

Ejercicio 3

Luis Para

Tenemos las Siguientes Clases:

```
1 class Bebida{  
2     int a = 0, b =8  
3  
4     fun s(int x): int{  
5         a = b - x  
6         return t(a)  
7     }  
8  
9     fun t(int y): int {  
10        return a * y + b  
11    }  
12 }
```

```
1 class Cafe extends Bebida{  
2     Bebida caliente = new Marron()  
3  
4     fun q(int y): int{  
5         return caliente.s(a + b) - y  
6     }  
7 }
```

```
1 class Marron extends Cafe {  
2     int c = 3  
3  
4     fun s(int x): int {  
5         a = x -4  
6         c = a + b * x  
7         return t(a * b + c)  
8     }  
9  
10    fun t(int y): int {  
11        return c - y  
12    }  
13 }
```

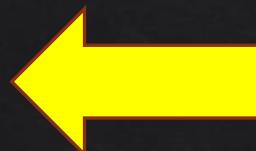
Y queremos ejecutar el siguiente código con Asociación estatica de métodos:

```
1 class Bebida{  
2     int a = 0, b = 8  
3  
4     fun s(int x): int{  
5         a = b - x  
6         return t(a)  
7     }  
8  
9     fun t(int y): int {  
10        return a * y + b  
11    }  
12 }
```

```
1 class Cafe extends Bebida{  
2     Bebida caliente = new Marron()  
3  
4     fun q(int y): int{  
5         return caliente.s(a + b) - y  
6     }  
7 }
```

```
1 class Marron extends Cafe {  
2     int c = 3  
3  
4     fun s(int x): int {  
5         a = x - 4  
6         c = a + b * x  
7         return t(a * b + c)  
8     }  
9  
10    fun t(int y): int {  
11        return c - y  
12    }  
13 }
```

```
1 Bebida e = new Cafe()  
2 Bebida pres = new Marron()  
3 Cafe o = new Marron()  
4  
5 print (e.s(1) + pres.s(1) + o.s(1))
```



Comenzemos creando la Instancia e (lineal):

```
1 class Bebida{  
2     int a = 0, b = 8  
3  
4     fun s(int x): int{  
5         a = b - x  
6         return t(a)  
7     }  
8  
9     fun t(int y): int {  
10        return a * y + b  
11    }  
12 }
```

```
1 class Cafe extends Bebida{  
2     Bebida caliente = new Marron()  
3  
4     fun q(int y): int{  
5         return caliente.s(a + b) - y  
6     }  
7 }
```

```
1 class Marron extends Cafe {  
2     int c = 3  
3  
4     fun s(int x): int {  
5         a = x - 4  
6         c = a + b * x  
7         return t(a * b + c)  
8     }  
9  
10    fun t(int y): int {  
11        return c - y  
12    }  
13 }
```

e			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```
1 Bebida e = new Cafe()  
2 Bebida pres = new Marron()  
3 Cafe o = new Marron()  
4  
5 print (e.s(1) + pres.s(1) + o.s(1))
```

Ahora creamos la instancia pres (linea 2):

```
1 class Bebida{  
2     int a = 0, b = 8  
3  
4     fun s(int x): int{  
5         a = b - x  
6         return t(a)  
7     }  
8  
9     fun t(int y): int {  
10        return a * y + b  
11    }  
12 }
```

```
1 class Cafe extends Bebida{  
2     Bebida caliente = new Marron()  
3  
4     fun q(int y): int{  
5         return caliente.s(a + b) - y  
6     }  
7 }
```

```
1 class Marron extends Cafe {  
2     int c = 3  
3  
4     fun s(int x): int {  
5         a = x - 4  
6         c = a + b * x  
7         return t(a * b + c)  
8     }  
9  
10    fun t(int y): int {  
11        return c - y  
12    }  
13 }
```

e			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

pres			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

```
1 Bebida e = new Cafe()  
2 Bebida pres = new Marron()  
3 Cafe o = new Marron()  
4  
5 print (e.s(1) + pres.s(1) + o.s(1))
```

Ahora creamos la instancia o (linea 3):

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }
```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }
```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }
```

e			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

pres			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))
```

Ahora comenzamos la linea 5 con e.s(1):

pres			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Buscamos en la table de E de donde es el Metodo:

pres			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Ejecutamos el metodo s de Bebida en e:

pres			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

ins		
2		
1		
0	x	1

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Modificamos a en la tabla de e:

pres			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

ins		
2		
1		
0	x	1

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7

```

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13

```

Entramos en la t de bebida con y de valor a de e:

pres			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

Bebida.t		
1	y	7

Bebida.s		
0	x	1

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Retornamos a de e multiplicado por y + b de e total(49 + 8):

pres			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

Bebida.t		
1	y	7

Bebida.s		
0	x	1

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Bebida.t retorna 57:

pres			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```



```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Bebida.s retorna 57:

pres			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```



Print: 57 + ? + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Ahora vamos con pres.s:

pres			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

Print: 57 + ? + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

En la tabla de pres, Podemos ver de donde es la funcion s

pres			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

Print: 57 + ? + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

En la tabla de pres, Podemos ver de donde es la funcion s

pres			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

Print: 57 + ? + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Entramos a las de Bebida con $x = 1$

pres			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

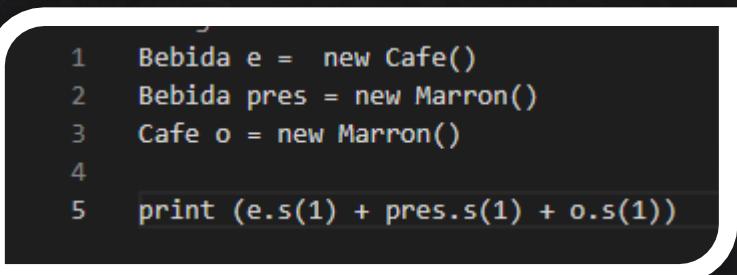
o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```



Print: 57 + ? + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Modificamos la a de pres (8 -1)

pres			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```



Print: 57 + ? + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Entramos a t. Por como vemos en pres, tambien en Bebida

pres			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

Bebida.s		
1	y	7
Bebida.s		
0	x	1

Print: 57 + ? + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Ejecutamos linea 10 de Bebida. Con a y b de pres.

pres			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

Print: 57 + ? + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }
```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }
```

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))
```

Bebida.t		
1	y	7

Bebida.s		
0	x	1

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }
```

Retornamos el resultado 57

pres			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```



Print: 57 + ? + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Nuevamente retornamos el resultado 57

pres			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```



Print: 57 + ? + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Ahora procedemos con o.s

pres			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

Print: 57 + 57 + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Buscamos en o de donde es el metodo s. (Bebida)

pres			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

Print: 57 + 57 + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Entramos al s de Bebida con x = 1

pres			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```



Print: 57 + 57 + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Modificamos la clase o (8-1)

pres			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```



Print: 57 + 57 + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Entramos a t, tambien de bebida con y = 7(valor de a de o)

pres			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

Bebida.t		
1	y	7

Bebida.s		
0	x	1

Print: 57 + 57 + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Retornamos $a^*y + b$ con a y b de o :

pres			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

Bebida.t		
1	y	7

Bebida.s		
0	x	1

Print: 57 + 57 + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Retornamos 57:

pres			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```



Print: 57 + 57 + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Nuevamente retornamos 57:

pres			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```



Print: 57 + 57 + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Y con eso temenos que print imprime

pres			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

Print: $57 + 57 + 57$
= Print 171

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

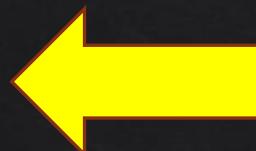
Ahora queremos aplicar Asociación dinamica de metodos:

```
1 class Bebida{  
2     int a = 0, b = 8  
3  
4     fun s(int x): int{  
5         a = b - x  
6         return t(a)  
7     }  
8  
9     fun t(int y): int {  
10        return a * y + b  
11    }  
12 }
```

```
1 class Cafe extends Bebida{  
2     Bebida caliente = new Marron()  
3  
4     fun q(int y): int{  
5         return caliente.s(a + b) - y  
6     }  
7 }
```

```
1 class Marron extends Cafe {  
2     int c = 3  
3  
4     fun s(int x): int {  
5         a = x - 4  
6         c = a + b * x  
7         return t(a * b + c)  
8     }  
9  
10    fun t(int y): int {  
11        return c - y  
12    }  
13 }
```

```
1 Bebida e = new Cafe()  
2 Bebida pres = new Marron()  
3 Cafe o = new Marron()  
4  
5 print (e.s(1) + pres.s(1) + o.s(1))
```



Comenzemos creando la Instancia e (lineal):

```
1 class Bebida{  
2     int a = 0, b = 8  
3  
4     fun s(int x): int{  
5         a = b - x  
6         return t(a)  
7     }  
8  
9     fun t(int y): int {  
10        return a * y + b  
11    }  
12 }
```

```
1 class Cafe extends Bebida{  
2     Bebida caliente = new Marron()  
3  
4     fun q(int y): int{  
5         return caliente.s(a + b) - y  
6     }  
7 }
```

```
1 class Marron extends Cafe {  
2     int c = 3  
3  
4     fun s(int x): int {  
5         a = x - 4  
6         c = a + b * x  
7         return t(a * b + c)  
8     }  
9  
10    fun t(int y): int {  
11        return c - y  
12    }  
13 }
```

e			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```
1 Bebida e = new Cafe()  
2 Bebida pres = new Marron()  
3 Cafe o = new Marron()  
4  
5 print (e.s(1) + pres.s(1) + o.s(1))
```

Ahora creamos la instancia pres (linea 2):

```
1 class Bebida{  
2     int a = 0, b = 8  
3  
4     fun s(int x): int{  
5         a = b - x  
6         return t(a)  
7     }  
8  
9     fun t(int y): int {  
10        return a * y + b  
11    }  
12 }
```

```
1 class Cafe extends Bebida{  
2     Bebida caliente = new Marron()  
3  
4     fun q(int y): int{  
5         return caliente.s(a + b) - y  
6     }  
7 }
```

```
1 class Marron extends Cafe {  
2     int c = 3  
3  
4     fun s(int x): int {  
5         a = x - 4  
6         c = a + b * x  
7         return t(a * b + c)  
8     }  
9  
10    fun t(int y): int {  
11        return c - y  
12    }  
13 }
```

e			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

pres			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

```
1 Bebida e = new Cafe()  
2 Bebida pres = new Marron()  
3 Cafe o = new Marron()  
4  
5 print (e.s(1) + pres.s(1) + o.s(1))
```

Ahora creamos la instancia o (linea 3):

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }
```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }
```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }
```

e			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

pres			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))
```

Comenzamos con a

pres			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7

```

Ahora comenzamos la linea 5 con e.s(1):

pres			
Variables	Funciones		
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables	Funciones		
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables	Funciones		
a	0	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Buscamos en la table de e de donde es el Metodo:

pres			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Ejecutamos el metodo s de Bebida en e:

pres			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	0	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

ins		
2		
1		
0	x	1

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Modificamos a en la tabla de e:

pres			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

ins		
2		
1		
0	x	1

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7

```

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13

```

Entramos en la t de bebida con y de valor a de e:

pres			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

Bebida.t

1	y	7
---	---	---

Bebida.s

0	x	1
---	---	---

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Retornamos a de e multiplicado por y + b de e total(49 + 8):

pres			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

Bebida.t		
1	y	7

Bebida.s		
0	x	1

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Bebida.t retorna 57:

pres			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```



```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Bebida.s retorna 57:

pres			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```



Print: 57 + ? + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Ahora vamos con pres.s:

pres			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

Print: 57 + ? + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Buscamos de donde es la funcion s en pres

pres			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

Print: 57 + ? + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Es en Marron. Asi que entramos en s de marron.

pres			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```



Print: 57 + ? + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Cambiamos la a de pres (Linea 5 Marron).

pres			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```



Print: 57 + ? + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Cambiamos la c de pres (Linea 6 Marron).

pres			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

Print: 57 + ? + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```



```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Entramos a t de Marron(según pres) con y = -19:

pres			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

Marron.s		
1	y	-19
Marron.s		
0	x	1

Print: 57 + ? + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Retornamos $c - y = 5 - (-19) = 24$

pres			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

Print: 57 + ? + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

Marron.s		
1	y	-19
Marron.s		
0	x	1

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Retornamos 24

pres			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

X

Marron.s		
1	X	-19
0	X	1

Print: 57 + ? + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Nuevamente Retornamos 24

pres			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

Print: 57 + ? + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```



Ahora procedemos con o.s

pres			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

Print: 57 + 24 + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Buscamos de donde es la s en o (Marron)

pres			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

Print: 57 + 24 + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Entramos a s de Marron con x = 1

pres			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	0	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```



Print: 57 + 24 + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Modificamos a de o en linea 5 de marron

pres			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	3	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```



Print: 57 + 24 + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Modificamos c de o en linea 6 de marron

pres			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

o			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```



Print: 57 + 24 + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Procedemos a entrar a t. Según o, de Marron tambien

pres			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

o			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

Print: 57 + 24 + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

Marron.s		
0	X	1

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Entramos a t de marron con y = -19

pres			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

o			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

Print: 57 + 24 + ?

Marron.t		
1	y	-19
Marron.s		
0	x	1

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

Retornamos 24

pres			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

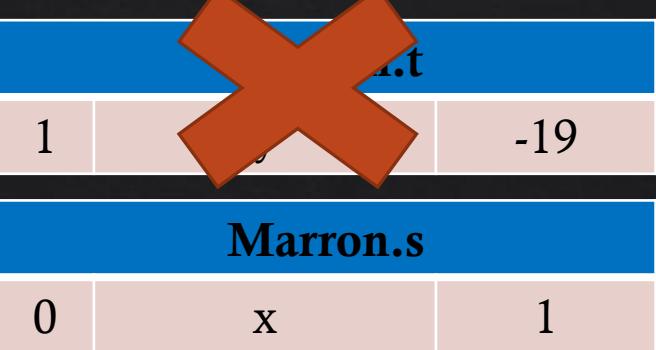
```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

o			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

Print: 57 + 24 + ?



```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

Nuevamente Retornamos 24

pres			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

o			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

Print: 57 + 24 + ?

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```



Ahora Podemos ver cuanto da print

pres			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

e			
Variables		Funciones	
a	7	Beb.	s
b	8	Beb.	t
Caliente.a	0	Cafe	q
Caliente.b	8		
Caliente.c	3		

```

1 Bebida e = new Cafe()
2 Bebida pres = new Marron()
3 Cafe o = new Marron()
4
5 print (e.s(1) + pres.s(1) + o.s(1))

```

o			
Variables		Funciones	
a	-3	Marron	s
b	8	Marron	t
c	5	Cafe	q
Caliente.a	0		
Caliente.b	8		
Caliente.c	3		

Print: 57 + 24 + 24
= Print 105

```

1 class Bebida{
2     int a = 0, b =8
3
4     fun s(int x): int{
5         a = b - x
6         return t(a)
7     }
8
9     fun t(int y): int {
10        return a * y + b
11    }
12 }

```

```

1 class Marron extends Cafe {
2     int c = 3
3
4     fun s(int x): int {
5         a = x -4
6         c = a + b * x
7         return t(a * b + c)
8     }
9
10    fun t(int y): int {
11        return c - y
12    }
13 }

```

```

1 class Cafe extends Bebida{
2     Bebida caliente = new Marron()
3
4     fun q(int y): int{
5         return caliente.s(a + b) - y
6     }
7 }

```