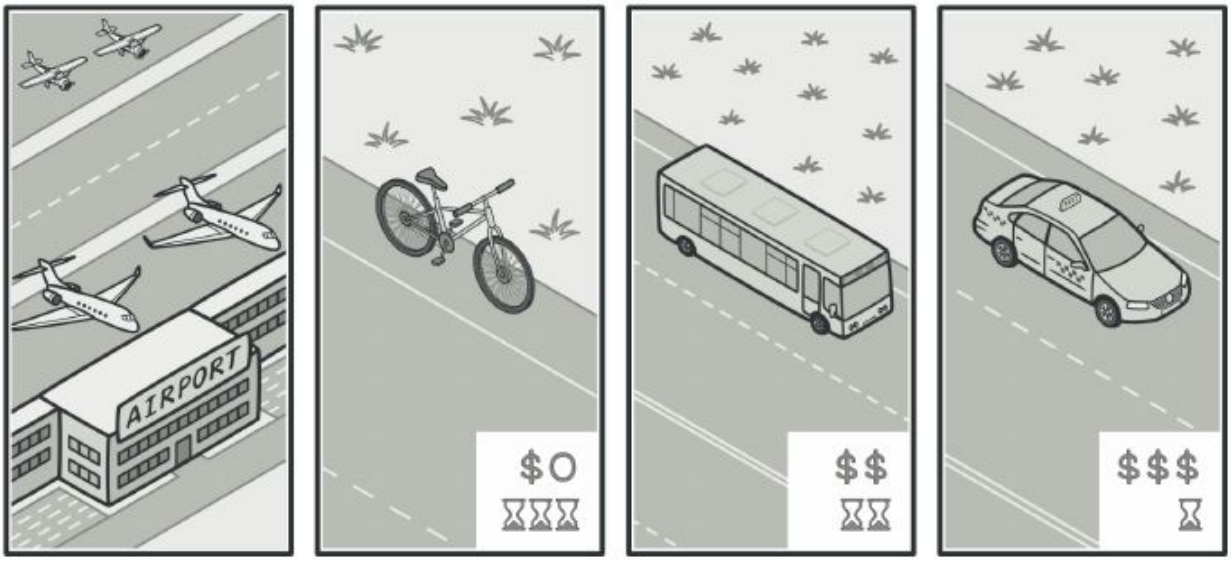
UNIVERSIDAD NACIONAL DE COLOMBIA
INGENIERÍA DE SOFTWARE II
Practice 3

Professor: Yury Niño Roa

1. Imagine that you have to get to the airport. You can catch a bus, order a cab, or get on your bicycle. These are your transportation strategies.

Implement a main method in which you can pick one of the strategies depending on factors such as budget or time constraints using Strategy Pattern.



Remember:

- Identify an algorithm that's prone to frequent changes.
- Declare the strategy interface common to all variants of the algorithm.
- One by one, extract all algorithms into their own classes. They should all implement the strategy interface.
- Add a field for storing a reference to a strategy object.
- Clients of the context must associate it with a suitable strategy that matches the way they expect the context to perform its primary job

2. Use this repository to add a new franchise for Paisa Pizzas in Medellin, include local ingredients.
   https://github.com/bethrobson/Head-First-Design-Patterns/tree/master/src/headfirst/designpatterns/factory/pizzafm

3. Provide a custom implementation of the Builder pattern.

4. Choose two antipatterns and provide examples in code.
   https://sourcemaking.com/antipatterns/software-development-antipatterns

5. Write code for the Decorator pattern reviewed in class and use it to provide the sophistication level according to these rules:

   Simple: just coffee.
   Elementary: with milk.
   Sophisticated: with milk and cook time > 2 minutes.
   Basic: without milk and cook time > 2 minutes.
   Advanced: with milk, mocha and a figure in the cover.