# TS Advanced: Regular Exam - 18.01.20

# Problem 1. Bank Office

export class BankOffice {  
 *// TODO: implement this class...*  
}

### Your Task

### Write a generic BankOffice class, which takes two types (T and V) which via generic constraints extend the Customer and Contract interfaces and supports the described functionality below. First import the interfaces.

Note: T extends Customer and V extends Contract

### Functionality

#### constructor()

Receives 1 parameter at initialization of the class, and should have 1 private property:

* **officeName** - string
* **allCustomers** - initially an **empty array** of type **T**

#### newCustomer({customer})

The customer is of type T

This function should create a new customer and add it to the customers array with all the data passed as arguments in an object.

* Check if the customer is already in the array you should throw an error:

"{firstName} {lastName} is already your customer”

#### presentContract({customerID}, {contract})

The contract is of type V, customerID - number

* Check if the given id corresponds to any customer in the customers array, if not throw a new error:

"There is no customer with {id}"

* Otherwise you should **add** the **properties** of the **contract** object to the properties of the customer with the customer ID from the arguments of the method.

#### hasComplaints({customerID}, {complaints})

The customer is of type number, complaints - string or array of strings

* Check if the given id corresponds to any customer, if not throw a new error:

"Invalid customer"

* If the **complaints** are a string, just add a property **complaints** to the found customer. If the complaints are an array add them to the found customer in a property complaints and as value an object in the following format:

#### "1: {complaint}, 2: {complaint}, ... n: {complaint}"

#### showCustomerFile({customerID})

The customerID is of type number

* Check if the given id corresponds to any customer in the array, if not throw a new error:

#### "Invalid customer"

* Otherwise:
  + Check if the customer has any complaints. If he has **no** complaints and has not presented a contract, print the following message:

**"Office: {officeName}**

**Customer name: {firstName} {lastName}**

**Customer ID: {customerID}"**

* + If the customer has **no** complaints but has presented a contract print the following message:

**"Office: {officeName}**

**Customer name: {firstName} {lastName}**

**Customer ID: {customerID}**

**Valid contract: {validContract}**

**Salary: {salary}**

**Contract terms: {terms}"**

* + If the customer has **1** complaint and has not presented a contract, print the following message:

**"Office: {officeName}**

**Customer name: {firstName} {lastName}**

**Customer ID: {customerID}**

**Complaints: {complaints}"**

* + If the customer has **1** complaint but has presented a contract print the following message:

**"Office: {officeName}**

**Customer name: {firstName} {lastName}**

**Customer ID: {customerID}**

**Valid contract: {validContract}**

**Salary: {salary}**

**Contract terms: {terms}**

**Complaints: {complaints}"**

* + If the customer has **more than 1** complaint and has not presented a contract, print the following message:

**"Office: {officeName}**

**Customer name: {firstName} {lastName}**

**Customer ID: {customerID}**

**Complaints:**

**1. {complaints}**

**2. {complaints}**

**...**

**n. {complaints}"**

* + If the customer has **more than 1** complaint but has presented a contract print the following message:

**"Office: {officeName}**

**Customer name: {firstName} {lastName}**

**Customer ID: {customerID}**

**Valid contract: {validContract}**

**Salary: {salary}**

**Contract terms: {terms}**

**Complaints:**

**1. {complaints}**

**2. {complaints}**

**...**

**n. {complaints}"**

### Examples

|  |
| --- |
| Sample code usage |
| **let office = new BankOffice();**  **office.newCustomer({ firstName: 'Ivan', lastName: 'Ivanov', id: 1 })**  **office.presentContract({ validContract: true, salary: 3000, terms: 4 }, 1);**  **office.hasCompaints(1, ['Not enough money', 'Stolen money from account']);**  **console.log(office.showCustomerFile(1));** |
| Corresponding output |
| Office: Moskovska  Customer name: Ivan Ivanov  Customer ID: 1  Valid contract: true  Salary: 3000  Contract terms: 4  Complaints:  1. Not enough money  2. Stolen money from account |