第二周作业：JVM论述题

JVM整体架构

题目 01：请你用自己的语言介绍 Java 运行时数据区（内存区域）

- （1）堆
- （2）虚拟机栈
- （3）本地方法栈
- （4）方法区
- （5）运行时常量池
- （6）直接内存
- （7）为什么堆内存要分年轻代和老年代?

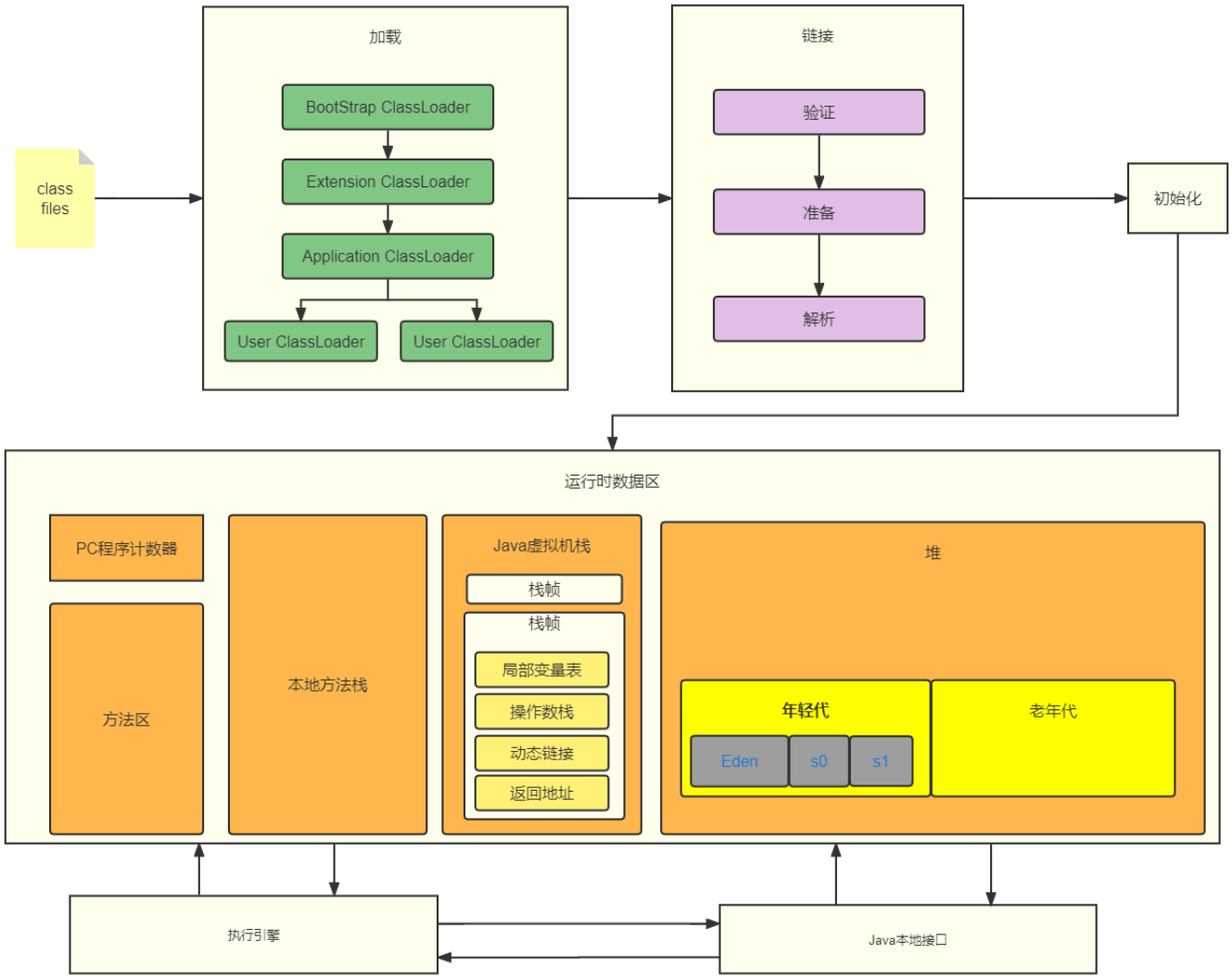
题目 02：描述一个 Java 对象的生命周期

- （1）解释一个对象的创建过程
- （2）解释一个对象的内存分配
- （3）解释一个对象的销毁过程
- （4）对象的 2 种访问方式是什么?
- （5）为什么需要内存担保?

题目 03：垃圾收集算法有哪些？垃圾收集器有哪些？他们的特点是什么?

- （1）垃圾收集算法
- （2）ParNew 收集器
- （3）Parallel Scavenge 收集器
- （4）Parallel Old 收集器
- （5）CMS 收集器
- （6）G1 收集器

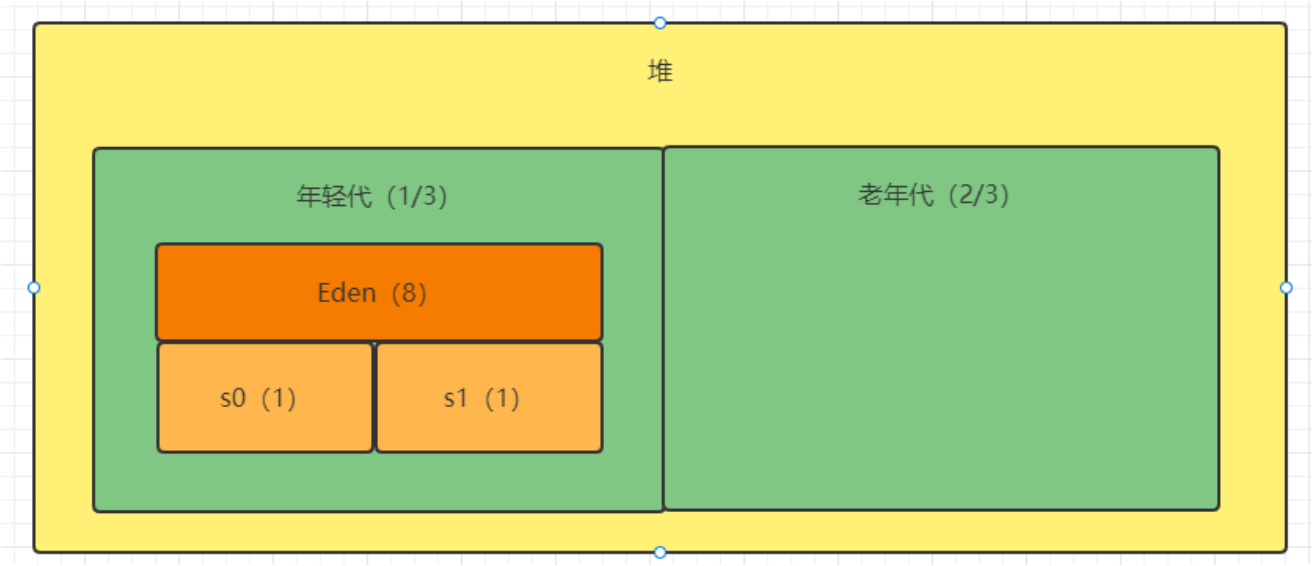
JVM整体架构



题目 01：请你用自己的语言介绍 Java 运行时数据区（内存区域）

（1）堆

堆是JVM启动时创建的一块内存区域，根据分代理论将其分为年轻代（young）和老年代（old），两者默认比例为1:2，年轻代分为Eden区、Survivor 1区和Survivor 0区，三者默认比例为8:1:1。



（2）虚拟机栈

虚拟机栈是线程私有的区域，每个Java方法都会创建一个栈帧，栈帧中保存着该方法的本地变量表、操作数栈、动态链接、返回地址。

如果线程请求分配的容量超过了Java虚拟机栈的最大容量（深度），则会抛出StackOverflowError的错误。

如果线程请求分配的内存大小总和超过了Java虚拟机栈的最大内存，则会抛出OutOfMemoryError的错误。

（3）本地方法栈

本地方法栈是线程私有的，用于管理本地方法的调用。

调用的方法是 native 修饰的，Java代码会通过 JNI 方式去调用 C++ 对应的代码，例如 Thread.start() 方法中调用的 start0() 方法

（4）方法区

方法区是线程共享的一个区域，保存着 class（类）的相关信息、运行时常量池、JIT编译之后的代码缓存。

在 jdk1.7 及以前，方法区的是实现是永久代，在 jdk 1.8 之后改为元数据区，将其放置到了直接内存中，其中得到运行时常量池放到了堆中。

（5）运行时常量池

常量池分为：class常量池、运行时常量池、字符串常量池

运行时常量池：一个class对象有一个运行时常量池

字符串常量池：保存字符串常量，底层实现数据结构是StringTable，使用哈希表+链表

字符串通过一下方式可进入字符串常量池：

- “”双引号创建的字符串，编译器直接进入字符串常量池
- 被final修饰的变量会变成常量在编译器进入字符串常量池中
- 字符串变量通过intern方法进入字符串常量池

（6）直接内存

直接内存不在运行时数据区，它的大小直接受制于本机内存大小

直接内存和堆内存的比较如下：

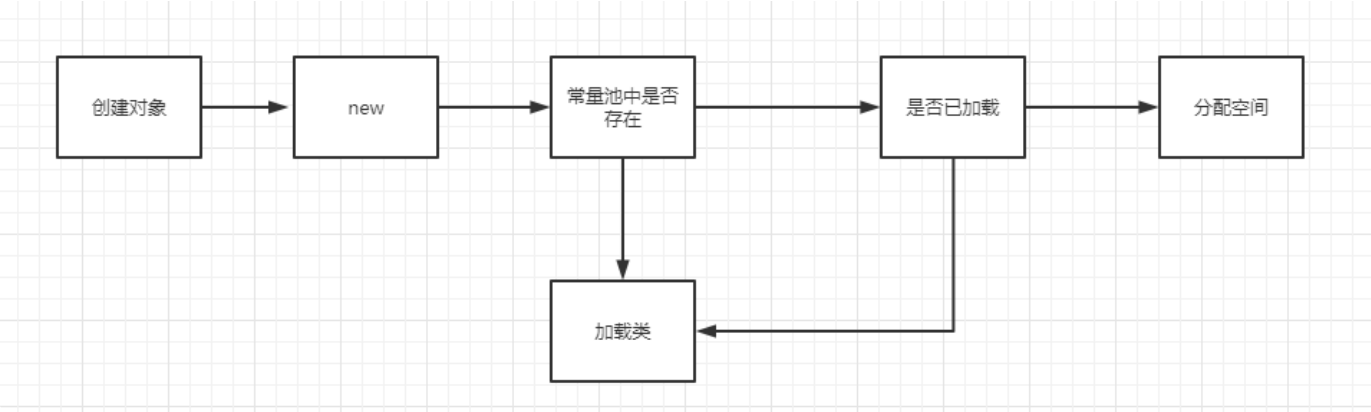
	性能	读写
堆内存	高	低
直接内存	低	高

（7）为什么堆内存要分年轻代和老年代?

堆根据分代收集理论分为年轻代和老年代，分代收集理论是根据两大假说所建立的：强分代假说和弱分代假说，大多数的对象都是朝生夕灭的，熬过越多次垃圾收集的对象越难以消亡。根据这两大假说说明收集器是需要进行分代收集的，然后根据年龄分配到不同的内存区域。

题目 02：描述一个 Java 对象的生命周期

（1）解释一个对象的创建过程



（2）解释一个对象的内存分配

对象的内存分配方式有两种：

- 指针碰撞
- 空闲列表

指针碰撞分配的空间是连续的，空闲列表分配的空间是不连续的。

（3）解释一个对象的销毁过程

判断一个对象是垃圾的方法有两种：

- 引用计数法
- 可达性分析

引用计数法：当对象被引用时，引用的值就会加一；当该对象不被引用时，就会减一，当值为0时，认为该对象是垃圾。存在循环引用问题

可达性分析：当对象被gc root引用时，该对象即为可达；当该对象不被任何gc root引用时，该对象即为不可达，也就是垃圾。

如果当JVM进行垃圾收集时，先会确认当前对象是否不可达，如果当前对象不可达，会将当前对象进行标记，之后会查看该对象是否执行了finalize方法使得当前对象可达，如果可达，该对象逃过垃圾收集，如果不可达，当前对象被回收，释放资源。

（4）对象的 2 种访问方式是什么？

- 使用句柄池中的句柄指向对象

- 使用直接指针指向对象

（5）为什么需要内存担保?

当新生代Eden区域存放不下新对象时就会进行 mintor GC ，此时会将存活的对象放入s0 / s1区域，此时尝试存放新的对象，如果还是不能存放下数据，就会将当前的存活对象放入老年代，然后将新对象存放到Eden区域。也就是说当新生代存放不下新的对象的时候，就会让老年代进行内存担保。

题目 03：垃圾收集算法有哪些？垃圾收集器有哪些？他们的特点是什么?

（1）垃圾收集算法

- 标记—整理算法
- 标记—清除算法
- 复制算法
- 分代算法

（2）ParNew 收集器

- 新生代并行，老年代串行
- Serial 收集器的多线程版本
- 因为多线程之间切换的开销问题，所以在单CPU情况下，性能较 Serial 收集器性能差

（3）Parallel Scavenge 收集器

- 吞吐量优先
- 新生代并行，老年代串行
- 新生代使用复制算法

（4）Parallel Old 收集器

- 使用标记—整理算法
- 吞吐量优先

（5）CMS 收集器

- 低延迟
- 可能产生内存碎片
- 可以同时执行用户线程

（6）G1 收集器

- 并发收集
- 多代收集
- 标记—整理算法