

10 秒で始める AI プログラミング学習サービス「Aidemy」

新コース「ブロックチェーン発展 II, III」を開設

－ ビットコインにおける UTXO・ブルームフィルタ・マークルツリーの実装 －

東大発スタートアップの株式会社アイデミー(本社:東京都渋谷区、代表取締役 CEO 石川聡彦)が提供する AI プログラミング学習サービス「Aidemy」(<https://aidemy.net/>)は、2018 年 7 月 31 日(水)より、新コース「ブロックチェーン発展 II」「ブロックチェーン発展 III」の提供を開始しましたのでお知らせいたします。



【「ブロックチェーン発展 II」コース概要】

ビットコインシステムにおいて、トランザクション(取引記録)とは最も重要な部分です。ここではトランザクションの構造を解説し、アカウント別の残高計算方法、なりすましを防ぐ仕組みを学びながら実装していきます。

なお、このコンテンツは3部に渡るブロックチェーン発展講座の2つ目の講座となります。

■**ブロックチェーン発展 II**コース: <https://aidemy.net/courses/7120>

■コース内容

1. 基礎講座の復習と概要

このコースは無料で公開しているブロックチェーン基礎講座を基に説明を進めます。今回の発展 II 講座を理解していただく上で必要なブロックチェーン基礎講座の要素をクイズ形式で復習します。また今回の講座の全体像を理解していただきます。

2. コイン識別方法

ビットコインモデルにおいて、効率的にコイン残高を記録し計算するために「コイン識別方式」が採用されています。ここではこの方式の概要とメリット、デメリットについてクイズ形式で学びます。

3. ビットコインの UTXO

コイン識別方式を基本として暗号通貨向けに発展させた仕組みである UTXO について見ていきます。UTXO とはコイン残高をデータとして保持したまま不正検証を行うシステムです。この仕組みを理解し、トランザクションから UTXO を読み取り、残高計算を実装していきます。

4. 電子署名

ビットコインにおいて、送金時に電子署名を実装することでなりすましを防ぐことができます。この電子署名を実装するために、UTXO のロック、アンロック処理について学習していきます。

ブロックチェーン発展 II

1.4.2 Scriptによるロック/アンロック

まずはScriptの具体例として、「2+3=5」という式が成立するかどうかをTrue/Falseで返すscriptを見てみましょう。

下図のように動作するScriptは、

```
2 3 OP_ADD 5 OP_EQUAL
```

という命令コードで書けます。Scriptはコード内の個々のアイテムを左から順番に処理を行うこと実行できます。2などの数値がある場合はそれをスタックの一番上加えます。(プッシュと言います。それに対してアイテムをスタックから除くことをポップと言います)。

OP_XXX などの表示(オペレーターと言います)は1つまたは複数の値をプッシュまたはポップし、それに対し何かしらの操作をするものです。例えば、OP_ADD という表示は加法オペレーターと呼ばれ、上にある2つの値を足し合わせ、結果をスタックにプッシュするという操作をします。また、OP_EQUAL というオペレーターは上にある2つの値が同じである場合にTRUEを返すという操作をしています。

```
1 # 用いる関数を定義しています
2 def add_number(num, stack):
3     stack.append(num)
4
5
6 def OP_ADD(stack):
7     temp = str(int(stack[-1]) + int(stack[-2]))
8     stack.pop()
9     stack.pop()
10    stack.append(temp)
11
12
13 def OP_EQUAL(stack):
14     if stack[-1] == stack[-2]:
15         temp = 1
16     else:
17         temp = 0
18     stack.pop()
19     stack.pop()
20     stack.append(temp)
21
```

コンソール

```
ScriptPubkey : ['3', 'OP_ADD', '5', 'OP_EQUAL']
ScriptSig : ['3', 'OP_ADD', '5', 'OP_EQUAL']
ロック解除成功
```

図1.4.2-1 Scriptにおける演算の例

▲「ブロックチェーン発展 II」の画面▲

■価格:980 円

【「ブロックチェーン発展 III」コース概要】

最も一般的なノードである SPV ノードはブルームフィルタを用いて、プライバシーに関するリスクを減らしながら必要なトランザクション情報を集めます。またブロックチェーンの個々のブロックは、マークルツリーという手法を用いて、そのブロックに格納されている全てのトランザクションを要約した情報を含ませています。ここではこの2つ「ブルームフィルタ」と「マークルツリー」の実装を行います。

なお、このコンテンツは3部に渡るブロックチェーン発展講座の3つ目の講座となります。

■ブロックチェーン発展 IIIコース:<https://aidemy.net/courses/7130>

■コース内容

1. ブルームフィルタ

SPV ノードにおける、トランザクションの探索フィルタである、ブルームフィルタについて理解と実装をしていきます。ブルームフィルタのアルゴリズムを理解し、実装をすることで、プライバシーに関するリスクを低減させられることを学んでいただけます。

2. マークルツリー

ブロックチェーンの個々のブロックは、そのブロックに格納されているすべてのトランザクションを要約した情報を含んでいます。そしてその要約にはマークルツリーという手法が用いられています。またマークルツリーは SPV ノードによって利用され、自分と関係のあるトランザクションをブルームフィルタを通じて、マークルパスという形式で受け取ります。このマークルパスの作成を実装していきます。

The screenshot shows the Aidemy IDE interface. On the left, there's a sidebar with the title "ブロックチェーン発展 III" and a sub-section "1.2.3 ノードのリストの作成". Below this, there's explanatory text and a diagram illustrating the Merkle tree structure. The diagram shows a sequence of transactions (トランザクション1 to n) being hashed into a Merkle root (マークルルート). Below the diagram, there's a tree diagram showing nodes (ノード1 to n) and their corresponding transactions (トランザクション1 to n).

On the right, there's a code editor showing a Python script named "solution.py". The script defines a "Tree" class and a "create_merkletree" function. The "Tree" class has attributes "data", "hash", and "transaction_path". The "create_merkletree" function takes a list of nodes and returns a list of pairs of nodes to be hashed together. The script also includes a main function that generates random transaction data and creates a list of "Tree" objects.

```
script.py solution.py
1 import random
2 from blockchain import make_hash
3
4 class Tree:
5     def __init__(self, transaction_data):
6         # dataにはtransaction_dataを代入してください
7         self.data = transaction_data
8         # hashにはtransaction_dataのハッシュ値を代入してください
9         self.hash = make_hash(transaction_data)
10        self.transaction_path = []
11
12
13 def create_merkletree(node_list):
14     if len(node_list) % 2 == 1:
15         # ノードの数が奇数個であるとき、最後のノードを複製し合計数を偶数個に変えてください
16         node_list.append(node_list[-1])
17
18     # 隣接するノードをまとめてペアを作成してください
19     pairs = [[node_list[i*2], node_list[i*2 + 1]] for i in range(len(node_list)//2)]
20     return pairs
21
22 # 適当なトランザクションデータを100作成しています。
23 transactions = [str(data).encode() for data in random.sample(range(1000), 100)]
24 transaction_list = [Tree(transaction_data) for transaction_data in transactions]
```

Below the code editor, there's a "コンソール" (Console) window showing the output of the script:

```
[{'hash': '588fcb8b49d039d70f82334b4237bd72ab8a01007402272459b8f8d1deb6374', 'transactions': [<__main__
```

▲「ブロックチェーン発展 III」の画面▲

■価格: 980 円

【Aidemy の概要】

Aidemy は正式公開から 3 ヶ月で会員登録ユーザー数 1 万人以上、コード実行回数 100 万回以上を記録した、日本最大級の先端技術のラーニングサービスです。 <https://aidemy.net/>

1. 10 秒で演習開始 - PC への環境構築は不要で、インターネットブラウザ上でプログラミングができます。
2. 今話題の技術を習得可能 - ディープラーニングや自然言語処理など、いま話題の技術を習得できます。
3. 無料から始められる - 一部の講座は完全無料にてご受講いただけます。



▲Aidemy の演習画面の例(左:コードを書きながら学習する問題, 右:クイズに答えながら学習する問題)▲

【Aidemy の教材の特徴】

1. 業界トップシェア技術を採用 - Python/numpy/pandas/scikit-learn/tensorflow などの技術が学べます。
2. 理論より実践重視 - 難しい数学の知識や理論もできるだけ直感的に理解できるような教材です。
3. 自動採点システム - 書いたプログラムは仮想環境上で自動的に採点されます。



▲Aidemy の教材(左:受講ルートページ, 右:受講コースページ)▲

【株式会社アイデミーについて】

株式会社アイデミー(旧社名 Goods 株式会社)は「社会とテクノロジーをつなぐ。」をミッションとする、2014 年創業のベンチャー企業です。大学での機械学習応用系の研究、クライアント企業のアプリケーション制作・データ解析を経て、2017 年 12 月に「10 秒で始める AI プログラミング学習サービス Aidemy」をリリースしました。Aidemy はサービス開始 3 ヶ月で会員数 1 万名超、コード実行回数 100 万回を突破した日本最大級の先端技術のラーニングサービスです。また、早稲田大学リーディング理工学博士プログラムでの AI 入門特別実践セミナーも担当し、代表取締役 石川聡彦の著書「人工知能プログラミングのための数学がわかる本」が KADOKAWA より 2018 年 2 月に発売されました。こうした事業を通じて、「世界 100 万人規模の先端 IT 人材の不足」という社会課題の解決に貢献して参ります。



▲株式会社アイデミー 代表取締役 CEO 石川 聡彦▲

【株式会社アイデミー概要】

会社名:株式会社アイデミー

所在地:東京都文京区本郷 7-3-1 東京大学アントレプレナープラザ 302

代表者:代表取締役 CEO 石川 聡彦

設立:2014年6月

URL:<https://aidemy.net/>

株主:経営陣, Skyland Ventures, UTEC, エンジェル投資家 他

事業内容:エンジニアのための AI プログラミング学習サービス「Aidemy」の提供



【本件に関する報道関係者、企業からのお問合せ先】

株式会社アイデミー

代表取締役 CEO 石川 聡彦

TEL:03-6868-0998 (平日 9:00-18:00)

e-mail: support@aidemy.net