

Writing classes

Java Reference

Fields, constructor & methods

A class 类 is a blueprint for objects. Its fields 字段 store data, its constructor 构造方法 sets up a new object, and its methods 方法 are the actions. `this.name` means "this object's name". Create an object with `new`.

```
public class Main {
    public static void main(String[] args) {
        Dog d = new Dog("Rex", 3);
        System.out.println(d.describe()); // Rex is 3 years old
        d.haveBirthday();
        System.out.println(d.describe()); // Rex is 4 years old
    }
}

class Dog {
    private String name;
    private int age;

    public Dog(String name, int age) { // constructor
        this.name = name;
        this.age = age;
    }

    public String describe() {
        return name + " is " + age + " years old";
    }

    public void haveBirthday() {
        age++;
    }
}
```

Encapsulation

Encapsulation 封装 means hiding data behind methods. Mark fields `private` so outside code can't touch them directly; expose an accessor 访问方法 (getter) to read, and a method to change them safely. The method can **guard** the data —here a deposit must be positive.

```
public class Main {
    public static void main(String[] args) {
        Account a = new Account(100);
        a.deposit(50);
    }
}
```

```
        a.deposit(-999);           // rejected by the guard
        System.out.println(a.getBalance()); // 150
    }
}

class Account {
    private int balance;           // hidden from outside

    public Account(int start) {
        balance = start;
    }

    public void deposit(int amount) {
        if (amount > 0) balance += amount; // guard keeps balance valid
    }

    public int getBalance() {      // accessor (getter)
        return balance;
    }
}
```