LendInvest Coding Test

Thank you for your interest in LendInvest. In this stage of the recruitment journey, you're going to code (the fun part!).

Exercise

In LendInvest we think everyone should have the opportunity to invest in property, which is why we're disrupting the status quo. This is lending and investing without the banks. Mortgages simplified. We connect people who want to invest their money with investments to those who want to borrow.

One of the important parts of our business is to give our investors a way to invest in a loan for them to earn a return (monthly interest payment).

Model

- Each of our loans has a start date and an end date.
- Each loan is split into multiple tranches.
- Each tranche has a different monthly interest percentage.
- Each tranche has a maximum amount available to invest. So once the maximum is reached, further investments can't be made in that tranche.
- As an investor, I can invest in a tranche at any time if the loan is still open, the maximum available amount was not reached and I have enough money in my virtual wallet.
- At the end of the month we need to calculate the interest each investor is due to be paid.

Scenario

- Given a loan (start 01/10/2023 end 15/11/2023).
- Given the loan has 2 tranches called **A** and **B** (**3%** and **6%** monthly interest rate) each with **1,000** pounds amount available.
- Given each investor has **1,000** pounds in his virtual wallet.
- "***Investor 1***" would like to invest **1,000** pounds in the tranche "**A"** on **03/10/2023**: this is allowed and the software should go on without errors.
- "***Investor 2***" would like to invest **1** pound in the tranche "**A"** on **04/10/2023**: the maximum amount for the tranche A is 1000, the investor 2 should not be allowed to invest - "***Investor 3***" would like to invest **500** pounds on the tranche "**B"** on **10/10/2023**: this is allowed and the software should go on without errors
- "***Investor 4***" would like to invest **1,100** pounds in the tranche "**B**" **25/10/2023**: the investor 4 does not have enough money to invest the requested amount, and the tranche is smaller than the amount requested.

- On **01/11/2023** the system runs the interest calculation for the period **01/10/2023** -> **31/10/2023**:
  - "***Investor 1***" earns **28.06** pounds
  - "***Investor 3***" earns **21.29** pounds

What we care about

Working code is not our only priority, we aim to write maintainable, clear to read and testable code. So please :

- Use OOP at your best designing a nice architecture and implement it keeping SOLID in mind, using - if necessary - design patterns, inheritance and all the nice features that OOP makes available and may serve your purpose, without over-engineering your solution.

- Write unit tests, we use PHPUnit, but feel free to use the tool you're most familiar with.

- Write tests for the provided scenarios

How you can solve the challenge:

- We don't need a front end or an api layer. This is a domain modeling exercise. Please write your classes and show they work testing the scenarios and writing unit tests.

- Use plain php, this exercise is not about knowing a specific framework but please feel free to use the testing framework you prefer.

- We don't need a database, don't bother with it.

- Please use at least PHP 8.0

Some Hints for the coding challenge:

The investor earnings are calculated as follows:

Daily interest rate = Interest rate / Days in a month
Invested period interest rate = Daily interest rate * Days invested
Earned interest = Invested amount / 100 * Invested period interest rate (1)

(1) for instance, for investor1 it's from the 3/10/2023 (included) to the end of month, so 31/10/2023

How to send us the challenge

Please, don't publish your challenge on Github, Bitbucket etc. Rather create a git bundle and send it via email:

The command to create a bundle is :

```
git bundle create lendinvest-test-YOURNAME.bundle master
```

To clone it once created (may you want to double check everything is

bundled): `git clone endinvest-test-YOURNAME.bundle -b master`