
三种都不对!!!

自身角度
怎样写能拿下工作就怎样写!
跟自身水平无关!

简历唯一的作用:
拿到面试机会

部署

计算机工作原理，操作系统原理，计算机网络原理

精通面向对象编程并已构建厚实知识体系并灵活运用学习新的知识

熟悉使用Linux操作系统并可根据命令逐步排错和软件性能优化

精通计算

熟悉MySQL调优和常用引擎模型和核心技术，熟练掌握编写sql语句与存储过程

精通JVM，JMM模型

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉分布式系统，熟悉中间件，熟悉数据库，熟悉网络，熟悉Linux系统

熟悉中间件

熟悉Spring Cloud技术栈

灵活运用

并发解决方案和编程思想

熟悉HTML5, CSS, Ajax, jQuery, Layui, ElementUI, Bootstrap,

熟悉分布式系统，熟悉中间件，熟悉数据库，熟悉网络，熟悉Linux系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

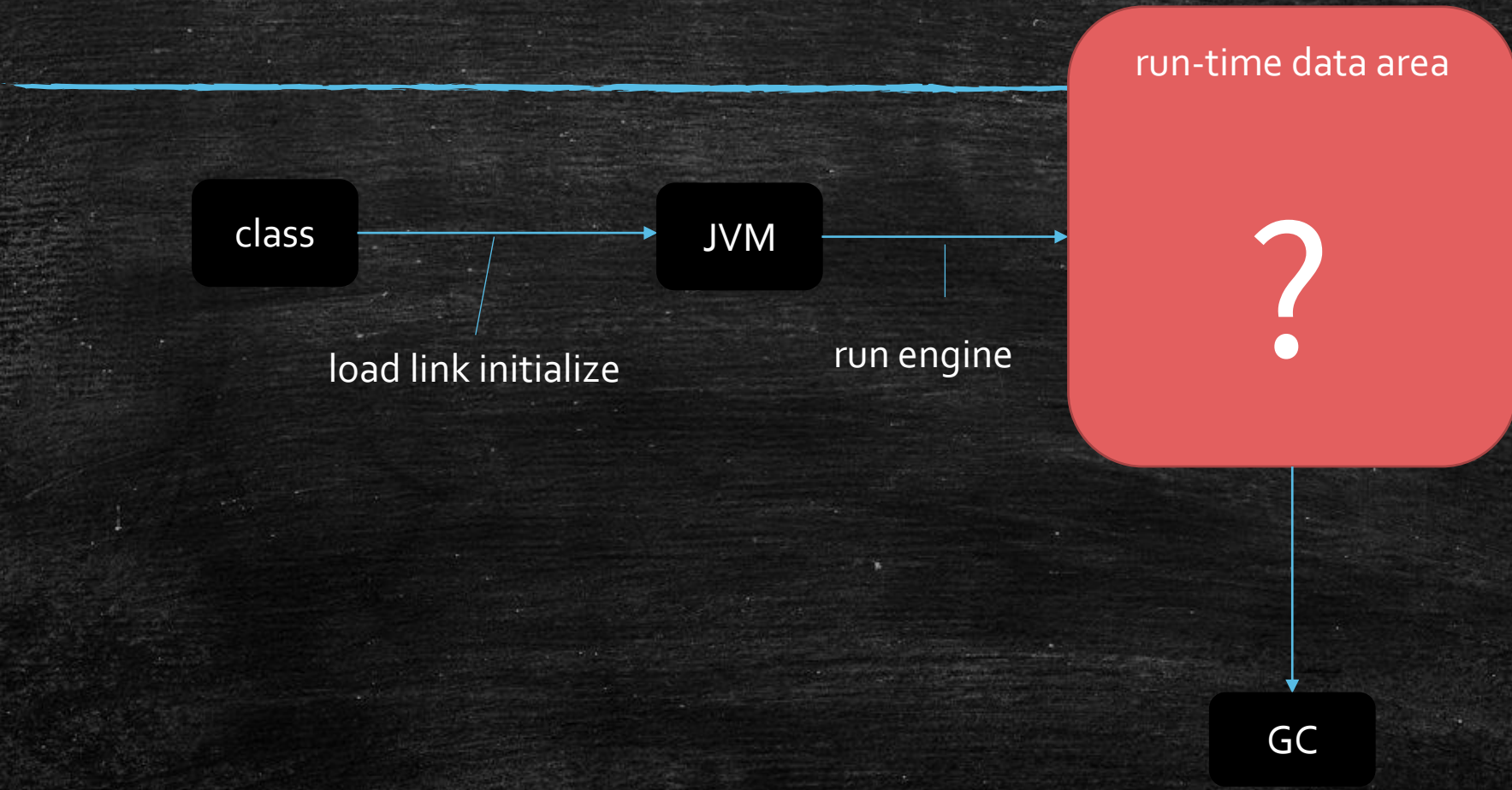
熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

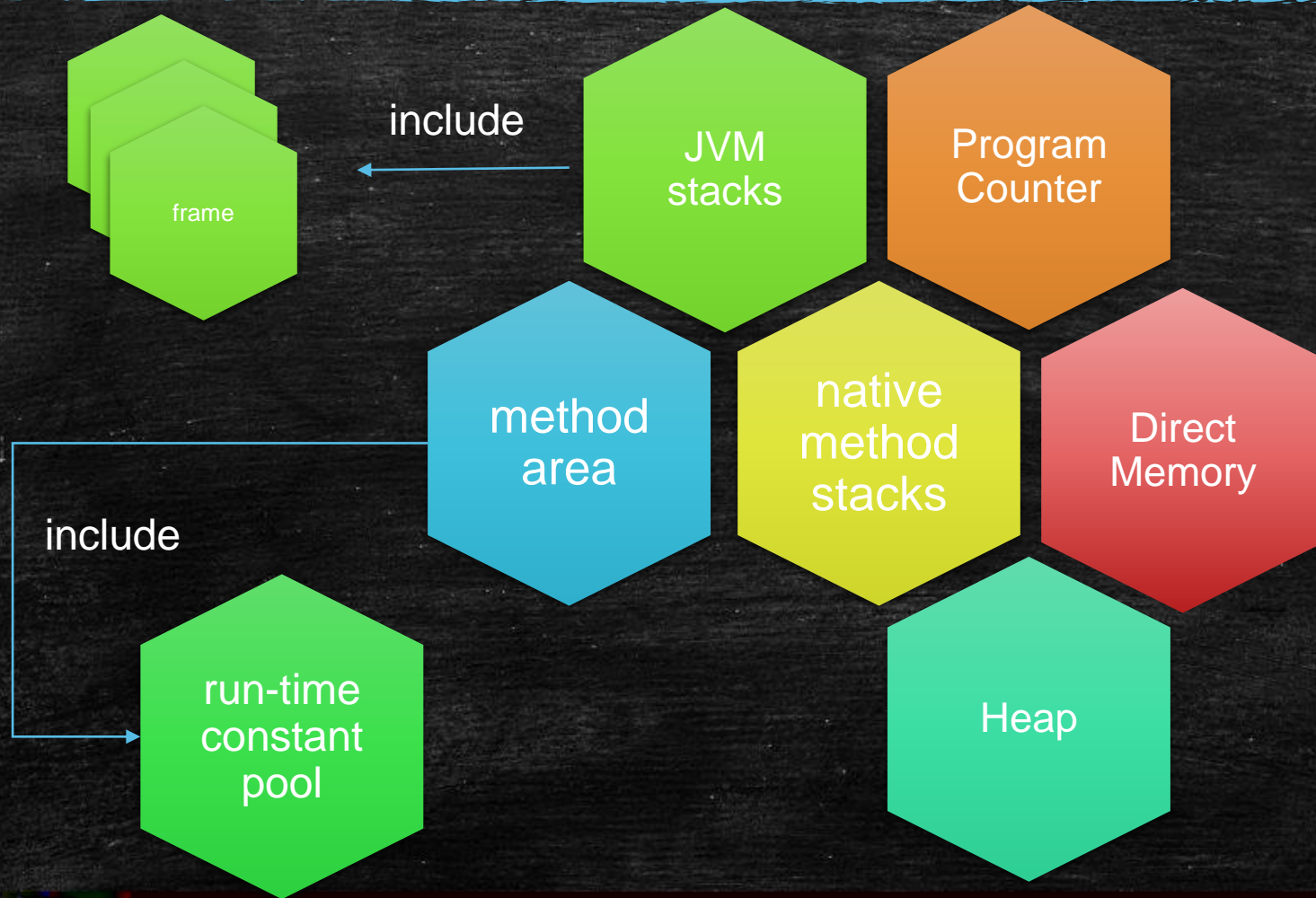
熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统

熟悉Linux系统，熟悉网络，熟悉数据库，熟悉中间件，熟悉分布式系统





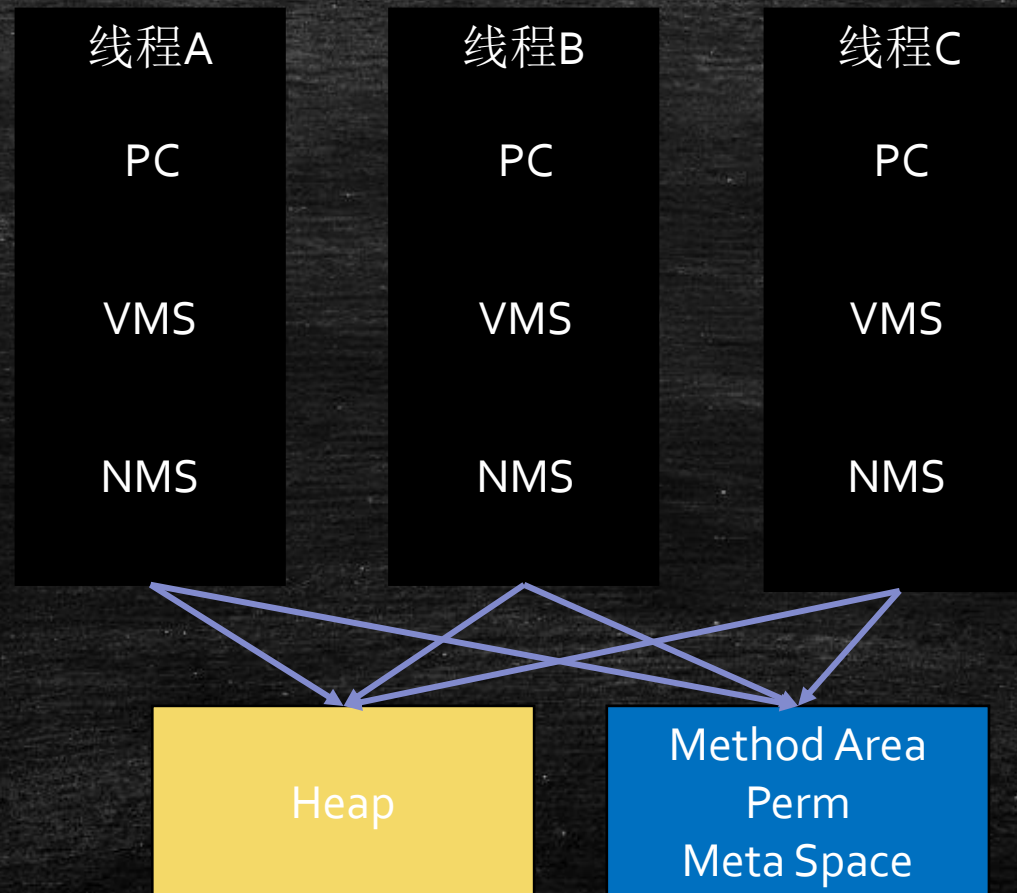
Run-time data areas

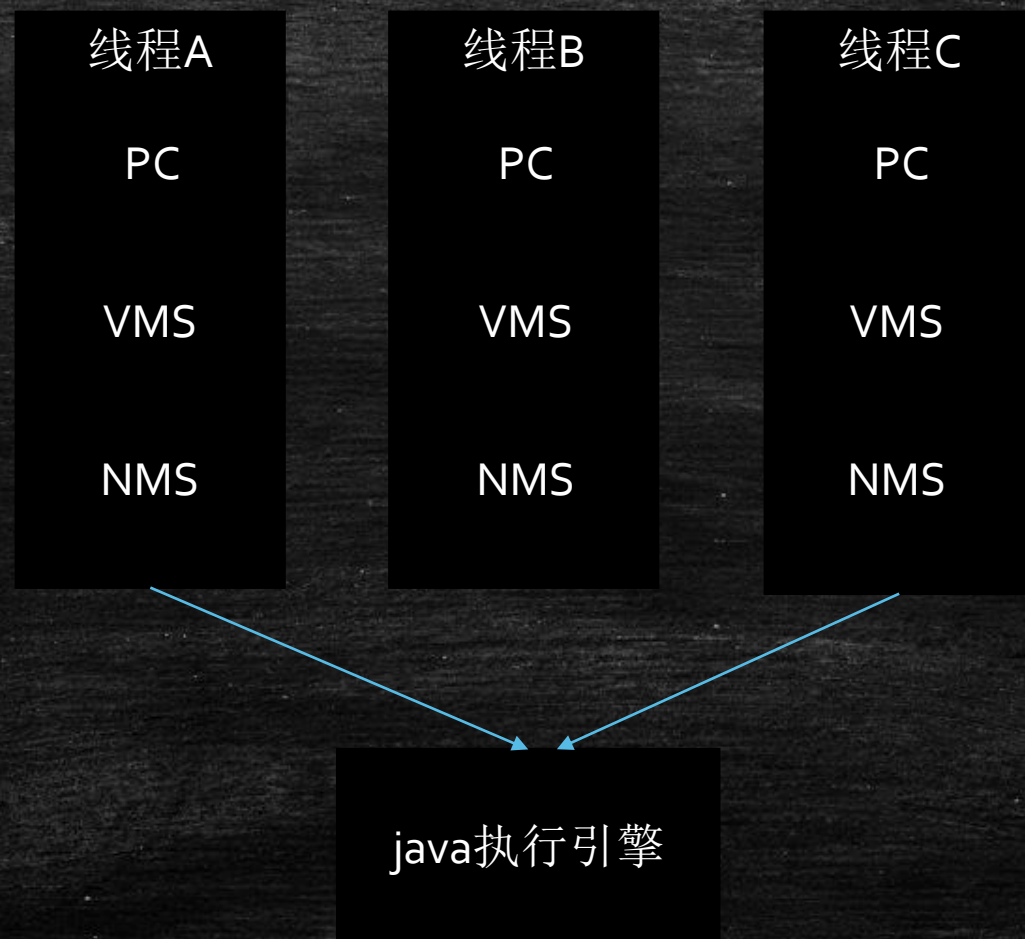


-
- Each Java Virtual Machine thread has its **own** pc (program counter) register.
 - At any point, each Java Virtual Machine thread is executing the code of a single method, namely the current method for that thread.
 - If that method is not native, the pc register contains the address of the Java Virtual Machine instruction currently being executed.

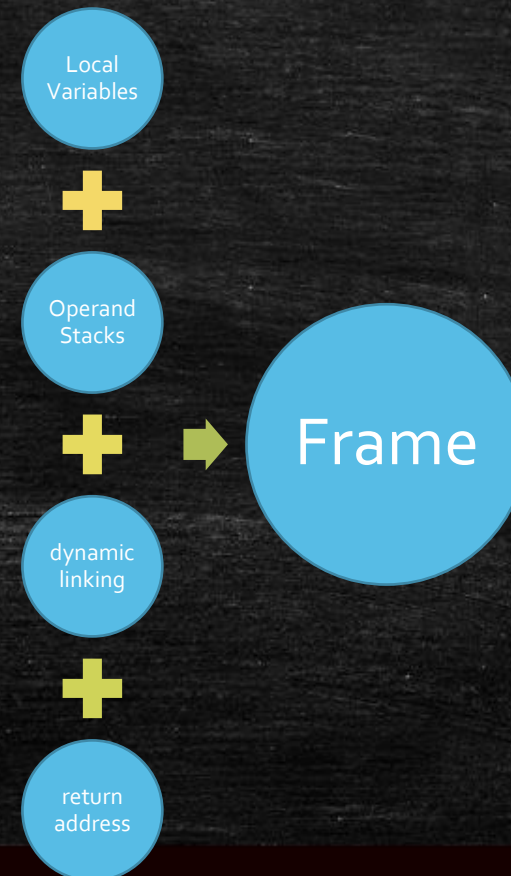


next:





- A frame is used to store data and partial results, as well as to perform dynamic linking, return values for methods, and dispatch exceptions.





JVM Stacks

Method Area

Heap

class

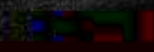
main

Local Variables

Operand Stacks

Dynamic Linking

Return Address



JVM Stacks

```
public void m1() {  
    int i = 200;  
}
```

	Bytecode	Exception
1	0 sipush 200	
2	3 istore_1	
3	4 return	

m1

Local Variables

Operand Stacks

Dynamic Linking

Return Address

this

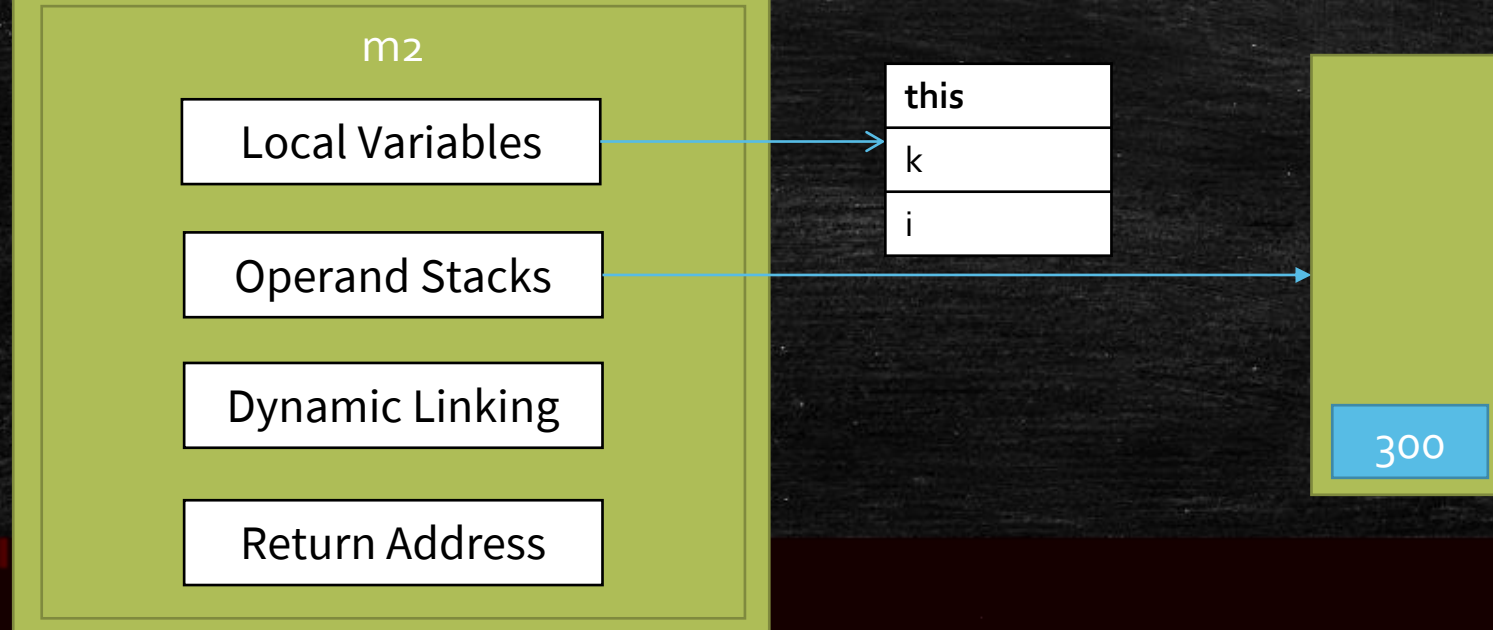
i

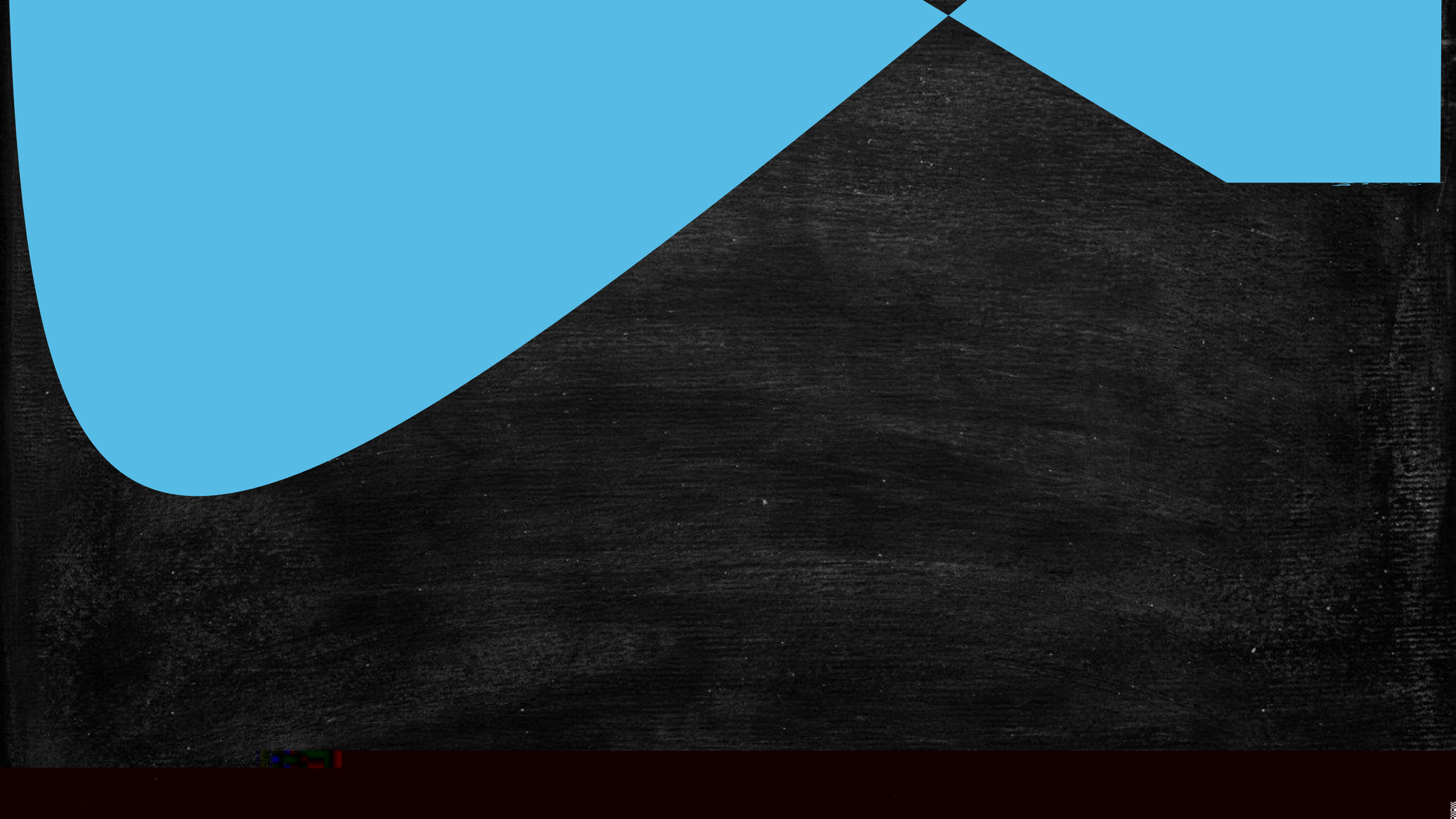
200

JVM Stacks

```
public void m2(int k) {  
    int i = 300;  
}
```

	Bytecode	Exception
1	0 sipush 300	
2	3 istore_2	
3	4 return	






```
public static void main(String[] args) {
    Hello_03 h = new Hello_03();
    h.m1();
}
```

```
public void m1() {
    //return 100;
}
```

```
public static void main(String[] args) {
    Hello_03 h = new Hello_03();
    h.m1();
}
```

```
public int m1() {
    return 100;
}
```

```
public static void main(String[] args) {
    Hello_03 h = new Hello_03();
    int i = h.m1();
}
```

```
public int m1() {
    return 100;
}
```

	Bytecode	Exception table	Misc
1	0 new #2 <com/mashibing/jvm/Hello_03>		
2	3 dup		
3	4 invokespecial #3 <com/mashibing/jvm/Hello_03.<init>>		
4	7 astore_1		
5	8 aload_1		
6	9 invokevirtual #4 <com/mashibing/jvm/Hello_03.m1>		
7	12 return		

	Bytecode	Exception table	Misc
1	0 new #2 <com/mashibing/jvm/Hello_03>		
2	3 dup		
3	4 invokespecial #3 <com/mashibing/jvm/Hello_03.<init>>		
4	7 astore_1		
5	8 aload_1		
6	9 invokevirtual #4 <com/mashibing/jvm/Hello_03.m1>		
7	12 pop		
8	13 return		

	Bytecode	Exception table	Misc
1	0 new #2 <com/mashibing/jvm/Hello_03>		
2	3 dup		
3	4 invokespecial #3 <com/mashibing/jvm/Hello_03.<init>>		
4	7 astore_1		
5	8 aload_1		
6	9 invokevirtual #4 <com/mashibing/jvm/Hello_03.m1>		
7	12 istore_2		
8	13 return		

```

public static void main(String[] args) {
    Hello_04 h = new Hello_04();
    int i = h.m(3);
}

```

```

public int m(int n) {
    if(n == 1) return 1;
    return n * m(n-1);
}

```

	Bytecode	Exception table	Misc
1	0	iload_1	
2	1	iconst_1	
3	2	if_icmpne 7 (+5)	
4	5	iconst_1	
5	6	ireturn	
6	7	iload_1	
7	8	aload_0	
8	9	iload_1	
9	10	iconst_1	
10	11	isub	
11	12	invokevirtual #4 <com/mashibing/jvm/Hello_04.m>	
12	15	imul	
13	16	ireturn	

JVM Stacks

m(1)

Local Variables

Operand Stacks

m(2)

Local Variables

Operand Stacks

m(3)

Local Variables

Operand Stacks

main

Local Variables

Operand Stacks

-
-
-
-
-

