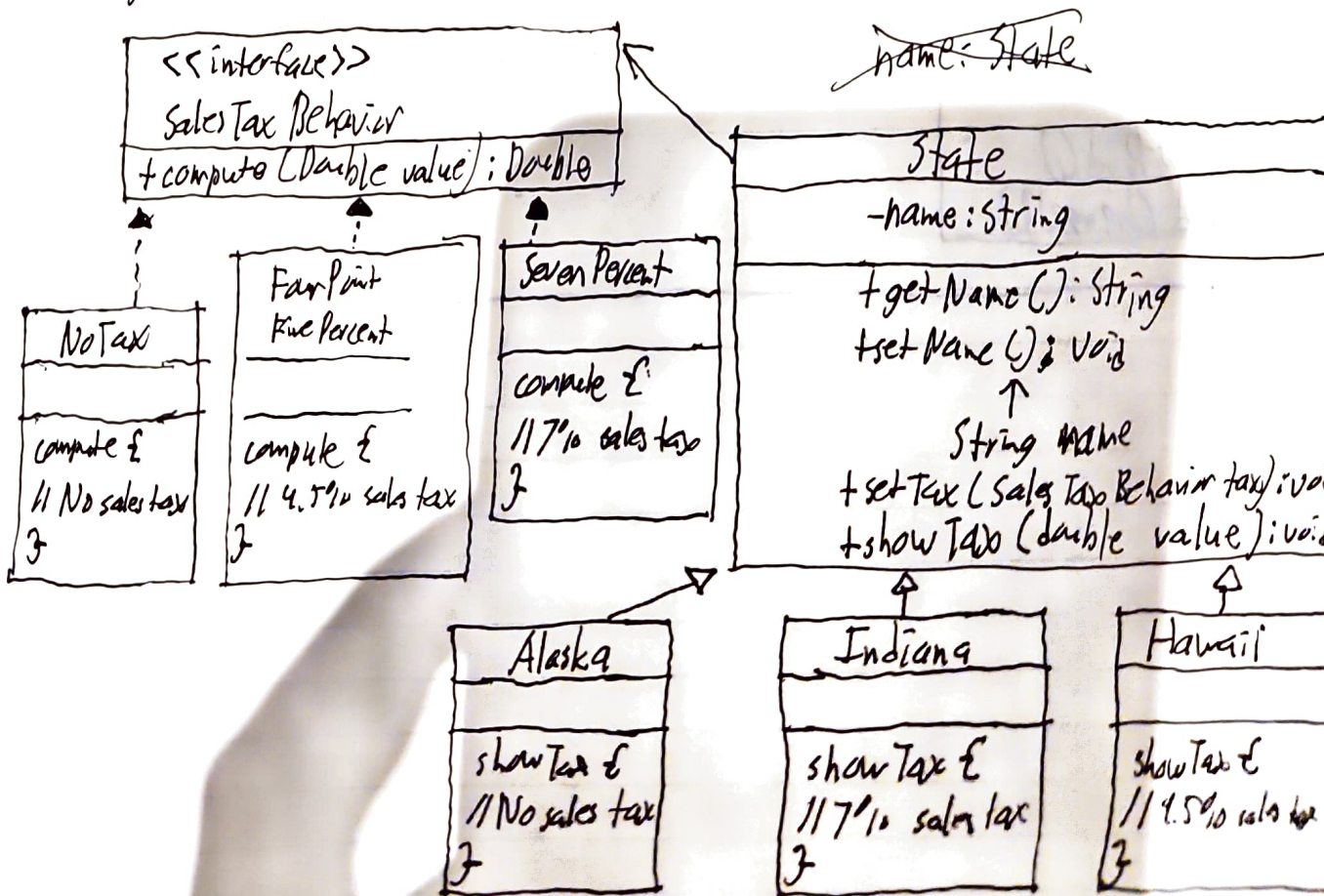# Problem 2:

1. IS-A: A Mallard Duck is a type of duck.
2. IS-A: A Redhead Duck is a type of Ducks, and also needs to swim, fly, etc.
3. IS-A: A Rubber Duck is a type of duck.

5. HAS-A: Fly Behavior is NOT a type of duck; Ducks have the ability To fly, so Fly Behavior would be a HAS-A relationship.

4. IS-A: Even if it's a decoy, Decoy Duck is categorized as a type of duck.

6. HAS-A: Quacking is NOT a type of duck, but rather a behavior a duck can do.

7. IS-A: Quack is a type of Quack Behavior.

8. IS-A: Squeaking is a type of Quack Behavior.

9. IS-A: Mute Quack, even if it is silent, is a type of QuackBehavior here.

Problem 1:

UML: State HAS-A Sales Tax

Indiana, Hawaii, Alaska ARE-A State

NoTax, Seven Percent, Four Point Five Percent ARE-A Sales Tax

name: State

```
<<interface>>
Sales Tax Behavior
+ compute (Double value): Double
```

```
NoTax

compute {
// No sales tax
}
```

```
FourPoint
Five Percent

compute {
// 4.5% sales tax
}
```

```
Seven Percent

compute {
// 7% sales tax
}
```

```
State
- name: String

+ get Name (): String
+ set Name (): Void
String name
+ set Tax (Sales Tax Behavior tax): void
+ show Tax (double value): void
```

```
Alaska

show Tax {
// No sales tax
}
```

```
Indiana

show Tax {
// 7% sales tax
}
```

```
Hawaii

show Tax {
// 4.5% sales tax
}
```

Problem 3: UML: A Auctioneer is subject, Three types of observers
In-person
Online
Phone
All channels

```
┌─────────────────────┐              observers          ┌─────────────────────┐   ┌─────────────────────┐
│   << interface >>   │───────────────────────────────▶│    <<interface>>    │   │    <<interface>>    │
│       Subject       │                                 │      Observer       │   │       Channel       │
├─────────────────────┤                                 ├─────────────────────┤   ├─────────────────────┤
│  registerObserver() │                                 │     update()        │   │     payBid()        │
│  removeObserver()   │                                 └─────────────────────┘   └─────────────────────┘
│  notifyObserver()   │
└─────────────────────┘
```

- Subject interface
  - registerObserver()
  - removeObserver()
  - notifyObserver()

- Observer interface
  - update()

- Channel interface
  - payBid()

- Auctioneer
  - registerObserver()
  - removeObserver()
  - notifyObserver()
  - getBid()
  - bidChanged()

- InPerson Channel
  - update()
  - payBid()

- Online Channel
  - update()
  - payBid()

- Phone Channel
  - update()
  - payBid()

subject
observers