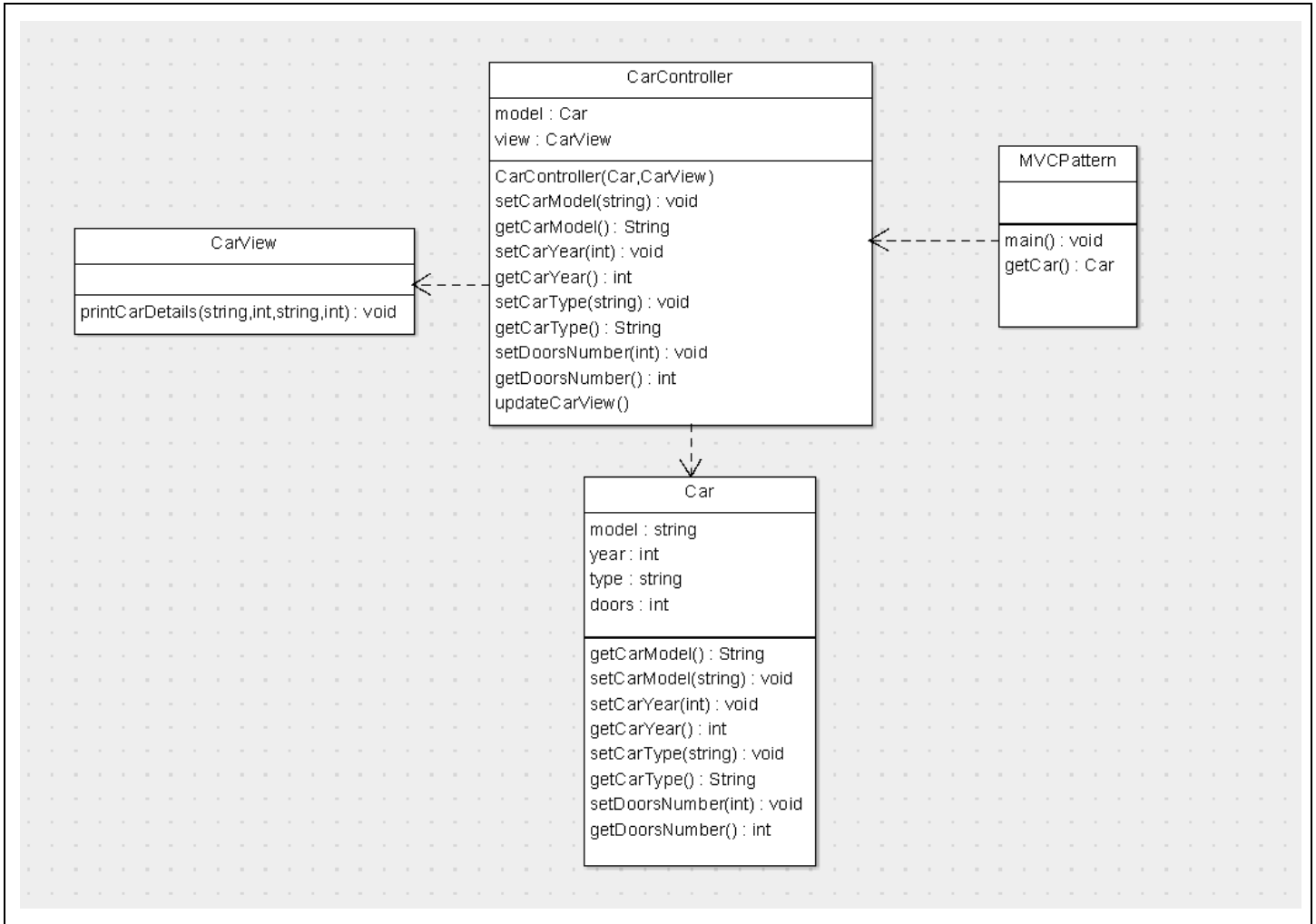


SUBIECT 5

Fie urmatoarea diagrama UML:



Sa se implementeze ierarhia de clase astfel incat codul `main()` descris mai jos:

```
void main()
{
    MVCPattern example = MVCPattern();
    example.main();
    system("pause");
}
```

Si codul main din MVCPattern:

```
void MVCPattern::main()
{
    Car model = this->getCar();
    CarView view = CarView();
    CarController controller = CarController(&model, &view);
    controller.updateCarView();
    controller.setCarModel("Jaguar");
    controller.updateCarView();
    controller.setCarType("XF");
    controller.updateCarView();
    controller.setCarYear(2015);
    controller.updateCarView();
    controller.setDoorsNumber(5);
    controller.updateCarView();
}
```

Sa afiseze:

```
-----
Car:
Model: Dacia | Year: 2000 | Type: 1310 | Doors Number: 4
-----
Car:
Model: Jaguar | Year: 2000 | Type: 1310 | Doors Number: 4
-----
Car:
Model: Jaguar | Year: 2000 | Type: XF | Doors Number: 4
-----
Car:
Model: Jaguar | Year: 2015 | Type: XF | Doors Number: 4
-----
Car:
Model: Jaguar | Year: 2015 | Type: XF | Doors Number: 5
```

Barem:

- 5p - Aranjarea codului in headere si fisiere cpp corespunzătoare.
- 3p - Scrierea clasei CarView
- 3p - Scrierea clasei MVCPattern
- 9p - Scrierea clasei CarController
- 5p - Scrierea clasei Car