Focus sur le passage de paramètres

Faculté des sciences, Université de Montpellier
Module HA8403I
Programmation par objets

Comprendre le passage de paramètres en Java

Passage de paramètre en Java

Un paramètre est toujours passé par valeur

Deux exemples :

- avec un paramètre de type primitif
- avec un paramètre de type construit (classe)

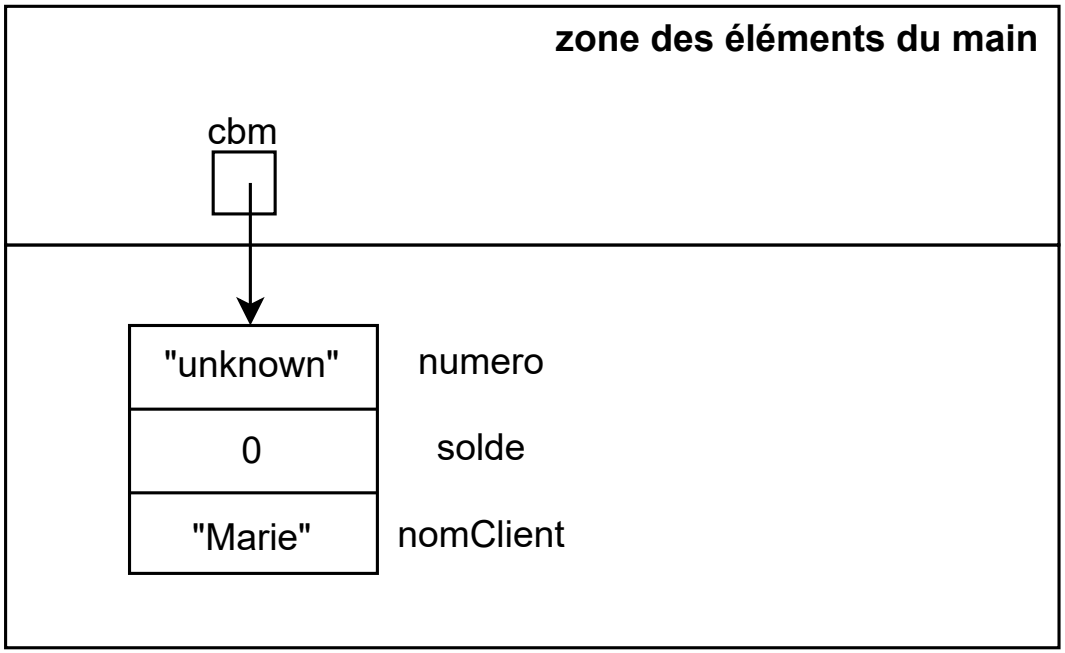
Paramètre de type primitif

```
public class CompteBancaire {
        private String numero = "unknown";
        private double solde;
        private String nomClient;

        public void setSolde(double nouveauSolde) {
                if (nouveauSolde  >= 0) {this.solde = nouveauSolde;}
        }

        // methode qui augmente le solde de val%
        public void augmenteLeSoldeDeValpourCent(double val){
                val = 1 + val/100;
                this.solde = this.solde * val;
        }
}
```

```
// Dans un main
CompteBancaire cbm = new CompteBancaire("Marie");
cbm.setSolde(100);
double v = 10;
System.out.println(v);
cbm.augmenteLeSoldeDeValpourCent(v);
System.out.println(v);
```

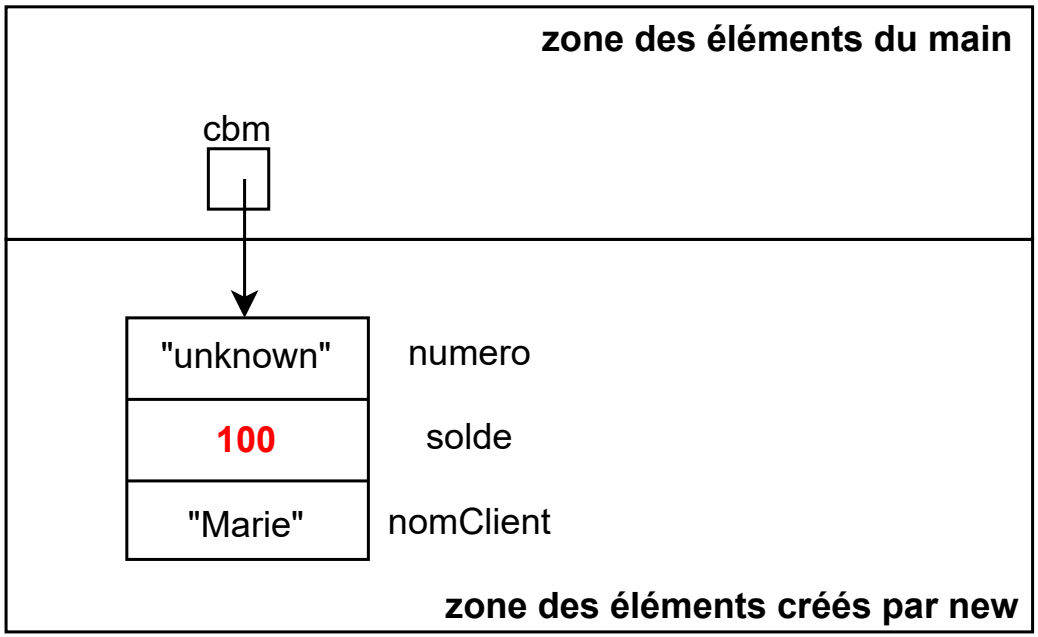
Paramètre de type primitif

CompteBancaire cbm = new CompteBancaire("Marie") ;



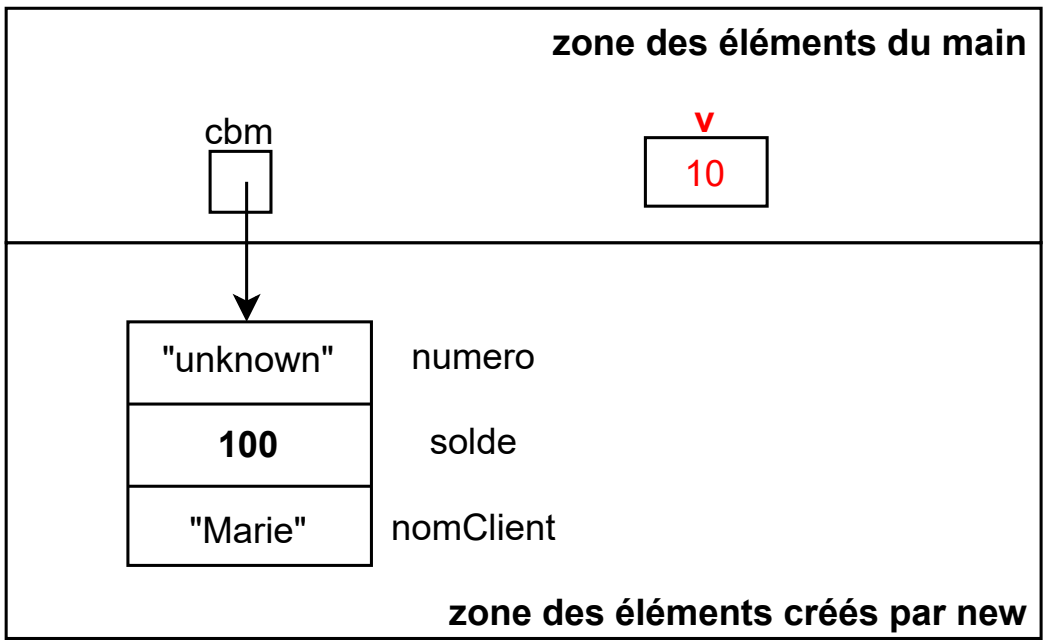
Paramètre de type primitif

cbm.setSolde(100) ;



Paramètre de type primitif

double v = 10 ;

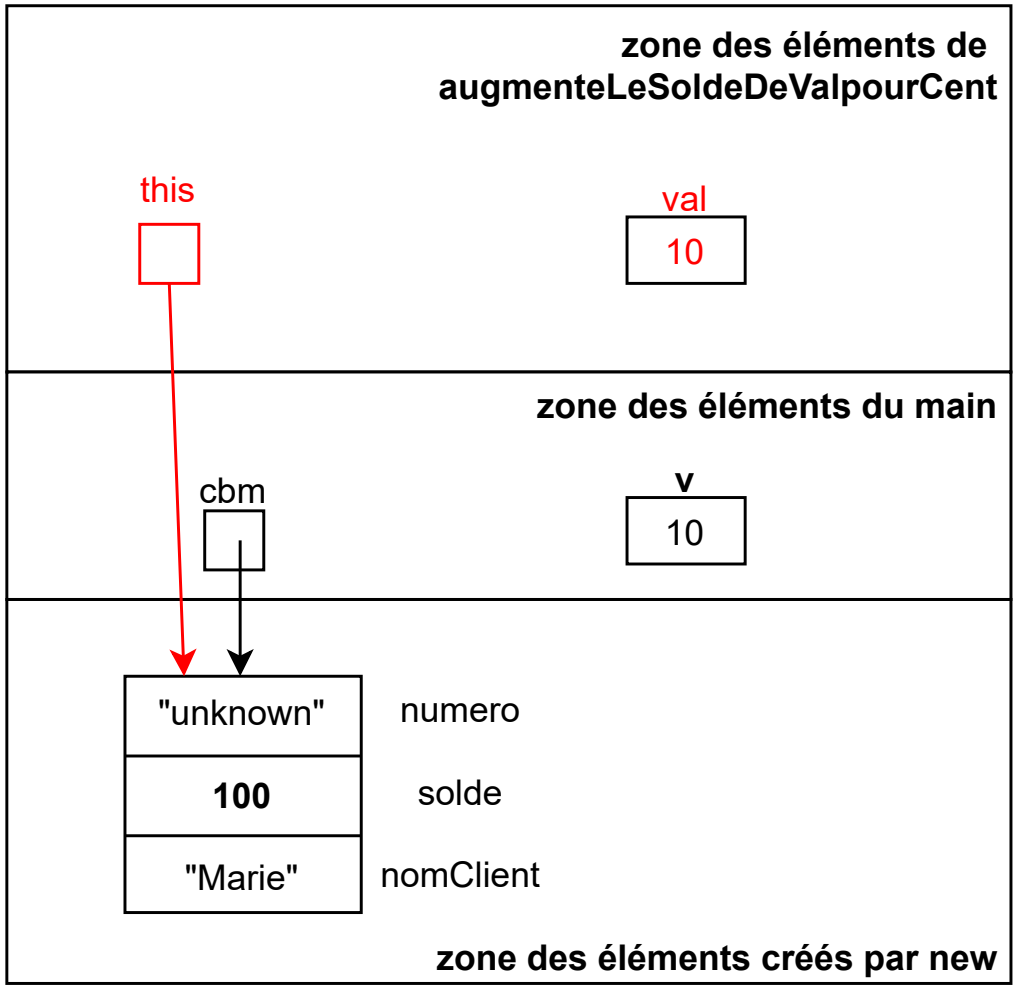


Paramètre de type primitif

Appel de `augmenteLeSoldeDeValpourCent(double val)`

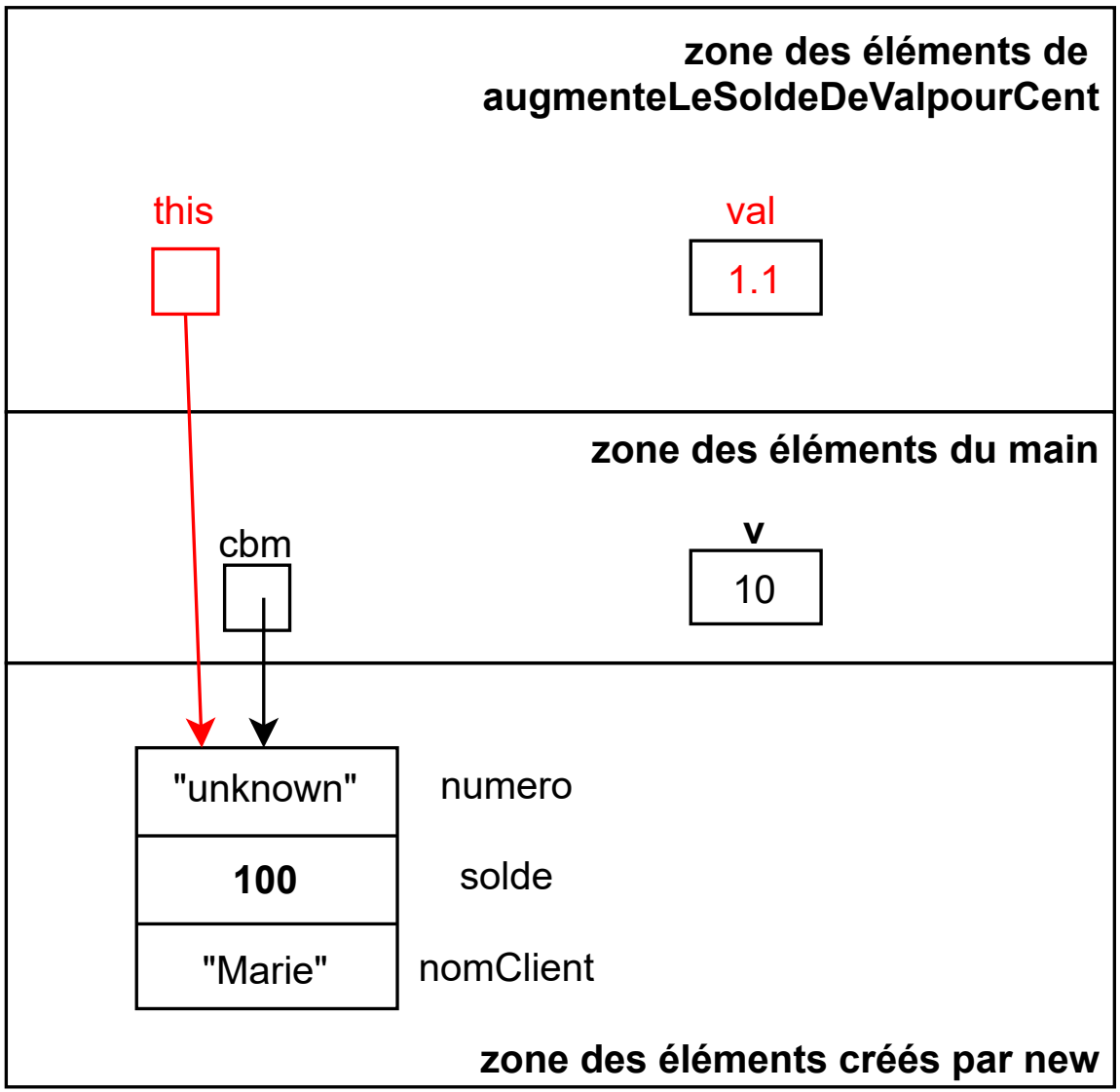
`cbm.augmenteLeSoldeDeValpourCent(v) ;`

la valeur du paramètre réel `v` est copiée dans le paramètre formel `val`



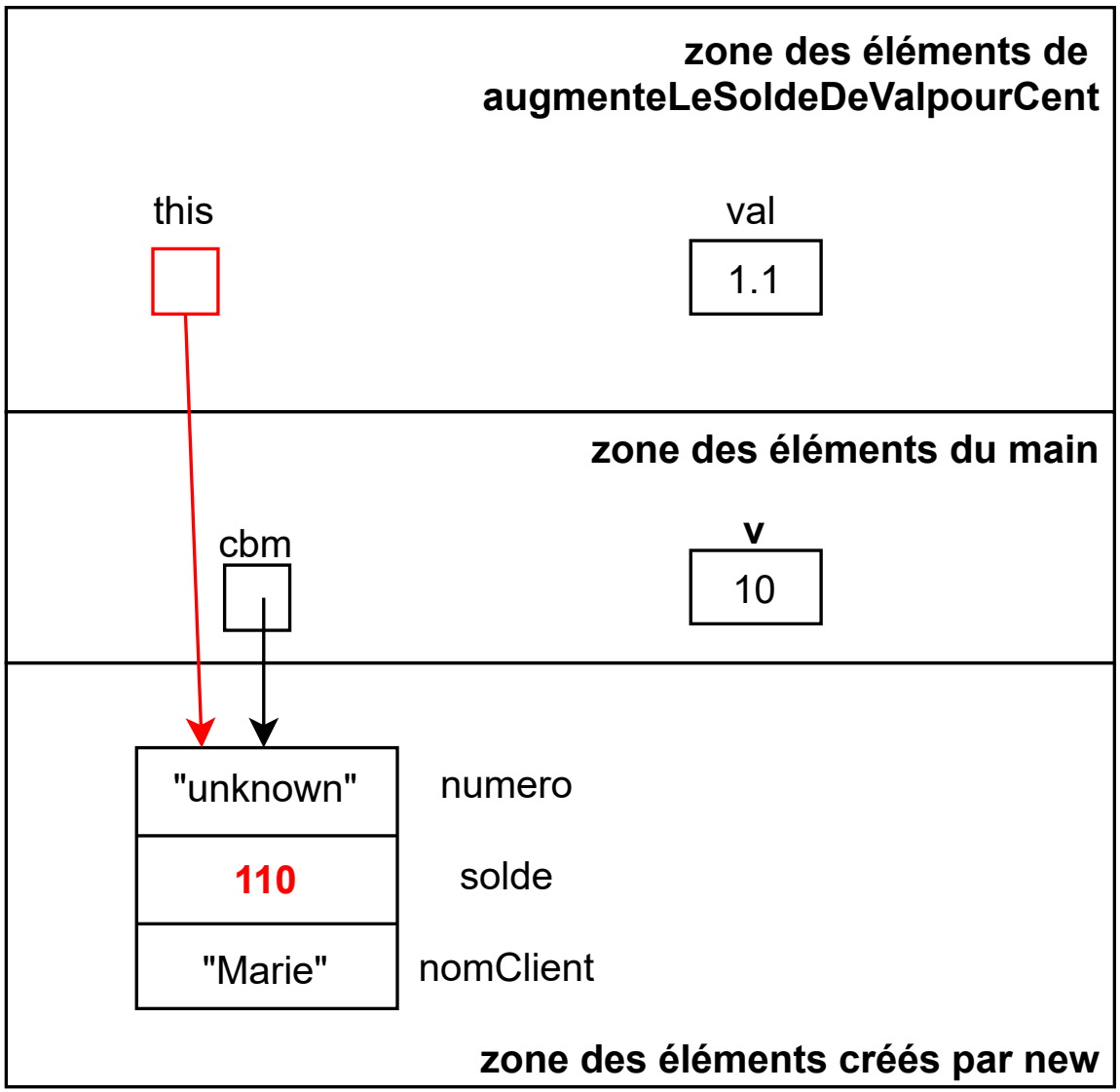
Paramètre de type primitif

val = 1 + val/100 ; `val` est modifiée



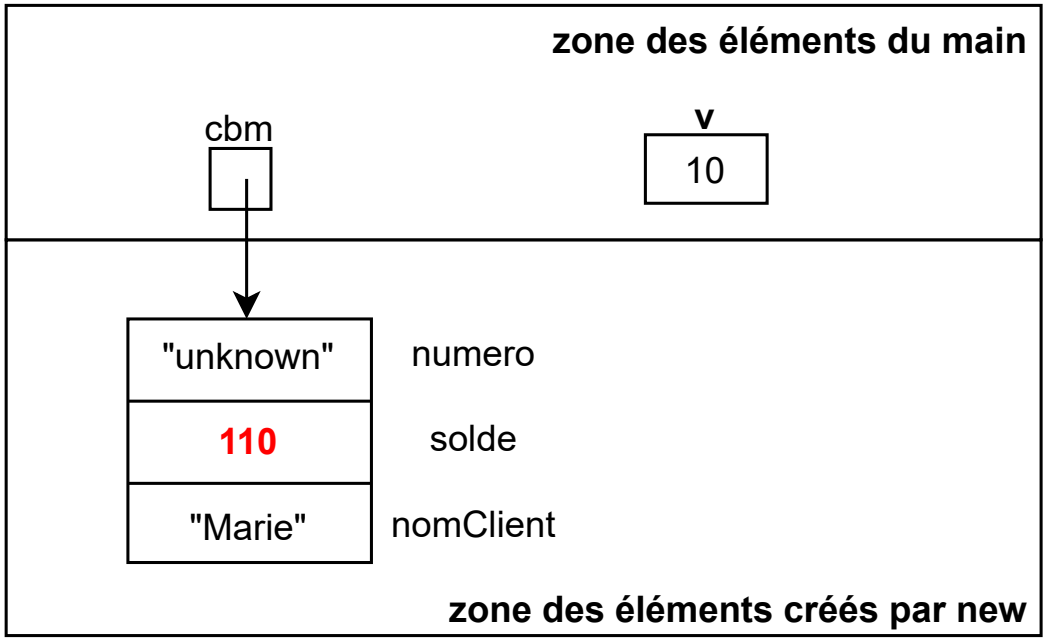
Paramètre de type primitif

this.solde = this.solde * val ;



Paramètre de type primitif

Et en revenant de `augmenteLeSoldeDeValpourCent`
.... dans le main la valeur de `v` n'a jamais été changée



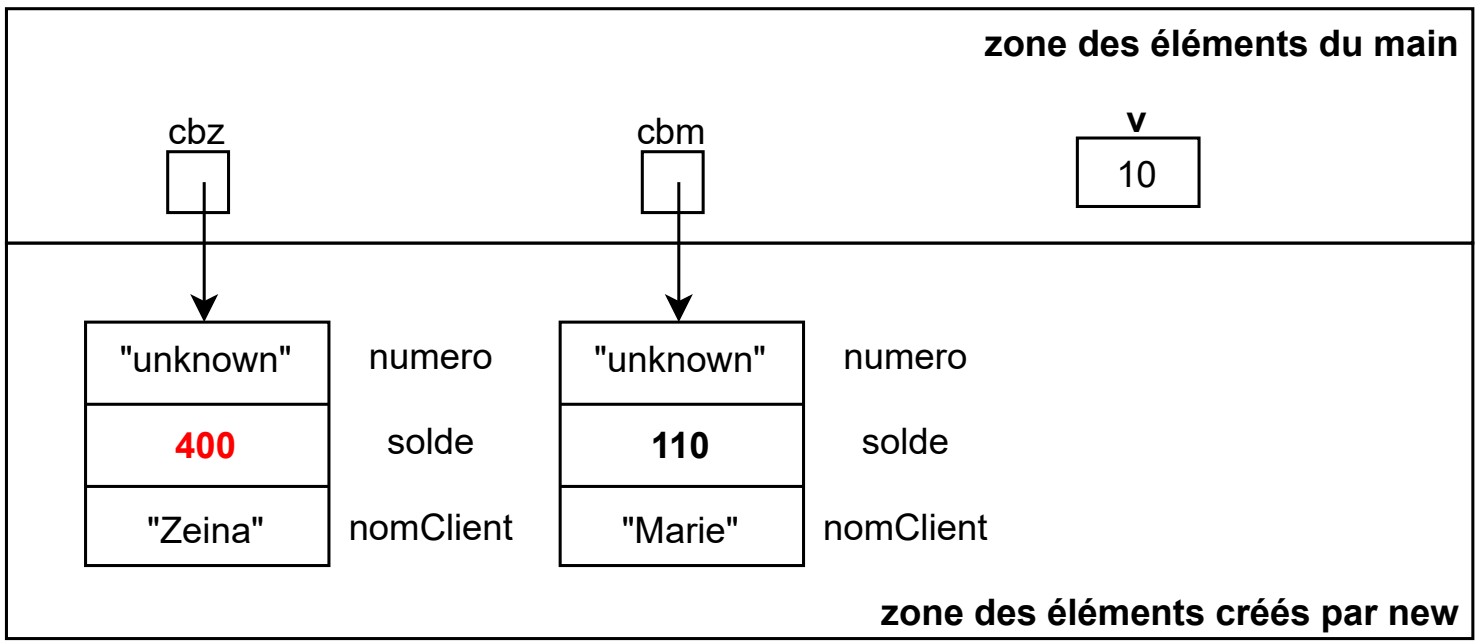
Paramètre de type référence d'objet

La référence ("adresse mémoire") de l'objet est passée par valeur.
On ne pourra pas modifier la variable référence, mais on peut modifier l'objet au travers de sa référence.

```
public class CompteBancaire{....
    public void virement(double v, CompteBancaire autreCompte) {
        if (v>=0 && this.getSolde()>=v) {
            this.solde = this.solde - v;
            autreCompte.solde = autreCompte.solde+v;
        }
        else System.out.println("virement impossible");
    }
}
// Dans la suite du main
   ....
   CompteBancaire cbz = new CompteBancaire("Zeina");
   cbz.setSolde(400);
   cbz.virement(200, cbm);
```

Paramètre de type référence d'objet

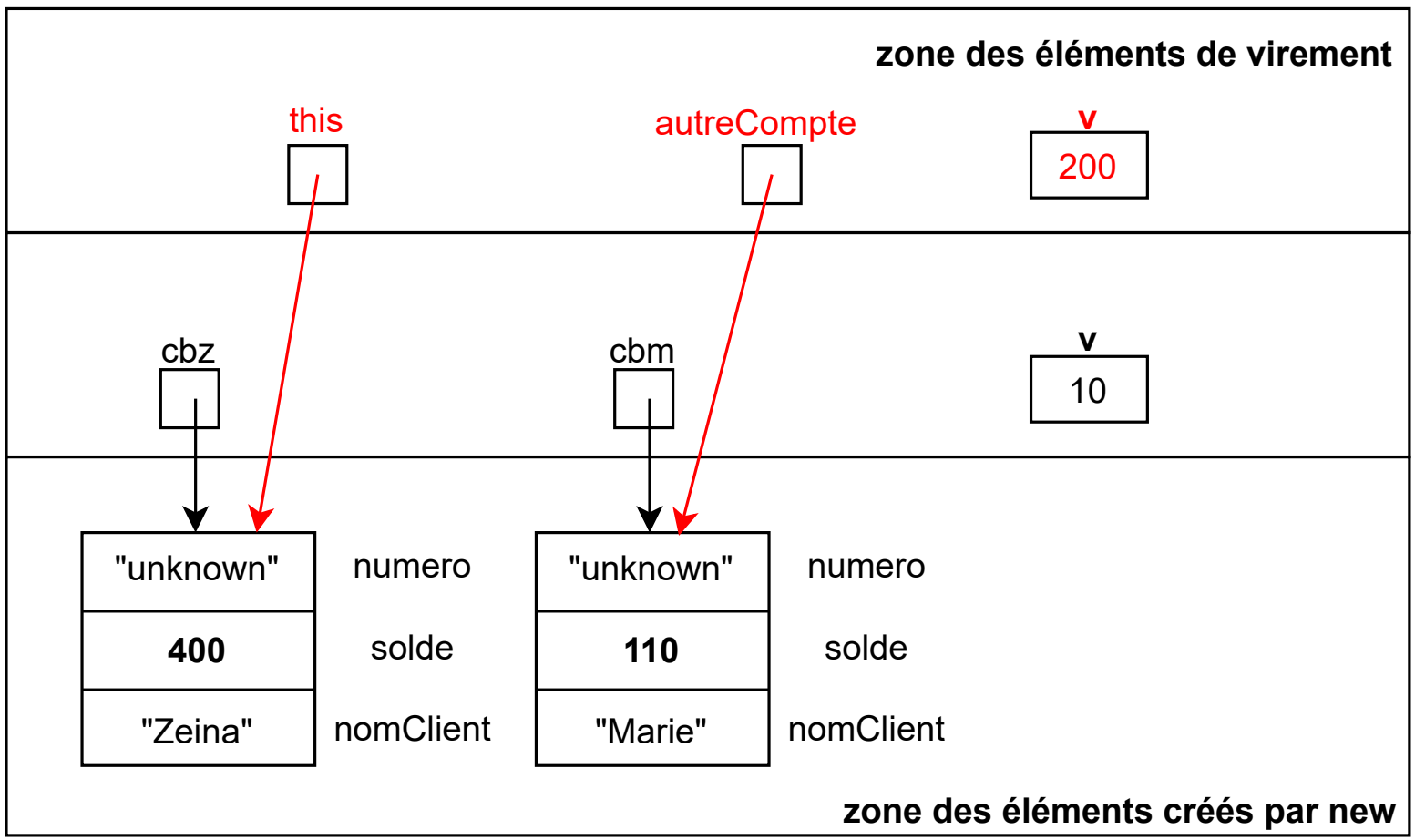
```
CompteBancaire cbz = new CompteBancaire("Zeina");
cbz.setSolde(400);
```



Paramètre de type référence d'objet

Appel de `virement(double v, CompteBancaire autreCompte)`

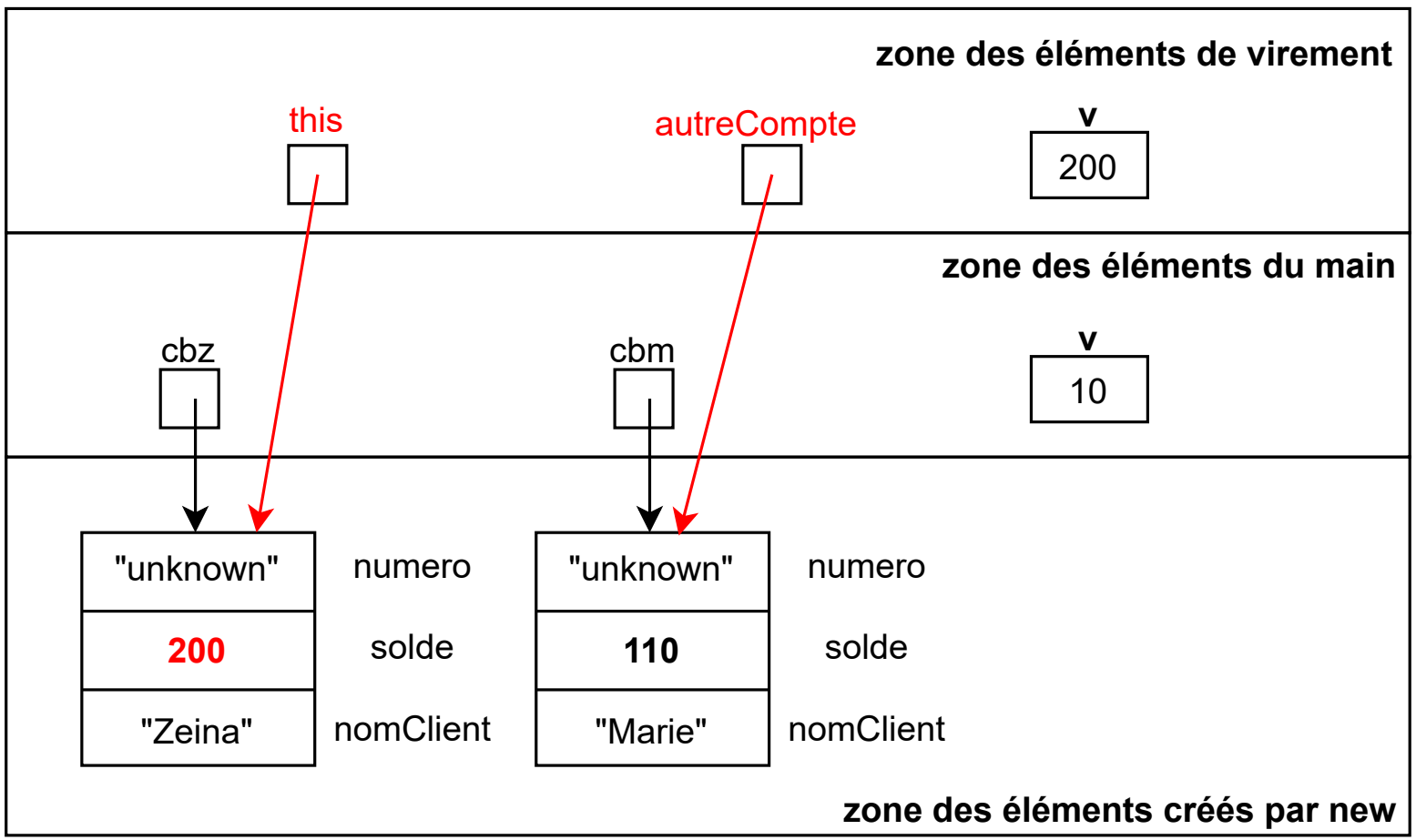
cbz.virement(200, cbm) ;



Paramètre de type référence d'objet

Dans `virement(double v, CompteBancaire autreCompte)`

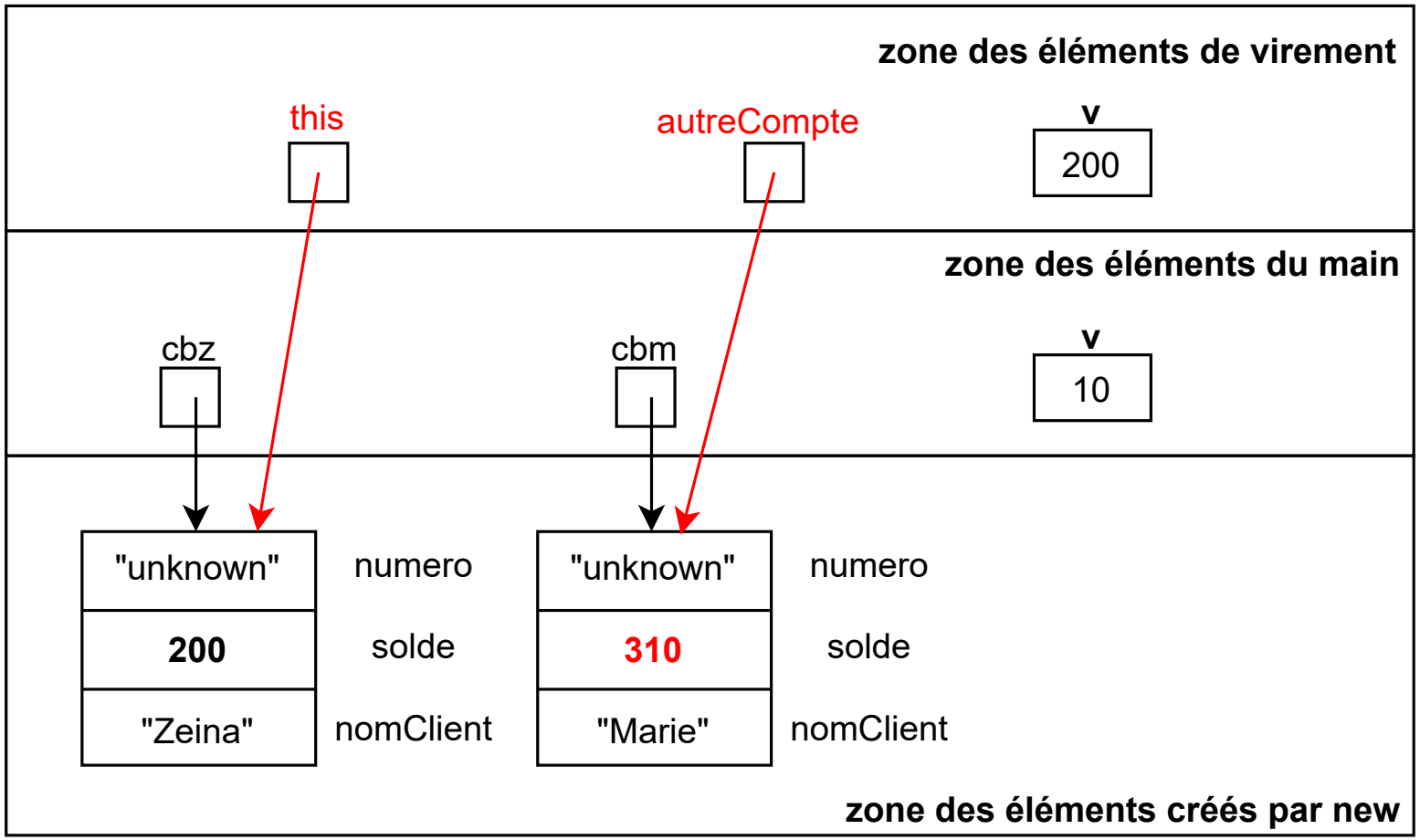
this.solde = this.solde - v ;



Paramètre de type référence d'objet

Dans `virement(double v, CompteBancaire autreCompte)`

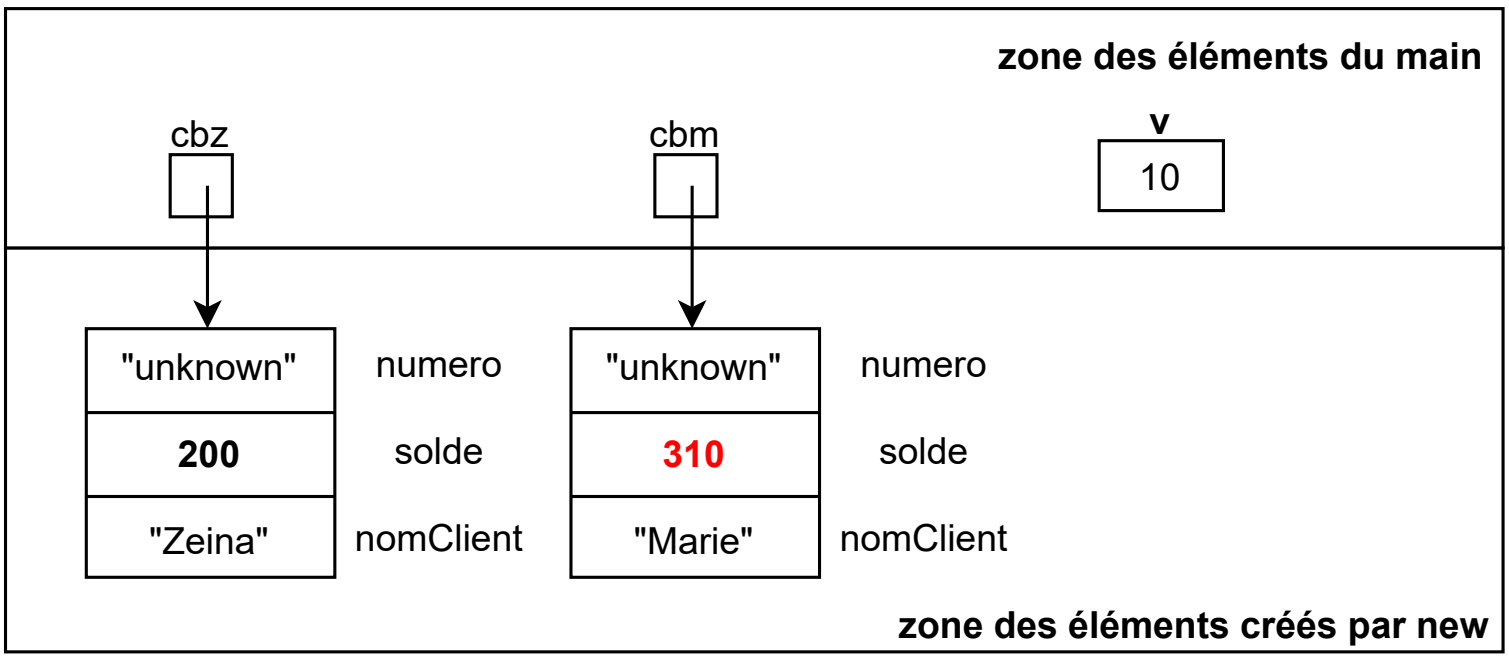
autreCompte.solde = autreCompte.solde+v ;



Paramètre de type référence d'objet

De retour dans le main en revenant de ...

`virement(double v, CompteBancaire autreCompte)`



Synthèse

Retenir, en Java :

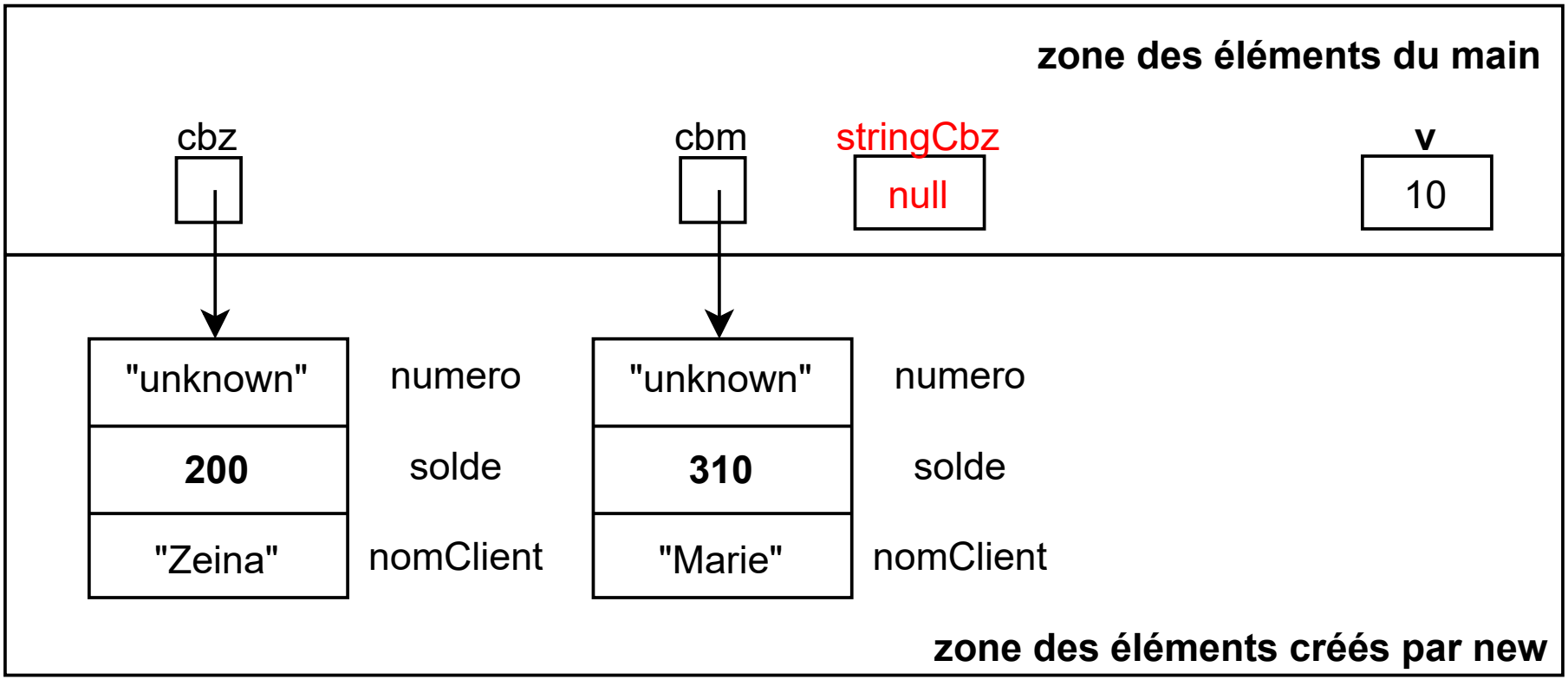
- Tout paramètre est passé par valeur : la valeur du paramètre réel (lors de l'appel) est copié dans le paramètre formel (celui de la déclaration)
- La valeur du paramètre réel ne peut donc être changée
- La valeur du paramètre formel peut être changée mais cela n'a pas d'influence sur celle du paramètre réel
- Lorsque le paramètre est une référence, l'objet désigné par la référence, lui, peut être modifié (c'est la référence qui n'est pas modifiée)

Retour de valeur par une méthode

```
public class CompteBancaire{....
        public String toString() {
                return "Client : "+this.nomClient
                        +" Numéro : "+this.numero+" solde = "
                        +this.solde;
        }
 }
// Dans la suite du main
   ....
   String stringCbz = null;
   stringCbz  = cbz.toString();
   .....
   System.out.println(stringCbz);
   .....
```

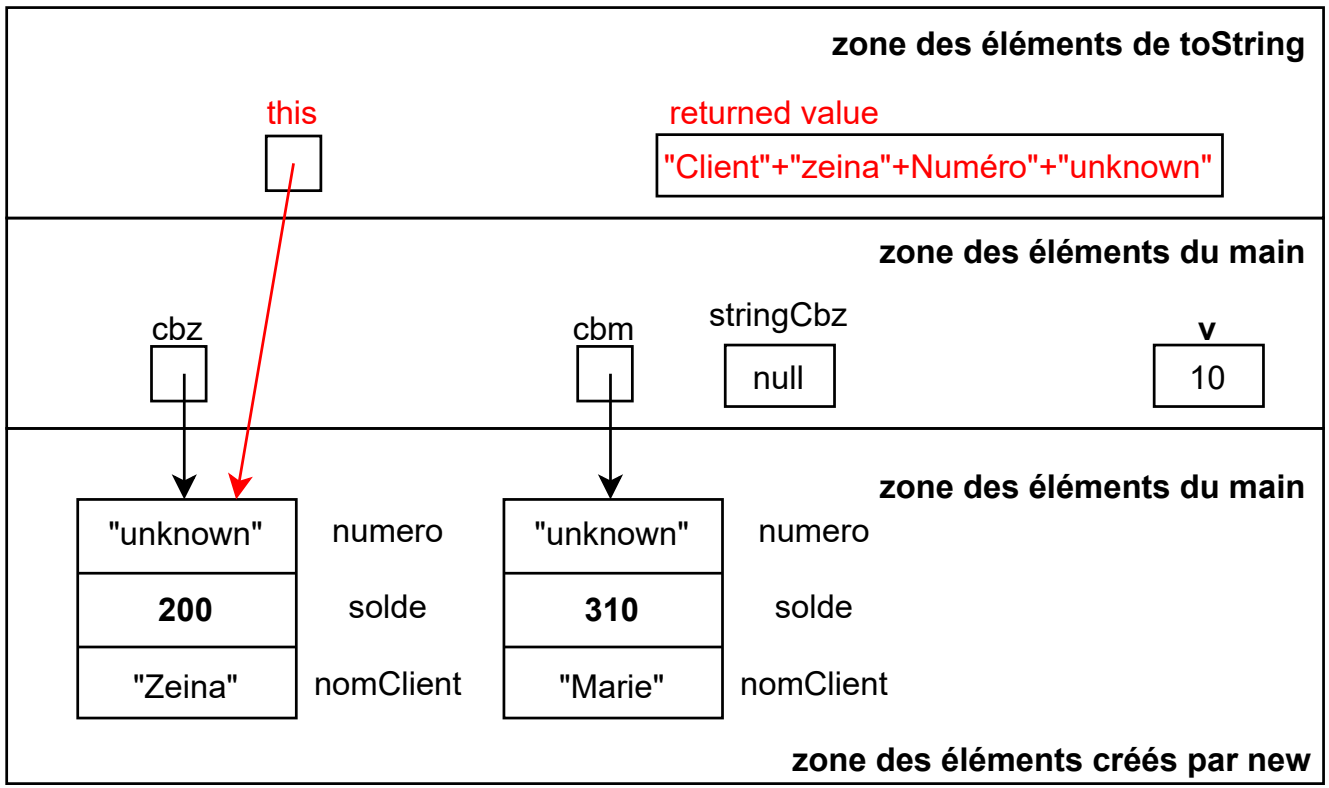
Paramètre de type référence d'objet

```
String stringCbz = null;
```



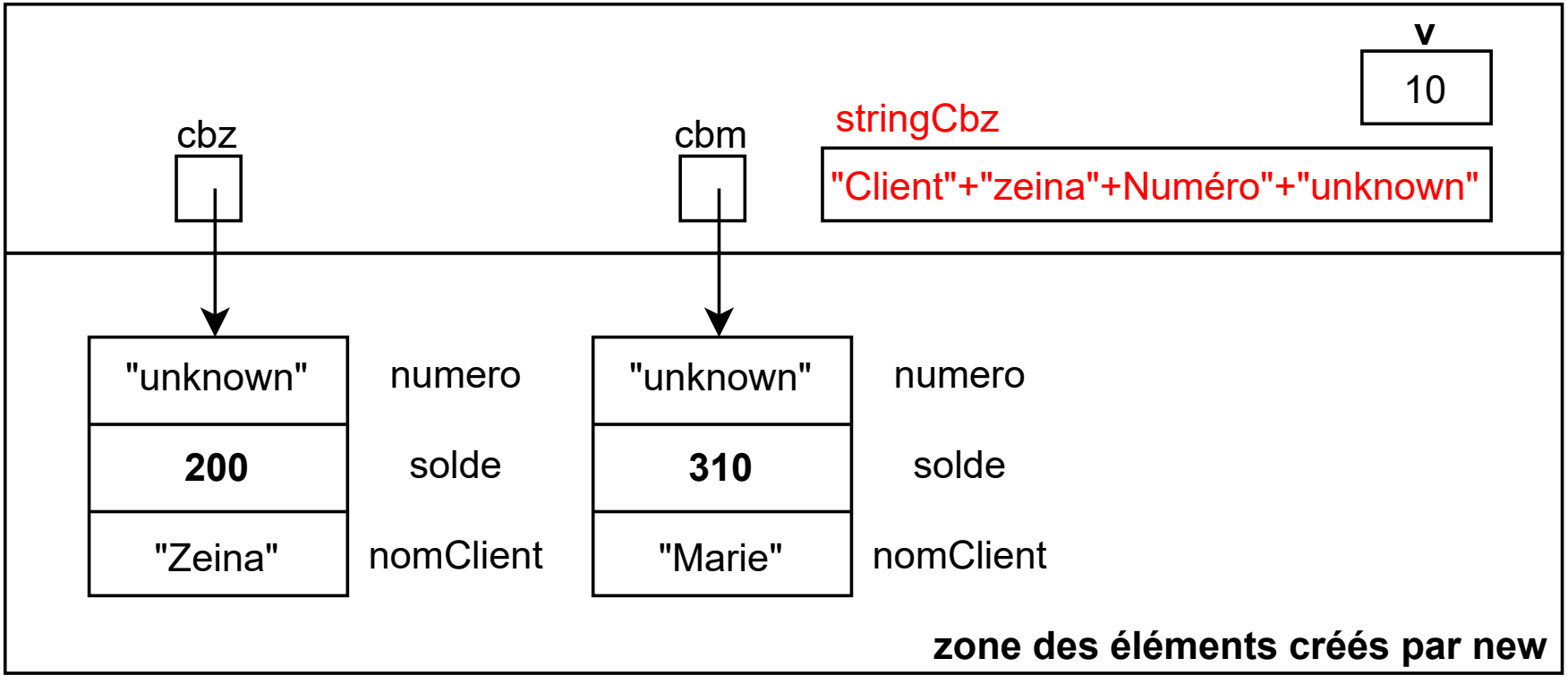
Paramètre de type référence d'objet

```
stringCbz = cbz.toString();
```



Paramètre de type référence d'objet

```
stringCbz = cbz.toString();
```



Synthèse

Retenir, en Java :

- Un espace est réservé dans la zone d'exécution de la méthode (comme pour les variables locales et les paramètres) ; il disparaît lorsque la méthode termine
- L'expression d'appel (par exemple cbz.toString()) est remplacée par la valeur retournée par la méthode