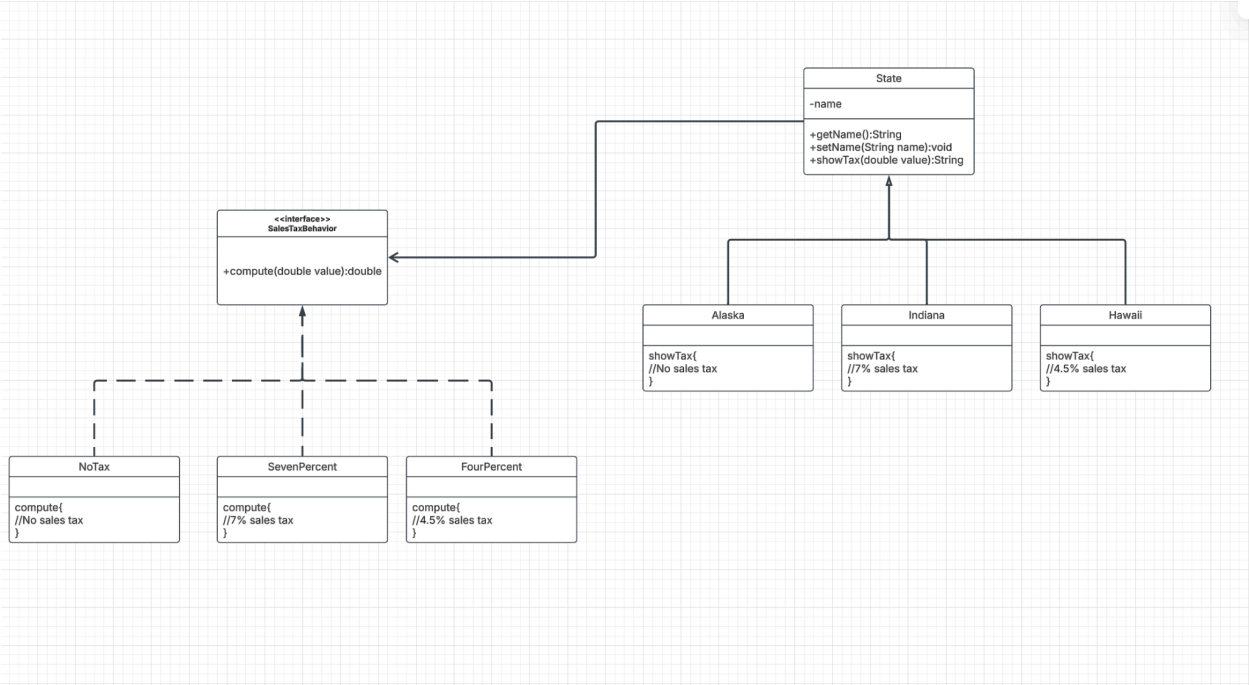
Thierry-Pascal Fleurant

Problem 1

UML class diagram



Problem 2

IS-A:

- Number 1, MallarDuck: It inherits from the Duck class therefore it's a IS-A relationship with the parent class. Furthermore, it only changes the display function to adapt to the type of duck it is.
- Number 2, RedheadDuck: It inherits from the Duck class therefore it's a IS-A relationship with the parent class. Furthermore, it only changes the display function to adapt to the type of duck it is.
- Number 3, RubberDuck: It inherits from the Duck class therefore it's a IS-A relationship with the parent class. Furthermore, it only changes the display function to adapt to the type of duck it is.
- Number 4, DecoyDuck: It inherits from the Duck class therefore it's a IS-A relationship with the parent class. Furthermore, it only changes the display function to adapt to the type of duck it is.
- Number 7, Quack: It implements the QuackBehavior interface therefore it is in fact a QuackBehavior.
- Number 8, Squeak: It implements the QuackBehavior interface therefore it is in fact a QuackBehavior.
- Number 9, MuteQuack: It implements the QuackBehavior interface therefore it is in fact a QuackBehavior.

HAS-A:

- Number 5. In this case, the Duck class has a FlyBehavior. It doesn't inherit it, it's only a composition where it contains an implementation of a FlyBehavior as one of its variables.
- Number 6. Similarly, in this case, the Duck class has a QuackBehavior. It doesn't inherit it, it's only a composition where it contains an implementation of a QuackBehavior as one of its variables.

Problem 3

UML class diagram