Modul Blockchain (BlCh)
Prüfung HS22

Nachname: _____ **Vorname:** _____

Nr.	Aufgabe	Max. Punkte	Erreichte Punkte
1	Blockchain / Stablecoin Basics	15	
2	Decentralized Finance (DeFi) Basics	10	
3	Non-Fungible Tokens (NFTs)	9	
4	Atomic Swaps / HTLC	6	
5	Solidity / Ethereum	3	
6	Merkle Tree / Root	4	

Total 47

Hinweise allgemein:

- Zeit: 60 min.
- Zuerst diese Hinweise, dann die Prüfungsteile ganz durchlesen.
- Diese Prüfung ist **"Closed Book".**
- Taschenrechner ist nicht erlaubt.
- Es sind keine Hilfsmittel zugelassen; auch kein Handy oder Smart Watch.
- Alle Antworten und Lösungen in diese Aufgabenblätter schreiben.
- Keinen Bleistift und keine rote Farbe verwenden.
- Die Antworten und Lösungen müssen nachvollziehbar sein. Bitte alles in leserlicher Schrift!
- Keine Fragen während der Prüfung. Bei Unklarheiten sind sinnvolle Annahmen zu treffen und diese klar zu dokumentieren.
- Die Bewertungspunkte der Aufgaben können nach der Korrektur noch etwas variieren.

Hinweise zum Prüfungsbeginn:

- Versehen Sie das Titelblatt mit Ihrem Namen.
- Den OST-Badge sichtbar auf den Tisch legen.
- Allfällig bereitliegende Prüfung nicht umdrehen oder umblättern, bis die Aufsichtsperson explizit das OK dazu gegeben hat.

Hinweise zur Prüfungsabgabe:

- Kontrollieren Sie nochmals, ob die Prüfung mit Ihrem Namen versehen ist. Geben Sie alle Blätter – auch von nicht gelösten Aufgaben – geordnet ab.
- Nach Prüfungsende: Drehen Sie Ihre Prüfung um und warten Sie, bis die Prüfung eingesammelt wird. Bleiben Sie dann sitzen bis die/eine Aufsichtsperson die Prüfungen gezählt hat und ein Zeichen zum Verlassen des Raums gegeben hat.

Aufgabe 1: Blockchain / Stablecoin Basics (16 Punkte)

a) (5P) Kreuzen Sie Zutreffendes an.

Bitte beachten Sie, dass **keine** Punkte für ein falsch gesetztes Kreuz abgezogen werden.

	Statement	True	False
1	Es gibt keine Blockchain, die Transaktionsdaten verschlüsselt/verschleiert.		
2	Mit proof-of-stake (PoS) sind keine 51% Angriffe möglich.		
3	Der Umstieg von proof-of-work (PoW) zu proof-of-stake (PoS) hat keinen grossen Einfluss auf den Strombedarf.		
4	Die meisten Smart Contract Entwickler verwenden bei Solana die Programmiersprache Solidity.		
5	In einer Blockchain kann es vorkommen, dass zwei Blöcke den gleichen Verweis zum vorherigen Block haben.		

b) (3P) Was verstehen Sie unter den Begriffen Stablecoin, CeFi, und DeFi? Erklären Sie die Begriffe mit je einem Satz.

Stablecoin:

CeFi:

DeFi:

c) (3P) Was muss man bezüglich der Volatilität beachten, wenn man eine Kryptowährung (z.B. Ethereum) bei einem Stablecoin hinterlegen möchte?

d) (4P) Beschreiben Sie einen Reentrancy Attack mit einem Pseudo Code Beispiel (Syntax ist nicht relevant).

Reentrancy attack:

Aufgabe 2: Decentralized Finance (DeFi) Basics (10 Punkte)

a) (4P) Wie wird der Preis bei einer Centralized Exchange (CEX) bestimmt? Und wie wird der Preis bei einer Decentralized Exchange (DEX) bestimmt?

Preis bei CEX:

Preis bei DEX:

b) (4P) Was verstehen Sie unter einem «Flash Loan»? Beschreiben Sie einen Use-case bei einer DEX, wo «Flash Loans» eingesetzt werden können?

Flash Loan:

Use-case DEX:

c) (2P) Was verstehen Sie unter «Impermanent Loss» bei einer DEX? Erklären Sie «Impermanent Loss» in 2 Sätzen.

Impermanent Loss:

Aufgabe 3: Non-Fungible Tokens - NFTs (9 Punkte)

a) (5P) Kreuzen Sie Zutreffendes an.

Bitte beachten Sie, dass **keine** Punkte für ein falsch gesetztes Kreuz abgezogen werden.

	Statement	True	False
1	Es gibt dedizierte Blockchains speziell für NFT Use Cases.		
2	Ein «proof of attendance protocol» (POAP) kann durch ein NFT abgebildet werden.		
3	Eine Mitgliedschaft in einem DAO kann durch einen NFT abgebildet werden.		
4	Wird ein NFT über einen dezentralen Marktplatz gehandelt, kann man nicht unterscheiden ob der Käufer und Verkäufer dieselbe Person ist, da dieselbe Person mehrere Adressen besitzen kann.		
5	Ein NFT, welcher über eine 256-bit basierten Id identifiziert wird, ist immer eindeutig und es kann auf der gesamten Blockchain keinen zweiten NFT geben mit derselben Id.		

b) (4P) Sie entwickeln ein UI, welches alle Ethereum NFTs aus einem bestimmten Contract auflistet, und anzeigt wer der Owner dieser NFTs ist. Es stehen ihnen die folgende Funktionen zur Verfügung:

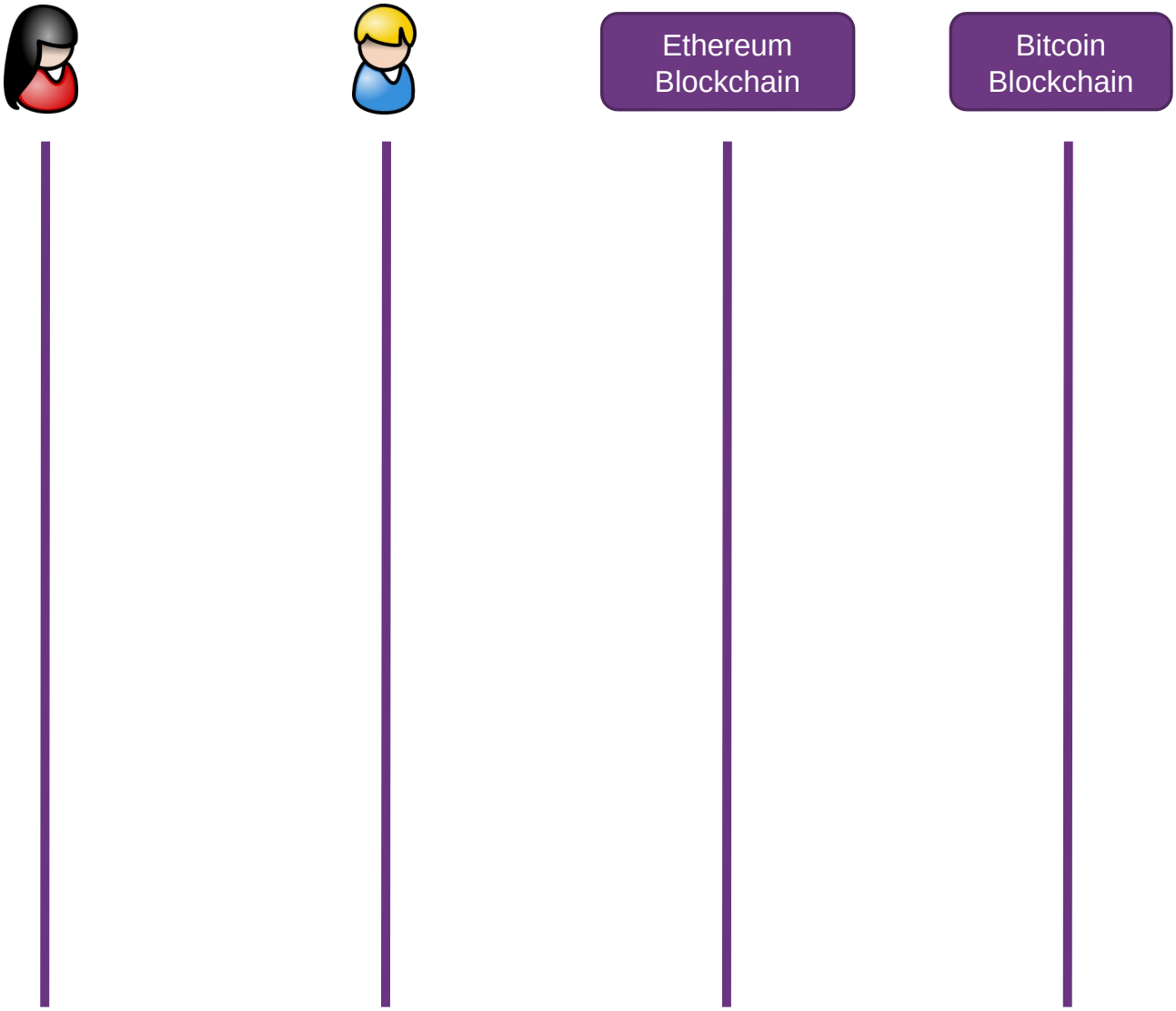
- ```
  /**
  * @dev Returns the owner of the `tokenId` token.
  */
  function ownerOf(uint256 _tokenId) returns (address)
  ```
- ```
  /**
  * @dev Returns the total amount of tokens stored by the contract.
  */
  function totalSupply() returns (uint256)
  ```
- ```
  /**
  * @dev Returns a token ID at a given `index` of all the tokens stored by the
  * contract. Use along with {totalSupply} to enumerate all tokens.
  */
  function tokenIdByIndex(uint256 _index) returns (uint256)
  ```

Beschreiben Sie mit einem Pseudo Code Beispiel (Syntax ist nicht relevant), eine mögliche Implementation welche alle NFTs mit entsprechendem Owner auflistet.
  ```

Aufgabe 4: Atomic Swaps / HTLC (6 Punkte)

Alice hat 10 ETHs und Bob hat 1 BTC. Alice möchte mit Bob 1 BTC für 10 ETHs tauschen. Wie kann Alice und Bob die ETHs und BTCs mit einem Atomic Swap und Hashed Time Lock Contracts (HTLC) austauschen?

(6P) Wer sendet welche Transaktionen an die entsprechenden Blockchains? Vervollständigen Sie die folgende Grafik eines erfolgreichen Austausches. Gehen Sie davon aus, dass die HTLC Smart Contracts und Adressen bereits existieren (Ethereum) resp. beiden bekannt sind (Bitcoin). **Alice** initiiert den Prozess.



Aufgabe 5: Solidity/Ethereum (3 Punkte)

Das folgende Code Snippet eines Ethereum Smart Contracts soll eine Zufallszahl zurück liefern:

```
//Generate random number between 0 & max
uint256 constant private FACTOR =
115792089237316195423570985008687907853269984665640564039457584007913129639935;
function rand(uint max) constant private returns (uint256 result){
    uint256 factor = FACTOR * 100 / max;
    uint256 lastBlockNumber = block.number - 1;
    uint256 hashVal = uint256(block.blockhash(lastBlockNumber));

    return uint256((uint256(hashVal) / factor)) % max;
}
```

a) (3P) Was könnte das Problem in diesem Code sein? (Hinweis: der Syntax ist korrekt)

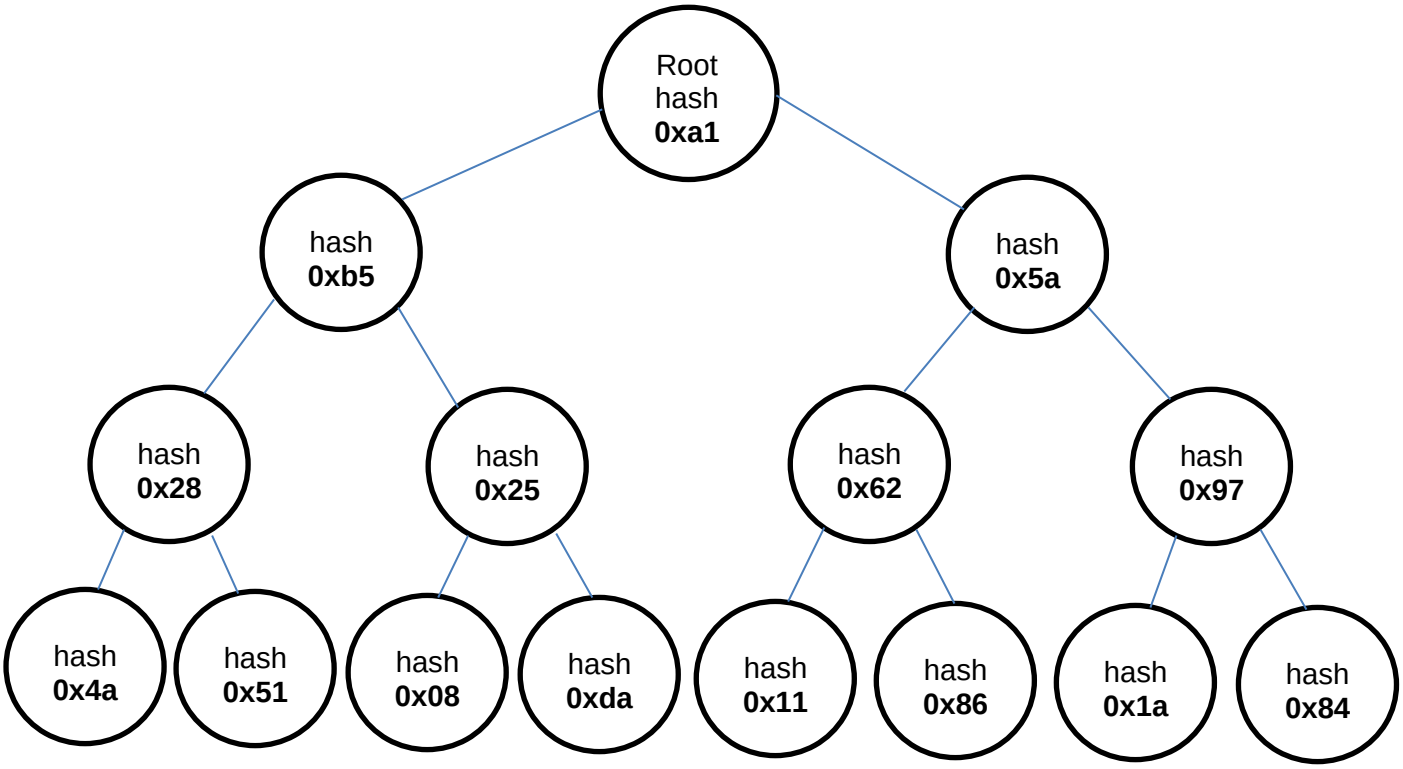
Problem:

Aufgabe 6: Merkle Tree / Root (4 Punkte)

(4P) In dem folgenden Merkle Tree mit der Root hash **0xa1** befinden sich Hashes von 8 Transaktionen. Sie erhalten die folgenden Merkle Proofs:

- Transaktionshash 0x08, proof: 0x25, 0xb5, 0x5a
- Transaktionshash 0x86, proof: 0x11, 0x97, 0xb5
- Transaktionshash 0x4a, proof: 0x25, 0x5a, 0xb5
- Transaktionshash 0x5a, proof: 0xa1, 0xb5, 0x28

Sind das korrekte Merkle Proofs?



Lösung:

Transaktionshash 0x08, proof: 0x25, 0xb5, 0x5a:
Transaktionshash 0x86, proof: 0x11, 0x97, 0xb5:
Transaktionshash 0x4a, proof: 0x25, 0x5a, 0xb5:
Transaktionshash 0x84, proof: 0x1a, 0x62, 0xb5: