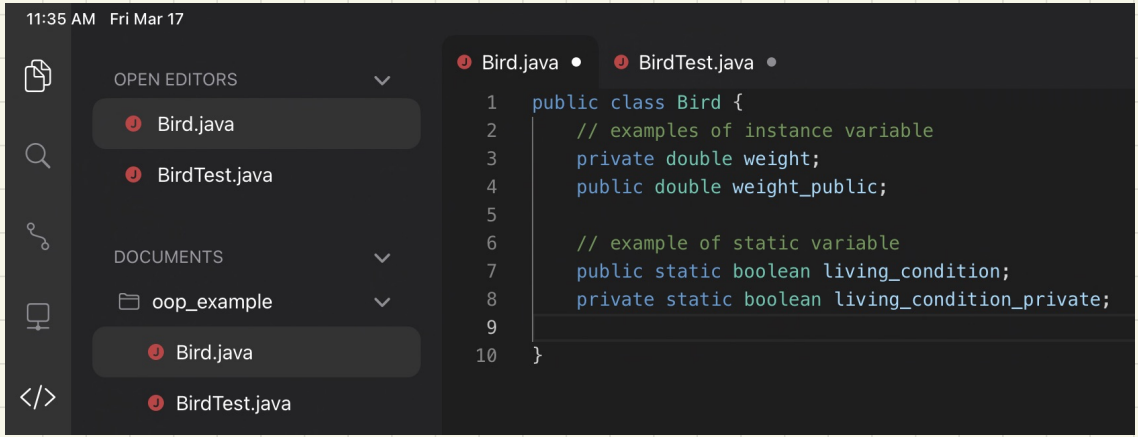
How to create a class?

1. Define instance variable

2. Define static variable



3. Create constructor

Bird.java • BirdTest.java •

```java
public class Bird {
    // examples of instance variable
    private double weight;
    public double weight_public;

    // example of static variable
    public static boolean living_condition;
    private static boolean living_condition_private;

    // create constructor that used to initialize instance variable
    // constructor allow you to create an object
    public Bird(double initial_weight) {

    }
}
```

Constructor is a method that has the same name as the class and no return type. It is used to initialize instance variables.

When you create an object, you will need to call the constructor in the client program.

With a constructor, you can create an object

Bird.java BirdTest.java

```java
public class BirdTest {

    public static void main(String args[]) {

        Bird little_bird = new Bird(10);
    }
}
```

4. Initialize instance variable inside the constructor

Bird.java • BirdTest.java

```java
public class Bird {
    // examples of instance variable
    private double weight;
    public double weight_public;

    // example of static variable
    public static boolean living_condition;
    private static boolean living_condition_private;

    // create constructor that used to initialize instance variable
    // constructor allow you to create an object
    public Bird(double initial_weight) {
        this.weight = initial_weight;
        this.weight_public = 0;
    }
}
```

5. Write some instance method

```java
public double getWeight() {
    return this.weight;
}

public void eat(double amount) {
    this.weight += amount;
}

private double private_get_weight() {
    return this.weight;
}
```

What is mutator method and accessor method

6. Write some static method

```
public static void change_living_condition() {
    living_condition = !living_condition;
}
```

The complete Bird class and BirdTest class

```java
public class Bird {
    // examples of instance variable
    private double weight;
    public double weight_public;

    // example of static variable
    public static boolean living_condition;
    private static boolean living_condition_private;

    // create constructor that used to initialize instance variable
    // constructor allow you to create an object
    public Bird(double initial_weight) {
        this.weight = initial_weight;
        this.weight_public = 10;
    }

    public double getWeight() {
        return this.weight;
    }

    public void eat(double amount) {
        this.weight += amount;
    }

    private double private_get_weight() {
        return this.weight;
    }

    public static void change_living_condition() {
        living_condition = !living_condition;
    }
}
```

Bird.java BirdTest.java

```java
public class BirdTest {

    public static void main(String args[]) {

        Bird little_bird = new Bird(10);

        System.out.println("The weight_public is " + little_bird.weight_public);

        System.out.println("The living condition of Bird is " + Bird.living_condition);

        little_bird.eat(10);
        System.out.println("The little_bird's weight is " + little_bird.getWeight());

        Bird.change_living_condition();
        System.out.println("The living condition of Bird is " + Bird.living_condition);
    }
}
```

PROBLEMS EXECUTION TERMINAL Split View

Program input..

```
Running BirdTest.java remotely..

The weight_public is 10.0
The living condition of Bird is false
The little_bird's weight is 20.0
The living condition of Bird is true
```

You should know the following:

1. Call public instance variable inside class: this.var or var
2. Call public instance variable in client program: object.var
3. Call private instance variable inside class: this.var or var
4. Call private instance variable in client program: not allowed
5. Call public static variable inside class: var
6. Call public static variable in client program: Class.var
7. Call private static variable inside class: var
8. Call private static variable in client program: not allowed
9. Call public instance method inside class: this.method() or method()
10. Call public instance method in client program: object.method()
11. Call private instance method inside class: this.method() or method()
12. Call private instance method in client program: not allowed
13. Call public static method inside class: method()
14. Call public static method in client program: Class.method()
15. Call private static method inside class: method()
16. Call private static method in client program: not allowed

You cannot only use static method/variable inside static method

1. Example: use this.weight_public or weight_public inside the bird class
2. Example: print(little_bird.weight_public) in the BirdTest class
3. Example: use this.weight or weight inside the bird class

5. Example: use living_condition inside the bird class
6. Example: use Bird.living_condition in the BirdTest class
7. Example: use living_condition_private inside the bird class

9. Example: use this.getWeight() or getWeight() inside the bird class
10. Example: use little_bird.eat(10) in BirdTest class
11. Example: use this.private_get_weight() or private_get_weight() inside the bird class

13. Example: use change_living_condition() inside the the bird class
14. Example: use Bird.change_living_condition() in the BirdTest class
15. We seldom use private static method

Quick demo of writing movingparticle

Method overload and constructor overload:
You can have two method/constructor with the same name but different parameter type or number of parameter

Passing reference type to method:

You make a copy of the reference when it passes to the method. You can change the instance variable of the object, but you cannot let the passed in reference refers to another variable