# Contract Classes API

Class Variable
Type Method

`TermContract`, `MTMContract`, and `PrepaidContract` are inherits from `Contract` class, so all of them have common public attributes: `start(datetime.date)` and `bill(Optional[Bil])` and common public methods: `new_month`, `bill_call`, and `cancel_contract`.

## TermContract

`TermContract` is a class for term contracts. This class has its unique public attributes: `current(datetime.date)`, `end(datetime.date)`, and `deposit(float)`. To instantiate `TermContract`, you have to send two arguments which are `start` date and `end` date, the type of those arguments must be `datetime.date` and the `end` date can not be earlier than `start` date.

To call `new_month`, you have to send 3 arguments: `month(int)`, `year(int)`, and `bill(Bill)`. The `month` and `year` can not be earlier than the `start` date. When `new_month` is called, it will replace its own `bill` to the given `bill` object and set rates on term contract's minute cost then set `current` to the given `month` and `year`.If `new_month` is called in the same month and year as `start` date, it will charge a term deposit and monthly fee into that month's bill, and charge only term contract's monthly fee on following months. also `current` will reset to current date.

To call `bill_call`, you have to send 1 argument: `call(Call)`. when `bill_call` is called, it will add used free minutes on its bill first and add billed minutes after using all of the free minutes.

To call `cancel_contract`, you do not have to send any argument. if `cancel_contract` is called after the end date, it will return the rest of the deposit that was subtracted by that month's bill(negative return value indicates that customer gets a refund and positive indicates that customer has to pay.) if `cancel_contract` is called before the end date, it will return that month's bill and the deposit will be forfeited.

# MTMContract

`MTMContract` is a class for month-to-month contracts. This class does not have its unique public attributes. To instantiate `MTMContract`, you have to send one argument which is `start` date, the type of this argument must be `datetime.date`.

To call `new_month`, you have to send 3 arguments: `month`(`int`), `year`(`int`), and `bill`(`Bill`). The `month` and `year` can not be earlier than the `start` date. When `new_month` is called, it will replace its own `bill` to the given `bill` object and set rates on month-to-month contract's minute cost and charge a month-to-month contract's monthly fee.

To call `bill_call`, you have to send 1 argument: `call`(`Call`). when `bill_call` is called, it will add billed minutes in its `bill.`

To call `cancel_contract`, you do not have to send any argument. When `cancel_contract` is called, it will return that month's bill.

# PrepaidContract

`PrepaidContract` is a class for term contracts. This class has its unique public attributes: `balance`(`float`). To instantiate `PrepaidContract`, you have to send two arguments which are `start` date and `balance`, the types of those arguments must be `datetime.date` and `float` respectively, and the `balance` can not be negative

To call `new_month`, you have to send 3 arguments: `month`(`int`), `year`(`int`), and `bill`(`Bill`). The `month` and `year` can not be earlier than the `start` date. When `new_month` is called it will replace its own `bill` to the given `bill` object and set rates on prepaid contract's minute cost. If `new_month` is called for the first time it will add the prepaid contract's monthly fee, and from the second time it is called, it will create a new bill with the rest of month's balance and save the rest of month's balance on its `balance` and set rates on prepaid contract's minute cost.

To call `bill_call`, you have to send 1 argument: `call`(`Call`). when `bill_call` is called, it will add billed minutes in its `bill.`

To call `cancel_contract`, you do not have to send any argument. if `cancel_contract` is called when there is balance left, it will return 0 and if `cancel_contract` is called when there is no balance left and customer have to pay, it will return that month's bill

# Implementation of methods

## TermContract

The TermContract has public attributes:

- start (datetime.date): it is used to know when is the start date of the contract
- bil (Optional[Bill]): it is optional because when the contract is just made it is None. it indicates current month's bill
- current (datetime.date): it is used to track the current to date so that we can figure out if the current date is after the end date of the contract or not.
- end (datetime.date): it is used to know when is the end date and compared to current date.
- deposit (float): it is used to save deposits in the class.

### __init__

This method is used to instantiate TermContract class objects.
This method has 2 parameters: start and end both of them are datetime.date type and these are used to know the start date of the contract and end date of the contract.
It calls its parent class' initializer and input the given values into its public attributes.

### new_month

This method will be used to replace previous bill to new bill and set the given to properties of the contract.
This method takes 3 parameters: month(datetime.date), year(datetime.date), and bill(Bill). The month and year is used to know if it is the first month of the contract or not and the bill is used to replace last month's bill with the current month's bill.
First, replace its bil attribute to the given bill object. Secondly, it checks if it is the first month of the contract or not and if it is the first month, it adds the sum of term deposit and term contract's monthly fee in its bill's fixed cost. Lastly, it sets the rate and type of new bill to the term contract's minute cost and its type respectively and reset the current attribute to the given month year, the day parameter is 1 because it was mentioned that the billing cycle starts on the first day of the month.

### bill_call

This method is used to add billed minutes to its bill depending on the contract's properties.
This method takes 1 parameter which is call(Call) this parameter will be used to get the duration of the given call.
Firstly, it converts given duration to minutes because the duration of the call is in seconds.
Secondly, it checks if the sum of used free minutes and given minutes are greater or equal to

100. if it is, it adds billed minutes that difference of rest of free minutes and given minutes, and sets its bill's used free minutes to 100. if it is less than 100, it just simply adds given minutes to its bill's billed minutes.

### cancel_contract

This method is used to cancel the contract and return a certain value depending on the contract's properties.
This method takes no parameters at all
It checks if the contract is canceled after the end date and if it is, it returns the rest of the balance and the negative return value means that customer gets a refund and positive return value means that customer still has to pay its bill. it is canceled before the end date, it will just return that month's bill.

## MTMContract

The MTMContract has public attributes:

- start (datetime.date): it is used to know when is the start date of the contract
- bil (Optional[Bill]): it is optional because when the contract is just made it is None. it indicates current month's bill

### __init__

This method is used to instantiate MTMContract class objects.
It calls its parent class' initializer and input the given values into its public attributes.
This method takes only 1 parameter start (datetime.date). It inputs the given value to its attributes and sets the bill attribute to None.

### new_month

This method will be used to replace previous bill to new bill and set the given to properties of the contract.
This method takes 3 parameters: month(datetime.date), year(datetime.date), and bill(Bill). The month and year will not be used at all and the bill is used to replace last month's bill with the current month's bill.
This method replaces the previous bill to the given bill and sets the rates and charges a monthly fee to its bill. And it just sets the rates of its new bill to month-to-month contract's minute rate.

### bill_call

This method is used to add billed minutes to its bill depending on the contract's properties.
This method takes 1 parameter which is call(Call) this parameter will be used to get the duration of the given call.

it just simply charges given minutes to billed minutes of its bill.

## cancel_contract

This method is used to cancel the contract and return a certain value depending on the contract's properties.
This method takes no parameters at all
it simply sets its start date to None and returns the current month's bill.

# PrepaidContract

The PrepaidContract has public attributes:

- start (datetime.date): it is used to know when is the start date of the contract
- bil (Optional[Bill]): it is optional because when the contract is just made it is None. it indicates current month's bill
- deposit (float): it is used to track the rest of the deposit in the contract and negative value of deposit means that the customer still have credits

## __init__

This method is used to instantiate PrepaidContract class objects.
This method has 2 parameters: start(datetime.date) and deposit(float). start is not going to be used in this class and balance is used to set the contract's balance attribute.
It calls its parent class' initializer and input the given values into its public attributes, and sets its balance attribute to the negative value of the given balance because the negative value means that the customer still has the credits.

## new_month

This method will be used to replace previous bill to new bill and set the given to properties of the contract.
This method takes 3 parameters: month(datetime.date), year(datetime.date), and bill(Bill). The month and year will not be used at all and the bill is used to replace last month's bill with the current month's bill.
Firstly, it checks if the bill attribute is None or not to know if it is the first time this method is called or not. if it is, it simply replaces the bill and adds the prepaid contract's monthly fee into its bill's fixed cost. if it is not, it will keep the track of the rest of the balance and add 25 credits if the balance is less than 10, and it replaces the previous bill to new bill than it adds its balance into its bill's fixed cost because if the fixed cost is balance which is negative, we can get rest of the balance when we use get_cost method. At last, it just set the rates of its new bill to prepaid contract's minute rate.

## bill_call

This method is used to add billed minutes to its bill depending on the contract's properties.
This method takes 1 parameter which is call(Call) this parameter will be used to get the duration of the given call.
it just simply charges given minutes to billed minutes of its bill.

## cancel_contract

This method is used to cancel the contract and return a certain value depending on the contract's properties.
This method takes no parameters at all
It checks if the customer still has balance or not if the customer doesn't have balance, it will return the positive value which indicates that the customer still has to pay the given amount and if it is not it will return 0 because the rest of the balance will be forfeited.