JS Advanced: Retake Exam - 10.12.19

Exam problems for the "JavaScript Advanced" course @ SoftUni. Submit your solutions in the SoftUni Judge system at https://judge.softuni.bg/Contests/Compete/Index/.

Problem 3. Christmas Dinner

```
class ChristmasDinner {
   // TODO: implement this class...
```

Your Task

Write a **ChristmasDinner class**, which supports the described functionality below.

Functionality

constructor()

Should have these 4 properties:

- **budget** number comes from the constructor
- dishes empty array
- products empty array
- guests empty object

Important! Use accessors to validate that the budget (the budget cannot be negative number). If the class is initialized with negative budget throw an error with the following message:

"The budget cannot be a negative number"

shopping([product])

In this method you should go shopping for the needed products to prepare for the dinner. The **product** will come as an array that has the type of the product and the price for it

If you don't have enough money to buy the product, you should throw an error

```
"Not enough money to buy this product"
```

Otherwise you should push the type of the product in the products array, decrease the budget with the price of the product and return a message:

"You have successfully bought {type}!"

















recipes({recipe})

In this method you will receive a recipe in the following format: { recipeName: string, productsList: array of strings }.

The products require to prepare the meal are in contained the **productsList**. If all the products from the **productsList** are present in the products array you should push an object in the following format: { recipeName, productsList } to the dishes array and return a message:

"{recipeName} has been successfully cooked!"

If one or more products needed to make the meal are absent from the products, throw an error

"We do not have this product"

inviteGuests(name, dish)

In this method you would invite guests to the dinner. The name is string and so is the dish

If the dish is not present in the dishes array, throw a new error

"We do not have this dish"

And if the guest is already present in the guest object, throw a new error:

"This guest has already been invited"

Otherwise create a **new key and value pair** to the guest object with key the **name** of the guest and value - the dish in the format: { {guestName}: {dish} } and return a message:

"You have successfully invited {name}!"

showAttendance()

This function should return a sting with all the guest, the dish they are having and the products the dish is made of, separated by a comma and space (product, product) in the following format:

```
"{name} will eat {dish}, which consists of {products}
"{name} will eat {dish}, which consists of {products}
. . . "
```

Submission

Submit only your ChristmasDinner class.















Examples

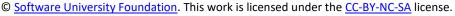
This is an example how the code is **intended to be used**:

```
Sample code usage
let dinner = new ChristmasDinner(300);
dinner.shopping(['Salt', 1]);
dinner.shopping(['Beans', 3]);
dinner.shopping(['Cabbage', 4]);
dinner.shopping(['Rice', 2]);
dinner.shopping(['Savory', 1]);
dinner.shopping(['Peppers', 1]);
dinner.shopping(['Fruits', 40]);
dinner.shopping(['Honey', 10]);
dinner.recipes({
    recipeName: 'Oshav',
    productsList: ['Fruits', 'Honey']
});
dinner.recipes({
    recipeName: 'Folded cabbage leaves filled with rice',
    productsList: ['Cabbage', 'Rice', 'Salt', 'Savory']
});
dinner.recipes({
    recipeName: 'Peppers filled with beans',
    productsList: ['Beans', 'Peppers', 'Salt']
});
dinner.inviteGuests('Ivan', 'Oshav');
dinner.inviteGuests('Petar', 'Folded cabbage leaves filled with rice');
dinner.inviteGuests('Georgi', 'Peppers filled with beans');
console.log(dinner.showAttendance());
```

Corresponding output

Ivan will eat Oshav, which consists of Fruits, Honey

















Petar will eat Folded cabbage leaves filled with rice, which consists of Cabbage, Rice, Salt, Savory

Georgi will eat Peppers filled with beans, which consists of Beans, Peppers, Salt



















