**JS Advanced: 30.08.2018**

**Problem 3. Vacationer (Simple Class)**

*The travel agency decides that they want to keep some information about their clients who get registered on their website. Your task for this additional functionality is to implement a simple class about the average vacationer.*

Write a JavaScript class **Vacationer** that keeps information about a **vacationer** that is registered on the travel agency’s website. The class must keep the vacationer’s **fullName**, **idNumber**, **creditCard** and **wishList**. You must name your properties as listed above.

|  |
| --- |
| **class Vacationer {**  **// TODO: implement this class**  **}** |

**Class Properties**

The class must have the following **properties**:

* **fullName** – an **object** with the following format:

|  |
| --- |
| **{  firstName: String,**  **middleName: String,**  **lastName: String }** |

* **idNumber** – a **String**.
* **creditCard** – an **object** with the following format:

|  |
| --- |
| **{  cardNumber: Number,**  **expirationDate: String,**  **securityNumber: Number }** |

* **wishList** – an **array**, which keeps **Strings** that contain the desired destinations.

**Class Implementation**

The class constructor receives the **fullName** of the vacationer and an **optional object parameter** that describes the **creditCard** details. If the **creditCard** parameter is **missing**, you must use the following **default values**: **cardNumber** – 1111, **expirationDate** – “”, **securityNumber** – 111. You must initialize the **wishList** property as an empty array. You must call the **generateIDNumber()** function in the constructor in order to set the value for the **idNumber.**

Implement the following features:

* Accessor **fullName** – used to get and set the value for the full name of the vacationer. The **input** for the set accessor comes as an **array of Strings**.  
  A full name is considered **valid** when it consists of **exactly 3 names**. Also each name must start with a **single capital Latin letter**, followed by **lower-case Latin letters only**. This is an example of a valid name – Ivan Ivanov Ivanov. This is an example of an invalid name – Iv@n Ivanov.   
  First of all, in case that the number of given names is wrong, throw an error: "**Name must include first name, middle name and last name"**. Secondly, in case that a name with a wrong format is given, throw the following error: **"Invalid full name"**.
* Function **generateIDNumber()** – used to set the value for the id of the vacationer.   
  The id number must be generated individually for each vacationer using the following formula:

**231 \* firstName’s first letter’s ASCII code + 139 \* middleName length**

If the **last name** of the vacationer **ends with a vowel** ("a", "e", "o", "i", "u"), add **8** to the end of the above calculated id number. Otherwise, add **7**.

* Function **addCreditCardInfo(input)** – updates the details for the vacationer’s credit card. The **input** comes an **array of Strings**.  
  In case that **less arguments** are received, throw an error: **"Missing credit card information"**. However, if you received enough arguments, but the **cardNumber** and the **securityNumber** are not **numbers**, throw the error: **"Invalid credit card details"**.
* Function **addDestinationToWishList(destination)** – adds a destination to the **wishList** array. The **destination** is always a single **string** – either the name of a city, or the name of a country. Every time a destination is added to the wishList, the wishList must be **sorted** by destinations **names’ length** in **ascending** order (starting from the destination with the shortest name to the one with the longest name).   
  If a destination that already exists in the wish list is to be added, throw the error: **"Destination already exists in wishlist"**.
* Function **getVacationerInfo()** – returns a String that contains information about the vacationer in the following format. If the wish list is empty, print **“empty”**. If there are destinations added, print them all joined with **“, ”**. The format is:

**“Name: {firstName} {middleName} {lastName}  
 ID number: {idNumber}  
 Wishlist:   
 empty/destinations, joined with ‘, ’  
 Credit Card:  
 Card Number: {cardNumber}  
 Expiration Date: {expirationDate}  
 Security Number: {securityNumber}”**

Scroll down to see **examples**.

**Examples**

|  |
| --- |
| **Sample code usage** |
| // Initialize vacationers with 2 and 3 parameters  let vacationer1 = new Vacationer(["Vania", "Ivanova", "Zhivkova"]);  let vacationer2 = new Vacationer(["Tania", "Ivanova", "Zhivkova"],  [123456789, "10/01/2018", 777]);  // Should throw an error (Invalid full name)  try {  let vacationer3 = new Vacationer(["Vania", "Ivanova", "ZhiVkova"]);  } catch (err) {  console.log("Error: " + err.message);  }  // Should throw an error (Missing credit card information)  try {  let vacationer3 = new Vacationer(["Zdravko", "Georgiev", "Petrov"]);  vacationer3.addCreditCardInfo([123456789, "20/10/2018"]);  } catch (err) {  console.log("Error: " + err.message);  }  vacationer1.addDestinationToWishList('Spain');  vacationer1.addDestinationToWishList('Germany');  vacationer1.addDestinationToWishList('Bali');  // Return information about the vacationers  console.log(vacationer1.getVacationerInfo());  console.log(vacationer2.getVacationerInfo()); |
| **Corresponding output** |
| Error: Invalid full name  Error: Missing credit card information  Name: Vania Ivanova Zhivkova  ID Number: 208398  Wishlist:  Bali, Spain, Germany  Credit Card:  Card Number: 1111  Expiration Date:  Security Number: 111  Name: Tania Ivanova Zhivkova  ID Number: 203778  Wishlist:  empty  Credit Card:  Card Number: 123456789  Expiration Date: 10/01/2018  Security Number: 777 |

**Submission**

Submit **only** your class **Vacationer**.