**2. Shady Car Dealership**

class ShadyCarDealership {

    //TODO...

}

Write a class **ShadyCarDealership**, which implements the following functionality:

**Functionality**

**Constructor**

Should have these **4** properties:

* **dealerName –** String
* **availableCars –** Array
* **soldCars –** Array
* **revenue –** **default: 0**

**At the initialization of the ShadyCarDealership class,** the **constructor** accepts the **dealerName.** The **revenue** has a **default value of 0!** The rest of the properties must be **empty!**

**Hint:** You can add more properties to help you finish the task.

#### addCar (model, year, mileage, price) - This method should add a new car to the retailer. The method accepts 4 arguments:

* If any of the following requirements is **NOT fulfilled**, an **error** with the following message should be **thrown**: **"Invalid car!"**
  + **Model** – non-empty string;
  + **Year** – number **more** or **equal** 1950;
  + **Mileage** – positive number;
  + **Price** – positive number;

**Hint**: Zero is also a positive number.

* Otherwise, you should **add the car**, with properties: **{model, year, mileage, price}** to the **availableCars** array and **return**:

**"New car added: {model} ({year}) / {mileage} km. - {price}$"**

* When **returning** the result, the **Price** mustbe **rounded to the second decimal point!**

#### sellCar (model, desiredYear) – This method should search for a car with the given model in the availableCars array, and then sell it. Accepts 2 arguments.

* If a car with the given **model** cannot be found, an error with the following message should be **returned**:

**"{model} was not found!"**

* + If you **find the car** **with the given model**, you should look up its **year**. The buyer has a simple request. They are looking for a car with a year that is more recent or equal to their **desired year**. To ensure the sale of the car you must make a bargain:
  + If the found car’s year is **more recent** than or **equal** to the **desiredYear** – the price stays the same!
  + If the **difference** between the car’s **year** and the **desiredYear** is **5** years or **less** – the price gets deducted by **10%**!
  + If the **difference** between the car’s **year** and the **desiredYear** is **more** than **5** years – the price gets deducted by **20%**!
* You should **remove** the car from the **availableCars** array and **add** it to the **soldCars** array in the following format: **{model, year, mileage, soldPrice}**
* Finally, you must add the **soldPrice** to the **revenue** and **return**:

**"{model} ({year}) was sold for {soldPrice}$"**

**Note: soldPrice** must be **rounded** to the second decimal point!

#### prepareCarForSale (model) - This method should prepare a car for sale by cleaning it and reducing its mileage.

Accepts 1 argument.

* If a car with the given model **cannot be found** in the **availableCars** array, the following message should be **returned**:

**"{model} was not found for preparation!"**

* If you **find** the car with the given model, you should:
* **Reduce** the **mileage** by **50%**
* **Increase** the **price** by **30%**
* **Return** the updated car details in the format:

**"{model} ({year}) is prepared for sale with {mileage} km. - {price}$"**

**Note: price** mustbe **rounded** to the second decimal point!

**Note: mileage** and **price** mustbe **updated** to **availableCars** array!

#### salesJournal (criteria) – This method accepts 1 argument. It should sort the sold cars, based on a given criteria. The two possible criteria are – "year" or "model".

* If the given criteria **do not match** either of the possible criteria, an **error** with the following message should be **thrown**:

**"Invalid criteria!"**

* If the given criteria is **"year"** – the sold **cars** must be **sorted** by their **year** in **descending** **order**;
* If the given criteria is **"model"** – the sold cars must be **sorted alphabetically** by their **model**;
* Finally, **return** **all sorted** sold cars **separated** by **a new line** in format:

**"{DealerName} has a total income of {revenue}$**

**{soldCarsCount} cars sold:**

**{model} ({year}) / {mileage} km. / {price}$**

**{model} ({year}) / {mileage} km. / {price}$ "**

**…**

**Note: revenue and price must be rounded to the second decimal point!**

**Example**

|  |
| --- |
| **Input 1** |
| const dealership = new ShadyCarDealership('Shady Motors');  console.log(dealership.addCar('Honda CR-V', 2010, 120000, 15000));  console.log(dealership.addCar('VW Golf', 2011, 130000, 12000));  console.log(dealership.addCar('BMW X3', 2005, 220000, 9000));  console.log(dealership.addCar('Toyota Yaris', 2015, 80000, 18000)); |

|  |
| --- |
| **Output 1** |
| New car added: Honda CR-V (2010) / 120000 km. - 15000.00$  New car added: VW Golf (2011) / 130000 km. - 12000.00$  New car added: BMW X3 (2005) / 220000 km. - 9000.00$  New car added: Toyota Yaris (2015) / 80000 km. - 18000.00$ |

|  |
| --- |
| **Input 2** |
| const dealership = new ShadyCarDealership('Shady Motors');  console.log(dealership.addCar('Honda CR-V', 2010, 120000, 15000));  console.log(dealership.addCar('VW Golf', 2011, 130000, 12000));  console.log(dealership.addCar('BMW X3', 2005, 220000, 9000));  console.log(dealership.addCar('Toyota Yaris', 2015, 80000, 18000));  console.log(dealership.prepareCarForSale('Honda CR-V'));  console.log(dealership.prepareCarForSale('Honda Jazz')); |

|  |
| --- |
| **Output 2** |
| New car added: Honda CR-V (2010) / 120000 km. - 15000.00$  New car added: VW Golf (2011) / 130000 km. - 12000.00$  New car added: BMW X3 (2005) / 220000 km. - 9000.00$  New car added: Toyota Yaris (2015) / 80000 km. - 18000.00$  Honda CR-V (2010) is prepared for sale with 60000 km. - 19500.00$  Honda Jazz was not found for preparation! |

|  |
| --- |
| **Input 3** |
| const dealership = new ShadyCarDealership('Shady Motors');  console.log(dealership.addCar('Honda CR-V', 2010, 120000, 15000));  console.log(dealership.addCar('BMW X3', 2005, 220000, 9000));  console.log(dealership.addCar('Toyota Yaris', 2015, 80000, 18000));  console.log(dealership.prepareCarForSale('Honda CR-V'));  console.log(dealership.prepareCarForSale('BMW X3'));  console.log(dealership.sellCar('Honda CR-V', 2012));  console.log(dealership.sellCar('BMW X3', 2012));  console.log(dealership.sellCar('Toyota Yaris', 2012)); |

|  |
| --- |
| **Output 3** |
| New car added: Honda CR-V (2010) / 120000 km. - 15000.00$  New car added: BMW X3 (2005) / 220000 km. - 9000.00$  New car added: Toyota Yaris (2015) / 80000 km. - 18000.00$  Honda CR-V (2010) is prepared for sale with 60000 km. - 19500.00$  BMW X3 (2005) is prepared for sale with 110000 km. - 11700.00$  Honda CR-V (2010) was sold for 17550.00$  BMW X3 (2005) was sold for 9360.00$  Toyota Yaris (2015) was sold for 18000.00$ |

|  |
| --- |
| **Input 4** |
| const dealership = new ShadyCarDealership('Shady Motors');  console.log(dealership.addCar('Honda CR-V', 2010, 120000, 15000));  console.log(dealership.addCar('VW Golf', 2011, 130000, 12000));  console.log(dealership.addCar('BMW X3', 2005, 220000, 9000));  console.log(dealership.prepareCarForSale('Honda CR-V'));  console.log(dealership.prepareCarForSale('BMW X3'));  console.log(dealership.sellCar('Honda CR-V', 2012));  console.log(dealership.sellCar('BMW X3', 2012));  console.log(dealership.salesJournal('model')); |

|  |
| --- |
| **Output 4** |
| New car added: Honda CR-V (2010) / 120000 km. - 15000.00$  New car added: VW Golf (2011) / 130000 km. - 12000.00$  New car added: BMW X3 (2005) / 220000 km. - 9000.00$  Honda CR-V (2010) is prepared for sale with 60000 km. - 19500.00$  BMW X3 (2005) is prepared for sale with 110000 km. - 11700.00$  Honda CR-V (2010) was sold for 17550.00$  BMW X3 (2005) was sold for 9360.00$  Shady Motors has a total income of 26910.00$  2 cars sold:  BMW X3 (2005) / 110000 km. / 9360.00$  Honda CR-V (2010) / 60000 km. / 17550.00$ |