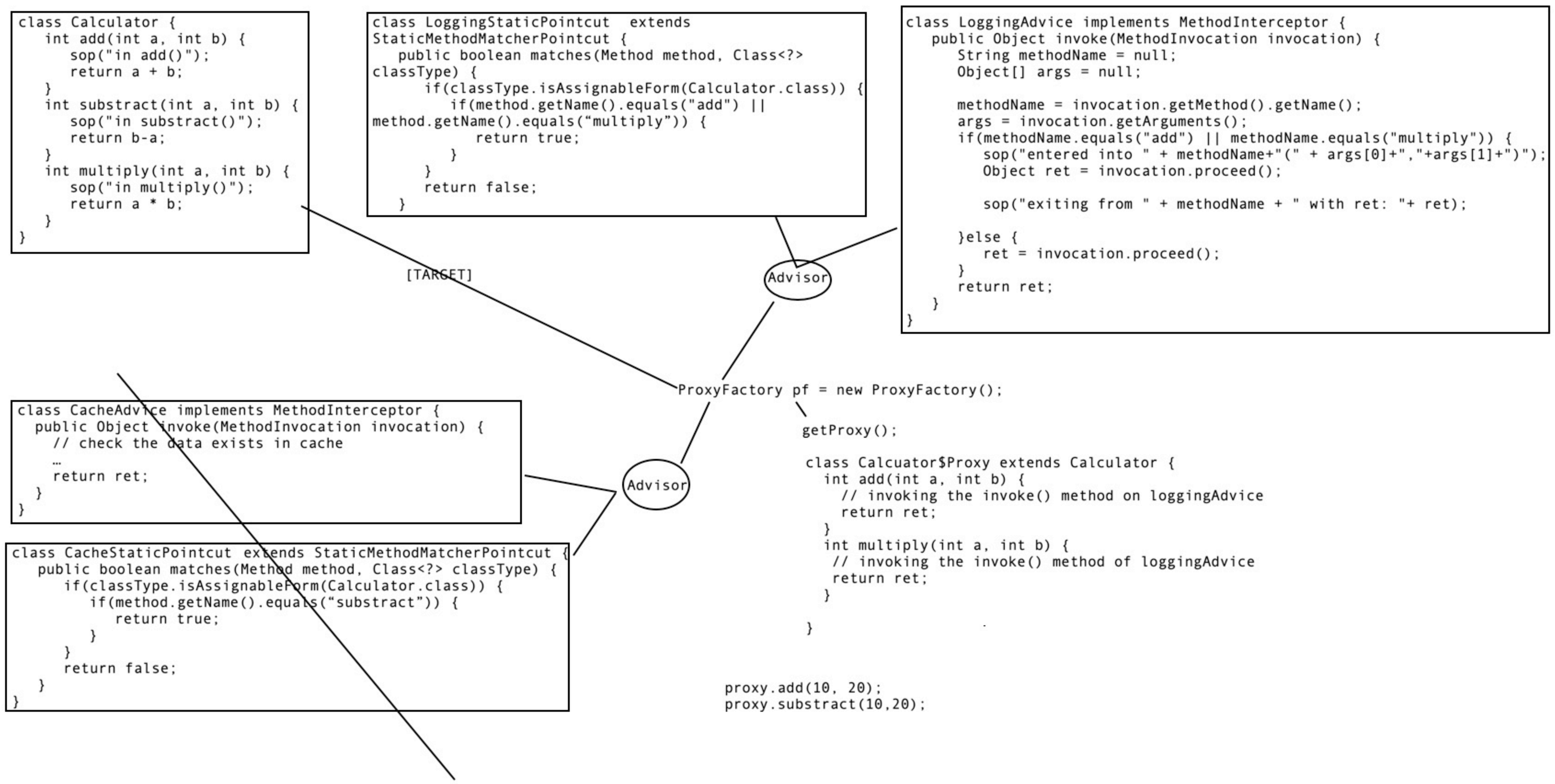

```
class Calculator {
   int add(int a, int b) {
      sop("in add()");
      return a + b;
   }
   int substract(int a, int b) {
      sop("in substract()");
      return b-a;
   }
   int multiply(int a, int b) {
      sop("in multiply()");
      return a * b;
   }
}
```

```
class LoggingStaticPointcut  extends
StaticMethodMatcherPointcut {
   public boolean matches(Method method, Class<?>
classType) {
      if(classType.isAssignableForm(Calculator.class)) {
         if(method.getName().equals("add") ||
method.getName().equals("multiply")) {
            return true;
         }
      }
      return false;
   }
```

```
class LoggingAdvice implements MethodInterceptor {
   public Object invoke(MethodInvocation invocation) {
      String methodName = null;
      Object[] args = null;

      methodName = invocation.getMethod().getName();
      args = invocation.getArguments();
      if(methodName.equals("add") || methodName.equals("multiply")) {
         sop("entered into " + methodName+"(" + args[0]+","+args[1]+")");
         Object ret = invocation.proceed();

         sop("exiting from " + methodName + " with ret: "+ ret);

      }else {
         ret = invocation.proceed();
      }
      return ret;
   }
}
```

```
ProxyFactory pf = new ProxyFactory();
```

```
class CacheAdvice implements MethodInterceptor {
  public Object invoke(MethodInvocation invocation) {
    // check the data exists in cache
    …
    return ret;
  }
}
```

```
class CacheStaticPointcut extends StaticMethodMatcherPointcut {
   public boolean matches(Method method, Class<?> classType) {
      if(classType.isAssignableForm(Calculator.class)) {
         if(method.getName().equals("substract")) {
            return true;
         }
      }
      return false;
   }
}
```

```
class Calcuator$Proxy extends Calculator {
  int add(int a, int b) {
    // invoking the invoke() method on loggingAdvice
    return ret;
  }
  int multiply(int a, int b) {
   // invoking the invoke() method of loggingAdvice
   return ret;
  }

}
```

```
proxy.add(10, 20);
proxy.substract(10,20);
```