



FACULTATEA DE
MATEMATICĂ
ȘI **INFORMATICĂ**

Modelarea Sistemelor Software

Student: Ciprian - Mihai Ceașescu

Grupa: 506

Specializarea: Inginerie Software



Cuprins

- Problemă
- Diagrama de clase
- Diagrama de secvență
- Design Patterns
- Frameworks
- Demo
- Concluzii - Why UML?



Problemă

PROJECT #1

Proiectați o parcare pe următoarele ipoteze:

1. Parcare are mai multe niveluri.
2. În parcare puteți parca motociclete, autoturisme și autobuze.
3. Parcare are locuri destinate celor 3 categorii de vehicule.
4. O motocicletă poate parca în oricare dintre cele 3 locuri.
5. O mașină poate parca fie într-un loc de mașină, fie într-un singur loc de autobuz.
6. Un autobuz poate parca în cinci locuri destinate autobuzelor care sunt consecutive și pe același rând.

Diagrama de clase

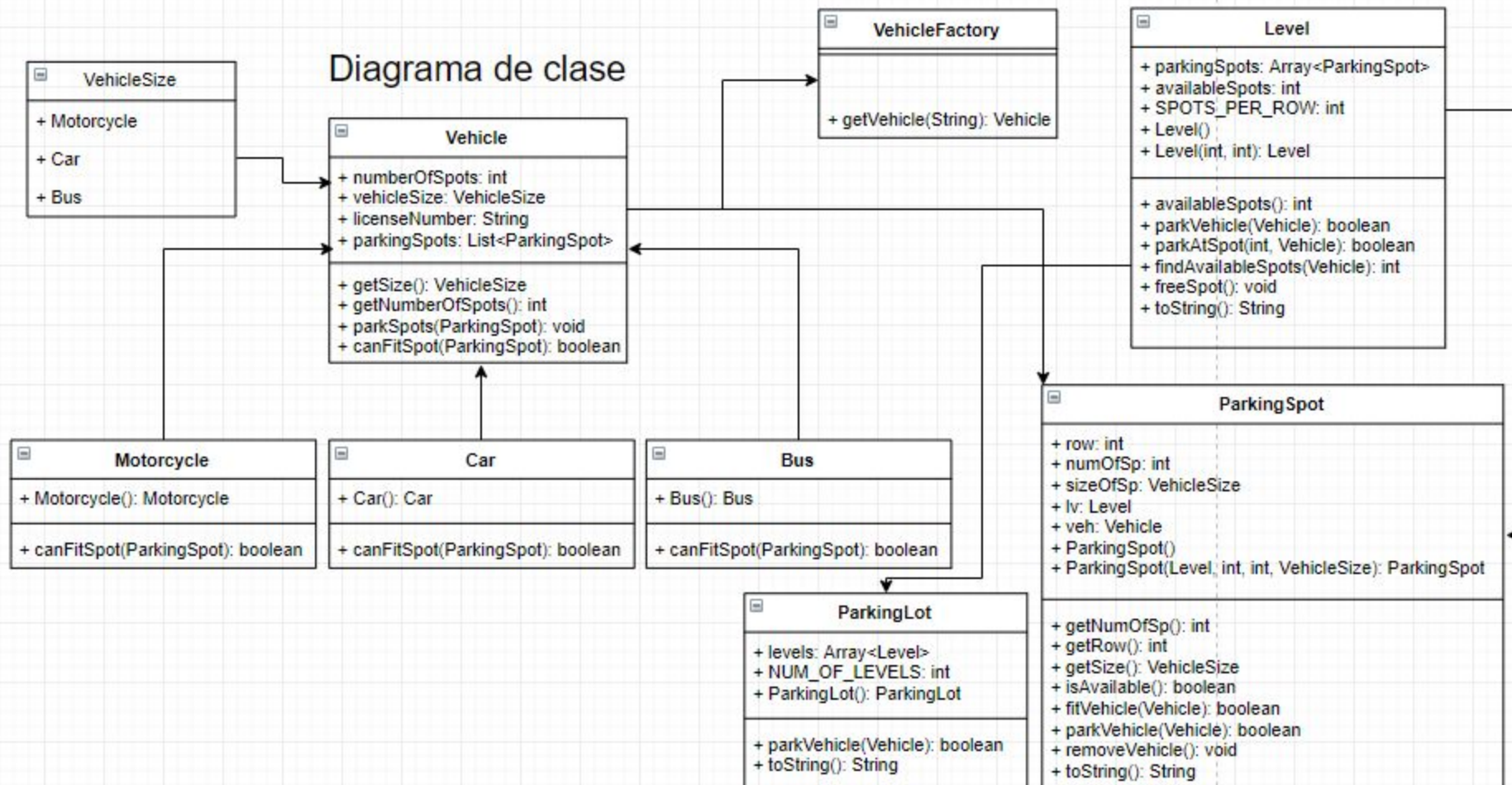
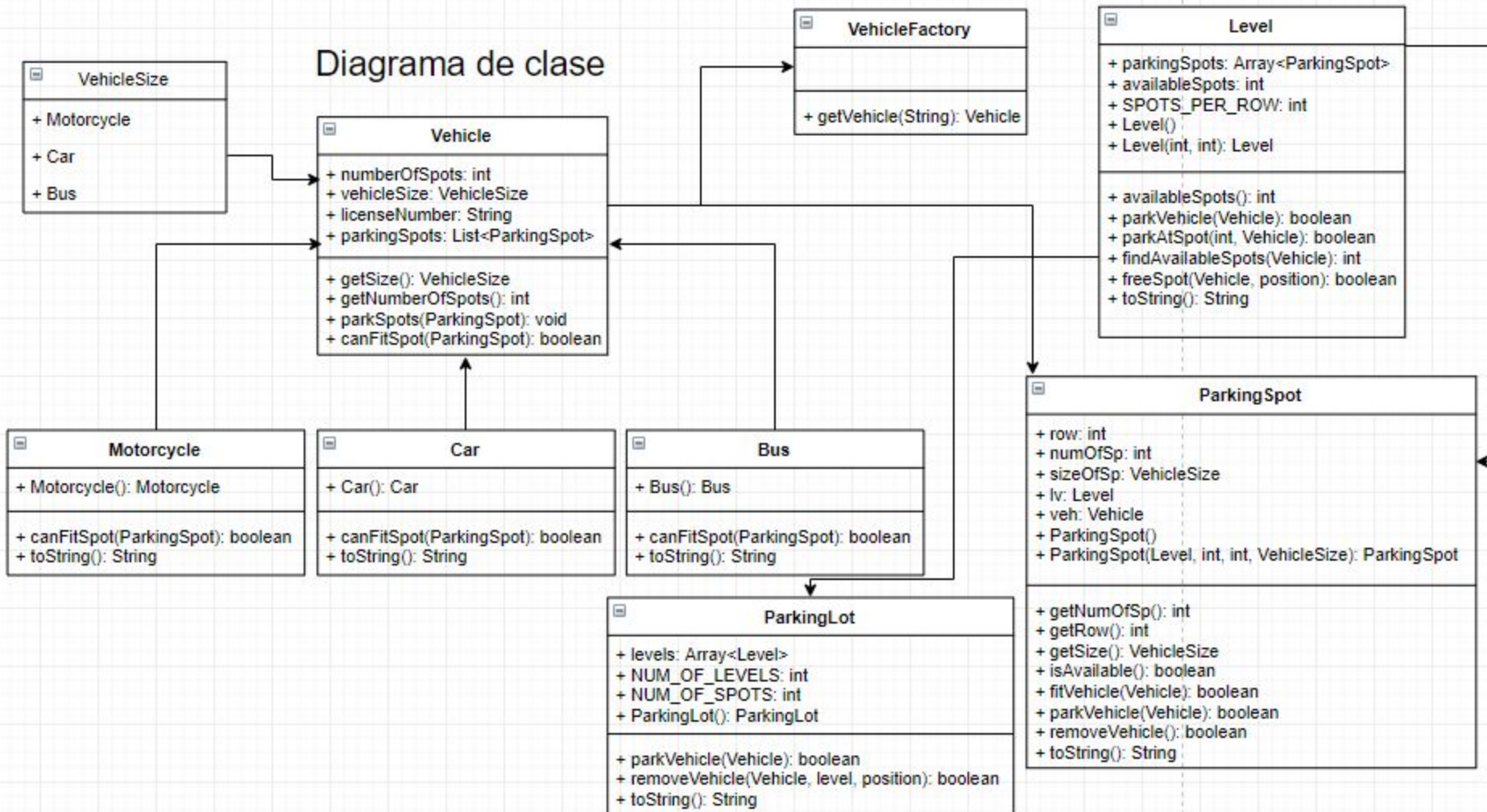


Diagrama de clase



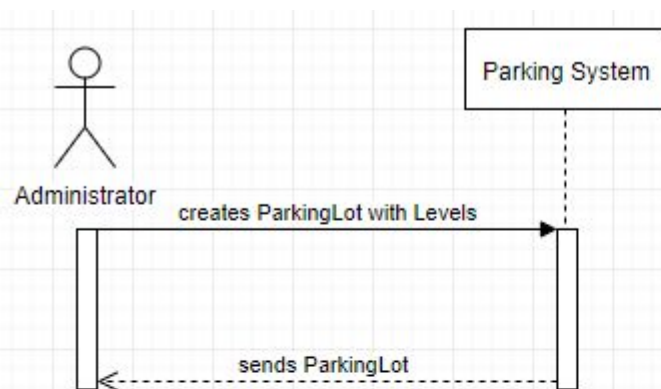
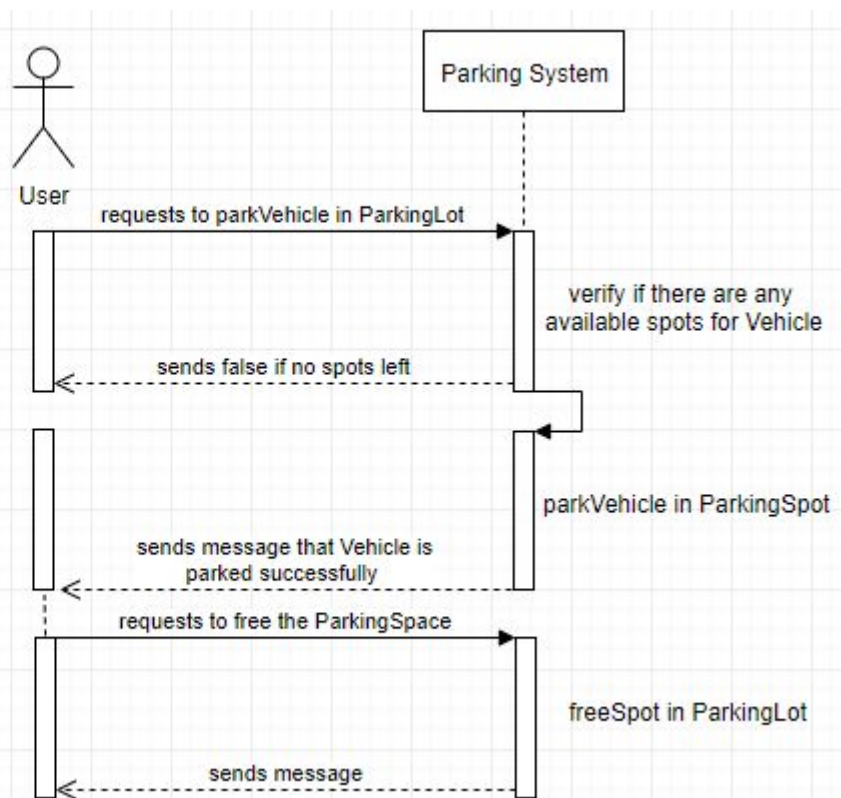


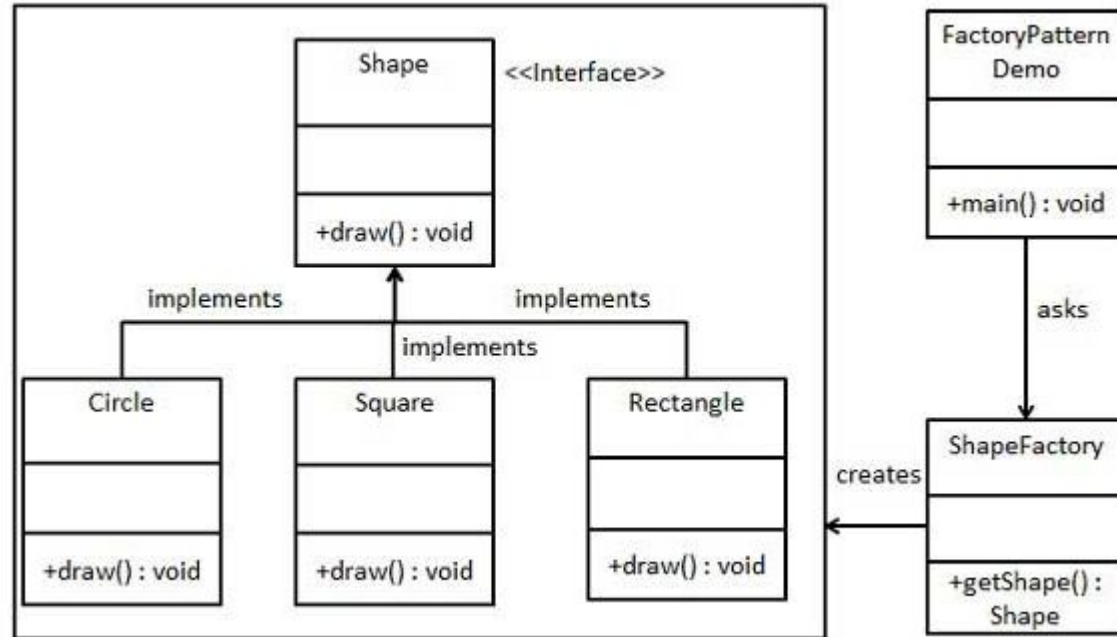
Diagrama de secvență



Design Patterns

Factory Design Pattern - este unul dintre cele mai utilizate modele de design, oferind o modalitate ușoară de a crea un obiect.

Instanțierea acestor obiecte, se realizează fără a expune logica utilizatorului, folosind o interfață comună.





Frameworks

Lombok - este un framework pentru limbajul Java care generează automat metode importante pentru diverse clase create, și anume:

- Getters
- Setters
- Constructors with no arguments
- Constructors with all arguments
- Builder design pattern



Frameworks

Class.java file:

```
import lombok.*;

@Setter
class Student {
    private String firstname;
    private String lastname;
    private String id;
    private int age;
    private String pin;
}
```



Frameworks

Class.class file:

```
class Student {  
    private String firstname;  
    private String lastname;  
    private String id;  
    private int age;  
    private String pin;  
  
    Student() {  
    }  
  
    public void setFirstname(String firstname) { this.firstname = firstname; }  
  
    public void setLastname(String lastname) { this.lastname = lastname; }  
  
    public void setId(String id) { this.id = id; }  
  
    public void setAge(int age) { this.age = age; }  
  
    public void setPin(String pin) { this.pin = pin; }  
}
```



Demo

Proiectul realizat este încărcat pe GitHub:

<https://github.com/cciprianmihai/-Modeling-of-Software-Systems.git>



Concluzii - Why UML?

- Standard - un design foarte cunoscut.
- Flexibil - elemente de modelare pentru domeniul vizat.
- Modele portabile - XMI format - suportat de diverse tool-uri.
- Diferite tipuri de diagrame - se pot folosi doar o parte din ele.
- Arhitectura proiectului este importantă.



Vă mulțumesc!