)

**[TransactionOutput.java]**

public TransactionOutput(PublicKey reciepient, float value, String parentTransactionId)

public boolean isMine(PublicKey publicKey)

**[StringUtil.java]**

public static String applySha256(String input)

public static byte[] applyECDSASig(PrivateKey privateKey, String input): private key와 거래 정보를 이용해서 signature 만듦

public static boolean verifyECDSASig(PublicKey publicKey, String data, byte[] signature): public key와 거래 정보, signature를 이용해서 signature가 올바른 것인지 확인

public static String getStringFromKey(Key key)

public static String getMerkleRoot(ArrayList<Transaction> transactions): transactions을 이용해서 merkle root 획득

public static String getDifficultyString(int difficulty)