Problem 3. SoftUni Parking

I. Project Structure

For this problem you should create a new package named **"softUniParking",** which should hold inside the two classes **both Car and Parking.** The Main class can also be inside this package however it is not a must it may also be outside the package. Your project structure should look like that:

Pay attention to name the package, all the classes, their fields and methods exactly the same way they are presented in the following document. It is also important to keep the project structure as described above.

II. Car

Create Java class **`Car that has the following structure:`**

```
public class Car {
    // TODO: implement this class
}
```

1. Fields

- **make: String**
- **model: String**
- **horsePower: int**
- **registrationNumber: String**

The class **constructor** should receive all the fields parameters (**make, model, horsePower and registrationNumber)**.

2. Methods:

- Method **toString()** which returns the information about a single Car object in the following format:

" Make: {make}"
" Model: {model}"
" HorsePower: {horse power}"
"RegistrationNumber: {registration number}

III. Parking

Write a Java class **Parking** that has **data field**, which stores objects of type Car with a corresponding unique **registration number**.

```
public class Parking {
    // TODO: implement this class
}
```

1. Fields

- **cars** – Map<String, Car>
- **capacity** - accessed only by the base class (responsible for the parking capacity).

The class **constructor** receive **capacity(int)** and should initialize the **cars** with a new **Map** instance**,** and set the value for the capacity.

2. Methods

- Method **addCar(Car car)** – first checks if there is already a car with tha provided car registration number and if there is the method returns the following message:
  `"Car with that registration number, already exists!"`
  Next checks if the count of the cars in the parking is more or equals than the capacity and if it is returns the following message:
  `"Parking is full!"`
  Finally if nothing from the previous conditions is true it just adds the current car to the cars in the parking and returns the message:
  `"Successfully added new car {Make} {RegistrationNumber}"`
- Method **removeCar(string registrationNumber)** – removes a car with the givven registration number. If the provided registration number does not exist returns the message:
  `"Car with that registration number, doesn't exists!"`
  Otherwise, removes the car and returns the message:
  `"Successfully removed {registrationNumber}"`
- Method **getCar(string registrationNumber)** – returns the **Car** with the provided registration number
- Method r**emoveSetOfRegistrationNumber(List<string> registrationNumbers)** – removes all cars having the provided registration numbers and procceeds the same way as the **RemoveCar()** method
- Method **getCount()** – returns the number of stored Car objects.

Examples

This is an example how the **Parking** class is **intended to be used**. Make sure to comment out the parts that throw an error!

Sample code usage

```
public static void main(String[] args) {

        //Initialize the Parking
        Parking parking = new Parking(5);

        //Initialize Car
        Car car = new Car("Skoda", "Fabia", 65, "CC1856BG");

        //Initialize second Car object
        Car car2 = new Car("Audi", "A3", 110, "EB8787MN");

        System.out.println(car.toString());
        //Make: Skoda
        //Model: Fabia
        //HorsePower: 65
        //RegistrationNumber: CC1856BG

        System.out.println(parking.addCar(car));
        //Successfully added new car Skoda CC1856BG

        System.out.println(parking.addCar(car));
        //Car with that registration number, already exists!

        System.out.println(parking.addCar(car2));
        //Successfully added new car Audi EB8787MN

        System.out.println(parking.getCar("EB8787MN").toString());
        //Make: Audi
        //Model: A3
        //HorsePower: 110
        //RegistrationNumber: EB8787MN

        System.out.println(parking.removeCar("EB8787MN"));
        //Successfullyremoved EB8787MN

        System.out.println(parking.getCount()); //1

        //Initialize listOfRegistrationNumbers
        List<String> regNums = new ArrayList<>();

        //add two registrationNumbers
```

```
        regNums.add("EB8787MN");
        regNums.add("invalid");

        // call method removeSetOfRegistrationNumber with our list
        parking.removeSetOfRegistrationNumber(regNums);

        System.out.println(parking.getCount()); //1
}
```

Submission

- Submit **single .zip file**, containing **repository package, with the two classes inside (Car and Parking) and the Main class**, there is no specific content required inside the Main class e. g. you can do any kind of local testing of you program there. However there should be **main(String[] args)** method inside: