



ABSTRACT CLASS

IN DART

Object-Oriented Programming

INTRODUCTION

จุดประสงค์



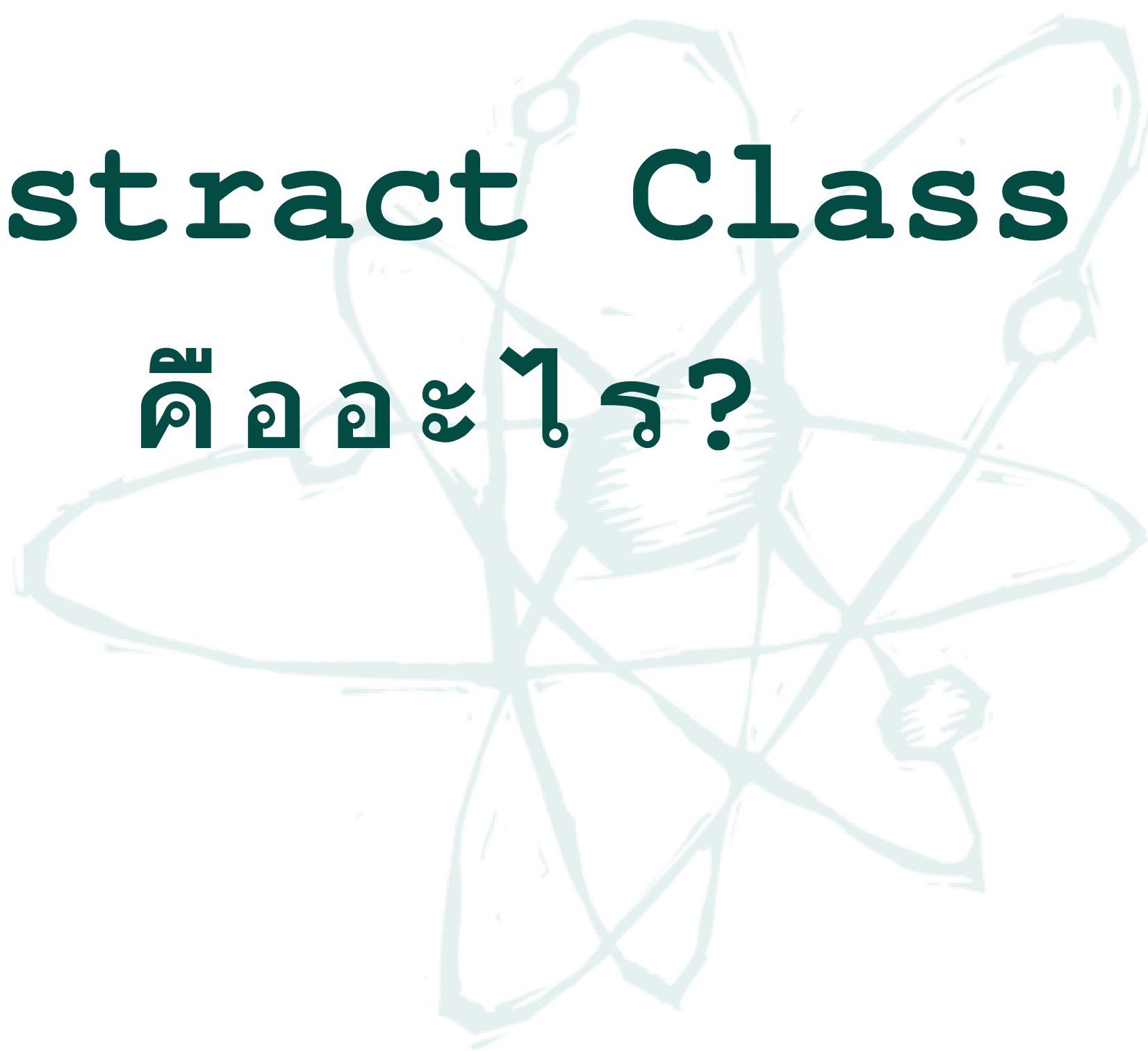
ในบทนี้จะได้เรียนรู้เกี่ยวกับ abstract class ใน ภาษา dart ก่อนที่จะลงลึกเนื้อหา
ในบทนี้ จะต้องมีความเข้าใจเกี่ยวกับ class, object, constructor, and inheritance.
ก่อนในบทที่ผ่านมาเราจะได้เรียนรู้การประกาศ คลาส กัน ซึ่ง คลาส เหล่านั้นเรียกว่า concrete
classes และ concrete class เหล่านั้นก็สร้างเป็น object เพื่อนำไปใช้งานได้ แต่ใน
abstract class นั้นสร้างเป็น object ไม่ได้





Abstract Class

คืออะไร?



Abstract Class คืออะไร?

Abstract Class คือ คลาส ที่ไม่สามารถสร้าง object หรือ สร้างเป็น
ตัวแปรได้ มันถูกใช้เพื่อบอกว่า คลาส นี้ มี method การทำงานอะไรบ้าง เพื่อ
ที่จะให้ คลาส อื่นนั้นสืบทอดการทำงานต่อไปนั่นเอง keyword ในการใช้ก็คือ
abstract



Abstract Method

คืออะไร?



Abstract Method คืออะไร?

Abstract method คือ method ที่มีเครื่องหมาย semicolon(;) ปิดท้าย และไม่มีการทำงานใน method เป็นแค่การประกาศไว้เพื่อบอกว่า class นี้มี method เท่านั้น และให้ class อื่น ที่สืบทอดต่อไปปรับเปลี่ยนข้างในกันเอง



ทำไมต้องงี้ ช้ Abstract Class ?

ทำไมต้องใช้ Abstract Class ?

คลาสลูก ที่สืบทอด abstract class ทั้งหมดจะต้องประกาศ method ที่มีอยู่ใน abstract class ทั้งหมด และเพราะการใช้สิ่งนี้จึงทำให้เกิด “Abstraction” ใน Dart programming language

ข้อมูลเพิ่มเติม: Abstraction คือการที่เราสร้าง class หนึ่งขึ้นมา และให้ class อื่นสืบทอดต่อกันไป หรือ เรียกว่าให้มันเป็นมาตรฐานเดียวกันก็ได้ในแต่ละ class

ตัวอย่าง

```
abstract class ClassName {  
    //Body of abstract class  
  
    method1();  
    method2();  
}
```



ตัวอย่าง (ต่อ)

```
abstract class Vehicle {
    // Abstract method
    void start();
    // Abstract method
    void stop();
}

class Car extends Vehicle {
    // Implementation of start()
    @Override
    void start() {
        print('Car started');
    }

    // Implementation of stop()
    @Override
    void stop() {
        print('Car stopped');
    }
}

class Bike extends Vehicle {
    // Implementation of start()
    @Override
    void start() {
        print('Bike started');
    }

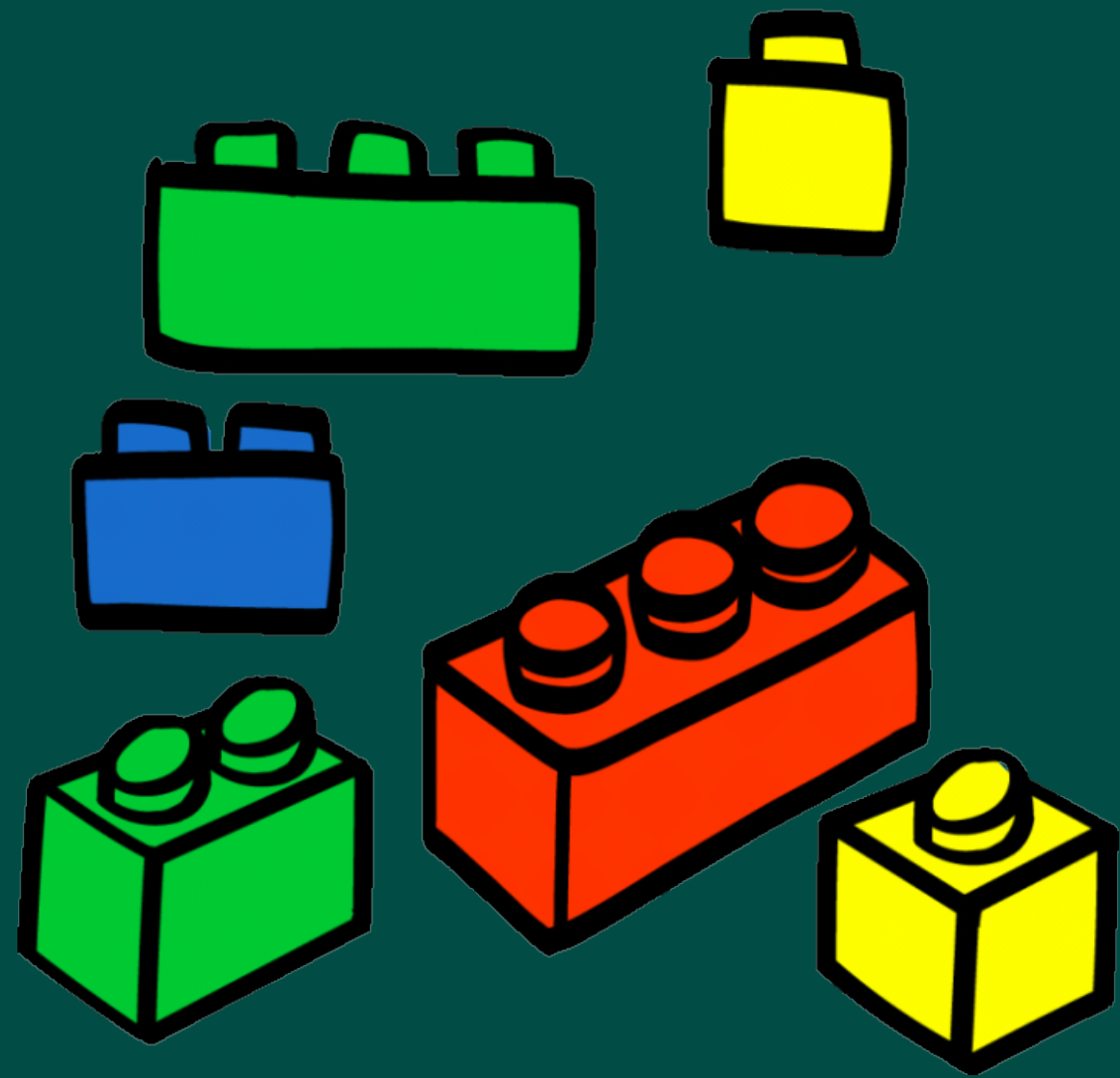
    // Implementation of stop()
    @Override
    void stop() {
        print('Bike stopped');
    }
}

void main() {
    Car car = Car();
    car.start();
    car.stop();

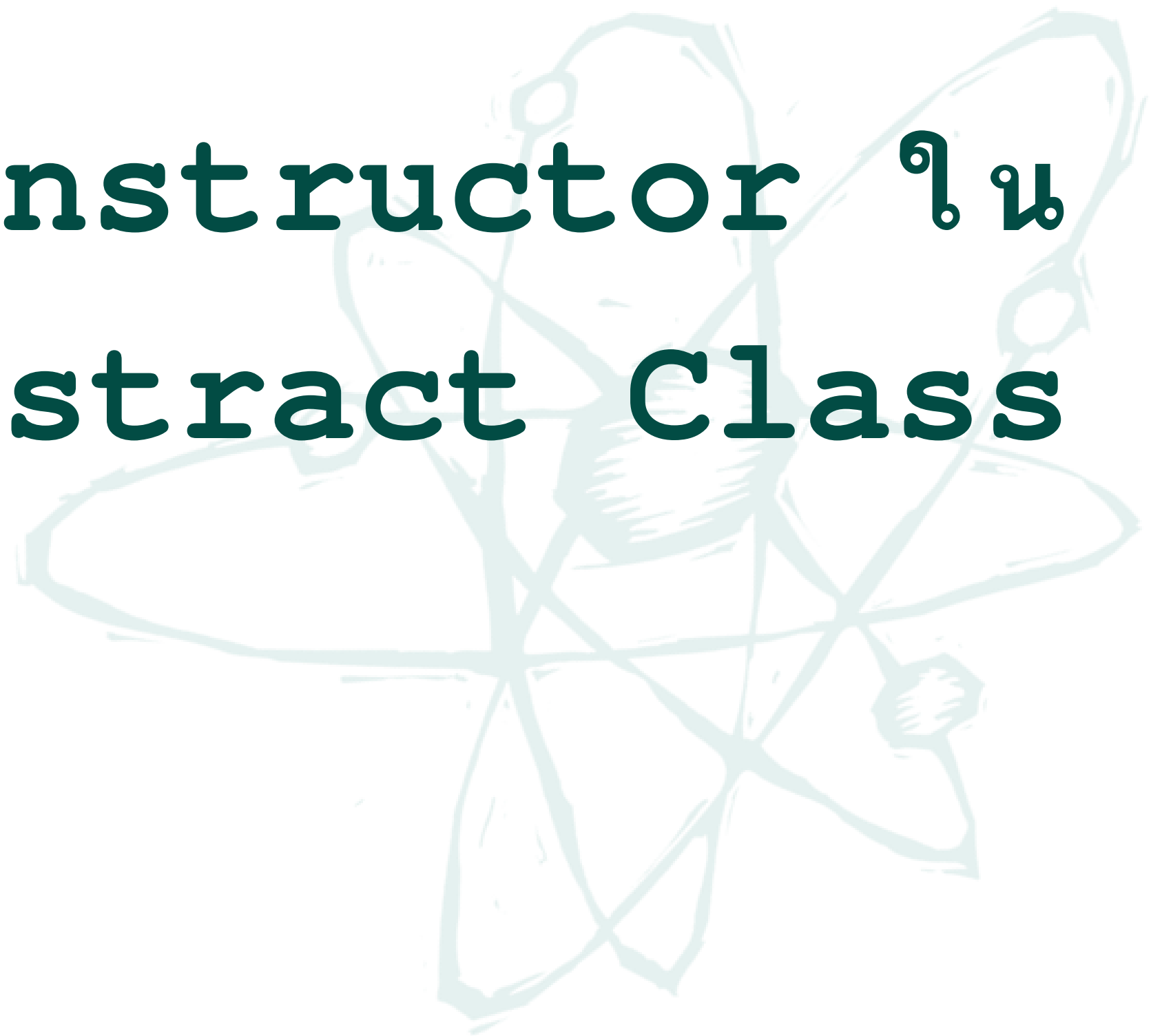
    Bike bike = Bike();
    bike.start();
    bike.stop();
}
```

ผลลัพธ์:

```
Car started
Car stopped
Bike started
Bike stopped
```



Constructor ใน Abstract Class



Constructor ใน Abstract Class

ใน Abstract class นั้นสร้าง object ไม่ได้ก็จริง แต่อย่างไรก็ตาม เราสามารถกำหนด constructor ใน abstract class ได้ และ constructor ใน abstract class จะถูกเรียกใช้เมื่อ object ของคลาสอื่นสร้างขึ้น โดยใช้ `super();`

ตัวอย่าง

```
abstract class Bank {
    String name;
    double rate;

    // Constructor
    Bank(this.name, this.rate);

    // Abstract method
    void interest();

    //Non-Abstract method: It have an implementation
    void display() {
        print('Bank Name: $name');
    }
}

class SBI extends Bank {
    // Constructor
    SBI(String name, double rate) : super(name, rate);

    // Implementation of interest()
    @override
    void interest() {
        print('The rate of interest of SBI is $rate');
    }
}

class ICICI extends Bank {
    // Constructor
    ICICI(String name, double rate) : super(name, rate);

    // Implementation of interest()
    @override
    void interest() {
        print('The rate of interest of ICICI is $rate');
    }
}

void main() {
    SBI sbi = SBI('SBI', 8.4);
    ICICI icici = ICICI('ICICI', 7.3);

    sbi.interest();
    icici.interest();
    icici.display();
}
```

ผลลัพธ์:

```
The rate of interest of SBI is 8.4
The rate of interest of ICICI is 7.3
Bank Name: ICICI
```

จุดสำคัญที่ต้องจำในเรื่องนี้

1. Abstract class สร้างเป็น object ไม่ได้
2. Abstract class มีได้ทั้ง abstract method และ concrete method (มีการทำงานข้างในซึ่งสืบทอดโดยคลาสลูกได้หากไม่มี keyword final นำหน้า)
3. สร้างขึ้นเพื่อบอกการทำงานในคลาส(abstract class)แล้วให้คลาสอื่นมาสืบทอด
4. Abstract method เป็นคลาสที่ไม่มีการทำงานข้างในหรือก็คือ มีเพียงแค่ semicolon(;) ปิดท้าย

แหล่งอ้างอิง

- <https://dart-tutorial.com/object-oriented-programming/abstract-class-in-dart/>
- <https://www.javatpoint.com/abstract-class-in-java>



จบการนำเสนอ