北京圣思园科技有限公司
http://www.shengsiyuan.com

主讲人：张龙

内部类

- Group related classes and thus reduce namespace clutter
  defined at a scope smaller than a package

- An inner class can be defined inside another class, inside a method, and even as part of an expression

- There are four types of inner classes
  - static inner classes (also called nested classes)
  - member inner classes
  - local inner classes
  - anonymous inner classes

静态内部类

- The simplest form of inner class
- Can’t have the same name as the enclosing class
- Compiled into a completely separate .class file from the outer class
- Can access only static members and methods of the enclosing class, including private static members
- Create an instance of a static inner class out of enclosing class:
  - **new outerclass.innerclass()**

成员内部类

- Defined in an enclosing class without using the static modifier
- Like instance variables
- Can access all members of the enclosing class
- Create an instance within the enclosing class

  this.new Innerclass();

- Create an instance out of the enclosing class

  (new Outerclass()).new Innerclass();

- Access members of the enclosing class within inner classes

  Outerclass.this.member

局部内部类

◆Defined within the scope of a method, even smaller blocks within methods

◆The least used form of inner class

◆Like local variables, can’t be declared public, protected, private and static

◆Can only access final local variables

匿名内部类

- Local inner classes which don’t have class names
- No key word class
- No key word extends and implements
- No constructors
- Implicitly extend a superclass or implement an interface