Modul Blockchain (BlCh)
Prüfung HS21

Nachname: _____ **Vorname:** _____

Nr.	Aufgabe	Max. Punkte	Erreichte Punkte
1	Blockchain Basics	11	
2	Decentralized Finance (DeFi) Basics	9	
3	Non-Fungible Tokens (NFTs)	14	
4	Atomic Swaps	8	
5	Solidity / Ethereum	4	

Total 46

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 Basics (11 Punkte)

a) (5P) Kreuzen Sie Zutreffendes an.

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

	Statement	True	False
1	Jede Blockchain verwendet proof-of-work (PoW).		
2	Mit proof-of-stake (PoS) sind keine 51% Angriffe möglich.		
3	Öffentliche Blockchains eignen sich schlecht um grosse Datenmengen zu speichern.		
4	Bei einer Blockchain hat ein Block jeweils einen Verweis zum vorherigen Block, so bildet sich eine Kette von Blöcken.		
5	In einer Blockchain kann es kurzfristig vorkommen, dass zwei Blöcke den gleichen Verweis zum vorherigen Block haben.		

b) (2P) Wofür kann man MetaMask verwenden?

d) (4P) Erklären Sie wie ein 51% Angriff funktioniert.

Aufgabe 2: Decentralized Finance (DeFi) Basics (9 Punkte)

a) (3P) Beschreiben Sie was man unter Flash Loan bei DeFi versteht.

b) (3P) In einer Centralized Exchange (CEX) wird der Preis per Orderbook bestimmt. Wie wird der Preis bei einer Decentralized Exchange (DEX) bestimmt?

b) (3P) Was versteht man unter Impermanent Loss bei DeFi?

Aufgabe 3: Non-Fungible Tokens - NFTs (14 Punkte)

a) (5P) Kreuzen Sie Zutreffendes an.

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

	Statement	True	False
1	Jede Blockchain unterstützt NFTs.		
2	Mit einem Transfer von einem NFT wird auch immer das Copyright übertragen.		
3	Ein NFT in Ethereum wird immer mit einem entsprechenden 256-bit basiertem Hash identifiziert.		
4	Wird ein NFT über einen dezentralen Marktplatz gehandelt, kann man nicht unterscheiden ob der Käufer und Verkäufer dieselbe Person ist.		
5	Kann in einem Smart Contract sichergestellt werden, dass bei einem Transfer von einem NFT auch eine entsprechende Anzahl an Tokens oder Coins überwiesen werden?		

b) (3P) In der folgenden Implementation eines NFT Marktplatzes hat sich ein schwerwiegender Fehler eingeschlichen im Use-case: Ein Angebot erstellen, und das Angebot wieder zurückziehen. Was ist der Fehler, und wie kann man den Fehler beheben?

```
function makeOffer(address nftContract, uint256 nftId) payable external {
  _offers[nftContract][nftId][msg.sender] = msg.value;
  emit Offer(nftContract, nftId, msg.sender, msg.value);
}
function withdrawOffer(address nftContract, uint256 nftId) payable external {
  uint256 val = _offers[nftContract][nftId][msg.sender];
  delete(_offers[nftContract][nftId][msg.sender]);
  payable(msg.sender).transfer(val);
  emit Withdraw(nftContract, nftId, msg.sender, val);
}
```

c) (6P) Sie entwickeln ein UI, welches Ethereum NFTs in einem bestimmten Contract auflistet, und zu welcher Adresse diese NFTs gehören. Es stehen ihnen die folgende Funktionen zur Verfügung:

- `function ownerOf(uint256 _tokenId)`
- `function totalSupply()`
- `function tokenIdByIndex(uint256 _index)`
- `tokenOfOwnerByIndex(address _owner, uint256 _index)`

Beschreiben Sie mit Stichworten oder in Pseudo Code (Syntax ist nicht relevant), eine mögliche Implementation welche alle `tokenIds` mit entsprechender Adresse auflistet.

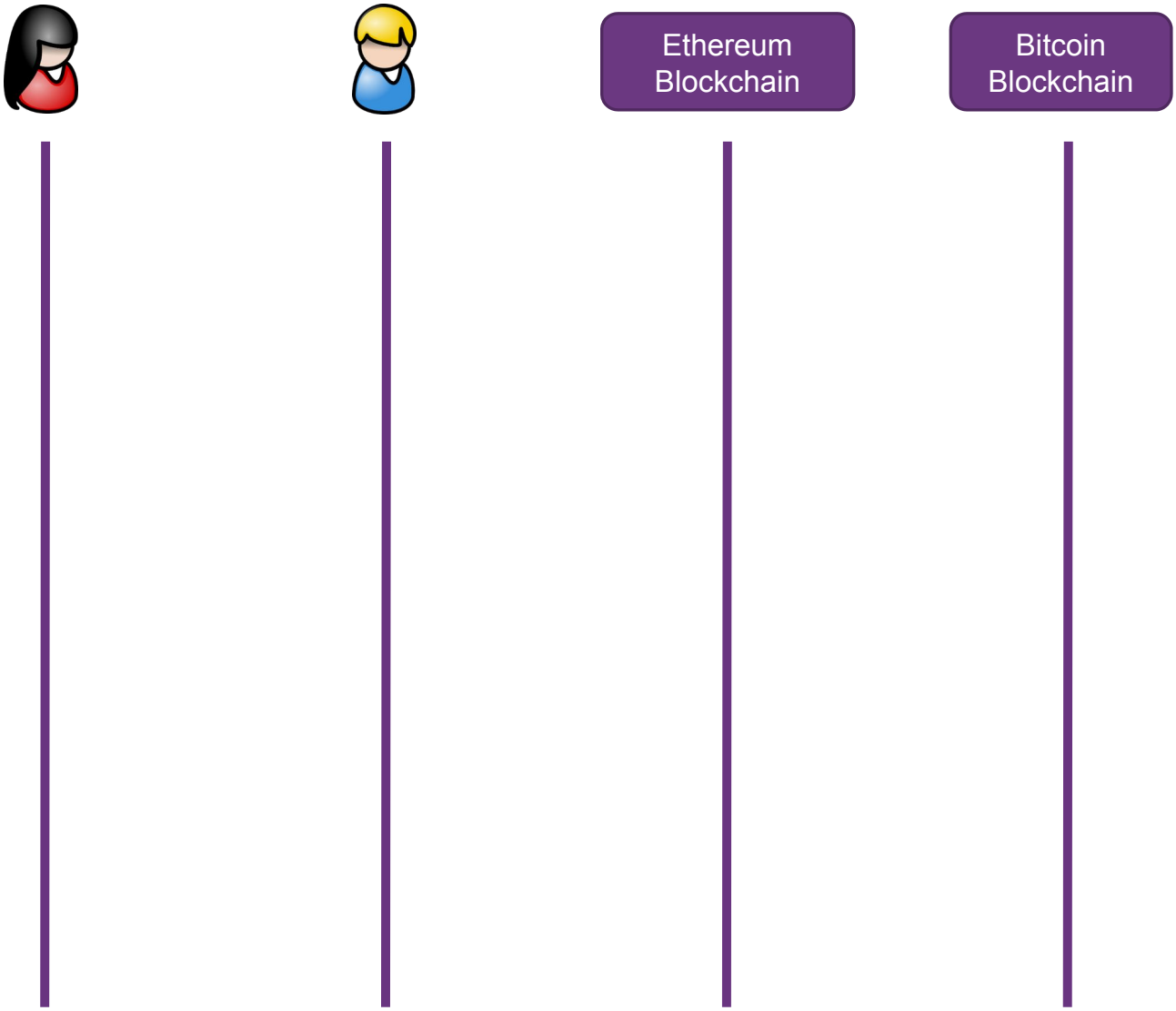
g) (2P) Was gibt es für Alternativen und die NFTs aufzulisten, wenn die Funktionen der ERC721 Erweiterung Enumerable: `totalSupply()`, `tokenByIndex(uint256 _index)` und `tokenOfOwnerByIndex(address _owner, uint256 _index)` nicht zur Verfügung stehen?

Aufgabe 4: Atomic Swaps (8 Punkte)

Alice hat 37 ETHs und Bob hat 1 BTC. Alice möchte mit Bob 1 BTC für 37 ETHs tauschen. Beschreiben Sie wie Alice und Bob die Coins austauschen können.

a) (2P) Welche Art von Contracts werden Alice und Bob verwenden wenn sie die Coins dezentral austauschen wollen?

b) (6P) Wer sendet welche der folgenden Informationen und Coins (`secret`, `hash(secret)`, `1BTC`, `37ETH`) an die entsprechenden Smart Contracts. Vervollständigen Sie die folgende Grafik eines erfolgreichen Austausches. Gehen Sie davon aus, dass die Smart Contracts und Adressen bereits existieren (Ethereum) resp. beiden bekannt sind (Bitcoin). **Bob** initiiert den Prozess.



Aufgabe 5: Solidity/Ethereum (4 Punkte)

Das folgende Snippet eines Ethereum Smart Contracts liefert den Index einer Adresse aus einem Array von Investorenadressen. Wird eine Investorenadresse nicht gefunden, wird die Länge des Arrays zurückgegeben:

```
function getEtherBalanceIndex(address _address) internal
returns (uint256) {
        for (uint256 i = 0; i < investors.length; i++) {
            if (investors[i] == _address) {
                return i;
            }
        }
        return investors.length;
    }
```

a) (4P) Was könnte das Problem in diesem Code sein? (Hinweis: der Syntax ist korrekt)

Problem: