2023 coding entry challenge

Given a list in json containing a series of objects, return the next most probable words in ascending alphabetical order (limited to at most 5) based on a list of statements.

You may assume that each statement is unique and would have **exactly one solution**.

A word here refers to a logical programmatical variable name.

For a key in the json mapping, the following permutations are possible:

1. *Empty string*: this represents an empty class
2. *Class containing more key-value pairs*: each key represents the variable name and each value represents the type
3. *List of strings*: if the string appears as a key in the list of classes, it would represent a polymorphic type. If the string does not appear as a key in the list of classes, it would represent an enum.

Note: the key-value pairs are case-sensitive.

Example 1:

Input:

```
classes = [
  {
    "Order": {
      "orderId": "String",
      "version": "Long",
      "orderType": "OrderType",
      "orderSide": "OrderSide",
      "status": "Status",
      "allocations": "List<Allocation>"
   }
  },
  {
    "OrderType": [
      "MarketOrderType",
      "LimitOrderType"
   ]
  },
  {
    "MarketOrderType": ""
  },
  {
    "LimitOrderType": {
      "price": "Double"
   }
  },
  {
    "OrderSide": [
```

```
      "Buy",
      "Sell"
   ]
  },
  {
    "Status": [
      "New",
      "Verifying",
      "Pending",
      "Working",
      "PartiallyFilled",
      "Filled",
      "Cancelled"
   ]
  },
  {
    "Allocation": [
      "LongAllocation",
      "EmptyAllocation"
   ]
  },
  {
    "LongAllocation": {
      "clientName": "String"
   }
  },
  {
    "EmptyAllocation": ""
  }
]

statements = [
  "Order.",
  "Order.order",
  "Order.allocations.",
  "Status.P",
  "MarketOrderType."
]
```

Output:

```
{
  "Order.": [
    "allocations",
    "orderId",
    "orderSide",
    "orderType",
    "status"
  ],
  "Order.order": [
    "orderId",
    "orderSide",
```

```
    "orderType"
  ],
  "Order.allocations.": [
    ""
  ],
  "Status.P": [
    "PartiallyFilled",
    "Pending"
  ],
  "MarketOrderType.": [
    ""
  ]
}
```

Explanation:

1. Order is an object containing a list of parameters. The top 5 most probable parameters are returned in alphabetical order.
2. Order.order represents a parameter under the Order object. We want it to return a list of most probable words that could fit, and they are the words that start with order.
3. Since Allocation is a polymorphic type, we do not return any probable words as we do not know which type Allocation belongs to.
4. Status refers to an enum. In this case, there are only 2 possible words that start with *P*.
5. MarketOrderType is an empty class. It does not have any parameters that can be completed.

Challenge submission

Please fork from one of the following templates to implement your challenge submission. There's a sample README.md in each of them to explain what needs to be done.

- Python: https://replit.com/@astitemi/codeit-2023-entry-challenge-python
- Java: https://replit.com/@astitemi/codeit-2023-entry-challenge-java
- Javascript: https://replit.com/@astitemi/codeit-2023-entry-challenge-js

We're evaluating your submission primarily based on correctness!

Do also add any assumptions you might be making so we can take them into consideration based on your understanding of the problem statement.

If you are using other languages, please feel free to consult the sample inputs in the templates and DIY.