2022 06 30 part 1

2022 06 30 part 2

2022 06 30 part 3

音质问题，没看。

----------------------------

2022 07 01 part 1 <https://youtu.be/TwYYZ25Jb7w>

oop

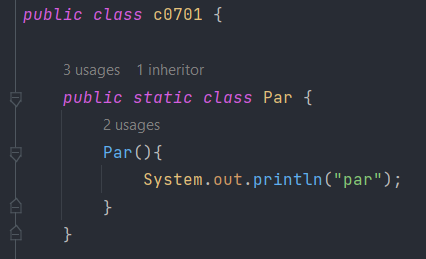
Polymorphism (override and overload): “change behavior of the function”, change ways to look at things.

Inheritance/extend "is a"

Encapsulation "has a" （encapsulation:everything needed for the class, we have it, just grab and use. dont care about inside fields, methods....）

(!!! reusability)

inner class need to be static



---Abstraction: no methods’ implementations. 有 { } 就是implemented.



abstract function必须要写 “abstract” keyword/修饰 .

如果class里有abstract function,他是abstract class.

Abstract class里可以只有concrete function.

Abstract class不可以实例化。可以static。可以被另一个abstract class extend.，

一个concrete class只有在@override 了abstract method时才能extend 一个abstract class.

@override 在compile time的时候check有没有真的在override

有些annotation是compile time,有些是run time

可以override abstract function.

Abstract class可以extend自concrete class.

---polymorphism 包含downcasting, upcasting, (override and overload):

两种允许的casting:

1.Parent p = new Child(); explicit/implicit upcasting.

这里的p，不能用Parent没有的method，但如果Child里override了function，走的是Child的定义。

1. Child c = (Parent)p; explicit downcasting.

这里从p可以downcast成Child，因为这个object本身 “is a ” Child.

upcasting总是可以的。

downcasting的object必须 ”is a” type。

-overload （polymorphism）

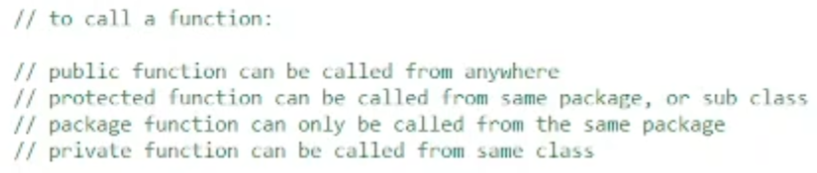
只要function名一样，其他有一点不一样，就算overload；但20年前return type也要一样。

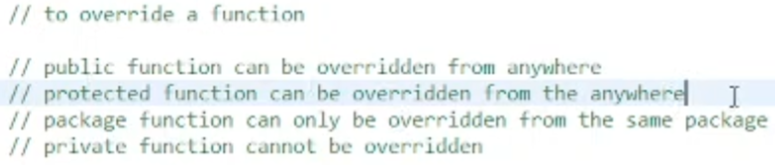
--------------------------

2022 07 01 part 1 <https://youtu.be/mlNsw9wxClc>

--Accessibilities (public private protected)

直接跳到30分钟看的结论：





--java bean， 也叫 POJO， POJO BEAN

The most basic operating unit of a java code.

1. Constructors (one from fields and one from superClass)
2. Getters && setters
3. Functions extended from Object class ( no logic in java bean. 要不然call functions的时候不知道是哪个bean。 )

Steps to create an object:

1. Declaration 2. instantiation (new keyword), 3. initialization (following constructor)

New keyword

1. Find the class from all loaded classes
2. Allocate memory for the instance
3. Create object
4. Call constructor to initialize object
5. Return the address

Constructor has NO, CANNOT have return type. Not even “void”.

Constructor DOES NOT construct/create an object, it ONLY initialize it. Without the object, cannot call constructor,

SO constructor is object level function.

Constructor will be called only once per each object for initializing.

Once initialized, CANNOT call constructor again.

If no constructor provided, then a default one, if any constructor provided, no more default one.

Constructor can be overloaded.

Constructor can NOT be overridden.

Constructor can be private. => its child class won’t be able to call super, CANNOT initialize.

Java.lang.Object : concrete class, CAN new Object. This class has its constructor…..

Object.hashCode() ‘s return type is int.

--------------------------

2022 07 05 part 1 <https://youtu.be/rlJqDDefBic>

Final, interface.

Everything in interface cannot be concrete, CANNOT implement anything.

By default, functions in interface is : abstract (no body), public.

By default, fields in interface is : public static final.

标准答案

Final: is a keyword, can be added in front of a function, or a class, or a variable.

Final variable ‘s reference cannot be modified, the object still can be changed.

Final class cannot be extended.

Final function cannot be overridden (but can be extended).

How many different ways to assign value to a final field?

Static:

1. Assign when define, static final field, MUST assign value when defined. ( in java, no constant, this is CONSTANT way. )
2. Assign in static block. ( static block will be executed after the class-loading ). but once assigned cannot reassign.

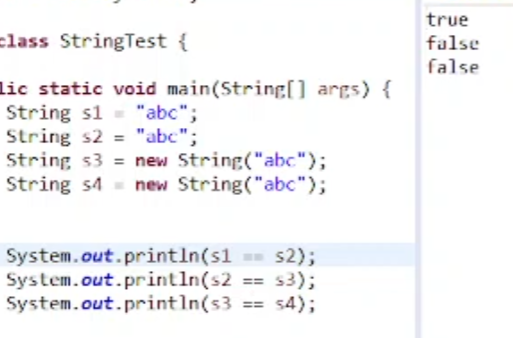
Non static:

1. Assign when define.
2. Assign in block
3. Assign in ALL constructors.

Class vs interface:

1. Class can implement interface, but not the other way around
2. A class can have concrete functions, but not in interface
3. A concrete class can be new, but not an interface / abstract class
4. Class canNOT multi-extend (diamond issue), interface is ok
5. Class can have private / protected / package, not in interface
6. Fields in class can be any, but interface fields is only public static final

String Pool: (because string is immutable)



Java check StringPool first when make a String object.

--------------------------

2022 07 05 part 2, <https://youtu.be/OcVXxuzpvHY>

Immutable class: the OBJECT of this class cannot be modified. Class itself CANNOT be immutable. CANNOT change class in runtime.

Multi-thread “conflict”: risk condition.

Thread safe: this shared resource is protected by 3 strategies:

1. asynchronization,
2. copy to each thread,
3. read-only

Immutable class(

private fields,

no setter only getter,

maybe default constructor,

use Collections.unmodifiableList() unmodifiableSet/Map(),

**For customized class type field, make sure it is also immutable,**

Should not provide any function to modify the field,

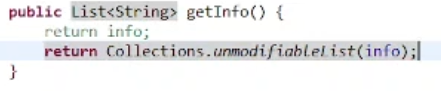
**final the class so no sub class can override the methods,**

No actual immutable class in java (reflection)

): read-only

In java: 2 types of classes: bean(no logic, field value matters), functional class(no field or internal use fields).

Immutable class normally used on Java Bean.



So that cannot call getInfo.add( element ).

This is static function, because we can directly call it.

Call add on this List now, will : 

In Collections class, all methods are static.

Class contains only static methods: util / helper class.

Example: Math, Arrays, Executors (always plural)

Non-static functions: object level.

Final variable ‘s reference 不能modify, 但是 object still can be changed.

Immutable class , 但 reference 可以。

-----------------

2022 07 05 part 3 <https://youtu.be/C3DBghT2Yxk>

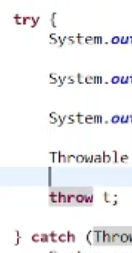
Exceptions

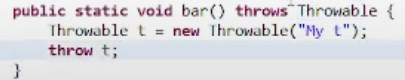
Throwable - concrete Class

Serializable Externalizable Cloneable Comparable - interfaces

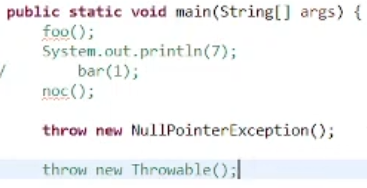
Throw~able == catch~able

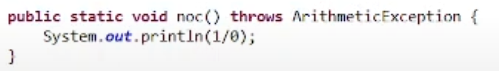
Handle Throwables:



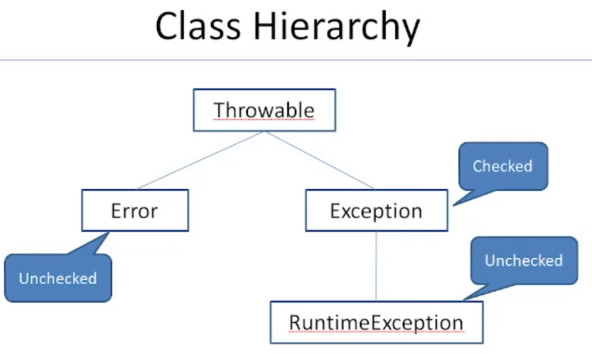
 可能会throw

call这个method的method也要handle （比如main）

 unchecked exception不用handle



这是RuntimeException，不一定把handling写出来

 checked Exception: Compile time exception.

Most famous RuntimeException: NPE: NullPointerException



Most famous Error:

java.lang.StackOverflowError,

java.lang.OutOfMemoryError

Exceptions Examples :

Checked Exceptions :

SQLException,

IOException

Unchecked Exceptions :

ArrayIndexOutOfBoundException,

NullPointerException,

ArithmeticException.

Throwable is checked. 代码上里必须handle

Error 也可以被throws



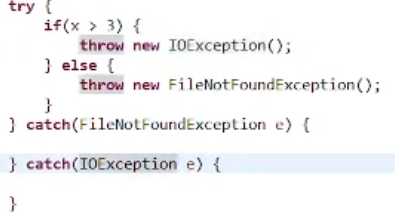






Concrete classes





如果多catch里的exceptions有inheritance, 要把child exception放上面，因为只能catch一个



如果没有继承关系，除了多catch，还可以

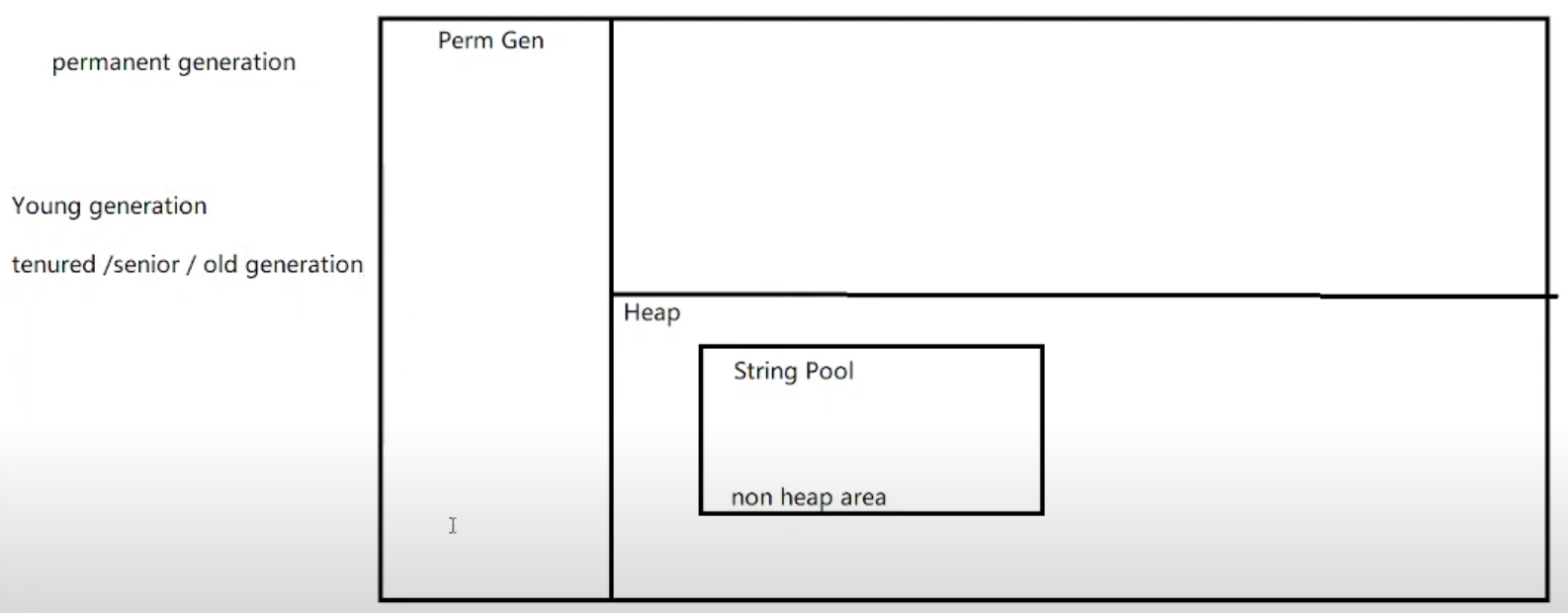


Catch出来记得print



==Perm Gen / Meta Space

JVM:



这些generation是给garbage collection用的。

在1.8之前，classes load到Perm Gen （is a special heap space separated from the main memory heap）.

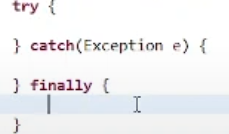
后期要load的class太多，放不下，会出问题，就放Meta Space.。 （是否还存在 有争议）

-----------------

2022 07 06 part 1 <https://youtu.be/U4c04CYMFuM>

跳着看的

Final, finally, finalize





-----------------

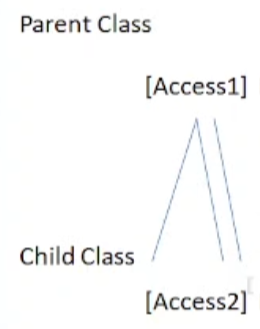
2022 07 06 part 2 <https://youtu.be/NpAkBl4rk74>

Clone and cloneable

In Object class

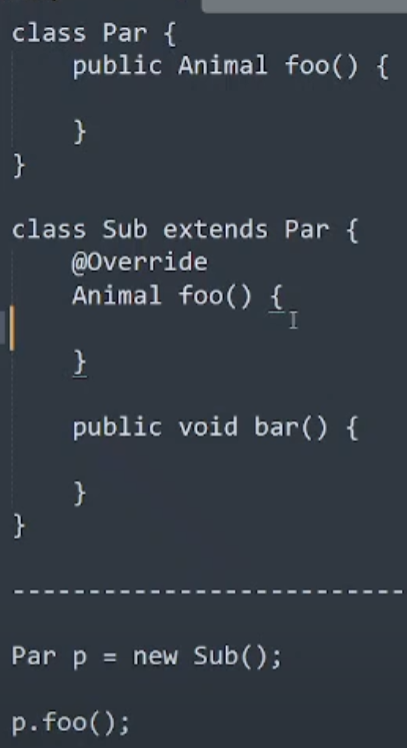


Native: concrete function

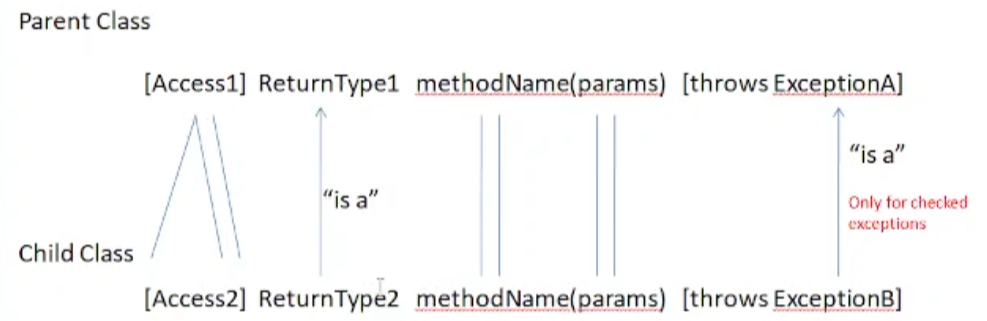


Private method in parent is not visible to child, so CANNOT override.

Other @Override, gets wider accessibility,



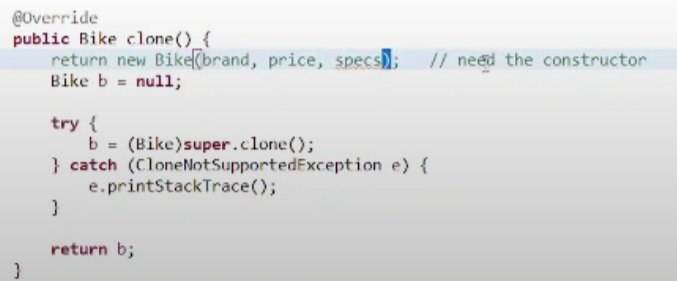
因为p是Sub object，所以p.foo()是在call Sub里Override的foo



-----------------

2022 07 06 part 3 <https://youtu.be/EFGV37QFNN8>

Clone,



为什么用try catch，因为如果是throws，所有用他的method都要跟着写throws CloneNotSupportedException。

需要clone的class要

JVM needs to see this, because “needs permission” to clone.

Marker interface: no functions, empty

Clone object:

Shallow copy (上面写的@Override clone方法)

different reference, but inside - same.

Deep copy

No easy way.

New Bike(...) 这不是deep copy，因为java pass by value(address) , Bike确实new了，但里面的class object没有重新new。

除非要掉 super.clone()，否则不用implement Cloneable



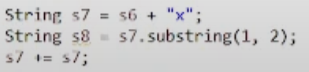
这里造了几个String？

1. String literal in String Pool
2. S5 / new String(“xyz”)



第二行造了几个String？

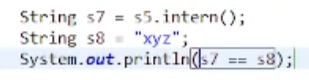
1. Xyz已经在spring pool里了，只造了 s6 / new String(“xyz”)



42:40 spring pool weird questions.

42:50 可以用javap看

String.intern()

 //returns true

47:15 开始，到结尾没看

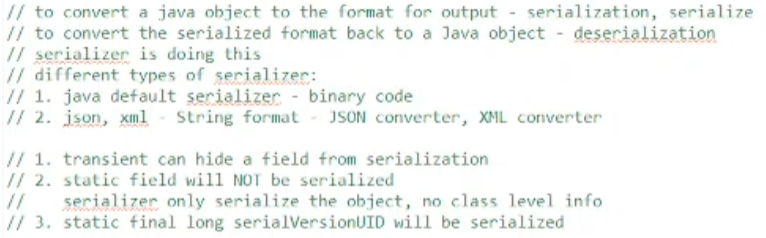
-- IO

-----------------

2022 07 06 part 4 <https://youtu.be/V8gYjPVgz2w>

IO, Serialize

没看





-----------------

2022 07 07 part 1 <https://youtu.be/l46zL_CldZo>

0 - 48:00 Serializable, Externalizable 没看

41:08 NIO : Non-blocking IO

48:00- Thread

Runnable : interface

Thread: concrete class



Runnable:

run()

Thread:

static:

sleep(long ms);

yield();

The thread which executes the yield method will enter in the Runnable state from Running state.

Once a thread pauses its execution, we can’t specify when it will get a chance again it depends on the thread scheduler.

currentThread();

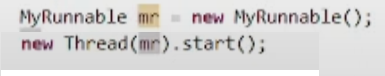
non-static:

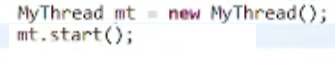
run(); define the job for the thread

start(); create the thread and the new thread will execute run()

join();

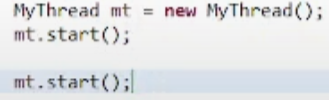
2 ways to create thread.





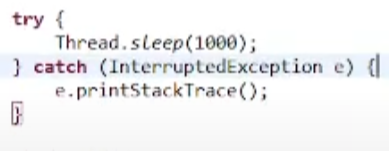
-----------------

2022 07 07 part 2 <https://youtu.be/6yDw9FkgHs4>



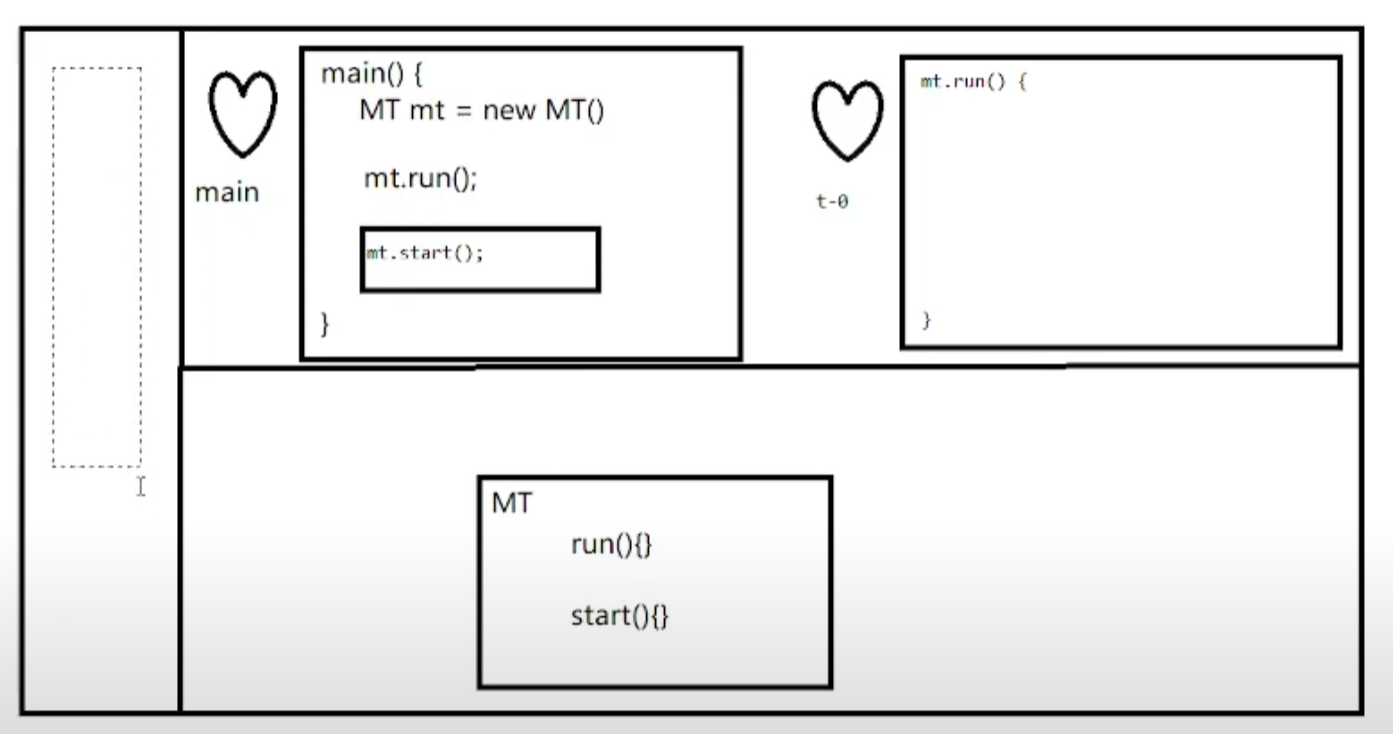
只能call一次start，要不然illegalThreadStateException

Static sleep

 需要try catch

这段代码在哪里，哪个thread就会sleep。在main里，main thread 就 sleep

Most famous deamon thread: Garbage Collector



左边的区域，load class的，以前（<1.8）是perm gen，现在是meta space.

右上是stack，右下是heap(放object)

Garbage Collector也是可以clean stack里的东西的，因为他们某种程度上也在heap里

Garbage: an object that has NO reference.

这里的清理：object还在heap，但reference没了。

Garbage Collection algorithms:

1. 要考虑garbage的定义：garbage check
2. 如何清理（多块，多彻底）

Different types of references in Java? GC related.

Shared resource (thread safe)



Synchronize 同步，意思是：一起做，但不一定同时开始

Synchronize keyword 没法保证methods们哪个先开始

Synchronized keyword哪个lock，必须是object

-----------------

2022 07 07 part 3 <https://youtu.be/FSQvlLKS1-k>

Deadlock，java.util.concurrent package

--DeadLock: 2 or more thread are waiting for the lock to be released by the other lock, in a loop.

Implements Runnable 比extends Thread的创造线程方法更好，因为他可以extend。

但代码要先写extends再写implements， 

为什么Thread.sleep(1000) 必须要用try catch 来，不能让run throws InterruptedException;

因为Runnable / Thread 里的run不throws任何东西，override的run（）的exception要跟parent有”is a”关系，也就不能用throws，

InterruptedException是checked Exception，必须要在代码上handle，所以只有try catch一种handle方法。

--java.util.concurrent package

Live Lock:

ReentrantLock

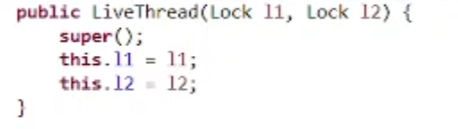


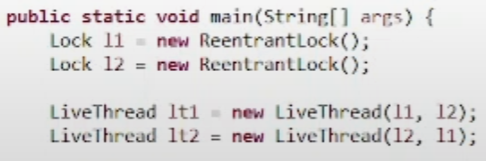
Lock: 这里不是synchronzed的“lock”，是一个interface

--ReentryLock







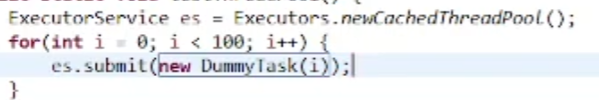


L1.trylock()

L1.unlock()

--Thread Pool



 Infinite ThreadPool

可以Submit runnable或者callable

 要不然要等一会才会关jvm，因为那些thread要等一会才会“没”。

但他不会把已经submit的task干掉。

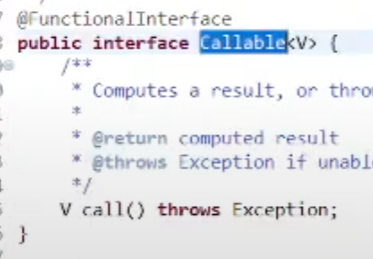


Thread pool 不一定是 multi pool

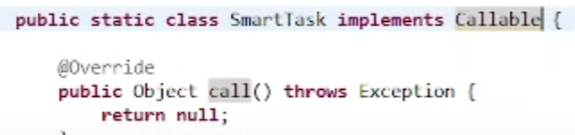


Finite pool

--Future, Callable

 return V:

Generic: variable for type



Runnable里有run()，run() return void

Callable里有call()，有自己的return type

 from java.util.concurrent

Future像promise，不一定有结果的时候就可以

里面有。List<Future> 可以保证结果的顺序跟传入的顺序一样

-----------------

2022 07 08 part 1 <https://youtu.be/wR9Xp1K0dZE>

-19:24 : 他们的考试, yield join

19:24 : Collection framework

Thread:

Static yield()

其实就是降低thread 的priority

Static，都能call ( Thread.yield(); )，在必要的时候才等，

ready to pause its execution.

The thread which executes the yield method will enter in the Runnable state from Running state.

Once a thread pauses its execution, we can’t specify when it will get a chance again it depends on the thread scheduler.

Void join()

t.join(): call了join()的thread（比如main），要等t线程（t不是“线程”，是对象，但a对象创造了新线程 : ( Thread a)）结束。

If a thread wants to pass its execution to give chance to remaining threads of the same priority then we should go for yield();

If a thread wants to wait until completing of some other thread then we should go for join()

--collections

-Generic

<Generic > ( variable for type ) must be a class

- (auto) boxing & auto-boxing

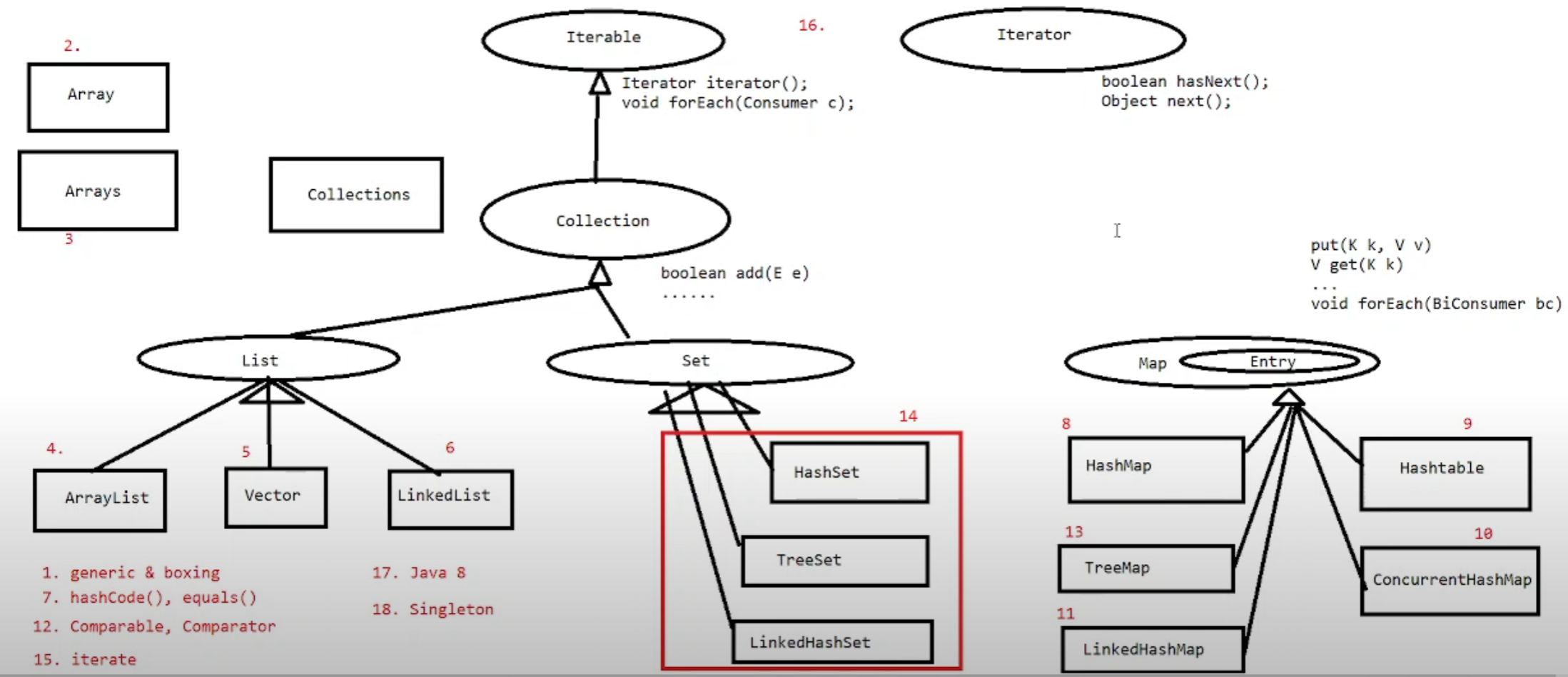
Convert int (primitive type) to Integer (class)

A java feature: boxing done automatically.

Boxing: primitive type -> wrapper class

Unboxing: 反过来

Only local primitive variables and references to object (i.e. variable declared in method) are stored in stack. Others are stored in heap



What is thread-safe:

different threads can access the **shared resources** without exposing erroneous behavior or producing unpredictable results.

the process to make our program safe to use in multithreaded environment,

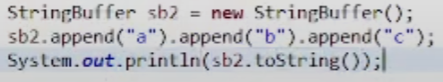
-- String Builder





Return type of this .append() : StringBuilder object.

-- String Buffer



Looks same.

String Builder & Buffer Difference: String Buffer, same functions, but have “synchronzed”.

Both Builder DP: build something piece by piece.

Always use StringBuilder, because fast.

<讲collection>

-----------------

2022 07 11 part 1

-----------------

2022 07 11 part 2

-----------------

2022 07 11 part 3

-----------------

2022 07 12 part 1

-----------------

2022 07 12 part 2

-----------------

2022 07 12 part 3

-----------------

2022 07 13 part 1

-----------------

2022 07 13 part 2

Singleton

</讲collection>

-----------------

2022 07 14 part 1

DDL

DDL: data definition Language

-----------------

2022 07 14 part 2

SQL

以下跳着看的

-----------------

2022 07 15 part 1 <https://youtu.be/9n0M7DerE2U>

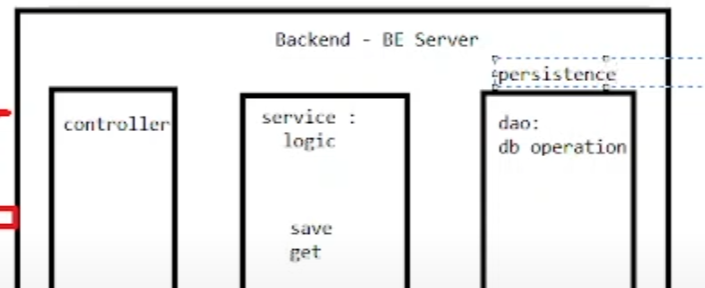
-2:04:04没看

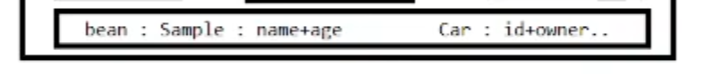
MVC: model view controller

Model: data model + logic

View: front view

Controller: receive requests and trigger what function



 bean 不是layer，对应table而已

3 layers,

Dao ( data access object ) also called persistent layer

DAO vs DTO (data transfer object)

-----------------

2022 07 15 part 2 <https://youtu.be/2dOb4Ktb2fw>

JDBC

JdbcUtil.java 管连接的

-----------------

2022 07 15 part 3 <https://youtu.be/BT2KddgchNE>

这个好好看了

Hibernate

JDBC:

1. Create connection
2. (with connection) write statement
3. (use statement) execute SQL
4. ResultSet (get result)

Hibernate (matching):

Table bean

Column field

Key column key field

define上述relation：

就no need to :

1. Write sql
2. Write try-catch
3. Write while{ ...ResultSet...}

Hibernate是framework， 这种tool叫ORM ( object relational mapping ) Tool

用hibernate步骤：

1. Import Hibernate dependency, 用maven
2. Tell Hibernate how to connect to DB
3. Define ORM ( object relational mapping )
4. Call hibernate API to operate DB

Hibernate internally uses JDBC

就像jdbcUtil.java， hibernate也有HibernateUtil

Hibernate 不用connection， 用session；session怎么创造，用SessionFactory

-factory

又是design pattern

Create things easily, with some but limited level of customization

-session

“Conversation”

Session holds information.

session里的info是stateful的。Session makes things stateful.

Stateful vs Stateless

东西有没有连续性/阶段性（就像lifecycle）

State: current status of everything

Session put variables and their values in HashMap.

在jdbc里，每次call get() ， 都query一下db。花时间，不一定必要。

Jdbc就是stateless的，everything individually executed, result not stored.

Hibernate keeps results, using sessions.

session需要知道怎么连数据库，Factory才能工作：

 “/”表示class path

里面是username, password, driver, url......

都传给了session-factory

hibernate里有dialect，针对certain DB，不用sql的

format\_sql: true: 有换行的SQL语句

ThreadLocal里面的东西，各个thread去拿的时候，拿到的都是copy。

用Copy的方法处理multi-thread.

ThreadLocal.initialValue.

-----------------

2022 07 18 part 1 <https://youtu.be/a07HmQiX3lk>

这个好好看了

前面讲什么retake：

Immutable class里面的customized class也要是immutable的。

Mapping是那个xml（extensible markup language）文件：

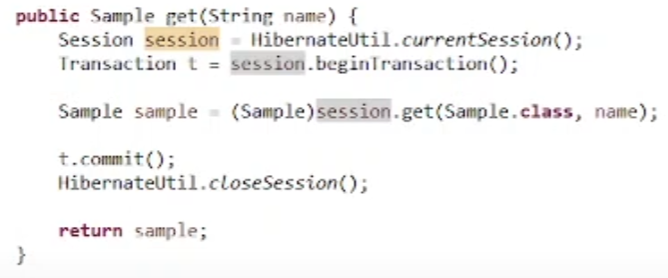
里面是哪个DB里的table 对应 哪个bean，还定义了primary key(叫id)

这个mapping也是要放到HibernateUtil里给factory看的

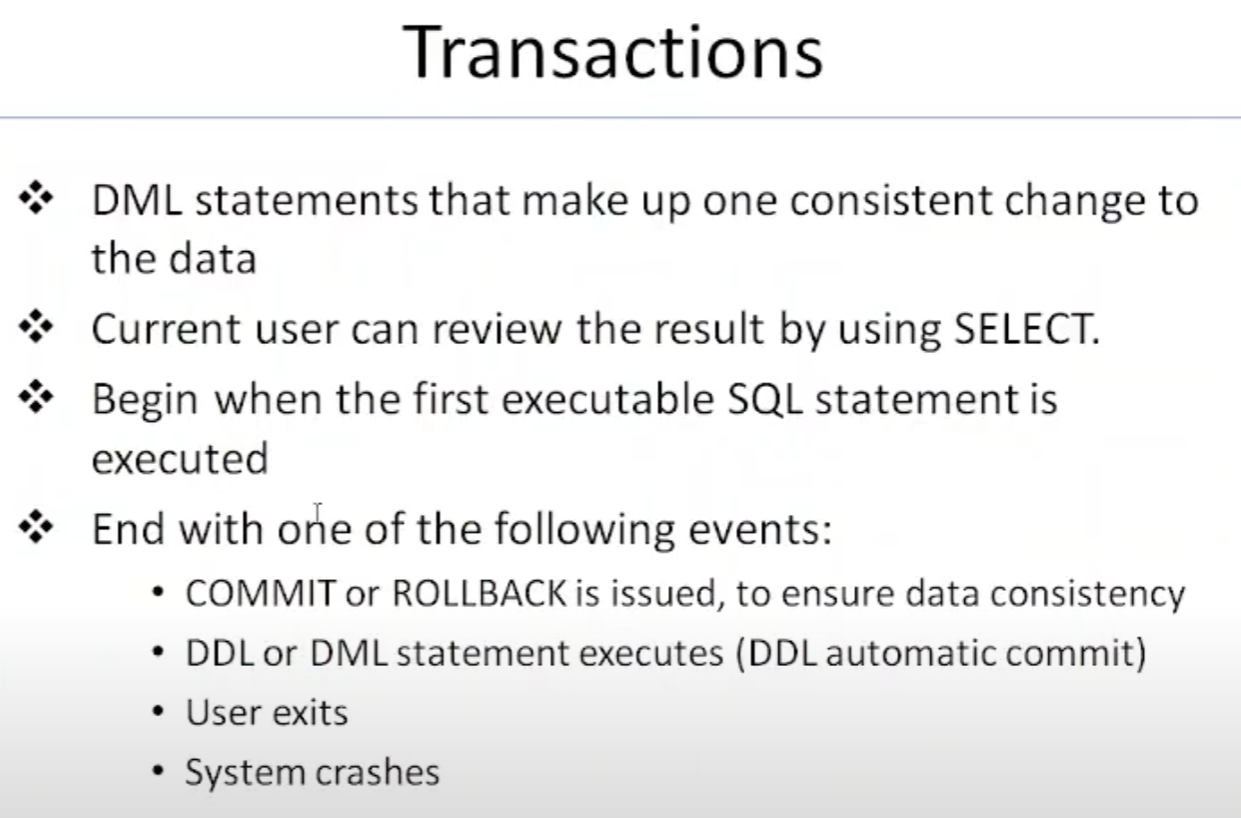
Hibernate DAO里定义了logic

每个logic：





Transaction：

 SQL.pptx 第51页

！！！What is transaction?

A group of operations. 并解释下面的ACID

4 important features of transaction:

Atomicity - all operations of this group, either all done (commit), or all not done (rollback). (苹果同步的故事)

Consistency - what do now will impact future. All operations need to have relationships.

Isolation - all transactions use same isolation level

Isolation levels:

- read uncommitted:

read other transaction info before the other transaction is committed. (dirty read: dirty data/ intermediate data)

No need for other transaction to finish, fast fetch.

Use only when only one operation. Or no rollback. So no dirty data created.

不等于auto save, auto save只是缺了rollback选泽。

- read committed （默认level）

等。

- repeatable read

Repeatly read a local value ( 第一次取了一个值，持续用 ).

At least within this transaction, everything correct. Could get old value.

- serializable （最安全，经典（都带）的level）

One by one, like synchronized.

One transaction working on a DB, other transaction CANNOT EVEN read, must pause.

Durability - the result will be persisted. Commit或者rollback了，这个transaction（的结果）就固定了.

Java, a thread CANNOT read value from another thread, because they have their own stack, own local variables. But if variable in heap is ok.

Java serializable: from JVM to outside, needs serializable.

Session is a thing to save states, has HashMap inside.

Transaction, everything before t.commit(), do them.

transaction不操作数据库，只保存operations, session是operate on DB.

@org.hibernate.MappingException unknown Entity

-----------------

2022 07 18 part 2 <https://youtu.be/e9o1QyeTczo>

跳着看的

都Hibernate 了就不用写sql语句了，hibernate帮弄。

Different hibernate entity object state (有具体的code)

1. Transient state: an ojbect only in java, but not in hibernate session for operation to DB
2. Persistence(attached) state: this object currently in session. (if commit, db will update)
3. Detached state: this object used to be in session, but not any more

Session has cache (session 可以保存state，stateful的表现)

HibernateUtil.closeSession() 之前的都是一个session。

Session factory level cache (session factory dependent) (cache separated by different session factory)

Second\_level\_cache （默认true）

Hibernate’s local cache: ehcache. （用的LRU - least recently used algorithm）

要让两个session都能share这里的cache，需要他们来自同一个session factory

Vs

Session level cache (1st level cache)

HQL: hibernate Query language

-----------------

2022 07 18 part 3 <https://youtu.be/fdf8-ntq3Dk>

HQL, get() vs load(), criteria, named query, 好好看了

1：00：40 - spring 跳着看的

--get() vs load()

1.

get() is eager,

马上去db查

load() is lazy

需要用到取来的object时再查，比如sout

之前，使用dummy/proxy object

2.Both get() and load() use cache

1. If id not exist, get() returns null, load() exception out

Load() should be in try-catch

--criteria



“age > 30 and age < 60”叫criteria

“age > 30”或“age< 60”叫criterion

用到了streamAPI, builder Design Pattern

--queryCache

Query 不用cache，直接去db取

但可以setCachable(true)



这里默认是false

QueryCache用的是factory ( 2nd ) level cache

--spring

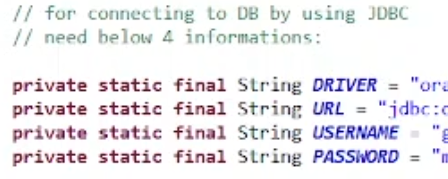
是framework，里面那些什么Spring IoC, Web, Data JPA, Security, AOP, Cache, Boot....都是library

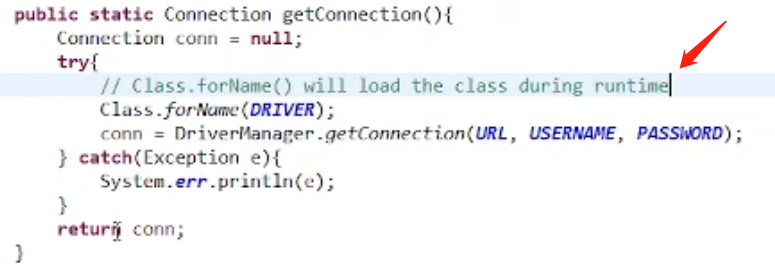
-----------------

2022 07 19 part 1 <https://youtu.be/1o9bk22yGOc>

跳着看的

From JdbcUtil.java





CAN load class during runtime.

SNAPSHOT version: 第一个test version

一个spring app可以有多个spring ioc container.

Container:

ioc container, create spring bean/components, wire them together, manage their life cycle.

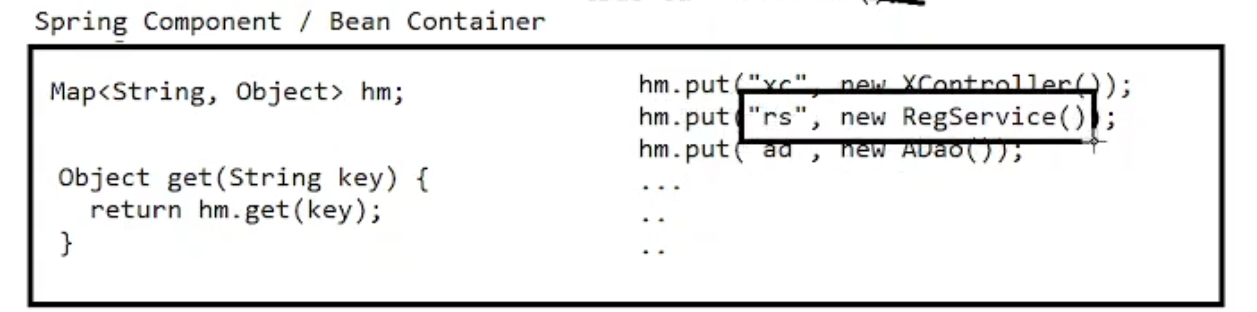
A Spring IOC Container is the BeanFactory or ApplicationContext .

If there are more than 1 application context, Spring will make sure the Singleton remains a singleton in the context of the container.

--spring ioc（inversion of control）

需要service, controller, dao ( 这些都是只需要一个就够了，所以singleton是一种solution )不用new，直接”manager” get （解决shared resource 问题，也省资源）

Manager就一个object，里面用Map



Spring给了个solution，我们就不用define manager了

这就是ioc（inversion of control），

这里的manager叫Spring Component Container,

Spring component 也叫spring bean

Spring bean/component:

The entire combination including the object/class （可以是service, controller, dao）+ a name, and some other features...

Hibernate entity:

Class + mapping relationship

But some class i dont want to put in the Spring Component/Bean Container:

Java Beans, DA BEAN.

有了ioc（inversion of control）以后，就不用造container，不用new， 不用get，只需告诉container: 1.class, 2. name, 3. use the spring bean/component object. Component们就可以直接用。

-----------------

2022 07 19 part 2 <https://youtu.be/PNKN38ADGhE>

跳着看的

51:54 - 1:18:00 spring bean scopes

To create a bean , @Autowire must be finished.

@Autowire : 去container找xxx object，就不用new了

spring bean scopes

以前有5个，现在就4个

1. Singleton ( 默认scope )

If any spring bean: 1 spring container, this 1 bean(including name, class, something else), 1 instance------this bean is a singleton.

1. Prototype

LIKE “new”.

Each time request an object from the container, new object will be created.

If @Autowire a **prototype bean** from 2 different places, we get 2 objects.

1. Request

Definition in web.

If one object created once per **request**, this bean is a request scope bean.

1. Session

Definition in web.

If one object created once per **session**, this bean is a request scope bean.

1. (global session)

If one object created once across multiple **sessions**, this bean is a request scope bean.

Singleton: Design Pattern, 1 scope, 1 definition, 1 instance

Java singleton: 1 JVM , 1 class , 1 instance

Spring singleton: 1 spring container, 1 spring component/bean, only 1 instance, we say this BEAN is a (spring?s) singleton bean

@Component: define a spring bean.

@Scope(“singleton”): defines a singleton bean. (默认就是singleton，所以从来不写)

@Scope(“prototype”): defines a prototype bean.

 如果我们把ADao define为singleton，aDao和aDao2会是同一个object；如果是prototype，就不同object。

Spirng IoC - an actual product that implements the idea of Inversion of Control

Ways to achieve IOC:

DI - Dependency Injection (spring ioc uses DI to achieve IOC)

DL - Dependency Lookup

Different injection types:

1. Constructor injection
2. Setter injection
3. Field injection

-----------------

2022 07 19 part 3 <https://youtu.be/em_9qMK9yhI>

Spring web, 我们也叫他MVC （model view controller）

Model: data+logic

View: front end

Controller: guide request form view to model (map request to logic)

以下咣咣跳

-----------------

2022 07 20 part 1



-----------------

2022 07 20 part 2

-----------------

2022 07 21 part 1

-----------------

2022 07 21 part 2

-----------------

2022 07 21 part 3