Java内

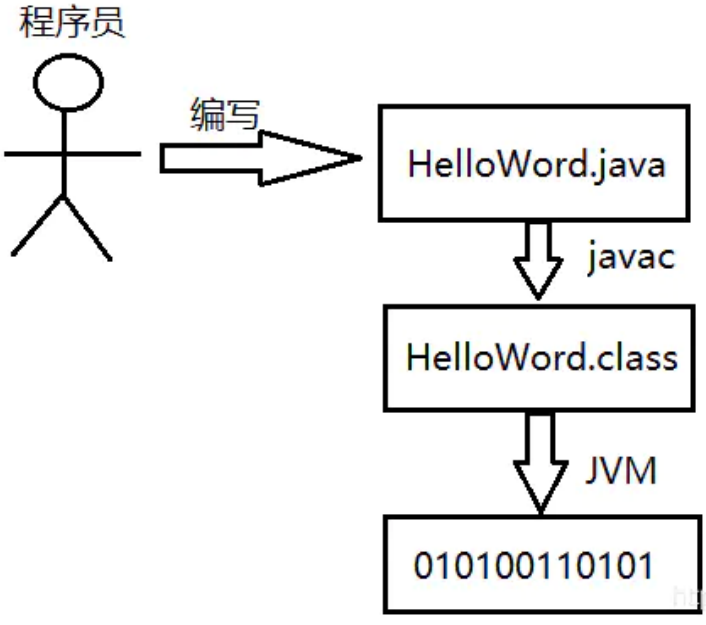
1. 们 发人员 写 Java代

- 了 二 ,
- He.. W d.ja a
- He.. W d.ja a , ,

Java 件

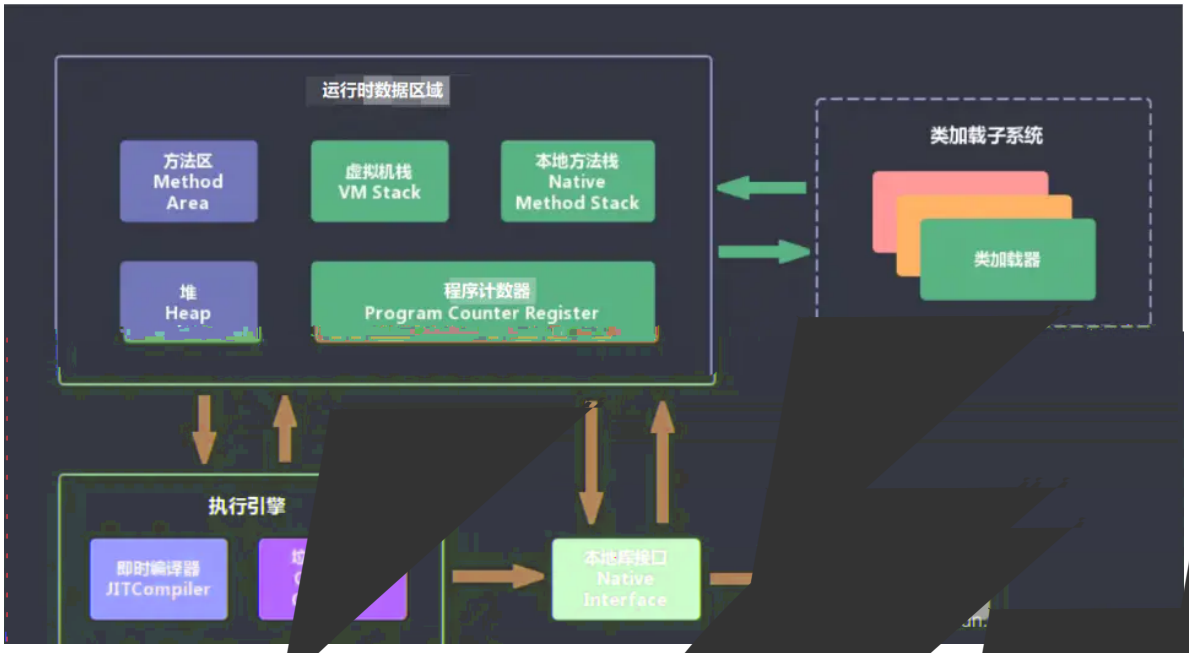
1. .ja a
2. ja ac .c.a : 么 c.a 为JVM .c.a
3. JVM (于)

（ 是一个大概的观念 抽 画的概念）



2. 为什么 java 台

- 个 中 (JVM)
- Ja a JVM 了

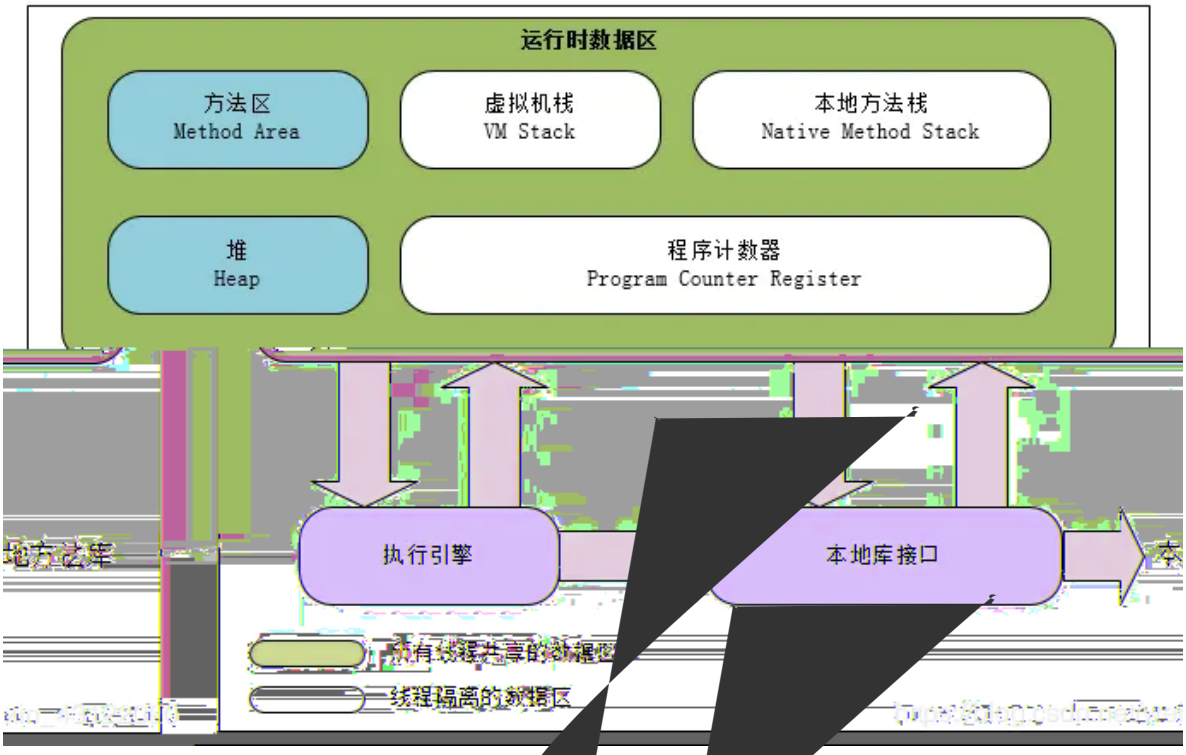


- JVM 两个 两个 : 两个 为C.a ec i
  ); 两个 i e da a a ea( ) Na ce(
  - C.a): (a..a g.Objec)
    R i e da a a ea中 e h d a ea
  - E ec i e gi e ( ) : c.a e
  - Na i e I e face( ): 与 a i e .ib a ie 互
  - R i e da a a ea( ): JVM
- : Ja a , (C.a L)
  中, (R i e da a a ea) , JVM
  , 不 ,
  (E ec i E gi e) , , CPU , 个 中
  (Na i e I e face) 个

5. 下JVM 区

- Ja a Ja a 中 为 个不
  些 , , 些 , 些
  Ja a 为 下 个 :

简单的 就是我们java 行时的东西是放在 的



- (P g a C e Regi e , 个 , 下 , , 个 ;

  为什么要线程计数器？因为线程是不具备记忆功能

- Ja a (Ja a Vi a. Machi e S ack) 个 Ja a 中 个 (S ack F a e) 于 ;

  栈帧就是Java虚拟机栈中的下一个单位

- (Na i e Me h d S ack) : 与 , 不 Ja a , 为 Na i e ;

  Native 关 字修 的方法是看不到的，Native 方法的源码大 分 是 C和C++ 的代码

- Ja a (Ja a Hea) : Ja a 中 , , 乎 ;
- (Me hed A ea) : 于

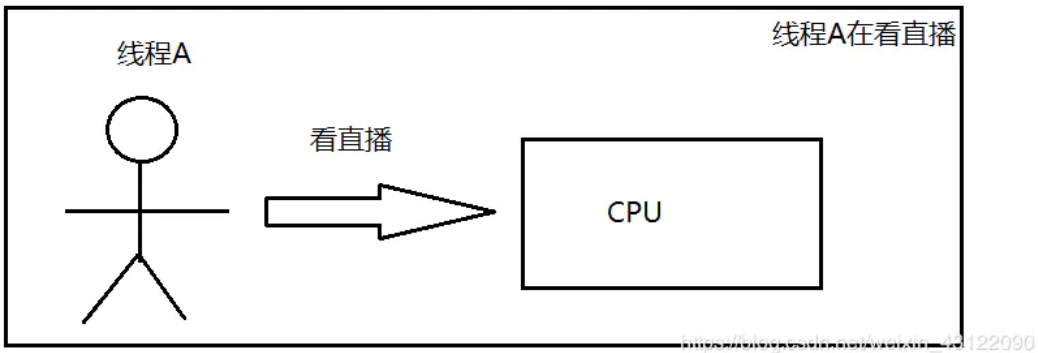
后 有 细的 明JVM 行时数据区

6. 介 下 ? ( )

1. , :
   ( )
2. 于Ja a , 个 中 , 为了 , 个 , 个 之 互不 , 之为 中 O OfMe E

总结：也可以把它叫做线程计数器

- **例** ： ja a中 ， ，
  ， CPU CPU上 ， 个 不 ， ， 个
  于 ，也 个
- **假** ：
  - A



    - ， B 了 个 ， A ， 了 A， A

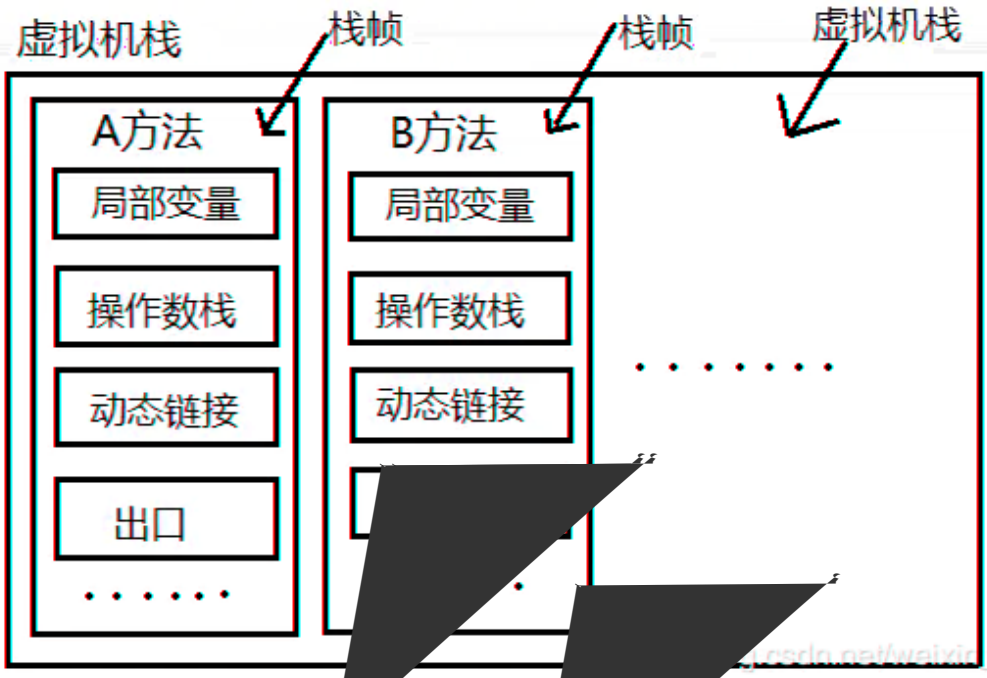
```
![在 插入图片描 ](https://user-gold-cdn.xitu.io/2020/4/13/171729fcc70da181?imageView2/0/w/1280/h/960/format/webp/ignore-error/1)

* 然后，视 电 结束， 时线程A究竟 干什么？ （线程是最小的执行单位，他不具备记忆功能，他只
  去干， 个记忆就由：**程序计数器来记录**）
![在 插入图片描 ](https://user-gold-cdn.xitu.io/2020/4/13/171729fcc90c8a88?imageView2/0/w/1280/h/960/format/webp/ignore-error/1)
```

7. 介 下Java ？（ ）

1. Ja a ，
2. Ja a ：每个方法在执行的同时 个 (S ack F a e) 于

- ： 中 ， ， 个 个 个 中 ，
  ， ， ，



:

1. : 临 8个 e Add e
(e Add e 中 e)
2. : , 中 个 i = 6*6,
, , 中
3. : 中, 个 e dd() , 中 , ,
4. : , e 不

个 另 个 , 会创 吗?

- : 个 中 , , 中 ,
个 个 , 个 下

向 什么 ?

- 么 , 中 么 , 中不 , 个

会创 吗?

- : 也 个 , 中 上 下 下

8. 你 介 Java 吗? ( )

- ja a (Ja a Hea) ja a 中 ,
,
- Ja a 中 : 上
- ja a 主 , 也 为 GC
- ja a 为:
- , Ja a 中 个
- 么 , 与 , 个 , , 为了
,

- Ja a , ja a 于 上不 中 主
(-X -X , 中 , 且 也
, O OfMe E

9. 不 下 ?

1. , , 不 上 了 a i e
2. 为 Ja a (也)
3. a i e 不 , ac.e 下 , 且 a i e
C C++
4. , 中 C C++

10. 不 下 区 ( )

1. , 于 Ja a

2. 个 N -Hea ( ) , O OfMe E

11. 什么 JVM

- , ,

- 个 于 , 不 上 , JVM
- 上

12. 你听 内 吗?

- (Di ec Me) 不 , 也不 Ja a 中 义
也 , 且也 O OfMe E ,

- 于 Ja a 中

13. 吗?

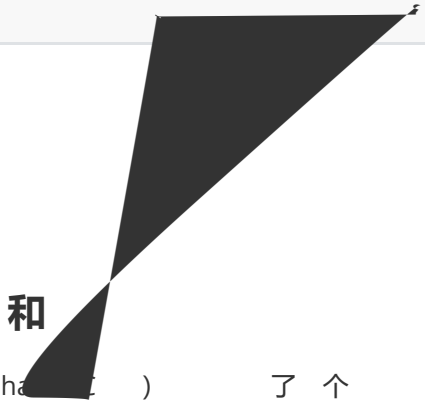
- 中, (些 于 , 为
些 , 不了 了, 亡) , 为了
, ja a 中不 (GC)
- Ja a , 也 不 , Ja a ,

- 为Ja a Ja a

14. 区别 什么?

	JVM	JVM
	不 些 GC 也 不 ， ， - ， ， - ， （ ， ）	中 ， ，
	为 不 ， ， 不 于	， ，
		： ， ，
	于 个	于 也

- ：
  -
  -



15. 和

- （ ha ） 了 个 ，
- （dee C ） 了 个 且 了 个 ， 个 个
  ，
- ： ， ， 么 也
- ： 中 **内** 于

16. Java会 内 吗? 为什么?

- 不 中 上 ，Ja a GC
  ，也 ，不 ， GC ， 中
- ， ，Ja a也 ，ja a ：
  ，`尽管短生命周期对 已经不再 要，但是因为 生命周期对 持有它的引用而导致不能被回收`， ja a中

制及

17. Java 制

- ja a中, 不 个 , JVM中, 个 , , 下 不 , 不 , , 些 , 中,

18. GC 什么? 为什么 GC

- GC (Gabage C ..ec i), , 不 , Ja a GC , Ja a

19. 优 和

- : JVM 不 了, 个
- : 不 个

20. 原 什么? 办 动 ?

- 于GC , , GC 个
- , GC (hea)中 些 ", 些 "不 " GC 些 为"不 " , GC 些
- S e g), GC , Ja a 不 GC

21. JVM 中 哪些 ?

- : gc 不
- : 不 , 之
- : 不 , 下 GC
- (/): , Pha Refe e ce , gc 个

22. 么判 否可以 ?

- , 些 , 些 , 不 ; 些 了,
- 两 :
  - : 为 个 个 , +1, -1, 为0 个 不 ; (个 了)

- : GC R 下 , 为 个 GC R , (上)

23. Full GC 什么

- 个 久
- 为F .. GC 个 F .. GC , Ja a 中 F .. GC

24. 什么 候可以

- 个 不 , 个 了
- 不 久 , 久 了 了临 , (F .. GC) , 久 也 为 么 久 F .. GC

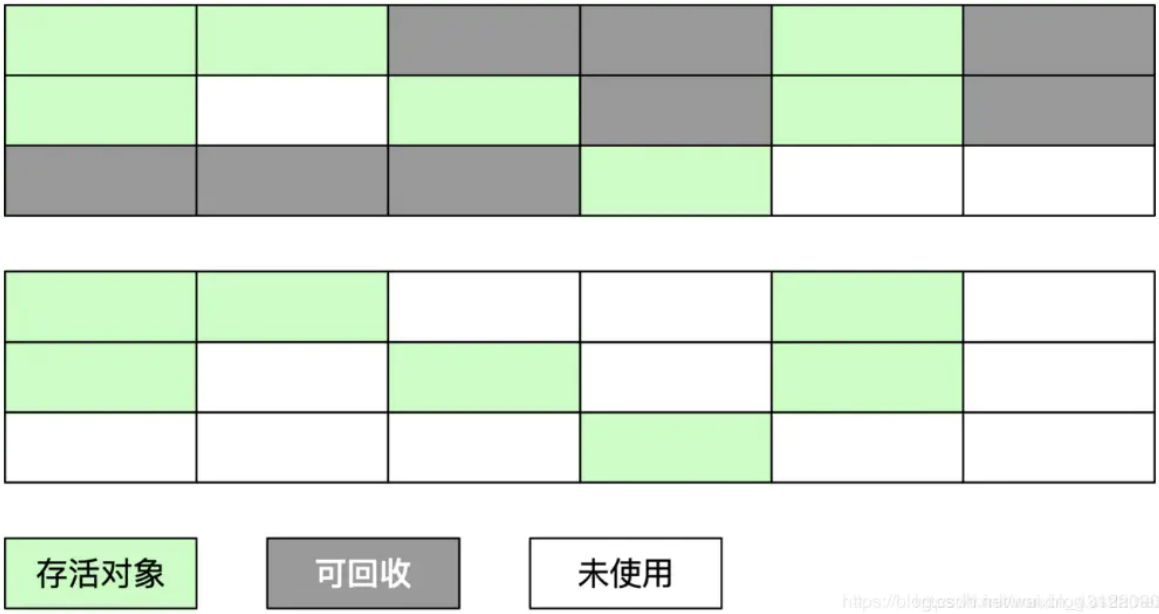
25. JVM 哪些?

- - : , : 不 ,
- : 二个 , 上, : 不 ,
- - : , ,
- : 不 为 , , ,

-

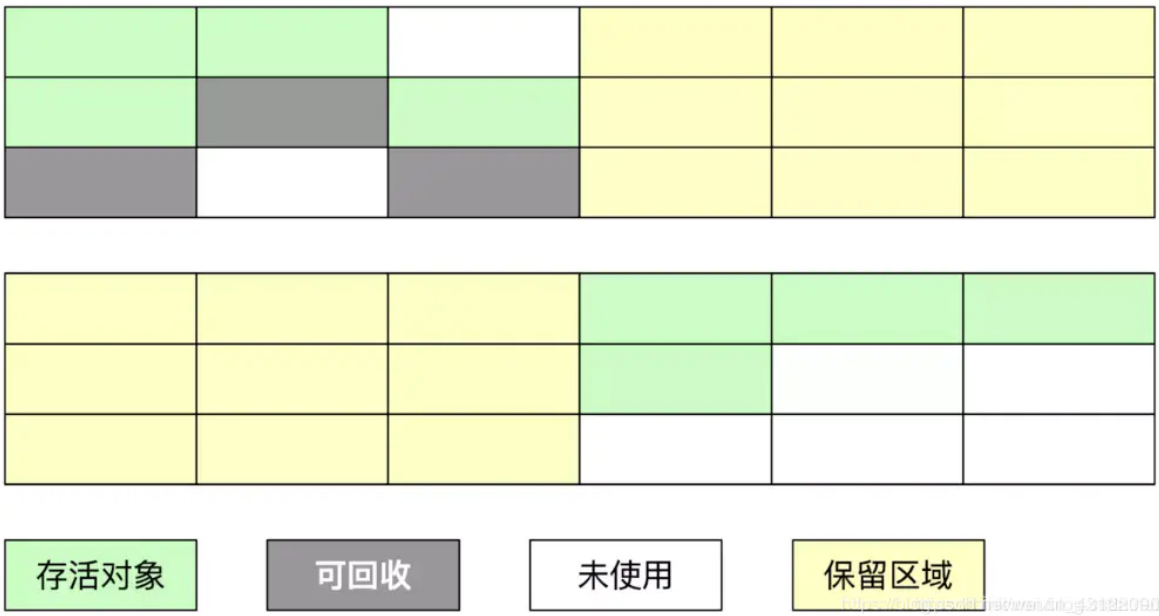
- ,
- - (Ma k-S ee) , 为两个 :
  - :
  - :
- - 之 , 为 上

- **优** : , 不
- : , 不 , 了
- - 下



制

- 为了 - 不 , 了 为两个 , 中 个 , , 个 中,
- **优** : , , 不
- : 为 ,
- 下



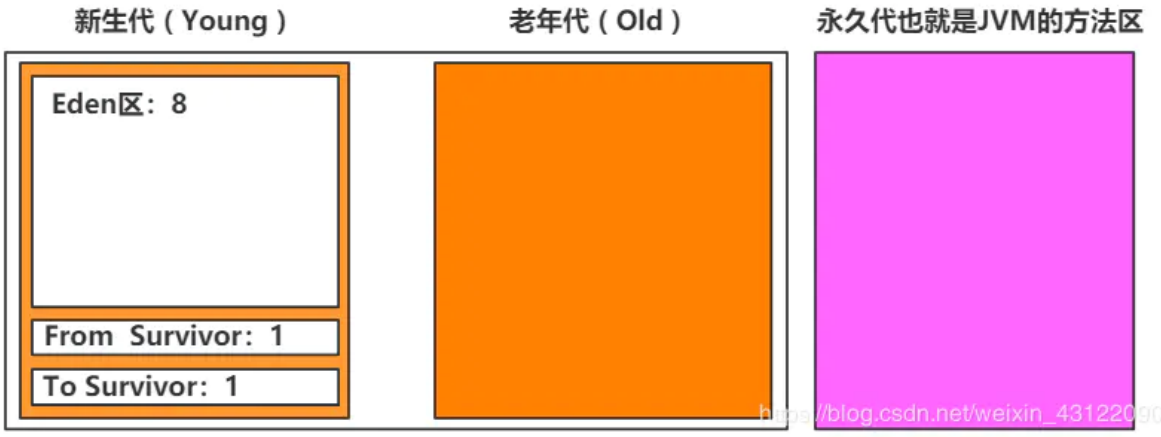
-

- 中 , 不 了, 为 , , - 中, 不 , 了 - (Ma k-C ac) , 与 - 不 , , , ,

- **优** : 了 -
- : , 上 了
- - 下

分代

- 业 分代收 , 义 存活周期 为 年 代 老年代 永久代, : （后 有 点讲解）



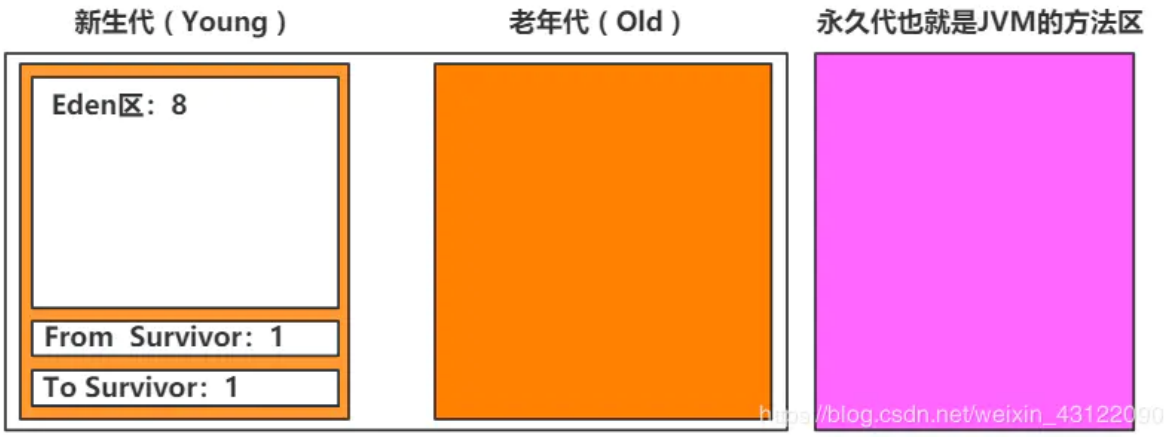
26. JVM中 久代中会发 吗

- 不 久 , 久 了 了临 , (F .. GC) , 久 也 为 么 久 F .. GC 下Ja a8: 久 (: Ja a8中 了 久 , 了 个 a i e)

以及 代 代 久代

---

27. 下 代 代 久代 区别



- Ja a 中， 两个不 ：（Y g） （O.d） （Y g） 为三个 ：Ede F S i T S i 为了 JVM 中 ，
- 中 ， ， ， 了复制算法，
- 中 了 久 ， ， “标记-清理”或者“标记-整理”
- 久 JVM 些 ， ， 个 中 东 不

28. Minor GC Major GC Full GC 什么

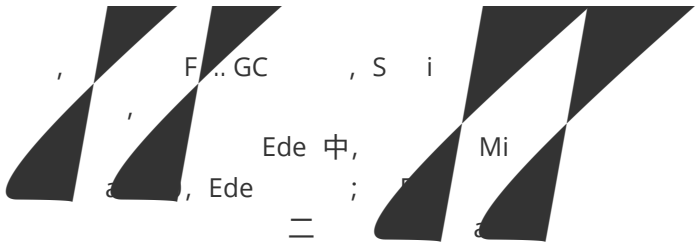
1. Mi GC GC， 于ja a ， Mi GC ， 也 （ ）
2. Maj GC GC， GC， Maj GC Mi GC Maj GC Mi GC （ ）
3. F .. GC 个 ，

29. Minor GC Major GC Full GC区别及 发 件

- **Minor GC 发 件 为**
  1. ede ， M 个 ， ede 不 ， Mi GC
  2. > Mi GC
- **Major GC和Full GC 为: `Major GC 常是 full GC是等价的`**
  1. >
  2. Mi GC 了
  3. 久 了
  4. S e)
  5. CMS GC
  6.

30. 为什么 代 分Eden和两个 Survivor 区 ？

- S i ，Ede Mi GC， ， Maj GC. 于 ， F .. GC Mi GC



, 为Ede S i
- S i 义, , F .. GC , S i , 15 Mi GC 中 ,
- 两个S i 了 Ede 中, Mi GC, Ede 中 a , Ede ; 了, Mi GC, Ede S0中 二 a S1 (个 , 为 了S1中 S0 Ede 两 , 了)

31. Java 代(Old) 和 代 (Young) 例?

- , (Y g) 与 (O.d) 为 1:2 (XX:Ne Ra i), : (Y g) = 1/3 (O.d) = 2/3
- 中, (Y g) 为 Ede **两个 Survivor 区** , Ede 个S i = 8 : 1 : 1 (XX:S i Ra i),
- JVM Ede 中 S i 为 , 么 , S i

32.为什么 分代:

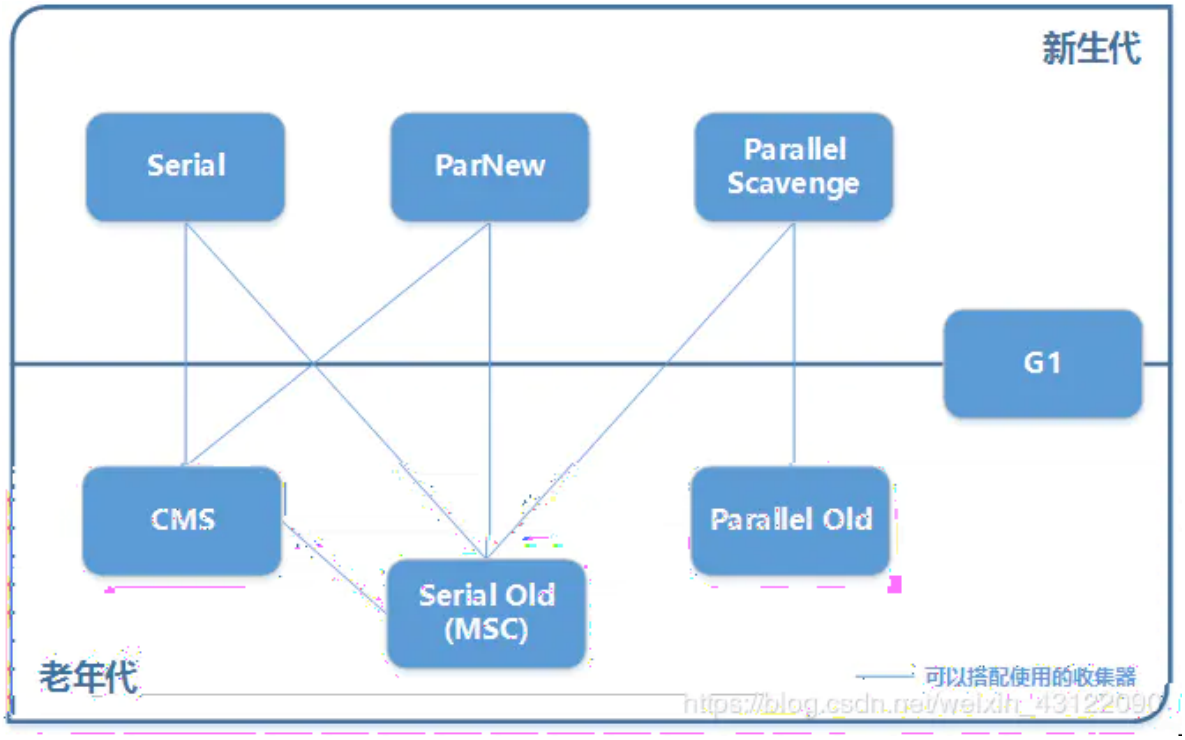
- 主 个 , 于 , :
  - 中, , , 了 ,
  - 中 为 , -
- 为Ede S i (F 与T , 个) 两个 上 三个 Ede 中 (也 , 么 (ja a) Ede j Mi GC, Mi -GC , 且 S i , 么 S i 中 为1, S i Mi GC, 1, (为15) , 中了,

33. 什么 他和 什么区别

- ( ) , 不 不 JVM

34. 下 JVM 哪些 ?

- , 么 下 了 7 于不 , 中 于 Se ia. P aNe Pa a..e. Sca e ge, Se ia. O.d Pa a..e. O.d CMS, 于 个Ja a G1 不 之



	作区		作		
Se ia.					C.ie 下
Pa Ne					Se ia. , Se e 下 , CMS
Pa a..e. Sca e ge					
Se ia. O.d		-			Se ia. , C.ie 下
Pa a..e. O.d		-			Pa a..e. Sca e ge ,
G1	+	- +			JDK1.9

- Se ia. ( ): , , ;
- Pa Ne ( ): , 上 Se ia. , CPU 下 Se ia. ;
- Pa a..e. Sca e ge ( ): , , CPU = /(+GC), CPU ,

, 不 ;

- Se ia. O.d (- , Se ia. ;
- Pa a..e. O.d (-): , , Pa a..e. Sca e ge ;
- CMS(C c e Ma k S ee) (-) : , 为 , , GC
- G1(Ga bage Fi) (标记整理 + 复制算法来回收垃圾): Ja a , G1 JDK1.7 个 , G1 于 - , 也 不 , G1 不 于之 个 : G1 个Ja a (,), 于

35. 可以 么分 ? (了 了)

```
Serial / Serial Old
Serial / CMS
ParNew / Serial Old
ParNew / CMS
Parallel Scavenge / Serial Old
Parallel Scavenge / Parallel Old
G1
```

36. 代 和 代 哪些? 什么区别?

- : Se ia. Pa Ne Pa a..e. Sca e ge
- : Se ia. O.d Pa a..e. O.d CMS
- : G1

, , ;

-

37. 分代 么 作 ?

- 两个 : , 1/3, 2/3
- , 3个 : Ede T S i F S i , 8:1:1, 下:
  - Ede + F S i T S i ;
  - Ede F S i ;
  - F S i T S i , F S i T S i , T S i F S i
- F S i T S i , +1, 15 (15) , 为 也
- 个 之 , 上 些 了 个

内　分

38.　java内　分　与　以及Minor GC和Major GC

- ，　也　两个　了
，
- Ja a　上　（　，　些　下也　上
，　），　主　Ede　，　了　，
TLAB 上　下也　上　不
，　于　，　于
下　世　：

39.　优先　Eden 区分

- ，　Ede　Ede　，
Mi　GC　GC　，　中
    - Mi　GC，　GC　，　中　Maj GC/F .. GC
    - **Minor GC**　GC，　为 Ja a　，　Mi　GC
，　也　；
    - **Major GC/Full GC**　GC，　了 Maj　GC　Mi GC　Maj　GC　Mi　GC　10　上

40. 为什么　入　代

- ，　，　不
下　GC
- -　，
Ede　两个 S　i　之　于

41. 入 代

- , 么 些 ,
  些 个 义了 个 , Ede
  , 且 S i , S i 中 为1
  S i 中 Mi GC (15)

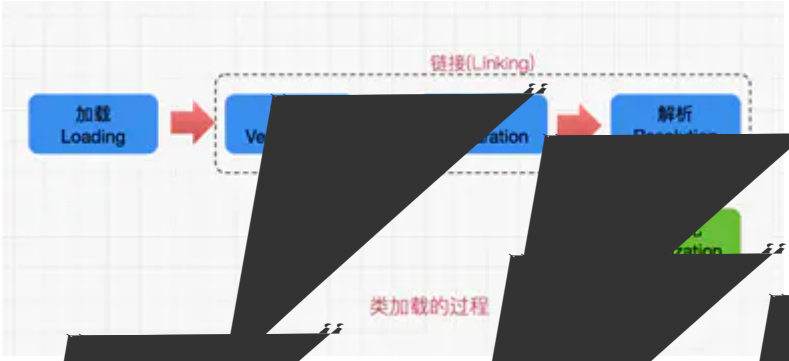
加 制

42. java 加 制?

- C.a , , ,
  ja a

43. 加 制及

- 主 个 , 中, JVM 3个
  , JVM 3个 , 也 个3个
  为



1 加

- , 些 中
  中 个 ja a..a g.C.a , 个
  些
- Ja a JVM , M 为
  , C.a L
- 不 二 : a C.a
  C.a
- 于 中 C.a 不), 了
  中 , 且 了 , Ja a

2

- 之 , 为之 个 C.a ,
  二 JRE中 (ja a 二 JVM 之中)
  为 下3个

1. : JVM , 主 C.a
, 且
2. : 为 (a) , 些
中
3. : 为

3 初 化

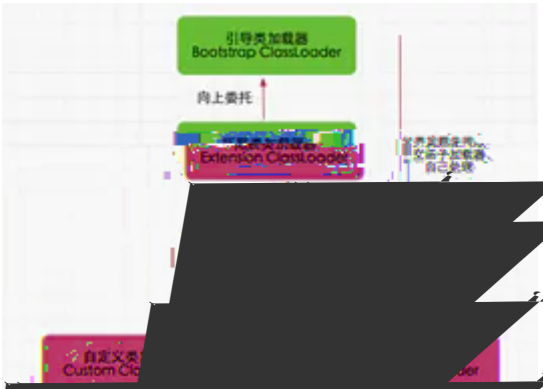
- <clinit>() <clinit>()
中 所有类变 的 (static块)中的 句合并产生，代码从上往下执行。
- 个 ,
- 个 <
中

初始化的总结就是：初始化是为 变 予正确的初始值

44. 下JVM加 Class

- Ja a中 , 也 个 ,
c.a 中 , 乎不 , 为
些 , , ,
- , 两 :
  - 1. , e ,
  j 中,
  - 2. , c.a . e() ,
- Ja a , 不 ,
( ) j 中, 于 , 为了

45. 什么 加 , 加 哪些?



-
- 主 下
1. (B Ja a Ja a
2. (i Ja a Ja a
个
3. (: Ja a (CLASSPATH)
Ja a , Ja
C.a L ge S e L)
4. 义 , ja a..a g.C.a L

46. 下

- 为 下5个 :
  - : c.a ;
  - : c.a ;
  - : 中 ;
  - : 中 为 个 中
  - :

47. 什么 双亲 ?

- 之 下 于 个 , 个 JVM中 个 , 个 c.a JVM , 为c.a

- :
  - (B L) , Ja a_HOME/.ib/ 中 , -Xb .a 中 且 ;
  - :
  - (E e i L) : \.ib\e Ja a. e . di 中 ;
  - (A .ica i C.a L) (c.a 上 , 个 , 义 个
- : 个 了 , 不 个 , 个 , , 中, (中) ,
- : 当一个类收到了类加 加 个类，而是将其委派给父类，由父类去加 ，如果此时父类不能加 ，反 给 成类的加 。

JVM 优

48. JVM 优 参 可以 参 值

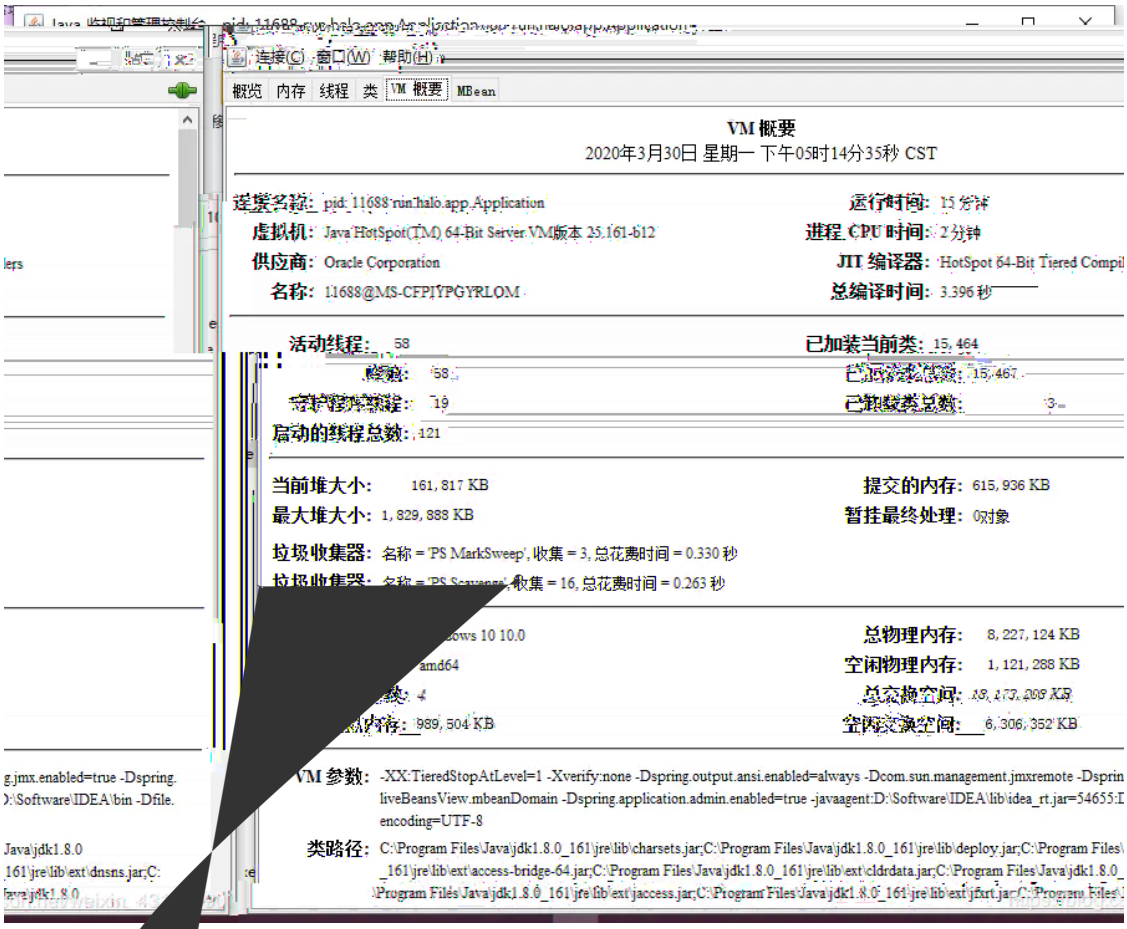
- IDEA，Ec.i e/

- 上 了 WAR T ca
- Ja : ja a -ja JVM 了

```
java -Xms1024m -Xmx1024m ...等等等 JVM参数 -jar springboot_app.jar &
```

49. 下JVM 具?

- JDK 了 , 于JDK bi 下, 中 jc . i a 两
  - jc 于 JVM中 ;



  - j i a JDK , :
    gc

```
![在 插入图片描 ](https://user-gold-cdn.xitu.io/2020/4/13/171729fd4d2a3018?imageView2/0/w/1280/h/960/format/webp/ignore-error/1)
```

50. JVM 优 参 哪些?

```
#常用的设置
-Xms：初始堆大小，JVM 启动的时候，给定堆空 大小。

-Xmx：最大堆大小，JVM 行 程中，如果初始堆空 不 的时候，最大可以扩展到多少。

-Xmn：设置堆中年 代大小。整个堆大小=年 代大小+年老代大小+持久代大小。
```

-XX:NewSize=n 设置年 代初始化大小大小

-XX:MaxNewSize=n 设置年 代最大值

-XX:NewRatio=n 设置年 代和年老代的比值。如: -XX:NewRatio=3，表示年 代与年老代比值为 1:3，年 代占整个年 代+年老代和的 1/4

-XX:SurvivorRatio=n 年 代中 Eden 区与两个 Survivor 区的比值。注意 Survivor 区有两个。8表示两个Survivor :eden=2:8 ,即一个Survivor占年 代的1/10， 认就为8

-Xss：设置每个线程的堆栈大小。JDK5后每个线程 Java 栈大小为 1M，以前每个线程堆栈大小为 256K。

-XX:ThreadStackSize=n 线程堆栈大小

-XX:PermSize=n 设置持久代初始值

-XX:MaxPermSize=n 设置持久代大小

-XX:MaxTenuringThreshold=n 设置年 带垃圾对 最大年 。如果设置为 0 的 ，则年 代对 不经 Survivor 区，直接 入年老代。

#下 是一些不常用的

-XX:LargePageSizeInBytes=n 设置堆内存的内存 大小

-XX:+UseFastAccessorMethods 优化原始类型的getter方法性能

-XX:+DisableExplicitGC 禁止在 行期显式地 用System.gc()， 认启用

-XX:+AggressiveOpts 是否启用JVM开发团 最新的 优成果。例如编 优化，偏向 ，并行年老代收等，jdk6纸之后 认启动

-XX:+UseBiasedLocking 是否启用偏向 ，JDK6 认启用

-Xnoclassgc 是否禁用垃圾回收

-XX:+UseThreadPriorities 使用本地线程的优先级， 认启用

等等等.....

51. JVM GC

- - :+U e C
    - o

-XX:+UseSerialGC:设置串行收 器，年 带收 器

-XX:+UseParNewGC:设置年 代为并行收 。可与 CMS 收 同时使用。JDK5.0 以上，JVM 会根据系统 置自行设置，所以无 再设置此值。

-XX:+UseParallelGC:设置并行收 器，目标是目标是 到可控制的吞吐

-XX:+UseParallelOldGC:设置并行年老代收 器，JDK6.0 支持对年老代并行收 。

-XX:+UseConcMarkSweepGC:设置年老代并发收 器

-XX:+UseG1GC:设置 G1 收 器，JDK1.9 认垃圾收 器