X = 1	Y = 2	Z = 6

```
class Abra {
   int a = X, b = Y

   fun cus(int x): int {
      a = b + x
      return pide(a)
   }

   fun pide(int y): int {
      return a - y * b
   }
}


class Cadabra extends Abra {
   Abra zo = new PataDeCabra()

   fun pide(int y): int {
      return zo.cus(a + b) - y
   }
}


class PataDeCabra extends Cadabra {
   int b = Y + Z, c = Z

   fun cus(int x): int {
      a = x - 3
      c = a + b * c
      return pide(a * b + x)
   }

   fun pide(int y): int {
      return c - y * a
}
```

Asociacion Estática de Métodos

```
Abra ho = new Cadabra()
Abra po = new PataDeCabra()
Cadabra cir = new PataDeCabra()

print(ho.cus(X + 1) + po.cus(Y + 1) + cir.cus(Z + 1))
```

Objeto	Metodos	Variables
ho (Cadabra)	pide = Abra::pide()	zo (PataDeCabra())
	cus = Abra::cus()	a = 1
		b = 2
ho.zo (PataDeCabra)	cus = Abra::cus()	a = 1
	pide = Abra::pide()	b = 2
		c = 6
po (PataDeCabra)	cus = Abra::cus()	a = 1
	pide = Abra::pide()	b = 2
		c = 6
cir (PataDeCabra)	cus = Abra::cus()	a = 1
	pide = Cadabra::pide()	b = 2
		c = 6

Output = -18

19			
18			
17			
16			
15			
14			
13			
12			
11			
10			
9			
8			
7			
6			
5			
4	out = 4 - 4 * 2	out = 5 - 5 * 2	out = 9 - 9 * 2
3	y = a	y = 5	y = 9
2	pide(a)	pide(a)	pide(a)
1	a = 4	a = 5	a = 9
0	x = 2	x = 3	x = 7
	ho.cus(X + 1) = -4	po.cus(Y + 1) = -5	cir.cus(Z + 1) = -9

X = 1	Y = 2	Z = 6

```
class Abra {
   int a = X, b = Y

   fun cus(int x): int {
      a = b + x
      return pide(a)
   }

   fun pide(int y): int {
      return a - y * b
   }
}


class Cadabra extends Abra {
   Abra zo = new PataDeCabra()

   fun pide(int y): int {
      return zo.cus(a + b) - y
   }
}


class PataDeCabra extends Cadabra {
   int b = Y + Z, c = Z

   fun cus(int x): int {
      a = x - 3
      c = a + b * c
      return pide(a * b + x)
   }

   fun pide(int y): int {
      return c - y * a
}
```

Asociacion Dinámica de Métodos

```
Abra ho = new Cadabra()
Abra po = new PataDeCabra()
Cadabra cir = new PataDeCabra()

print(ho.cus(X + 1) + po.cus(Y + 1) + cir.cus(Z + 1))
```

Objeto	Metodos	Variables
ho (Cadabra)	pide = Cadabra::pide()	zo (Abra())
	cus = Abra::cus()	a = 1
		b = 2
ho.zo (PataDeCabra)	cus = Abra::cus()	a = 1
	pide = Abra::pide()	b = 2
po (PataDeCabra)	cus = PataDeCabra::cus()	a = 1
	pide = PataDeCabra::pide()	b = 8
		c = 6
cir (PataDeCabra)	cus = PataDeCabra::cus()	a = 1
	pide = PataDeCabra::pide()	b = 8
		c = 6

Output = -68

19			
18			
17			
16			
15			
14			
13			
12			
11			
10			
9	out = -8 -4		
8	y = 8		
7	pide(a)		
6	a = 8		
5	x = 6	out = 48	out = -104
4	ho.zo.cus(a + b)	y = 3	y = 39
3	y = 4	pide(a * b + x)	pide(a * b + x)
2	pide(a)	c = 48	c = 52
1	a = 4	a = 0	a = 4
0	x = 2	x = 3	x = 7
	ho.cus(X + 1) = -12	po.cus(Y + 1) = 48	cir.cus(Z + 1) = -104