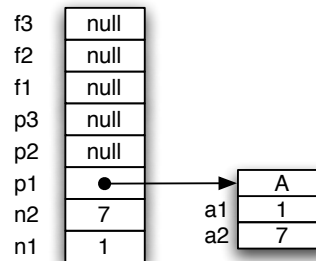


## Exécution jusqu'au Point 1

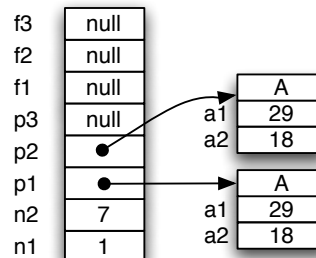
```
1  int n1;
2  int n2;
3  A p1, p2, p3;
4  B f1, f2, f3;
5
6  n1 = 1;
7  n2 = 7;
8  p1 = new A(n1,n2);
9  System.out.println("p1="+p1);
```



Rien à dire de spécial, le résultat est évident.

p1=(1,7)

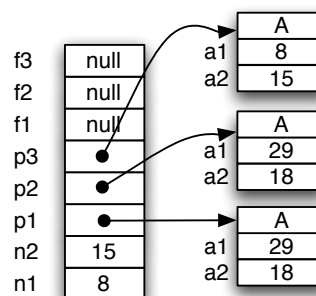
```
10 p2 = p1.met2(28,11);
11 p1.met1(28,11);
12 System.out.println("p1="+p1);
13 System.out.println("p2="+p2);
14 System.out.println(p1==p2);
15 System.out.println(p1.egale(p2));
```



Les deux objets sont syntaxiquement égaux (même valeur des attributs). Cependant les références ne sont pas égales (deux objets différents).

p1=(29,18)  
p2=(29,18)  
false  
true

```
16 n1 = n1 + n2;
17 n2 = n1 + n2;
18 p3 = new A (n1, n2);
19 System.out.println("p3="+p3);
```



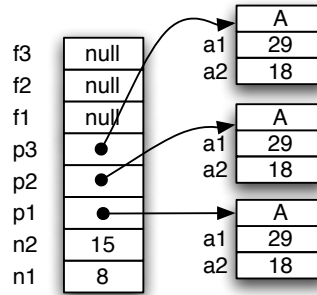
Il faut juste ne pas oublier de tenir compte de la première addition lorsque l'on calcule la seconde.

p3=(8,15)

```

20 p3.met1(21,3);
21 System.out.println("p3="+p3);
22 System.out.println(p1 == p3);
23 System.out.println(p1.egale(p3));

```



Ici encore rien de spécial. Les références ne sont pas égales bien que les objets le soit syntaxiquement.

```

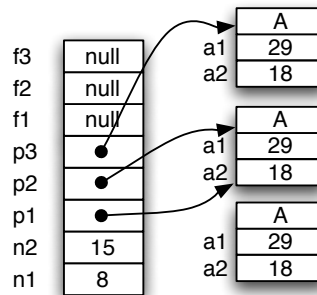
p3=(29,18)
false
true

```

```

24 p1=p2;
25 System.out.println(p1==p2);
26 System.out.println(p1.egale(p2));
27 System.out.println("");

```



Cette fois-ci, les références sont égales et les instances également.

```

true
true

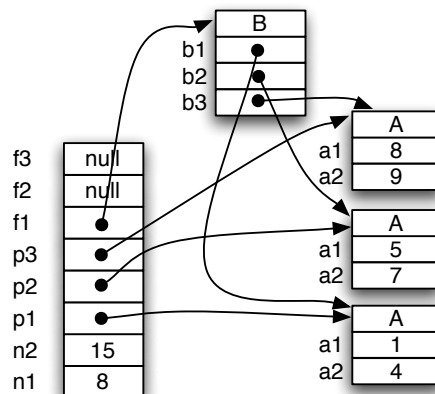
```

## Exécution jusqu'au Point 2

```

28 p1 = new A(1,4);
29 p2 = new A(5,7);
30 p3 = new A(8,9);
31 f1 = new B(p1,p2,p3);
32 System.out.println("f1="+f1);

```



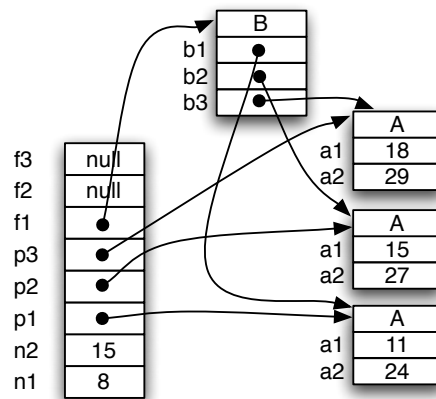
Rien de spécial.

```

f1=[(1,4);(5,7);(8,9)]

```

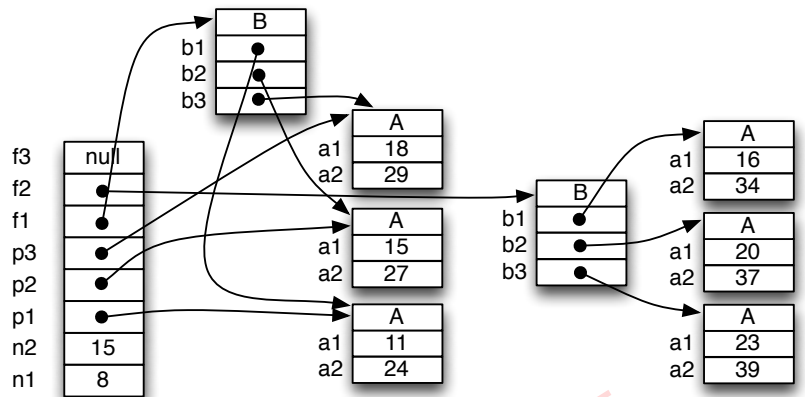
```
33 f1.met1(10,20);
34 System.out.println("f1="+f1);
```



Rien de spécial.

```
f1=[(11,24);(15,27);(18,29)]
```

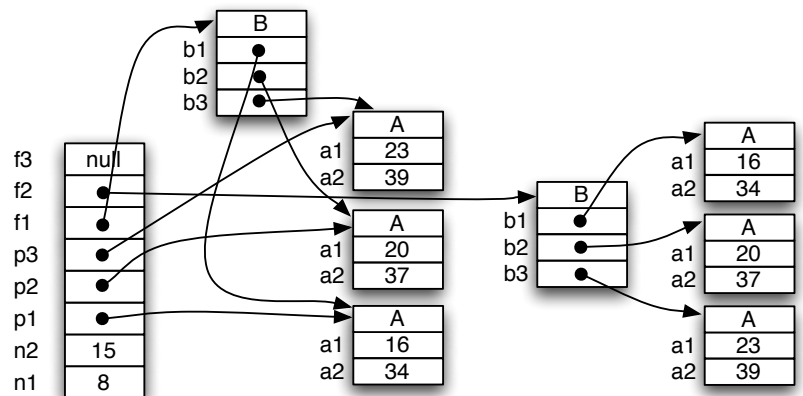
```
35 f2 = f1.met2(5,10);
36 System.out.println("f1="+f1);
37 System.out.println("f2="+f2);
38 System.out.println(f1==f2);
```



On crée une nouvelle instance de la classe B en recopiant un objet existant. Puis on a modifié la valeur de ces champs. Les deux références sont donc différentes.

```
f1=[(11,24);(15,27);(18,29)]
f2=[(16,34);(20,37);(23,39)]
false
```

```
39 f1.met1(5,10);
40 System.out.println("f1="+f1);
41 System.out.println(f1==f2);
42 System.out.println(f1.egale(f2));
43 System.out.println("");
```



Après avoir modifié les valeurs des attributs de l'instance f1, les deux objets sont syntaxiquement égaux, mais les références restent différentes.

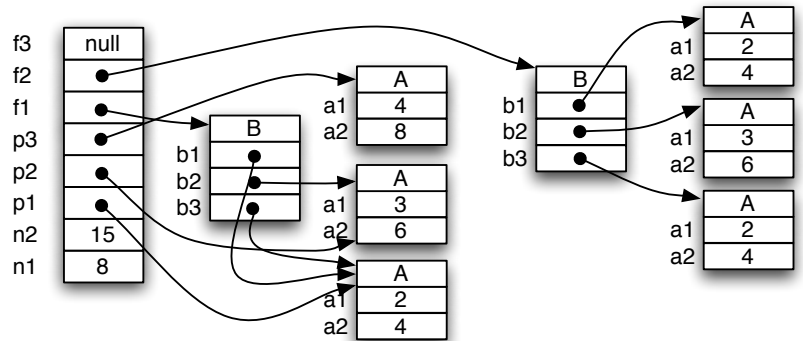
```
f1=[(16,34);(20,37);(23,39)]
false
true
```

## Exécution jusqu'au Point 3

```

44 p1 = new A(2,4);
45 p2 = new A(3,6);
46 p3 = new A(4,8);
47 f1 = new B(p1,p2,p1);
48 f2 = new B(new A(2,4),
49             new A(3,6),
50             new A(2,4));
51 System.out.println(f1==f2);
52 System.out.println(f1.egale(f2));
53 System.out.println("f1="+f1);

```



Rien de spécial, si ce n'est que le premier et le troisième attributs de `p1` référencent le même objet. Alors que concernant l'instance `f2`, les attributs ont la même valeur mais sont deux objets différents.

```

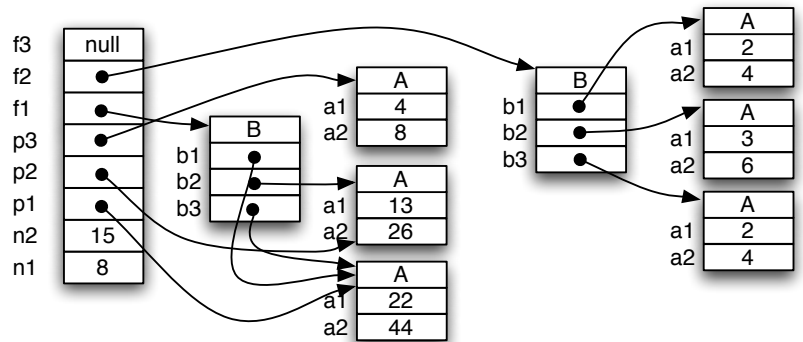
false
true
f1=[(2,4);(3,6);(2,4)]

```

```

54 f1.met1(10,20);
55 System.out.println("f1="+f1);
56 System.out.println("f2="+f2);

```



Comme le premier et le troisième attribut de l'instance `f1` référencent le même objet, cet objet sera mis à jour successivement deux fois lors de l'appel de la méthode `met1()`.

```

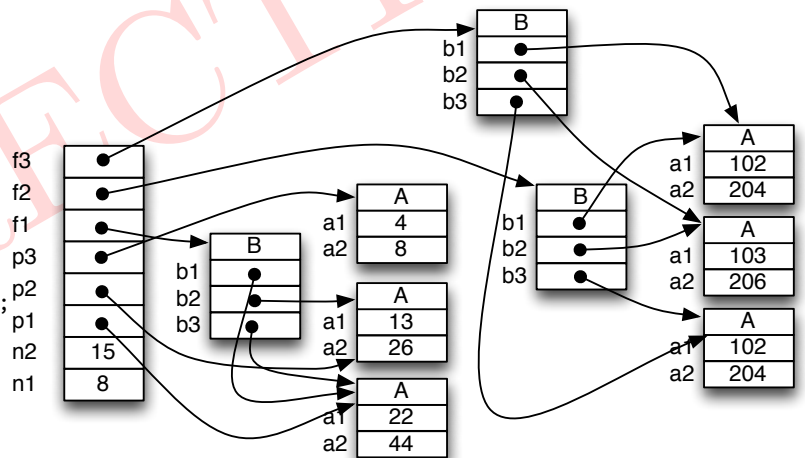
f1=[(22,44);(13,26);(22,44)]
f2=[(2,4);(3,6);(2,4)]

```

```

57 f3 = f2.met3(100,200);
58 System.out.println("f2="+f2);
59 System.out.println("f3="+f3);
60 System.out.println(f2==f3);
61 System.out.println(f2.egale(f3));

```



```

f2=[(102,204);(103,206);(102,204)]
f3=[(102,204);(103,206);(102,204)]
false
true

```

les références `f2` et `f3` sont différentes. Par contre les objets référencés sont syntaxiquement égaux.

### En résumé

Le résultat de l'exécution complète du programme est le suivant :

```
p1=(1,7)
p1=(29,18)
p2=(29,18)
false
true
p3=(8,15)
p3=(29,18)
false
true
true
true

f1=[(1,4);(5,7);(8,9)]
f1=[(11,24);(15,27);(18,29)]
f1=[(11,24);(15,27);(18,29)]
f2=[(16,34);(20,37);(23,39)]
false
f1=[(16,34);(20,37);(23,39)]
false
true

false
true
f1=[(2,4);(3,6);(2,4)]
f1=[(22,44);(13,26);(22,44)]
f2=[(2,4);(3,6);(2,4)]
f2=[(102,204);(103,206);(102,204)]
f3=[(102,204);(103,206);(102,204)]
false
true
```