Pair Programming Task

This task allows us to have some hands-on experience seeing you develop code, problem solve, and work as a team with someone from BJSS. To prepare, please write some code that can be used as a basis for pair programming.

For the interview, you should have prepared a solution which runs and acts as a starting point. It is important to remember: very little functionality in a working codebase is better than something near-fully functional which doesn't compile or run.

The Task

Buying a house in the UK incurs a tax which varies according to the price of the house and circumstances of the purchaser e.g., a higher rate is paid if you are buying a second home. In Scotland this is called Land and Buildings Transaction Tax (LBTT). We need you to write a piece of code which, when given the price of a house, will calculate the LBTT due to be paid.

A key part of any technical project is gaining understanding in unfamiliar business domains, for this reason we do not include how to calculate LBTT as part of this specification. LBTT can be complex to calculate as it varies according to the circumstances of the buyer so, for simplicity, you should assume:

- The buyer of the house currently owns a property and lives in it as their main residence.
- The buyer does not conduct any kind of business activity from their house i.e., it is purely for personal use.
- The buyer does not own any other properties.
- At the end of the purchase the buyer will still only own one house i.e., they intend to sell their current home at the same date they buy the new one.

Changes to how this tax is calculated have been made in recent years. There is information available on both the old and new calculations. We are happy for you to use whichever calculation you prefer. You will not be penalised in any way for whichever version you implement.

Expectations of the code

We are looking for you to prepare a starting point with which we can do some pair programming to add functionality. At a minimum the code should:

- Compile without errors
- Include unit tests which are runnable (either via an IDE or command line)
- Include unit tests which pass

To assist in keeping the functionality as simple as possible, the unit tests do not need to cover all possible scenarios, they should simply pass for the scenario they cover in your code e.g., if you only

have code which works for a certain price range of house then only this needs to be covered in the tests.

For the pair programming to be achievable in a small-time frame we ask you do not move far beyond this minimum e.g., adding a UI or other ancillary capabilities. The pair programming will focus on adding more logic to the LBTT calculation. Anything extra will be a wasted effort on your part or could get in the way of achieving the task.