obj.\_\_proto\_\_ === f.prototype;

**1，如果函数原型中存在原型的属性，通过原型方法操作原型属性 会对所有实例产生影响嘛?**  
如果在原型方法中使用this.a= 操作修改则只在当前实例上的a，如果使用的是f.prototype.a=修改则是修改所有实例里的a属性

new 是在堆内存的一个区域开辟一个空间，放置实例  
函数中this 指向了这个地址,

**2，Java在声明时extends 到底做了什么?**

答：加载父类静态代码块  
然后

当new son() 时  
 //加载父类非静态代码块

//父类构造函数  
 //然后才执行子类非静态代码块

//子类构造函数

**3，在new一个子类时到底有没有在内存空间里开辟空间放置一个父级实例?**  
《Java编程思想》， “

继承并不只是复制基类的接口，当创建了一个导出类(当A类继承B类时，A被称为导出类（derived class）)的对象时，该对象包含了一个基类的子对象，这个子对象与用基类直接创建对象是一样的.

二者的区别在于，用基类直接创建对象来自外部，而基类的子对象被包装在导出类对象的内部。

对于基类子对象的正确初始化也是至关重要的，而且也仅有一种方法来保证这一点：在构造器中调用基类构造器来执行初始化，而基类构造器具有执行基类初始化所需要的所有知识和能力。

第137页说“组合和继承都允许在新的类中放置子类对象，组合是显式的这样做，而继承则是隐式的做”。

**答案: 创建了一个对象但并没有新开辟一个空间，是在导出类对象中放置了一个基类子对象  
总结**: **JAVA只有通过在构造器中调用基类构造器来执行初始化才能保证基类的子对象被正确初始化包装在导出类对象的内部**

4。vue3 proxy 会不会套娃.

5,出现IllegalAccessException？  
  
要设置setAccessible(true) (非public 访问修饰符修饰的方法或构造函数)允许访问

6 Error:(6, 39) java: 未报告的异常错误java.lang.InstantiationException; 必须对其进行捕获或声明以便抛出  
要throws 抛出Exception

7 Java AutoConfigurationImportSelector怎么执行selectImports的？

8, 避免 Spring 的 AOP 的自调用问题。

(https://www.cnblogs.com/xd502djj/p/10940627.html)

在Spring的AOP代理下，只有目标方法由外部调用，目标方法才由（Spring 生成的代理对象）来管理，这会造成自调用问题。

在Spring的AOP代理下spring的代理对象才能发生回滚

若同一类中的其他没有@Transactional 注解的方法内部调用有@Transactional 注解的方法，有@Transactional 注解的方法的事务被忽略，不会发生回滚。见清单 5 举例代码展示。

清单 5.自调用问题举例

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | @Service  public class OrderService {      private void insert() {  insertOrder();  }  @Transactional      public void insertOrder() {          //insert log info          //insertOrder          //updateAccount         }  } |

insertOrder 尽管有@Transactional 注解，但它被内部方法 insert 调用，事务被忽略，出现异常事务不会发生回滚。

上面的两个问题@Transactional 注解只应用到 public 方法和自调用问题，是由于使用 Spring AOP 代理造成的。为解决这两个问题，使用 AspectJ 取代 Spring AOP 代理。