

## 1 有返回值的加法

```
public static double add() {  
    int i = 5;  
    double res = 5.0;  
    return res + i;  
}
```

1	0	iconst_5
2	1	istore_0
3	2	ldc2_w #4 <5.0>
4	5	dstore_1
5	6	dload_1
6	7	iload_0
7	8	i2d
8	9	dadd
9	10	dreturn

步骤为:

1. 定义常数 5，加载到局部变量 i 中 对应到字节码中 L1-L2
2. 建立运行时常量 5.0，加载到局部变量 res 中，对应 L3-L4
3. 将两个变量加载到栈中，对应 L5-L6
4. 将 int 类型转换为 double 类型，对应 L7
5. 进行 double 加法，对应 L8
6. 返回最终 double 结果

## 2 没有返回值的加法

```
public static void addNoReturn() {  
    int i = 5;  
    double d = 5.0;  
    double res = i + d;  
}
```

1	0	iconst_5
2	1	istore_0
3	2	ldc2_w #4 <5.0>
4	5	dstore_1
5	6	iload_0
6	7	i2d
7	8	dload_1
8	9	dadd
9	10	dstore_3
10	11	return

步骤:

- L1-L9 与前面的有返回值的加法相同
- L10 将计算得到的结果保存回到局部变量表中
- L11 空方法返回

与 带返回值的方法的区别是：带返回的方法可以不用在局部变量表中定义新的变量，因此也不需要计算结果保存回局部变量表中。

## 3 loop

<pre> public static void loop() {     int i = 4;     if (i == 0) i = 4;     for (int i = 0; i &lt;= i; ++ i); } </pre>	<pre> 1  0  iconst_4 2  1  istore_0 3  2  iload_0 4  3  ifne 8 (+5) 5  6  iconst_4 6  7  istore_0 7  8  iconst_0 8  9  istore_1 9 10  iload_1 10 11 iload_0 11 12 if_icmpgt 21 (+9) 12 15 iinc 1 by 1 13 18 goto 10 (-8) 14 21 return </pre>
--	--

步骤:

- L1-L3 定义一个 int 变量，并将其加载到对应的 局部变量中，加载到栈中
- L4 进行 if 判断，满足条件往下执行 L5-L6，不满足时 next 执行 8(L7)
- L7-L8 定义计数变量
- L9-L10 加载阈值变量和自增变量
- L11 比较并判断是否达到了跳出循环的条件，不满足往下执行，满足跳出循环体
- L12 进行自增
- L13 循环体执行完一轮，跳回判断阶段
- L14 退出方法体

测试了其他几种四则运算，但是和加法区别不是很大，为了节约自己和助教宝贵的时间，因此没有添加到作业中。