CSC148, Winter 2023
Assignment 1

This code defines four classes; Contract,MTMContract, TermContract and PrepaidContract. These classes represent contracts for a phone line for month-to-month, term-based, prepaid types and prepaid phone line contract.

'Contract' is an abstract class representing a phone contract.

- public attributes:
    1. 'start' is a 'datetime.date' object representing the starting date of the contract.
    2. 'bill' is an optional 'Bill' object representing the bill for the last month of call records loaded from the input dataset.
- Public methods
    1. '__int__(self, start: date time.date) -> None' is an initializes a new 'Contract' object with the given starting date.
    2. 'new_month(self, month: int, year: int, bill: Bill) -> None' is an advanced to a new month in the contract and sets the appropriate rate per minute and fixed cost. This method is abstract and must be implemented by subclasses.
    3. 'bill_call(self, call: Call) -> None' adds a given call to the bill.
    4. 'cancel_contract(self) -> float' returns the amount owed in order to close the phone line associated with this contract.

'MTMContract' is a subclass of 'Contract' representing a month to month phone contract.

- public attributes:
    1. 'start' is a 'datetime.date' object representing the starting date of the contract.
    2. 'end' is an optional 'datetime.date' object representing the ending date of the contract
    3. 'bill' is an optional 'Bill' object representing the bill for the last month of call records loaded from the input dataset.
- Public methods:
    1. '__int__(self, start: date time.date) -> None' is an initializes a new 'MTMContract' object with the given starting date and no bill or ending date
    2. 'new_month(self, month: int, year: int, bill: Bill) -> None' is an advanced to a new month in the contract and sets the appropriate rate per minute and fixed cost.

‘TermContract’ is a subclass of ‘Contract’ representing a term phone contract.

- public attributes:
    1. ‘start’ is a ‘datetime.date’ object representing the starting date of the contract.
    2. ‘end’ is an optional ‘datetime.date’ object representing the ending date of the contract
    3. ‘bill’ is an optional ‘Bill’ object representing the bill for the last month of call records loaded from the input dataset.
- private attributes:
    1. ‘_free’ is an integer representing the number of free minute in the contract per month
    2. ‘_curr’ is a ‘datetime.date’ object representing the current date in the contract. It used for keeping tract of how free minutes have been used.

-public methods:

1. ‘__int__(self, start: date time.date) -> None’ initialized a new ‘TermContract’ object with the given starting date and no bill or ending date
2. ‘new_month(self, month: int, year: int, bill: Bill) -> None’ advanced to a new month in the contract and sets the appropriate rate per minute and fixed cost.

‘PrepaidContract’ is a subclass of ‘Contract’ representing a prepaid phone contract.

- Public attributes:
    1. ‘start’ is a ‘datetime.date’ object representing the starting date of the contract.
    2. ‘end’ is an optional ‘datetime.date’ object representing the ending date of the contract
    3. ‘bill’ is an optional ‘Bill’ object representing the bill for the last month of call records loaded from the input dataset.
- public methods:
    1. ‘__int__(self, start: date time.date) -> None’ initializes a new ‘PrepaidContract’ object with the given starting date, a starting balance, and no bill or ending date.
    2. ‘new_month(self, month: int, year: int, bill: Bill) -> None’: advances to a new month in the contract and sets the appropriate rate per minute and fixed cost. For prepaid contracts, this method also deducts the fixed cost from the balance.
    3. ‘add_funds(self, amount: float) -> None’: adds the given amount to the current balance of the prepaid contract.

Overall, these classes provide a framework for modeling phone contracts of different types and keeping track of their associated bills and usage.