Lab 6 - Interfaces

Preparations

If you couldn't finish the previous exercise, you can copy and paste the pervious solution from the *labsSolutions* folder.

In this Lab you are going to create interfaces and improve the code base to make it more type safe.

Exercise 1: Create an `Iban` interface

1. Create an `Iban` interface. The interface should describe the shape returned by the `generateIban` function.
2. Use the interface as the return type of `generateIban`
3. Make sure that there are no compile errors.

Play around with the `Iban` interface. What happens if you add a property to it? What happens when you remove the explicit return type from the `Iban` function? Can you explain what's happening here?

Exercise 2: Create a `Customer` interface

1. Create an interface for a `Customer`. It should have a `firstName` (string), `lastName` (string) and an `insertion` (string). The `insertion` should be optional.
2. Make sure the `formatName` method accepts this new interface.
3. Test the code to see if it formats a name correctly.

Exercise 3: Create a `BackAccount` interface

1. Create an interface for a `BankAccount`. It should have a `customer` field (type `Customer`) and a `iban` field (type `Iban`)
2. Add a function `createBankAccount` with the following shape:

   ```
   function createBankAccount(customer: Customer): BankAccount {
       // TODO
   }
   ```

   It should create a new `Iban` using `generateIban` and return a new `BankAccount`.
3. Create some bank accounts in an array:

   ```
   const bankAccounts = [
       createBankAccount({firstName: 'Alfred', lastName: 'Kwak', insertion: 'Jodocus'}),
       createBankAccount({firstName: 'Brad', lastName: 'Pit'}),
       createBankAccount({firstName: 'Jack', lastName: 'Sparrow'})
   ]
   ```

4. Can you guess what the type of the `bankAccounts` variable is? Verify this by hovering over it in your code editor.

Exercise 4: If time permits

Add a method `toString` to the bank accounts created with the `createBankAccount` method. Use the `formatName` and `formatIban` functions to display them in the following order:

```
[NL15 TYPE 7608 1718] Alfred Jodocus Kwak
[NL65 TYPE 5016 0769] Brad Pit
[NL76 TYPE 3727 8486] Jack Sparrow
```

Hint You might need to add the `toString` method to the `BankAccount` interface.