Challenge 1: Implement a Banking Account

In this challenge, you will implement a Banking Account using the concepts of inheritance.

WE'LL COVER THE FOLLOWING

- Problem Statement
  - Task 1
  - Task 2
  - Task 3
- Coding Exercise

Problem Statement

Implement a basic structure of a *parent class*, `Account`, and a *child class*, `SavingsAccount`.

Task 1

Implement properties as **instance variables** and set them to `None` or 0.

`Account` has the following *properties*:

- `title`
- `balance`

`SavingsAccount` has the following *properties*:

- `interestRate`

Task 2

Create an **initializer** for `Account` class. The order of parameters should be the following:

```
Account("Mark", 5000)
```

where `Mark` is the `title` and `5000` is the account balance.

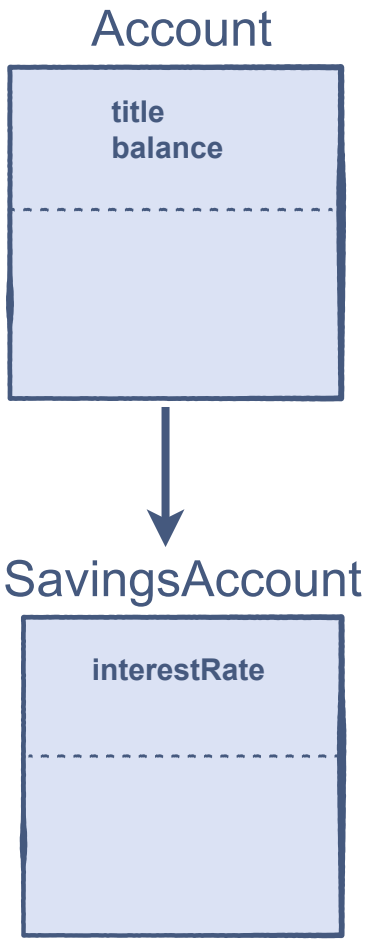
Task 3

Implement properties as **instance variables** and set them to `None` or 0.

Create an **initializer** for `SavingsAccount` class using the initializer of the `Account` class in the order below:

```
Account("Mark", 5000, 5)
```

where `Mark` is the `title` and `5000` is the `balance` and `5` is the `interestRate`.



Based and Derived Classes Structure

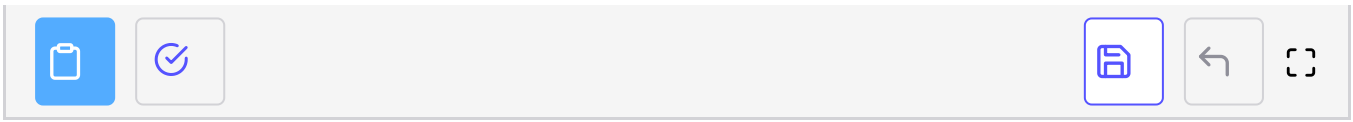
Coding Exercise

First, take a close look and design a step-by-step algorithm before trying the implementation. This problem is designed for your practice, so initially try to solve it on your own. If you get stuck, you can always refer to the solution provided in the solution review.

Good luck!

```
class Account:
    def __init__(self):
        # write your code here
        pass


class SavingsAccount():
    def __init__(self):
        # write your code here
        pass
```



The solution will be explained in the next lesson.