9-Classes & Objects

Author: Vincent Lau

Note: This material is intended for educational purposes only. All rights reserved. Any unauthorized sharing or copying of this material, in any form, to any individual or party, for any use without prior permission, is strictly prohibited.

Learning Objectives

> Gain basic understanding of Object, Class, Inheritance, Interface, and Package

> Understand the benefits of using objects to build code

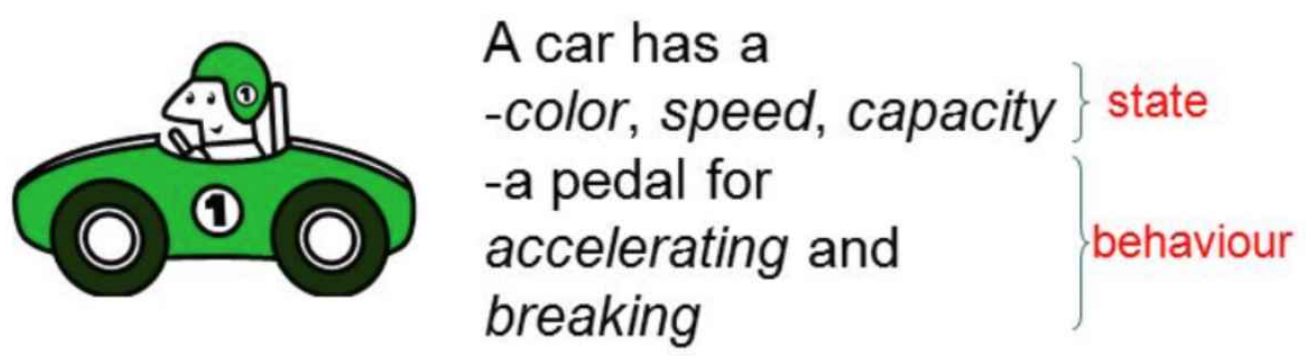
> Understand the benefits of data encapsulation

> Understand the benefits of inheritance and interface

What is an Object?

- An `object` is a software bundle of related **state and behavior**. Software objects are often used to **model real-world objects** in everyday life.

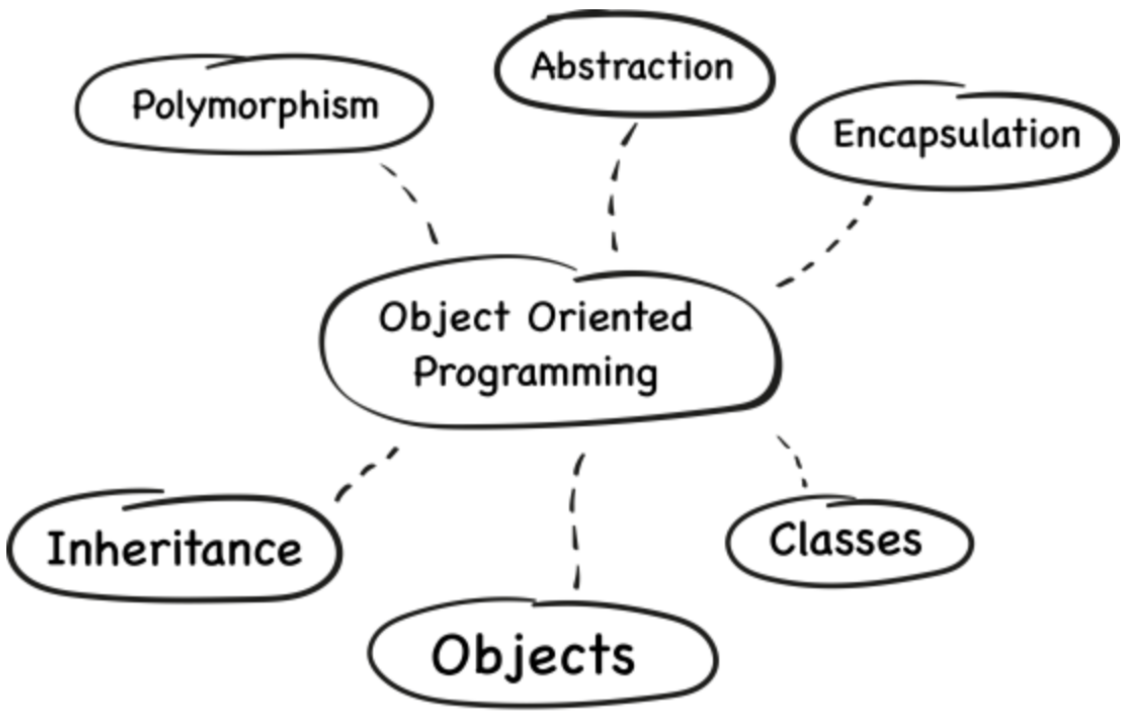
State and Behavior



- `Real-world objects` share two characteristics: They all have `state` and `behavior`.
  - For example, a dog has state (name, color, breed, hungry), and behavior (barking, fetching, wagging tail).
- Software objects are conceptually similar to real-world objects: they both consist of state and related behavior.
- An object stores its state in *fields* (variables in some programming languages) and exposes its behavior through *methods* (functions in some programming languages).
- Methods operate on an object's internal state and serve as the primary mechanism for object-to-object communication.

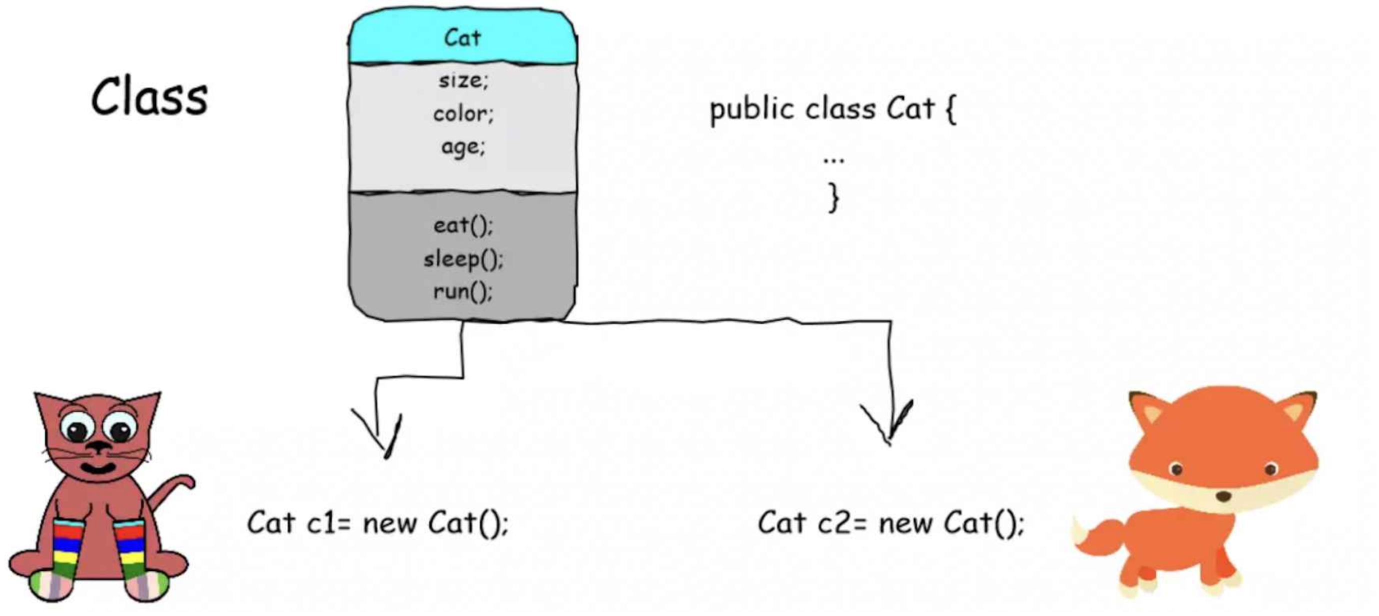
Object Oriented Programming - Java

- As one of the OOPs - Java, it complies with Encapsulation, Abstraction, Inheritance & Polymorphism fundamental programming concepts, called **`APIE`**. All these 4 concepts and programming mindset play a very important role throughout your developer life. In this course, we will go through it one by one seriously.



What is a Class?

- A class is a **blueprint** or **prototype** from which objects are created.
- It is **not a must to have the main method** in a class.
- The responsibility of creating and using new Cat objects may belong to some other class in your application.



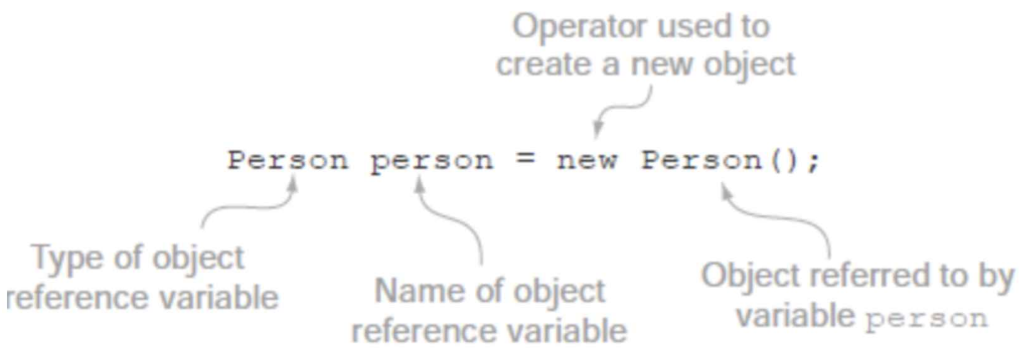
```java
public class Cat {
    private String name;
    private int weight;
    private String color;
    private int age;

    public void setName(String name) {
        this.name = name;
    }

    public void setWeight(int weight) {
        this.weight = weight;
    }

    public void setColor(String color) {
        this.color = color;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public void toString() {
        System.out.println("name: " + name +
                " weight: " + weight +
                " color: " + color
                " age: " + age +);
    }
}
```

- Another class with the "main" method to create two "Cat" Objects, which is produced by the Class of Cat. That's why we can say the Class Cat is a prototype.

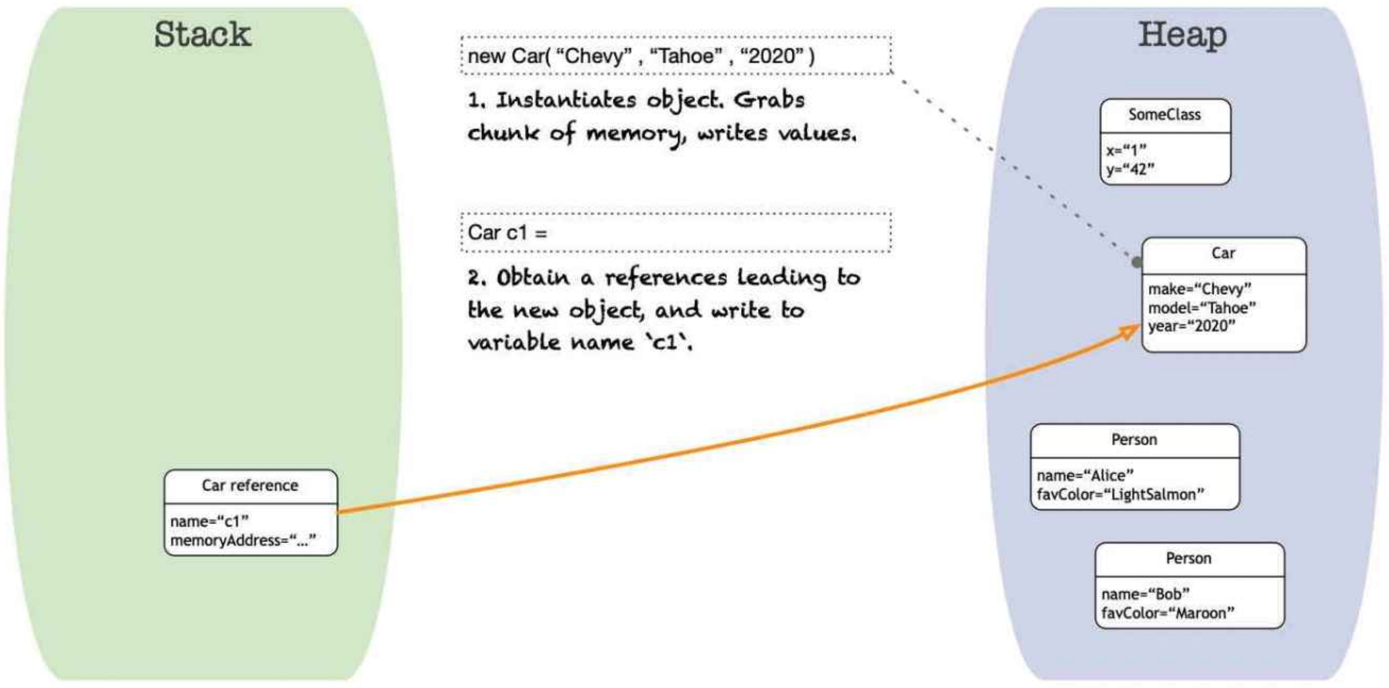
```java
public class CatDemo {

    public static void main(String[] args) {
        Cat cat1 = new Cat();
        Cat cat2 = new Cat();

        cat1.setName("Tommy");
        cat1.setWeight(4);
        cat1.setColor("Yellow");
        cat1.setAge(4);
        cat1.toString();

        cat2.setName("Ball");
        cat2.setWeight(6);
        cat2.setColor("Blue");
        cat2.setAge(7);
        cat2.toString();
    }

}

// Output:
// name: Tommy weight: 4 color: Yellow age: 4
// name: Ball weight: 6 color: Blue age: 7
```

Object References

- `Reference variables` are also known as *`object references`*. These terms can be used interchangeably.
- Objects are instances of classes, including both predefined and user-defined classes.


Operator used to create a new object
Person person = new Person();
Type of object reference variable
Name of object reference variable
Object referred to by variable person

- An object reference is, in fact, a **memory address** that points to a memory area where an object's data is located.
- When an object is instantiated with the **new** operator, a memory address value to that object is returned. That address is usually assigned to the reference variable.



Questions

- What are the two things that real-world and software objects both contain?
- Where is a software object's state is stored in?
- What is a software object's behavior exposed through?
- What is a class & object?
- What is the relationship between keyword "new", object reference, instance, memory address?