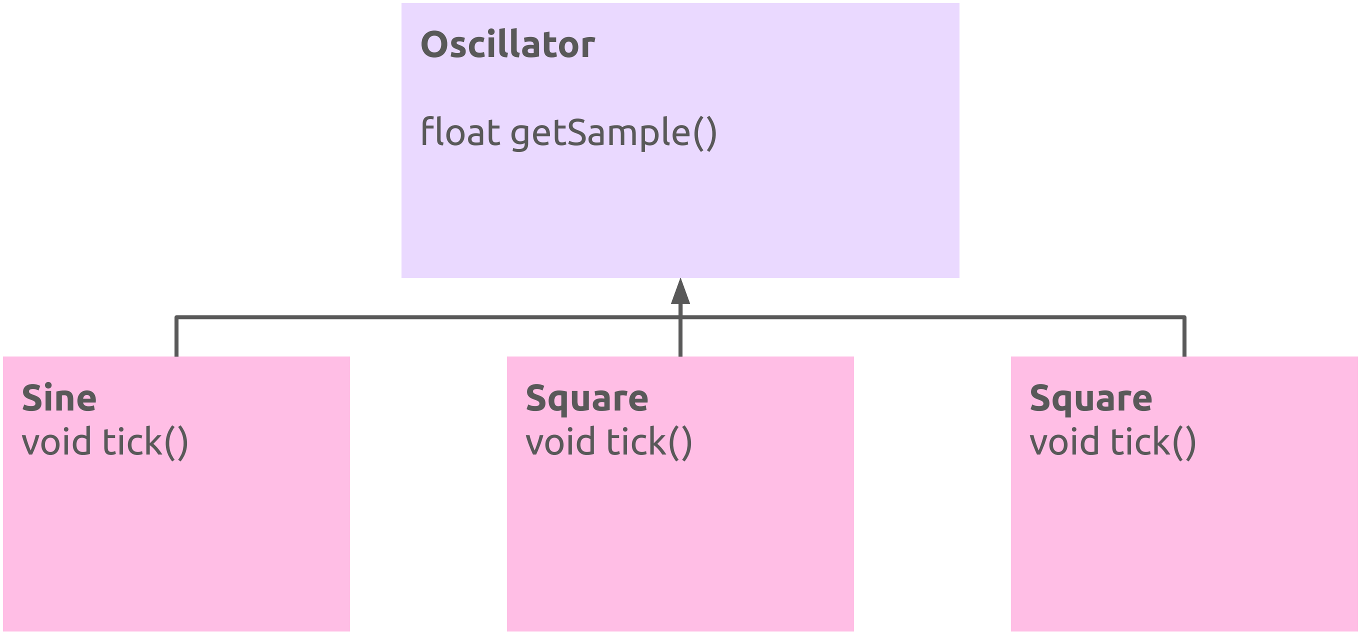
Virtual

- virtual method
- abstract class



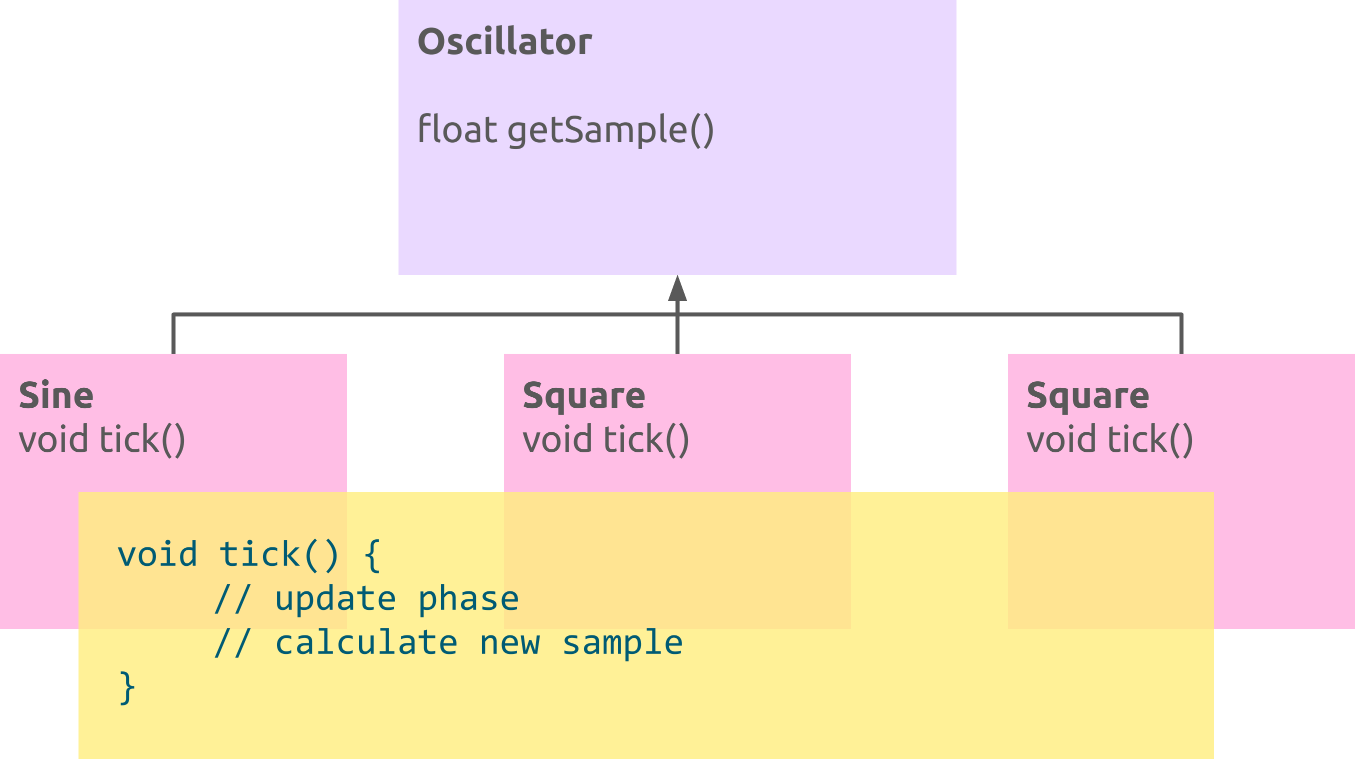
Oscillator

float getSample()

Sine
void tick()

Square
void tick()

Square
void tick()



Oscillator

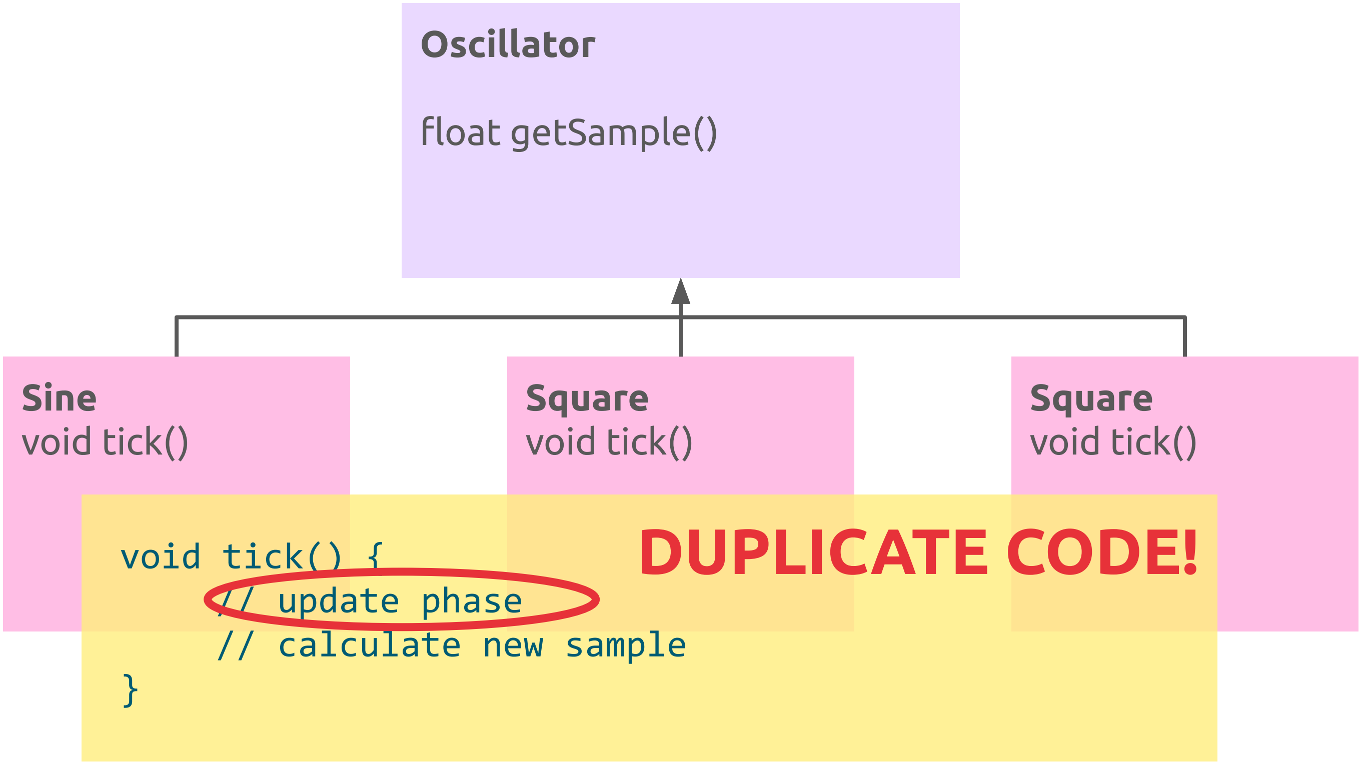
float getSample()

Sine
void tick()

Square
void tick()

Square
void tick()

```
void tick() {
     // update phase
     // calculate new sample
}
```



Oscillator

float getSample()

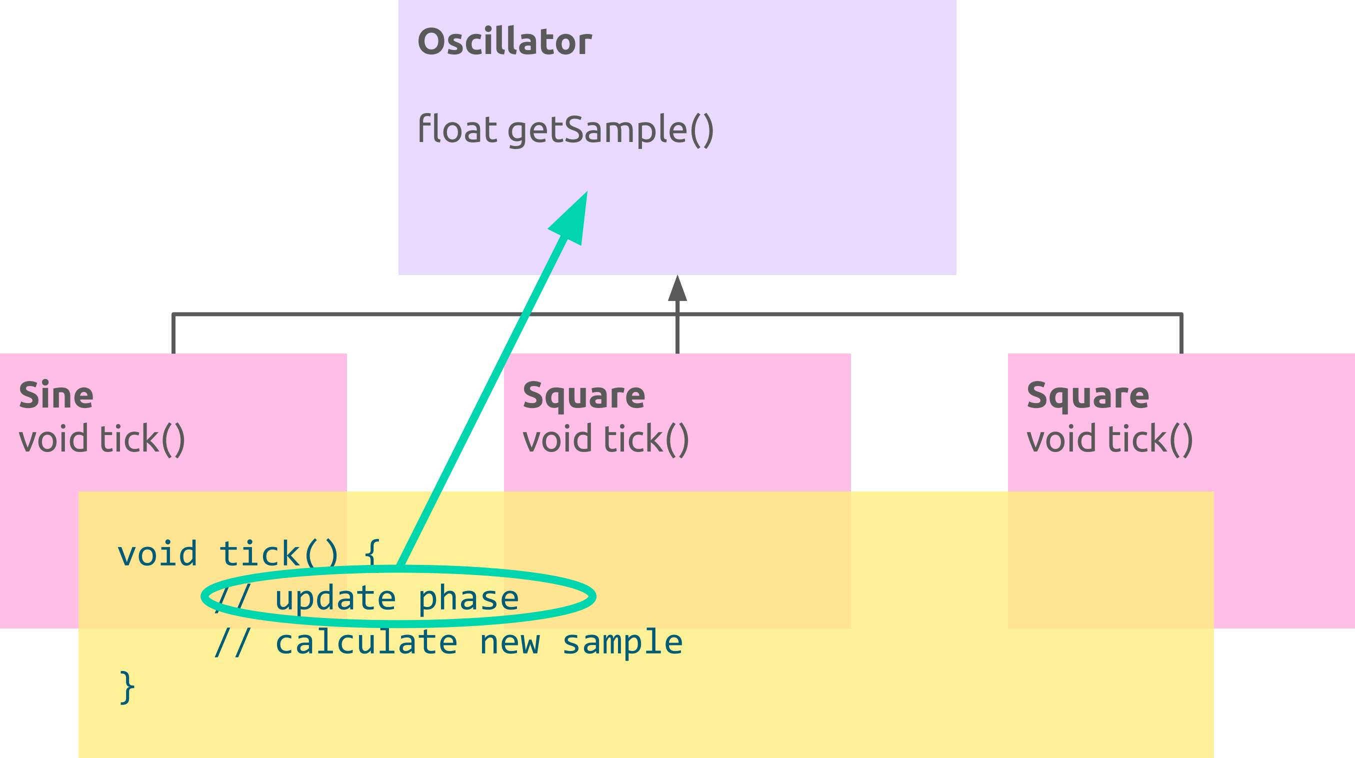
Sine
void tick()

Square
void tick()

Square
void tick()

```
void tick() {
     // update phase
     // calculate new sample
}
```

DUPLICATE CODE!



Oscillator

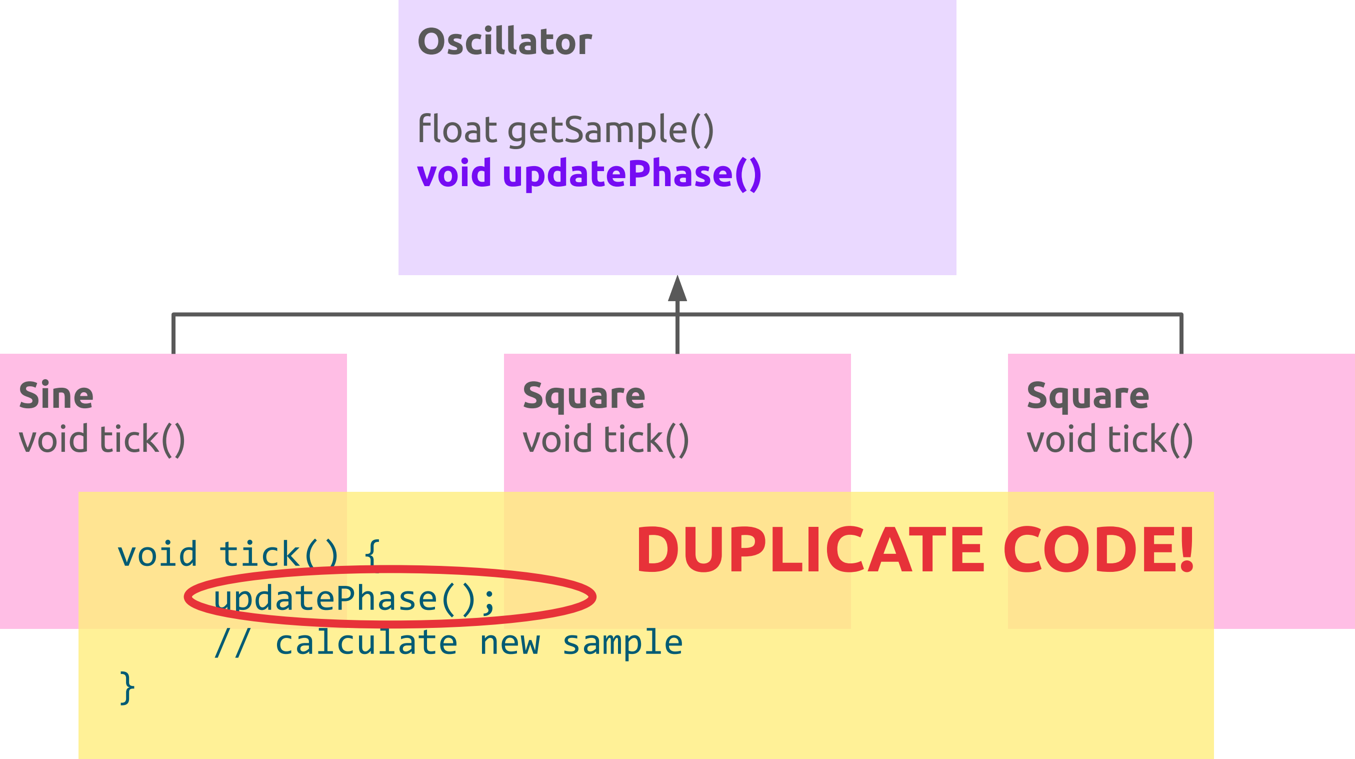
float getSample()

Sine
void tick()

Square
void tick()

Square
void tick()

```
void tick() {
     // update phase
     // calculate new sample
}
```



Oscillator

float getSample()

void updatePhase()

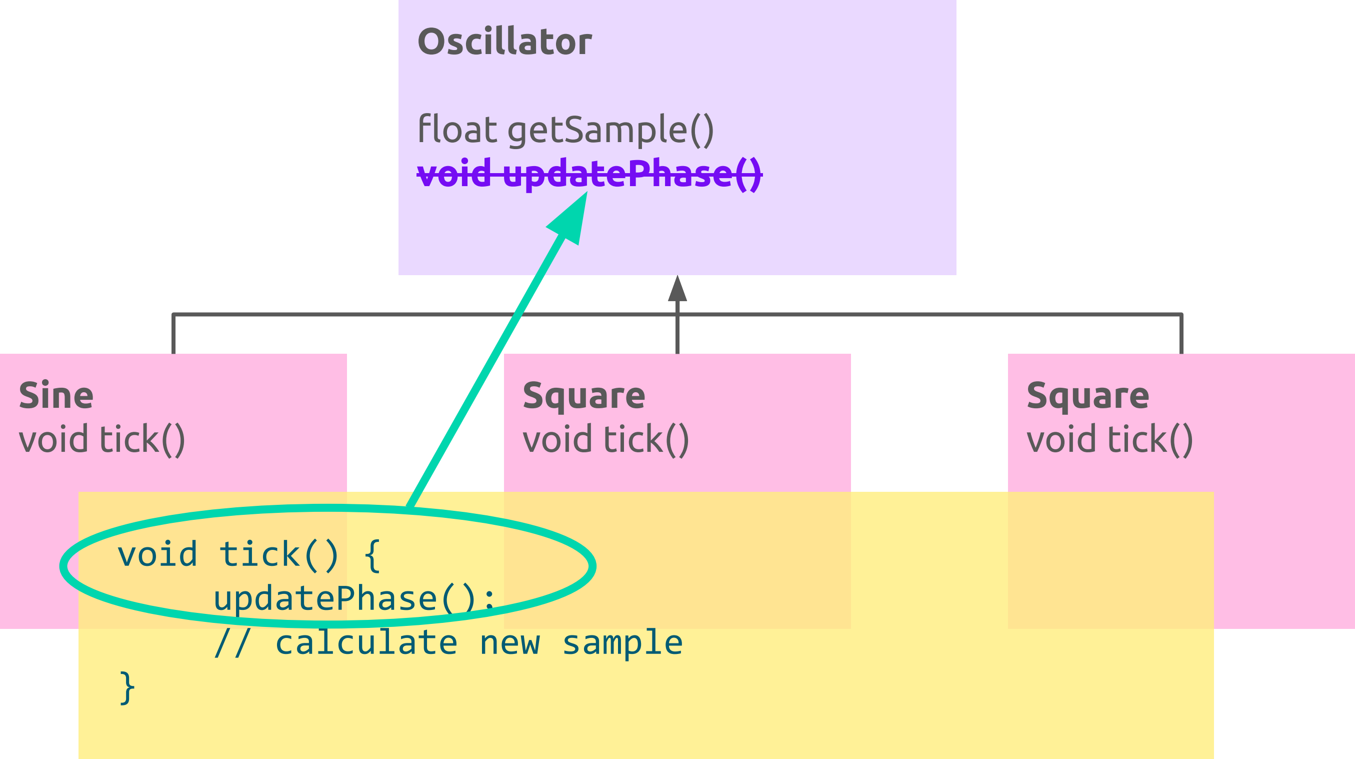
Sine
void tick()

Square
void tick()

Square
void tick()

```
void tick() {
    updatePhase();
    // calculate new sample
}
```

DUPLICATE CODE!



Oscillator

float getSample()

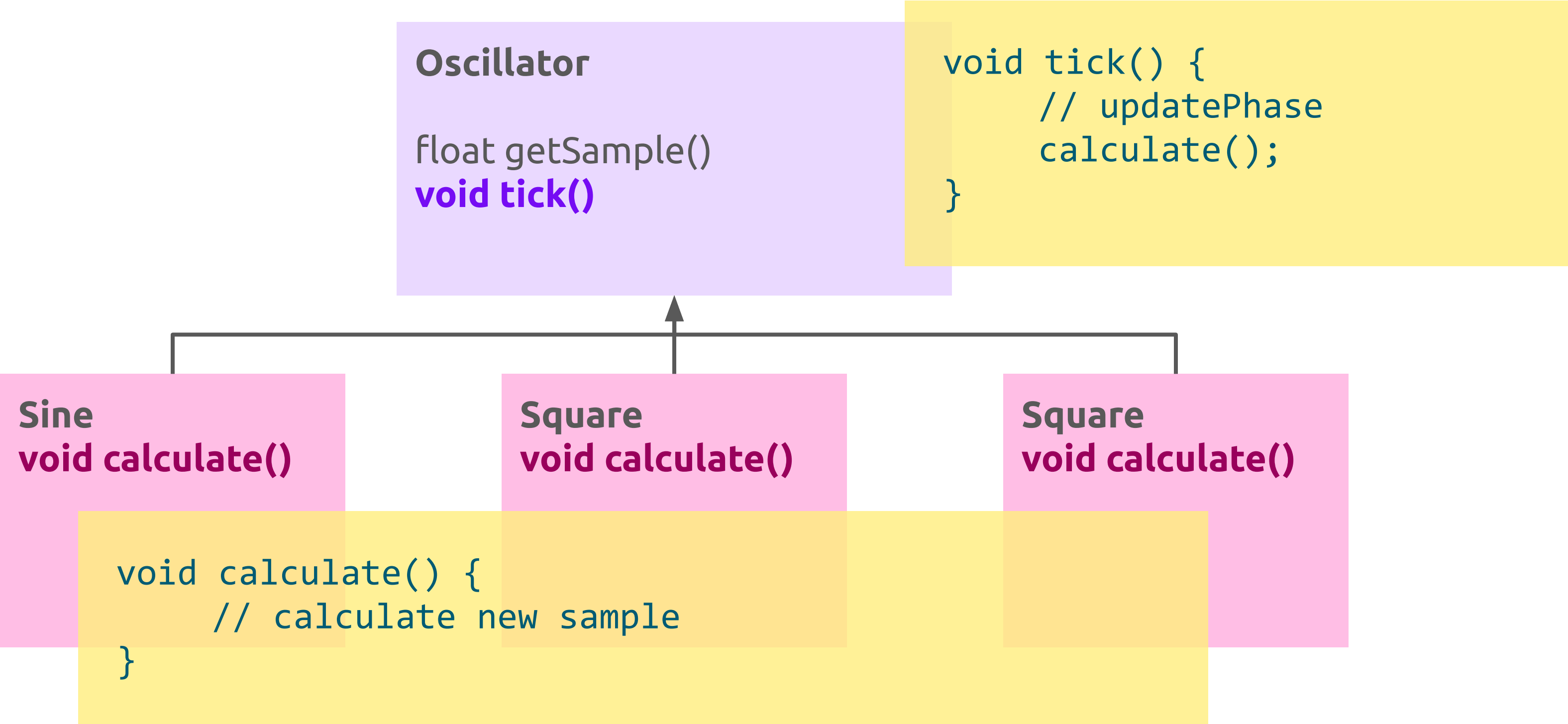
~~void updatePhase()~~

Sine
void tick()

Square
void tick()

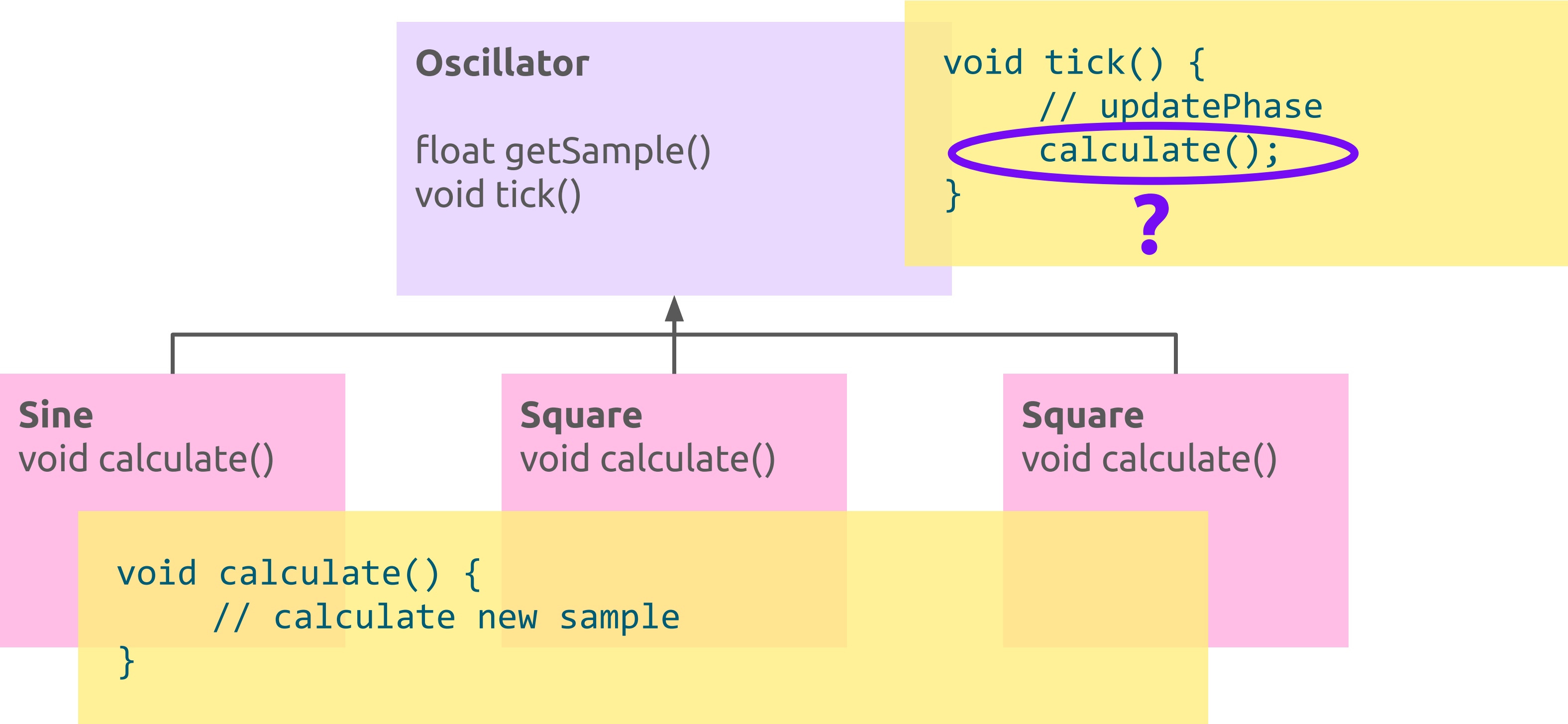
Square
void tick()

```
void tick() {
    updatePhase():
    // calculate new sample
}
```



```
void tick() {
     // updatePhase
     calculate();
}
```

```
void calculate() {
     // calculate new sample
}
```



Oscillator

float getSample()
void tick()

```
void tick() {
     // updatePhase
     calculate();
}
```

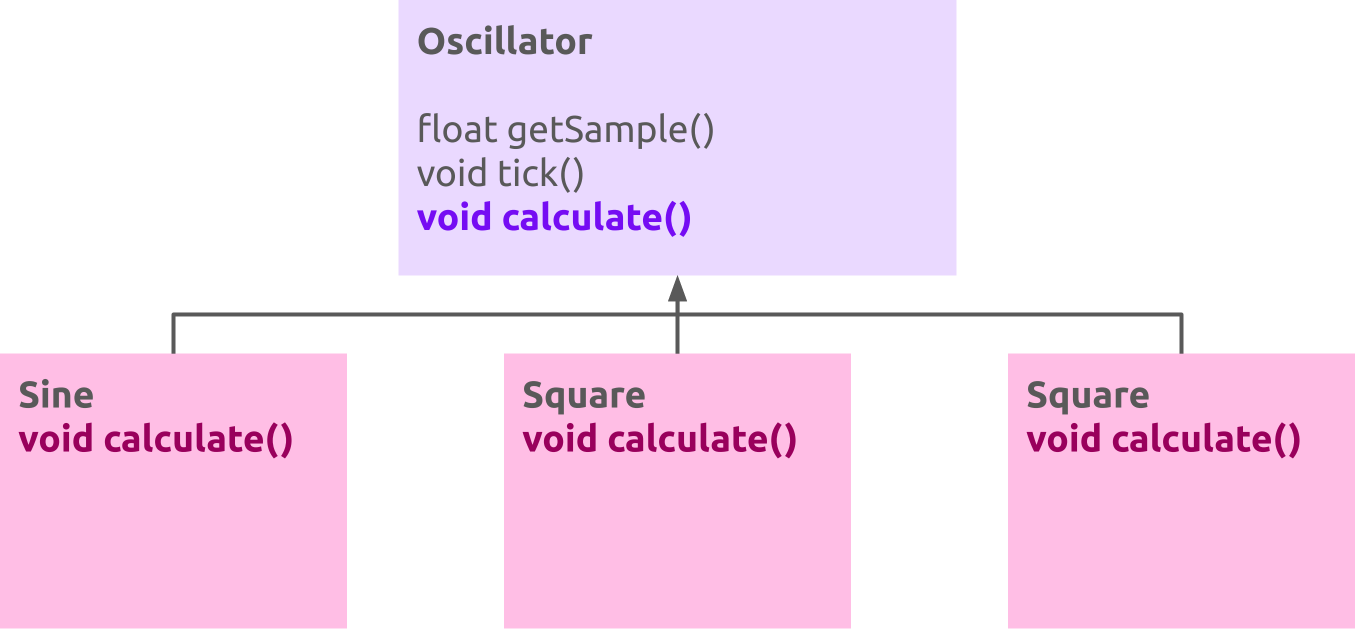
?

Sine
void calculate()

Square
void calculate()

Square
void calculate()

```
void calculate() {
     // calculate new sample
}
```



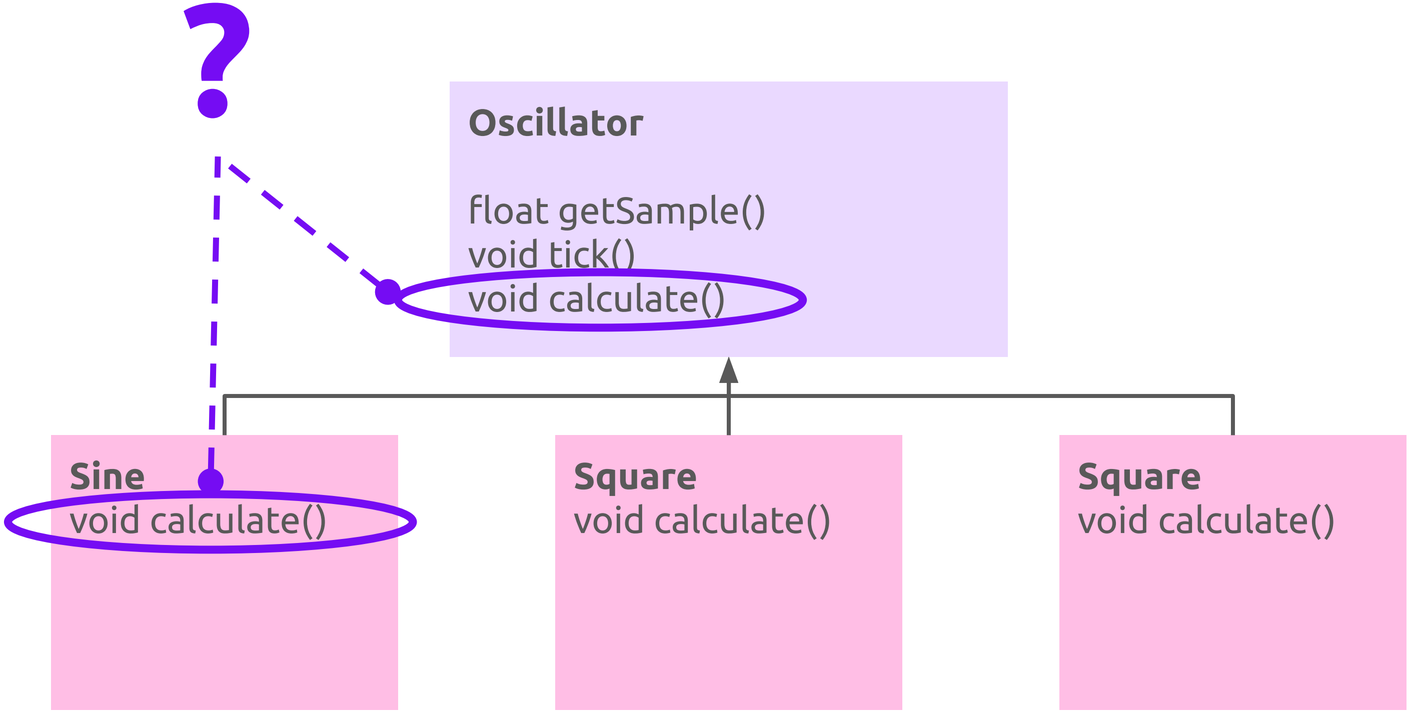
Oscillator

float getSample()
void tick()
void calculate()

Sine
void calculate()

Square
void calculate()

Square
void calculate()



?

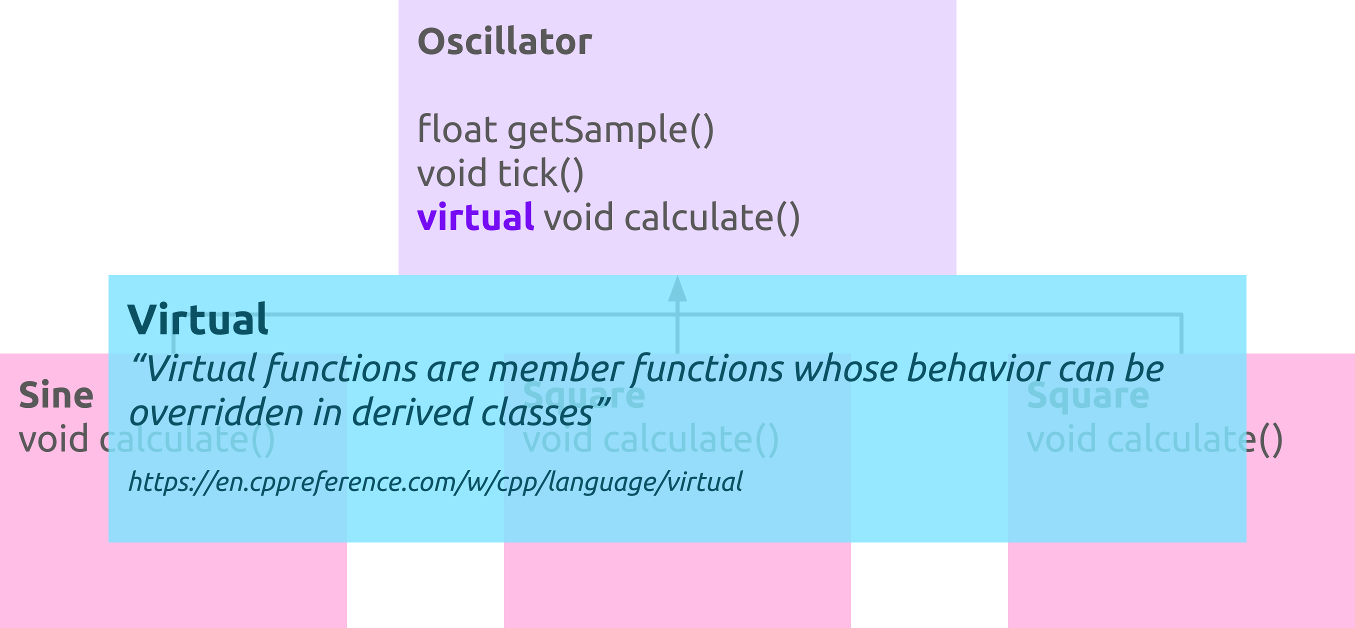
Oscillator

float getSample()
void tick()
void calculate()

Sine
void calculate()

Square
void calculate()

Square
void calculate()



Oscillator

float getSample()
void tick()
virtual void calculate()

Sine
void calculate()

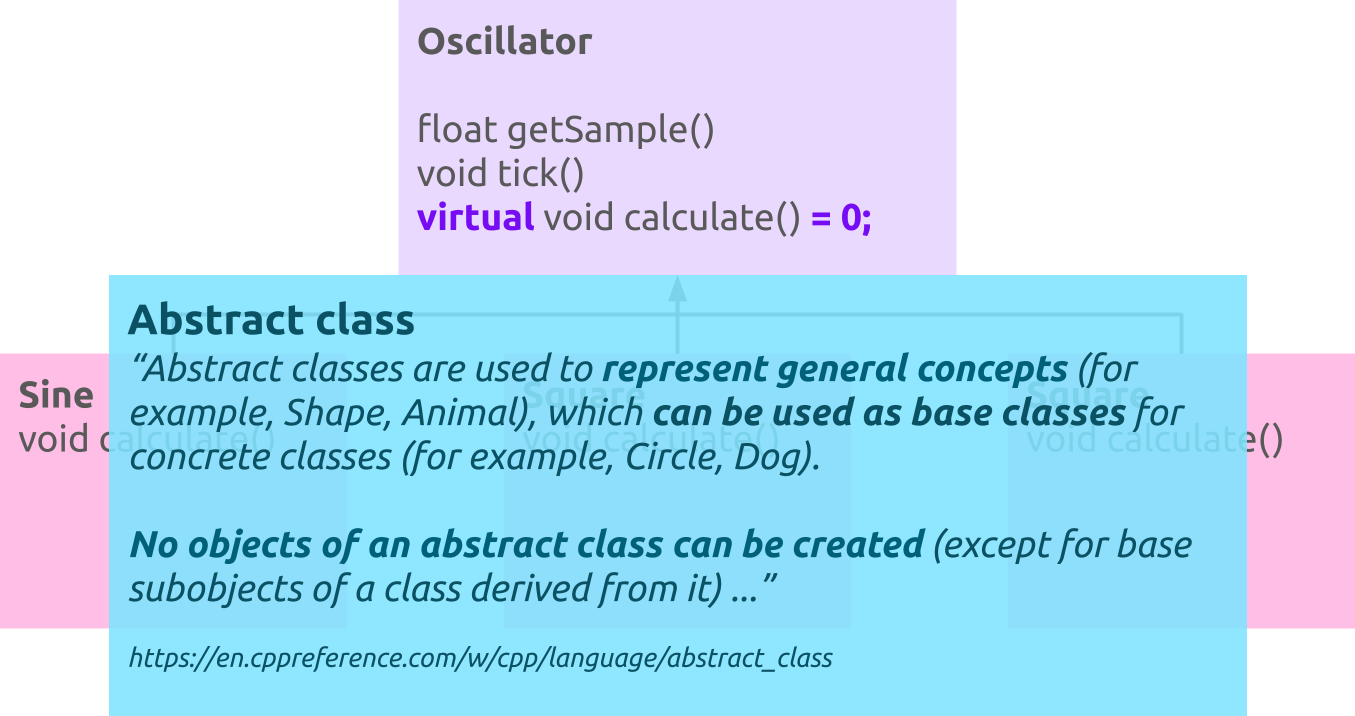
Square
void calculate()

Square
void calculate()

Virtual

"Virtual functions are member functions whose behavior can be overridden in derived classes"

https://en.cppreference.com/w/cpp/language/virtual



Oscillator

float getSample()
void tick()
virtual void calculate() **= 0;**

Sine
void calculate()

Square
void calculate()

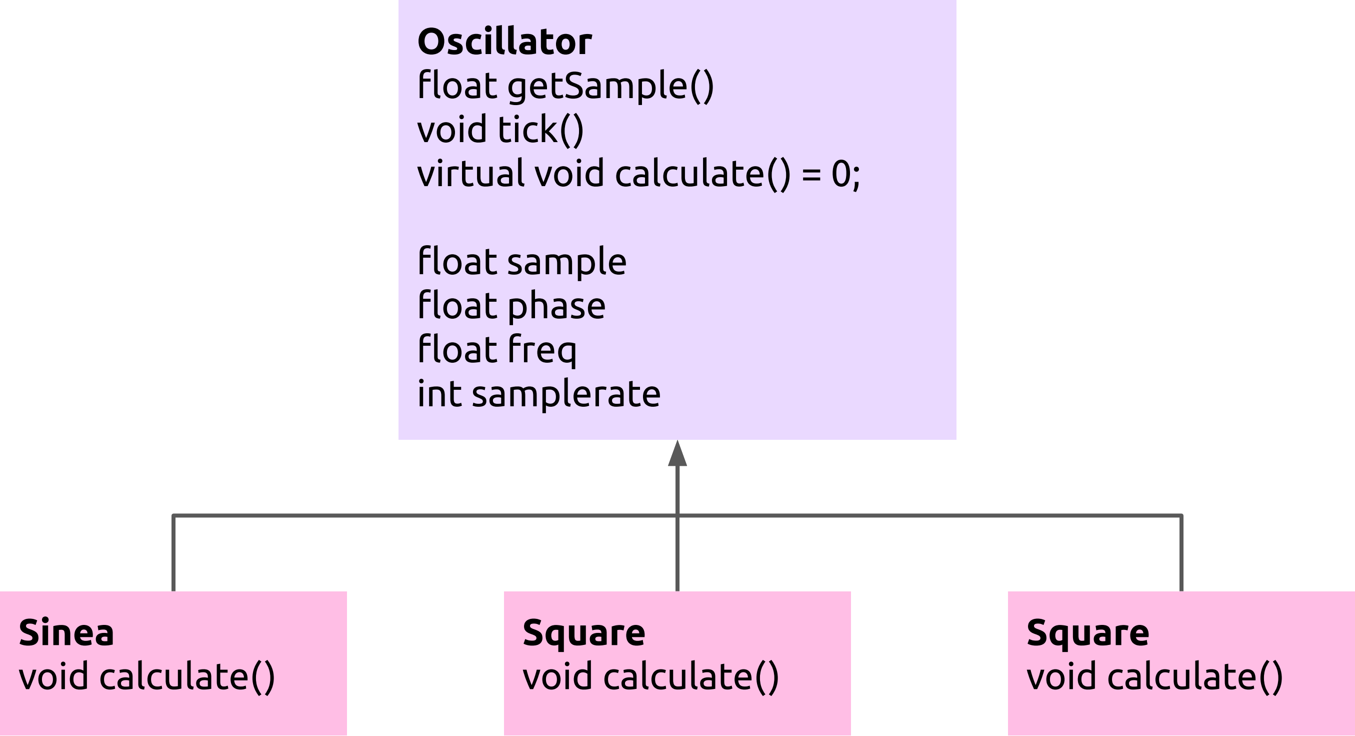
Square
void calculate()

Abstract class

*“Abstract classes are used to **represent general concepts** (for example, Shape, Animal), which **can be used as base classes** for concrete classes (for example, Circle, Dog).*

***No objects of an abstract class can be created** (except for base subobjects of a class derived from it) ...”*

https://en.cppreference.com/w/cpp/language/abstract_class



Oscillator
float getSample()
void tick()
virtual void calculate() = 0;

float sample
float phase
float freq
int samplerate

Sinea
void calculate()

Square
void calculate()

Square
void calculate()