OCL

Lab support material

Where do we start?

- If you have not done it yet, download USE (get link from MOODLE)
- Model a class diagram with USE
  - Use a common text editor for this
  - You have a small example in the end of the lecture slides for the syntax
    - You also have a few snippets to get you started (see next slides)
  - You also have USE's official documentation on the USE web site
- Finally, add the OCL constraints

File   Edit   Selection   Find   View   Goto   Tools   Project   Preferences   Help

FOLDERS
▼ CienciaViva
  ▶ Backup
  ▶ old
    adivinha.py
    condicoes.py
    consumo.py
    Dinossauros.py
    Ecoador
    exemplo.py
    pedra.py
    README.md

Marriage.use   ×   Sudoku.use   ×   Aeroporto.use   ×

```
 1  model Aeroporto
 2
 3  -- classes
 4
 5  class Airline
 6  attributes
 7    name : String
 8    profitRate: Real
 9  end
10
11  class Flight
12  attributes
13    cost : Real
14    number : Integer
15  operations
16    addPassenger(p: Passenger)
17  end
18
19  class Aircraft
```

FOLDERS
▼ CienciaViva
  ▶ Backup
  ▶ old
    adivinha.py
    condicoes.py
    consumo.py
    Dinossauros.py
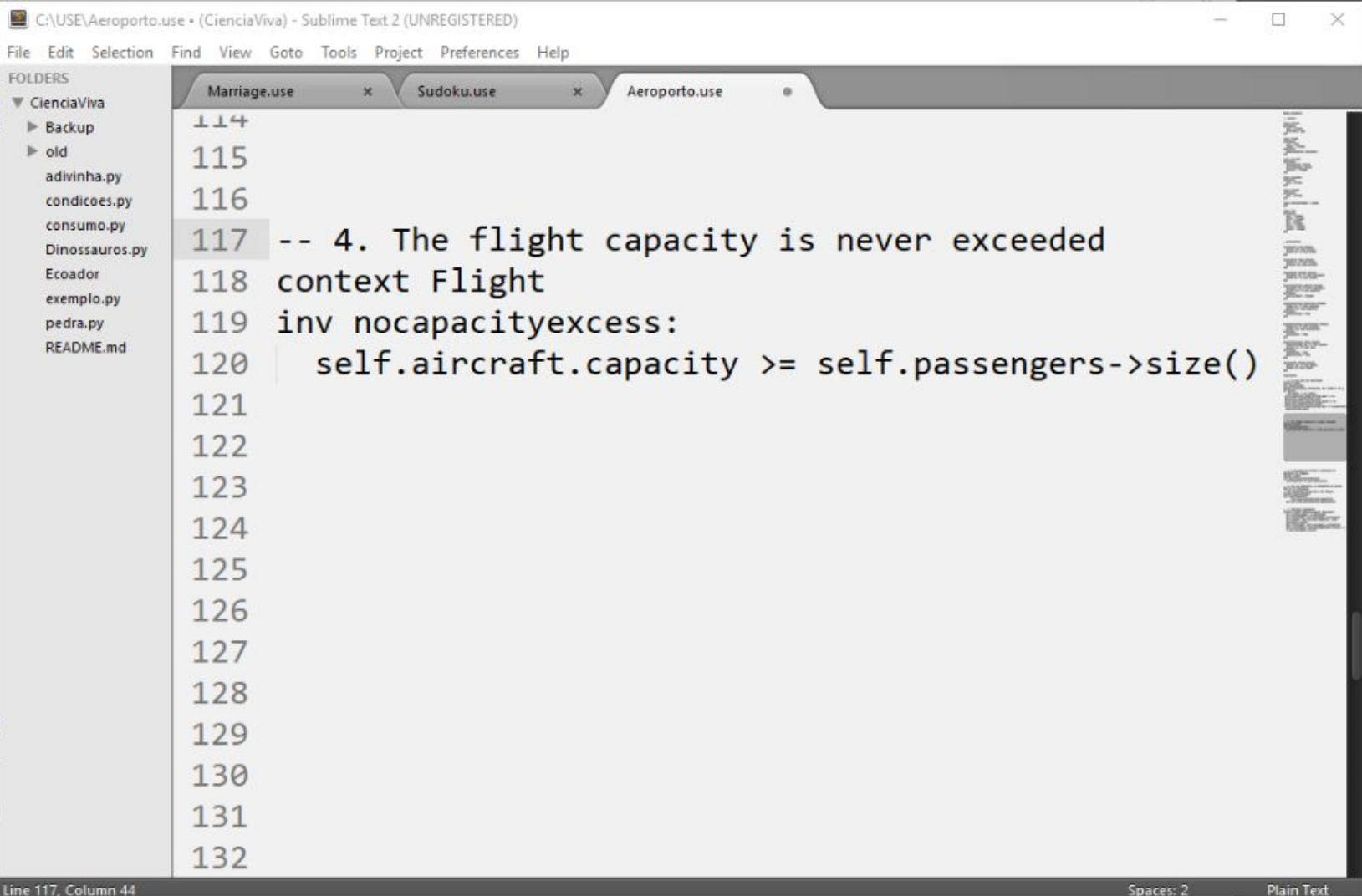   Ecoador
    exemplo.py
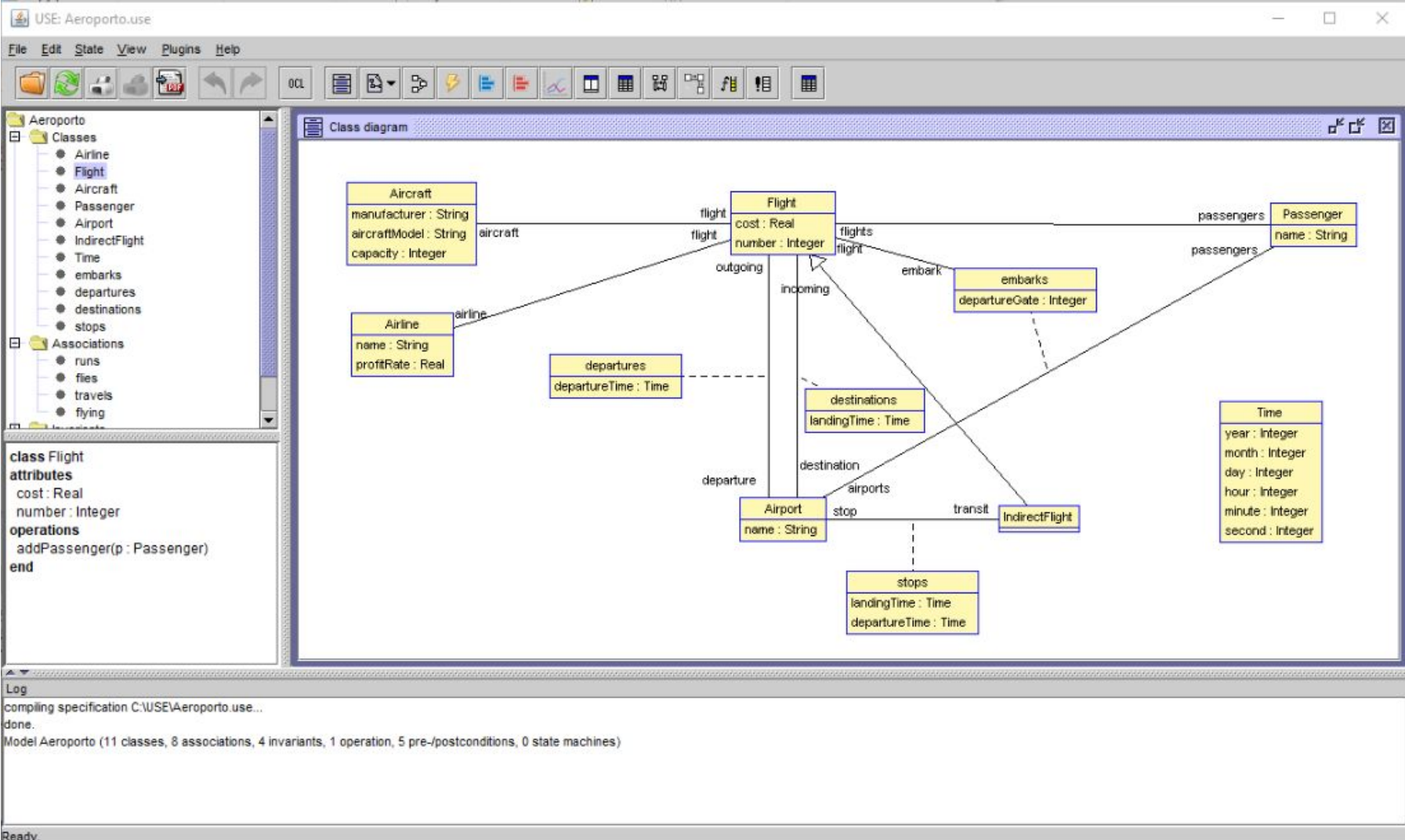   pedra.py
    README.md

Tabs: Marriage.use | Sudoku.use | Aeroporto.use

```
58  association flies between
59    Flight [1..*] role flight
60    Aircraft [1] role aircraft
61  end
62
63  association travels between
64    Passenger [0..*] role passengers
65    Flight [1..*] role flights
66  end
67
68  associationclass embarks between
69    Passenger [0..*] role passengers
70    Airport [1..*] role airports
71  attributes
72    departureGate : Integer
73  end
74
75  associationclass departures between
76    Flight [0..*] role outgoing
```

FOLDERS
▼ CienciaViva
  ▶ Backup
  ▶ old
    adivinha.py
    condicoes.py
    consumo.py
    Dinossauros.py
    Ecoador
    exemplo.py
    pedra.py
    README.md

Marriage.use   ×   Sudoku.use   ×   Aeroporto.use   ●

```
114
115
116
117  -- 4. The flight capacity is never exceeded
118  context Flight
119  inv nocapacityexcess:
120    self.aircraft.capacity >= self.passengers->size()
121
122
123
124
125
126
127
128
129
130
131
132
```

A model for the airport example

A model for the SUDOKU problem

File   Edit   State   View   Plugins   Help

Sudoku
- Classes
  - Board
  - Row
  - Column
  - Square
  - Field
- Associations
- Invariants
  - Field::allowedValue
  - Row::uniqueValuesRow
  - Column::uniqueValuesColumn
  - Square::uniqueValuesSquare
- Pre-/Postconditions
- Query Operations

Class diagram


```
Log
compiling specification C:\USE\Sudoku.use...
done.
Model Sudoku (5 classes, 7 associations, 4 invariants, 4 operations, 0 pre-/postconditions, 0 state machines)
```

Ready.