

Copilot

Yes, you're correct. It's a good practice to use private attributes for data encapsulation. Here's how you can modify the `InvalidKeyError` class:

```
class InvalidKeyError(ValueError):
    """Exception raised for errors in the input.

    Attributes:
        _message -- explanation of the error
    """

    def __init__(self):
        self._message = "Invalid key in params. Only 'T' and 'D' are allowed."
        super().__init__(self.message)

    @property
    def message(self):
        return self._message
```

Now, the `_message` attribute is private, and it can be accessed using the `message` property. This approach provides better data encapsulation and makes your code more robust. Good suggestion! 👍