Unit 6: OOP & TypeScript – Exercise 1: Bank Accounts

Assigned: N/A

Due: N/A

Completed: 12/14/22

Google Docs:

<https://docs.google.com/document/d/1UVoB2bhhRX4R1lF8g7qbMSESi2Arej_TlbNFXEPpV6g/preview>

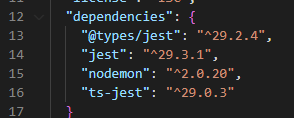
We started off by creating a JSON file with ‘npm init --yes’ (used --yes to skip the questions)



We then installed multiple packages, which can be done simply with spaces and added libraries to jest, ts-jest, @types/jest, and nodemon



These will now appear in the JSON file under dependencies



We then installed TypeScript globally with ‘npm install -g typescript’



We then added a TypeScript compiler with ‘npx tsc --init’



We finally added this and updated our JSON file “scripts” “test” to “jest”

Text

Description automatically generated

**Skills:** Classes, extended classes, and methods

Create a class called "BankAccount"

*Properties:*

* balance (number)
* interestRate (number)

*Methods:*

* addInterest() // increases the balance by adding the appropriate interest. returns void.

Next, create a subclass called "BankAccountWithFee"

It inherits all the properties and methods from BankAccount and adds the following.

*Properties:*

* fee (number) // a fixed dollar amount fee

*Methods:*

* applyFee() // subtracts the fee from the balance. returns void.

Create a few instances (a.k.a. objects) of each class, and test them out by calling the methods.

**BankAccount.ts**

class BankAccount {

    balance:number;

    interestRate:number;

    constructor(balance:number, interestRate:number) {

        this.balance = balance;

        this.interestRate = interestRate;

    }

    addInterest() : void {

        this.balance += this.balance \* this.interestRate;

    }

}

export {BankAccount}

**BankAccount.test.ts**

import { BankAccount } from "./BankAccount"

describe('Bank Account Tests', () => {

    test('Add 10% interest to $100 balance', () => {

        // arrange

        // instantiate your objects and setup you dependencies

        let testBankAccount = new BankAccount(100, 0.1);

        // You don't necessarily have to do this, but here we did an 'expect'

        // to confirm that our initial balance is correct

        expect(testBankAccount.balance).toBe(100);

        // Called the function

        testBankAccount.addInterest();

        // Did it work?

        expect(testBankAccount.balance).toBe(110);

    })

})

To run our test we run either ‘npm test’ or ‘npx jest’ and the results are the same.



Text

Description automatically generated