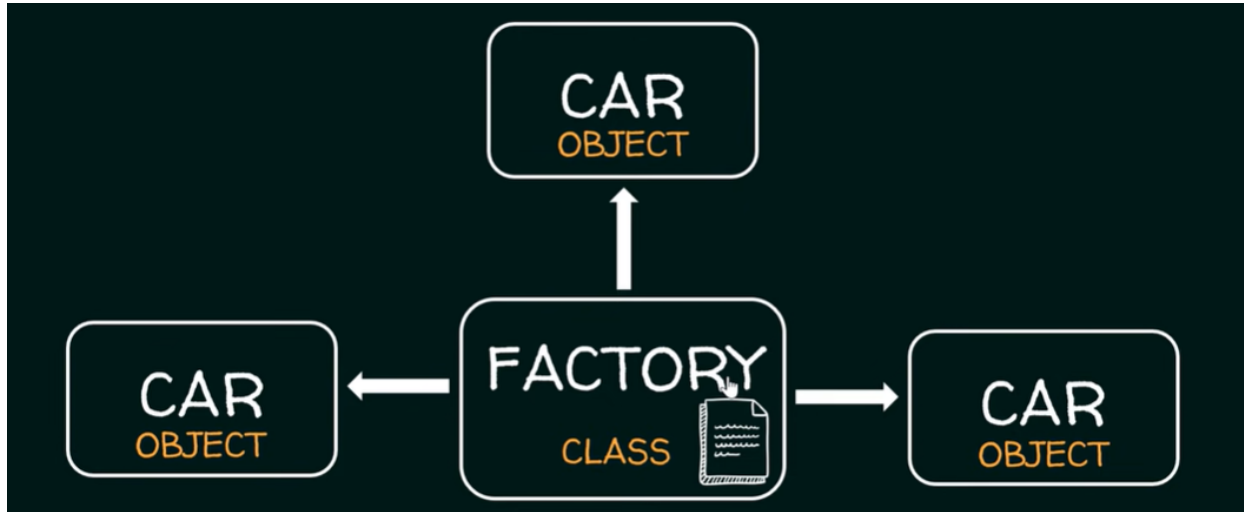# Lecture 4. Anatomy of Java Program

Java 程序剖析

Classes & Objects

Class is a blueprint to creat OBJECTS (类是创建对象的蓝图)

An object is an instance of a CLASS (一个对象是一个类的实例)

Class structure (类结构)

```
class class_name {     //'class' is a keyword(syntax)
    code block
}
```

Methods

Group of instructions to do a specific task (执行特定任务的一组指令)

We have a special method called 'main'

Method Structure

Each method consists of 4 main parts

```
return_type method_name( parameters.参数 ) {
    code block
}
```

Note: every method is written inside a Class.
→ A class is a cintainer of methods (类是方法的容器)

Calling a Method (调用方法)

Is basically using the method

```
method_name(give parameters);
```

→ The code block of this method will be excuted

Note: the main( ) method is automatically called when we run our java program

Access Modifiers (访问修饰符)

Specify how to access Classes & Methods — Discussed later in OOP

The four main access modifiers in Java:

- **public**

  公共访问修饰符是最开放的修饰符，允许任何其他类访问被修饰的元素。在Java中，public修饰的成员可以在任何地方被访问，无论是否处于同一个包内

- **private**

  私有访问修饰符限制了成员的最大可见性。private修饰的成员只能在声明它们的类内部访问，无法被任何其他类访问

- **protected**

  受保护访问修饰符限制了成员的访问性。protected修饰的成员只能被同一包内的类和子类访问。外部包的类不能访问protected成员

- **default (默认)**

  默认访问修饰符是当没有明确指定访问修饰符时的默认值。它的可见性仅限于同一包内的类。在不同包内的类无法访问默认修饰的成员

Naming Conventions(命名约定)

How to write names in programming

Pascal case convention (帕斯卡命名法):

标识符中的每个单词都以大写字母开头，并且没有使用空格或其他分隔符,也称为 "Upper Camel Case"（大驼峰命名法）
eg. ThisIsAName
Pascal case 用于类名，确保类名的首字母大写

Camel case convention(驼峰命名法)

- **Lower Camel Case(小驼峰命名法)**

  第一个单词的首字母小写，而后续每个单词的首字母都大写 → thisIsAName

  常用于变量名、方法名等非类名的标识符

- **Upper Camel Case(大驼峰命名法)**

  每个单词的首字母都大写，包括第一个单词。通常，大驼峰命名法用于类名和类型名 → ThisIsAName

Snake case convention(蛇形命名法):

用于标识符的命名，其中单词之间使用下划线 ⎽ 分隔，而且所有字母都通常使用小写字母 → this_is_a_name

Java Program Structure

```java
public class Main{
    public static void main(String[] args){


    }
}
```

Remember

- **Every Java program contains at least one class.**

- **Pascal case is used with classes.**

- **Camel case is used with methods.**

- **A method exists inside a class.(方法存在于类内部)**

- **main() is the starting point of execution of our program.(程序执行的起点)**

Packages

A container for Classes