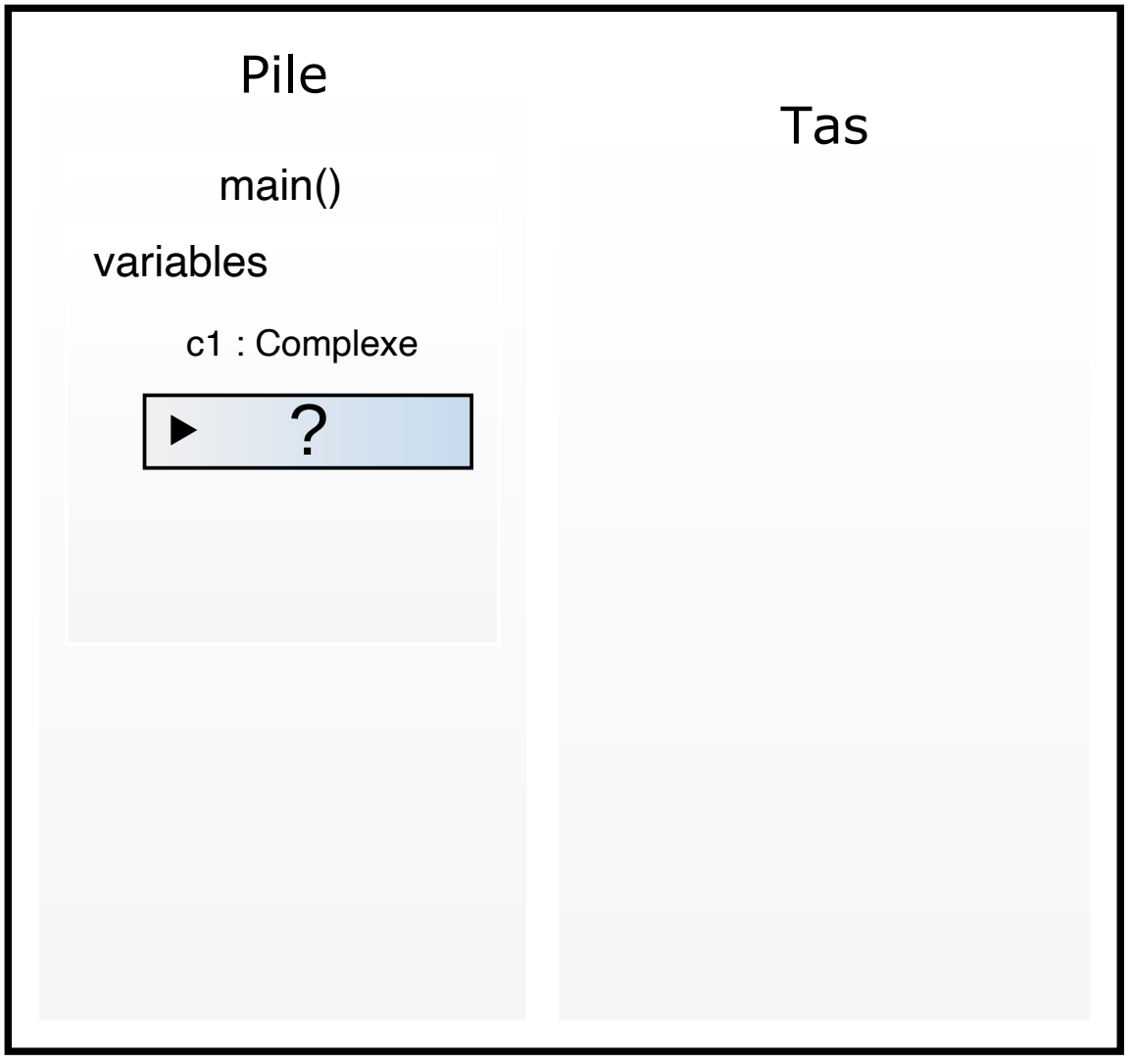
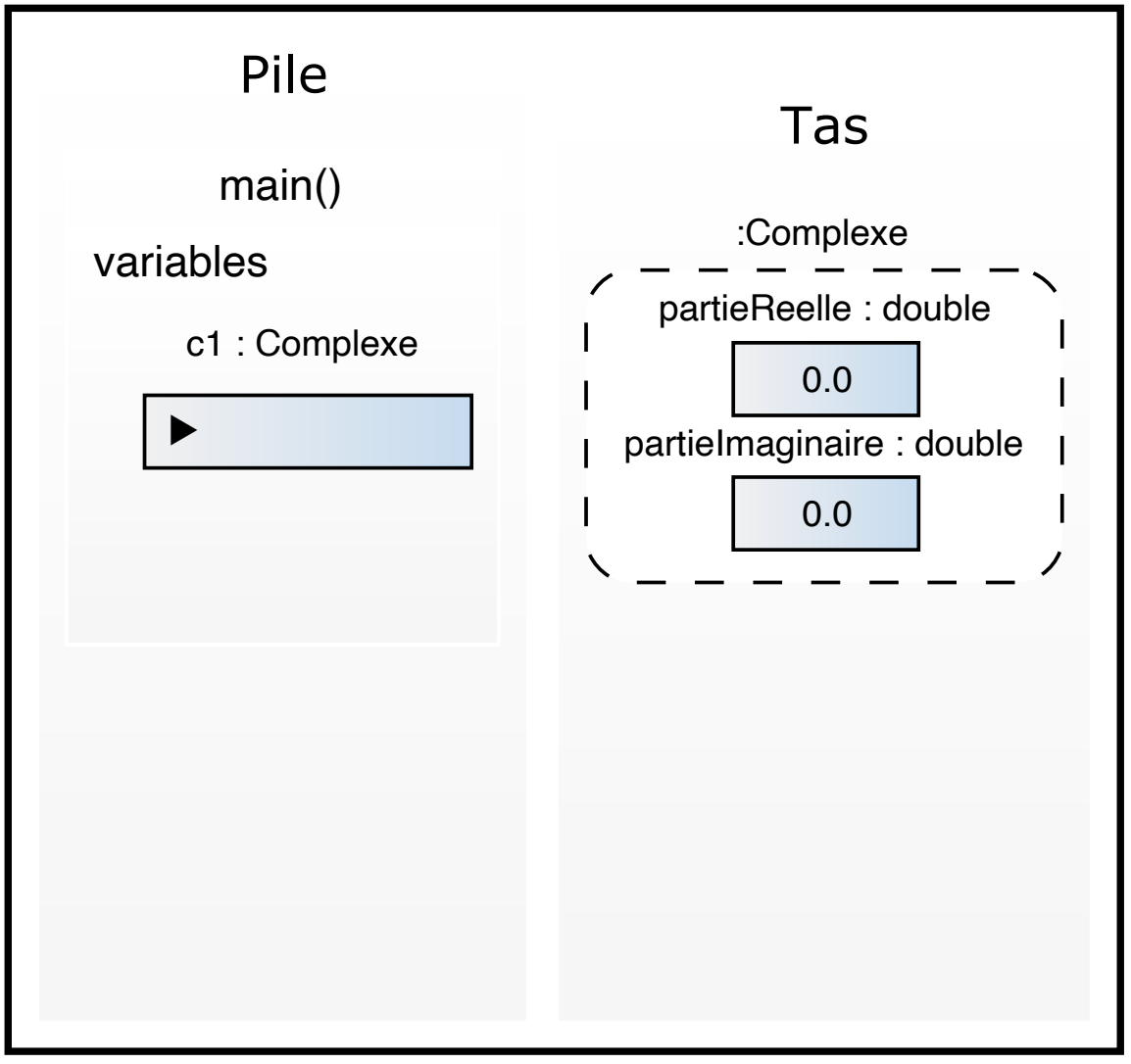

Pile
Tas
main()
variables



Ici, dans

- la méthode `main` de la classe `Test`,
  - Déclaration d'une variable
    - appelée `c1` et
    - de type Complexe

      - `Complexe c1 ;`

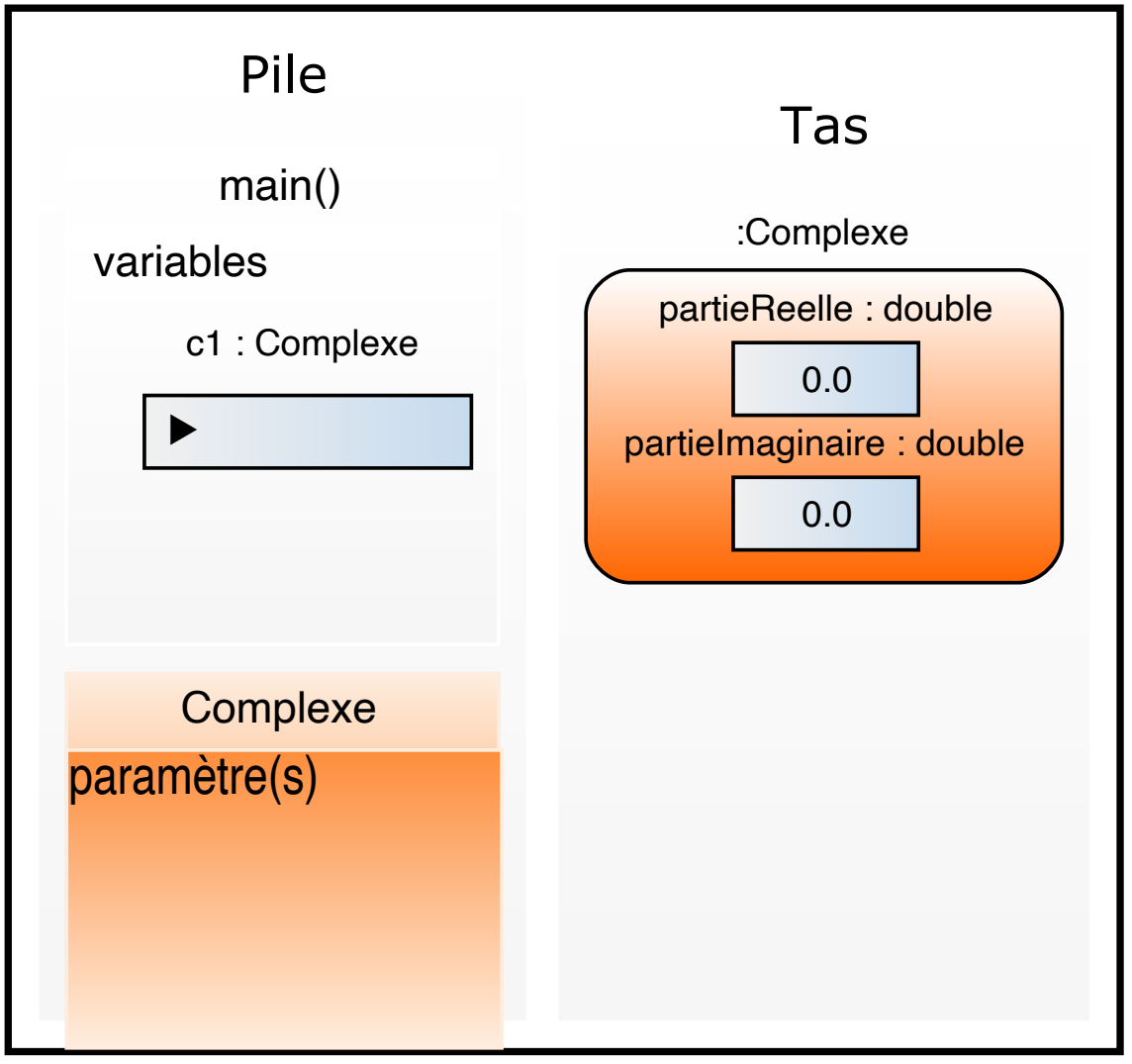


Ici, dans

- la méthode `main` de la classe `Test`,
  - 1ère étape de l'instanciation
    - Appel à l'opérateur `new` (création de l'instance)

      - `c1 = new Complexe() ;`

- Dans la classe Complexe
  - On se rend compte que la classe Complexe comporte 2 attributs :
    - `double partieReelle ;`
    - `double partieImaginaire ;`



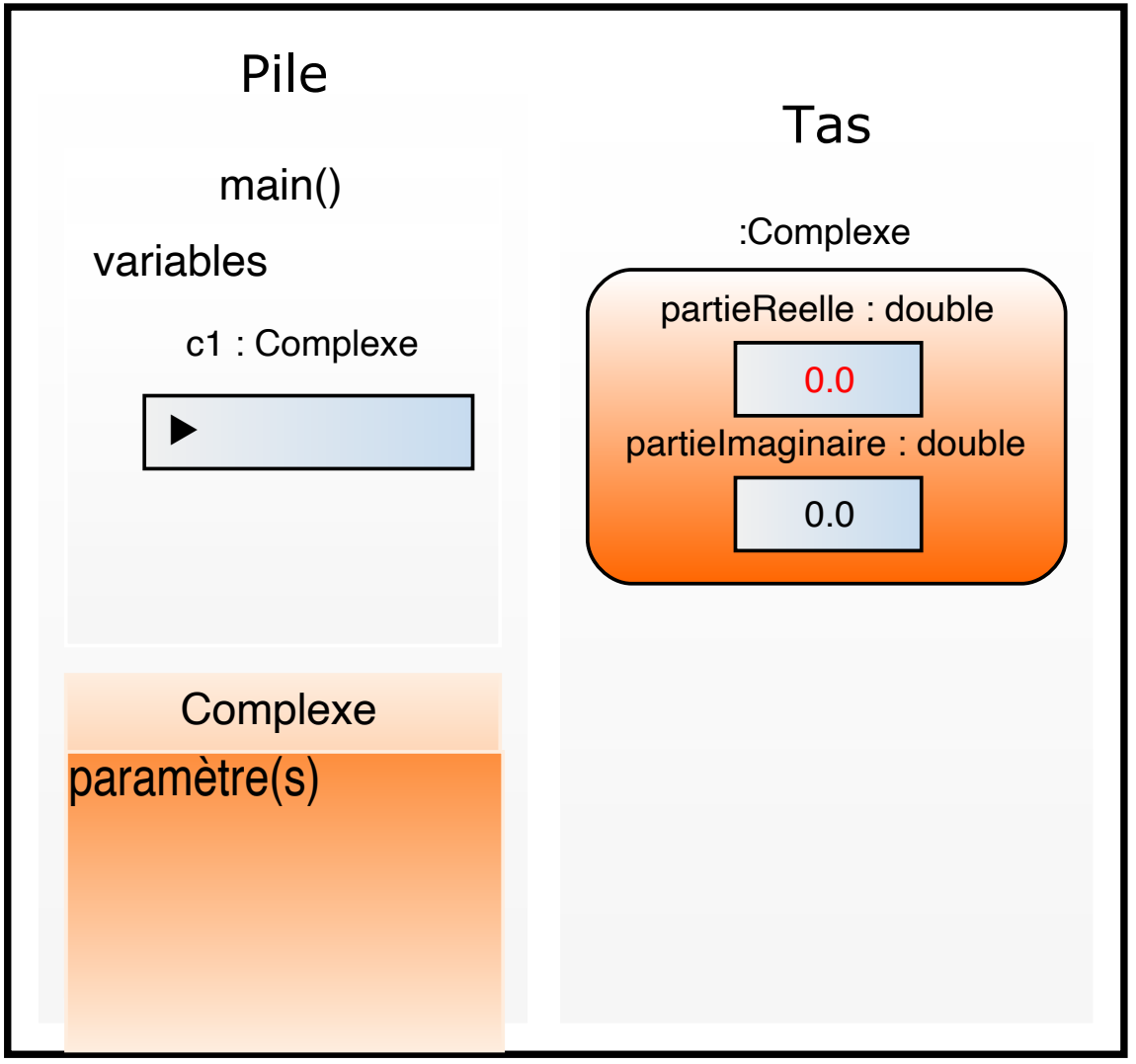
Ici, dans

- la méthode `main` de la classe `Test`,
  - 2ème étape de l'instanciation
    - Appel au constructeur Complexe() (ici pas de paramètre => constructeur par défaut

      - `c1 = new Complexe() ;`

- Dans la classe Complexe
  - Création d'un constructeur par défaut (qui ne prend pas de paramètre)

```
Complexe() {


}
```

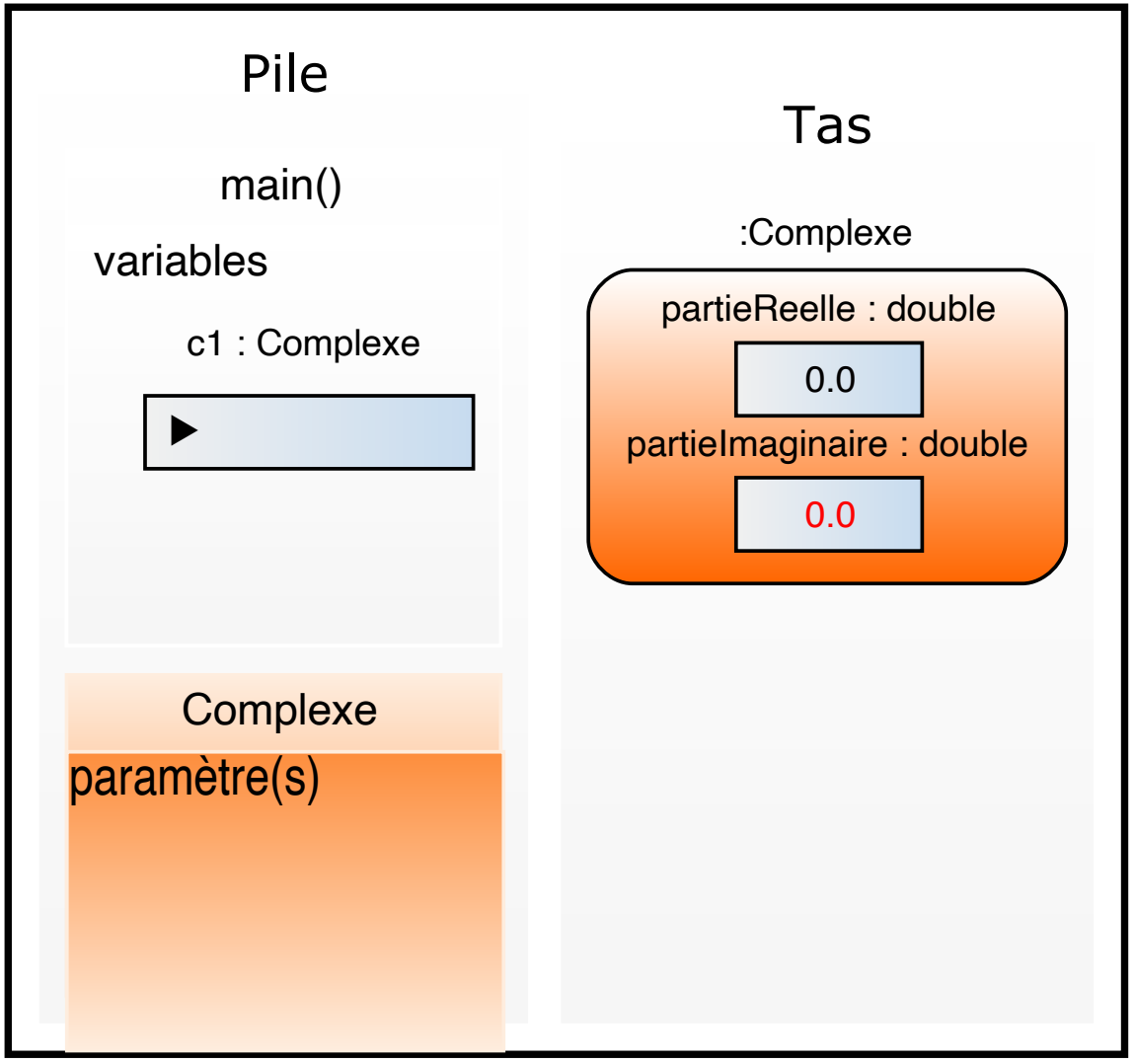


Ici, dans

- la méthode `main` de la classe `Test`,
  - 2ème étape de l'instanciation
    - Appel au constructeur Complexe() (ici pas de paramètre => constructeur par défaut
      - `c1 = new Complexe() ;`

- Dans la classe Complexe
  - Dans le constructeur, l'attribut partieReelle est mis à zéro

```
Complexe() {
   PartieReelle = 0.0 ;

}
```

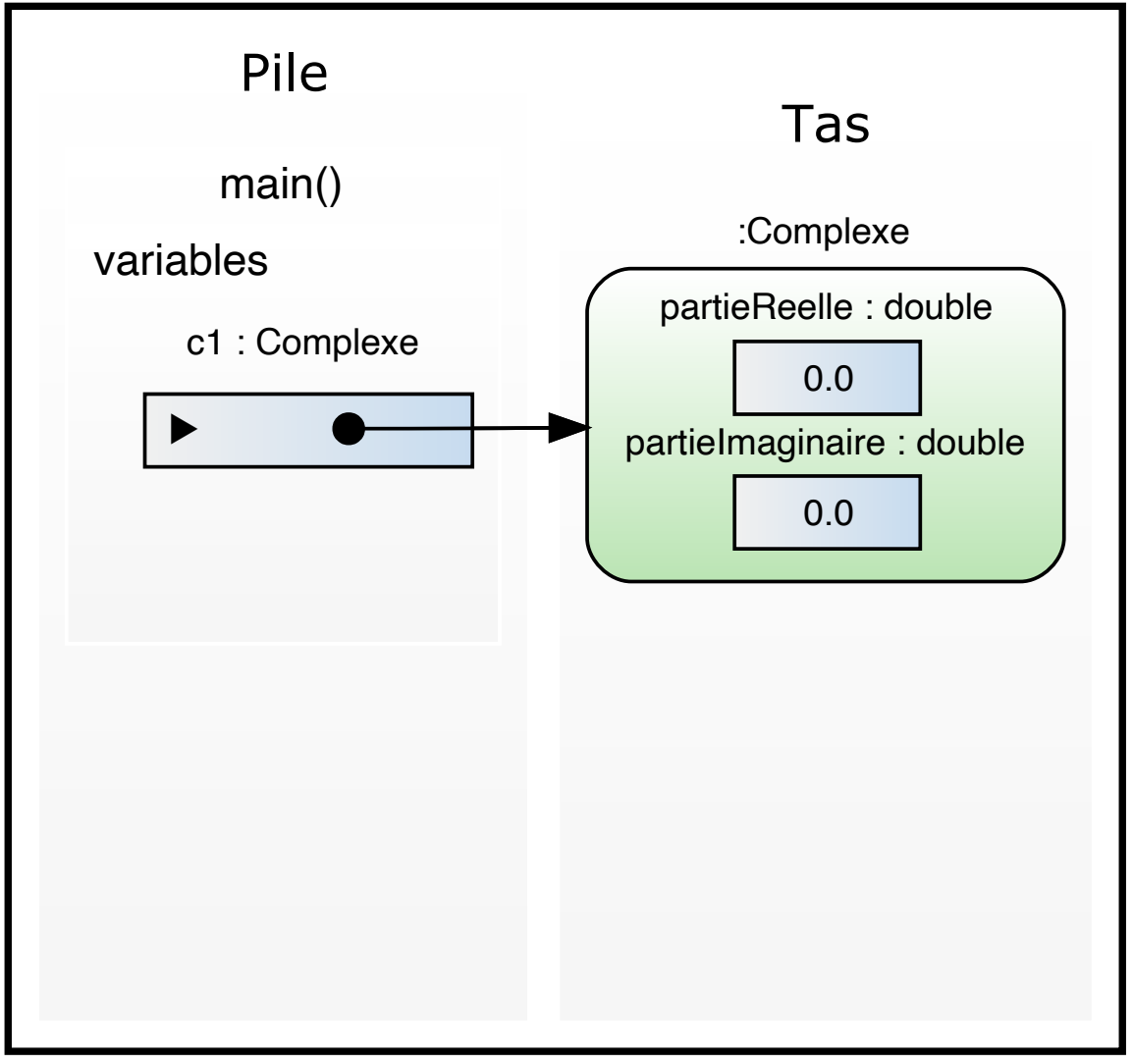


Ici, dans

- la méthode `main` de la classe `Test`,
  - 2ème étape de l'instanciation
    - Appel au constructeur Complexe() (ici pas de paramètre => constructeur par défaut
      - `c1 = new Complexe() ;`

- Dans la classe Complexe
  - Dans le constructeur, l'attribut partieImaginaire est mis à zéro

```
Complexe() {
   PartieReelle = 0.0 ;
   PartieImaginaire = 0.0 ;
}
```



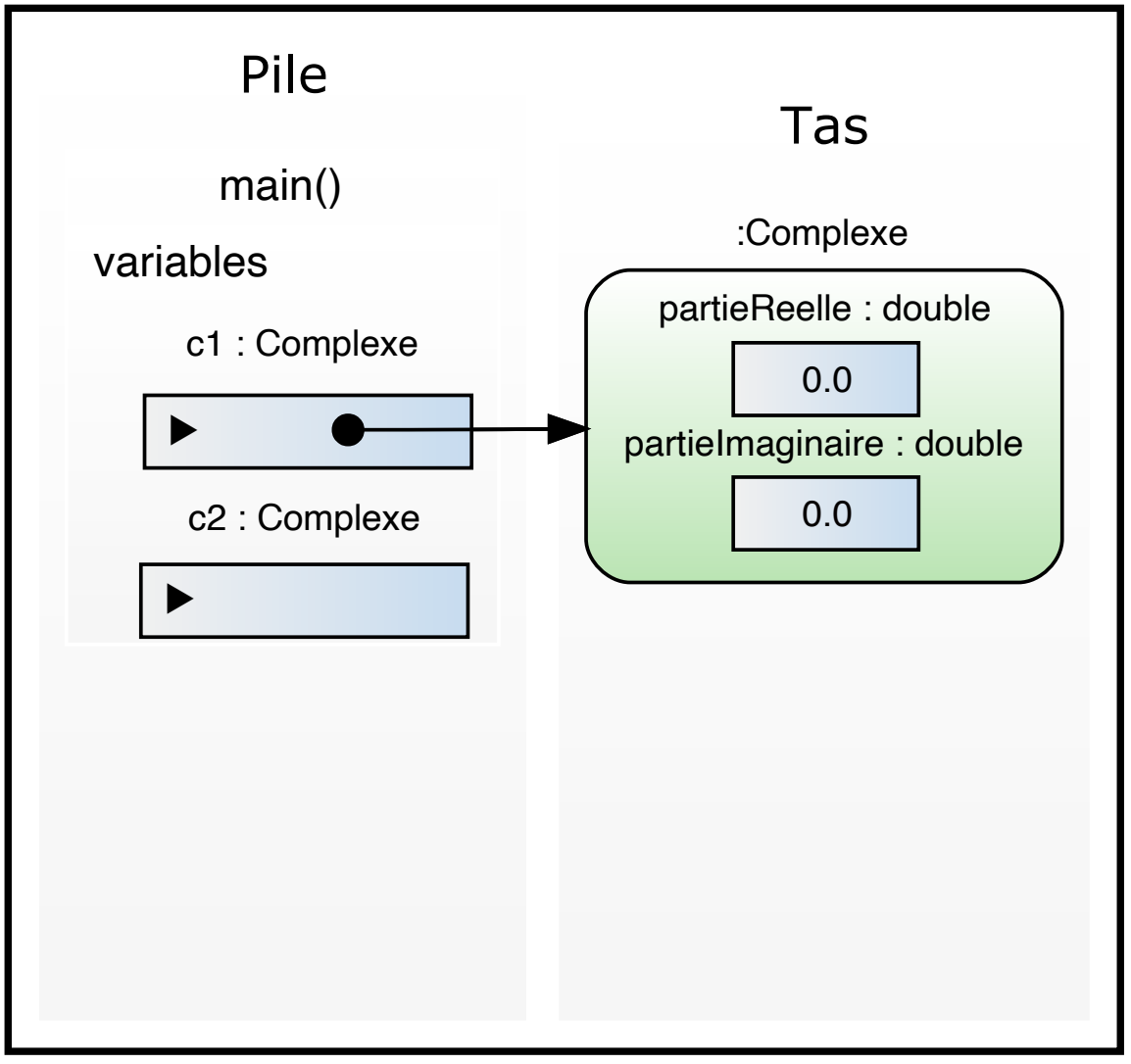
Ici, dans

- la méthode `main` de la classe `Test`,
  - 3ème étape de l'instanciation
    - Affectation de la variable c1 à l'instance.

      - `c1 = new Complexe() ;`

- Dans la classe Complexe
  - On sort du constructeur par défaut

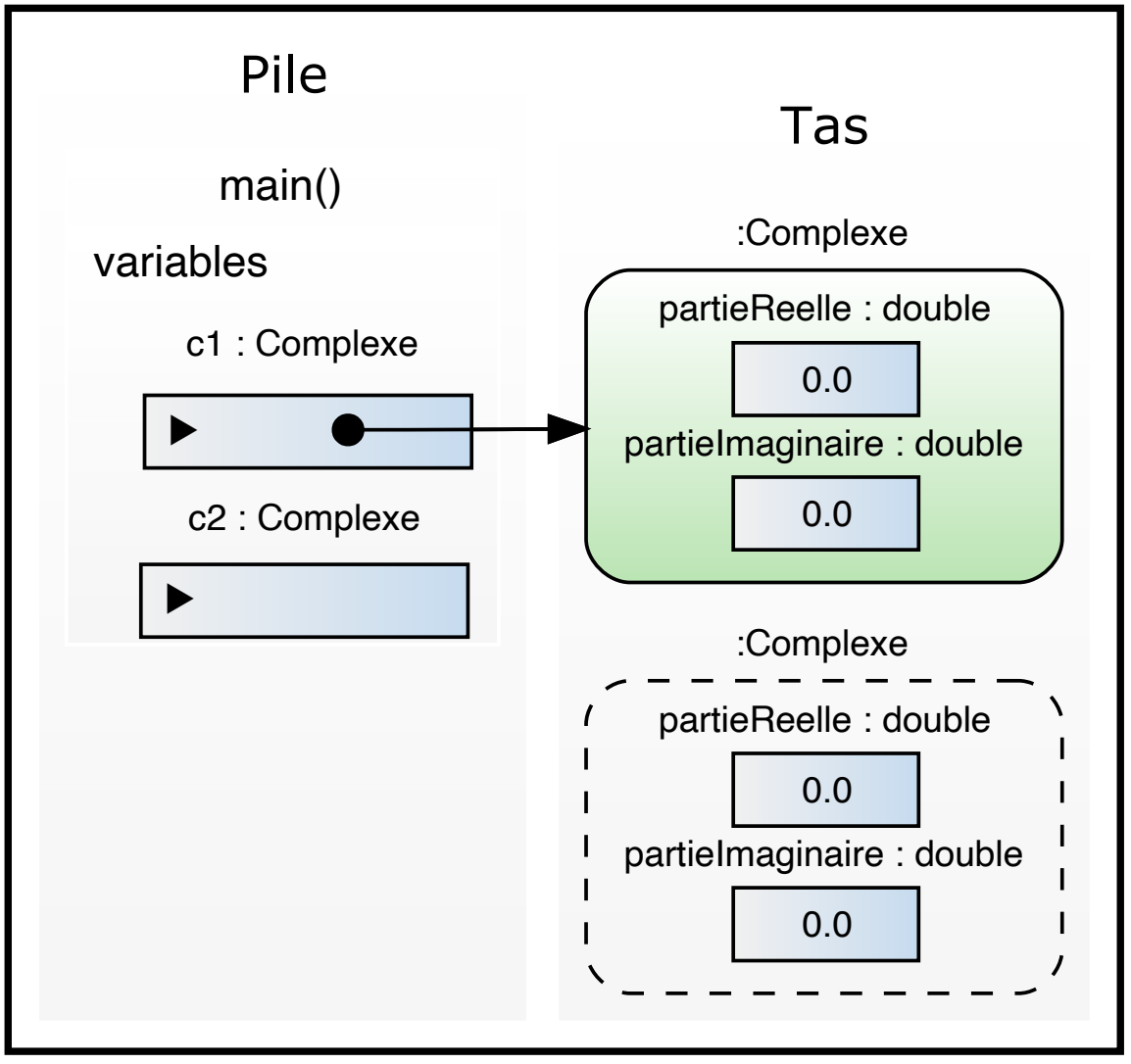
```
Complexe() {
   PartieReelle = 0.0 ;
   PartieImaginaire = 0.0 ;
}
```



Ici, dans

- la méthode `main` de la classe `Test`,
  - Déclaration d'une deuxième variable
    - appelée c2 et
    - de type Complexe

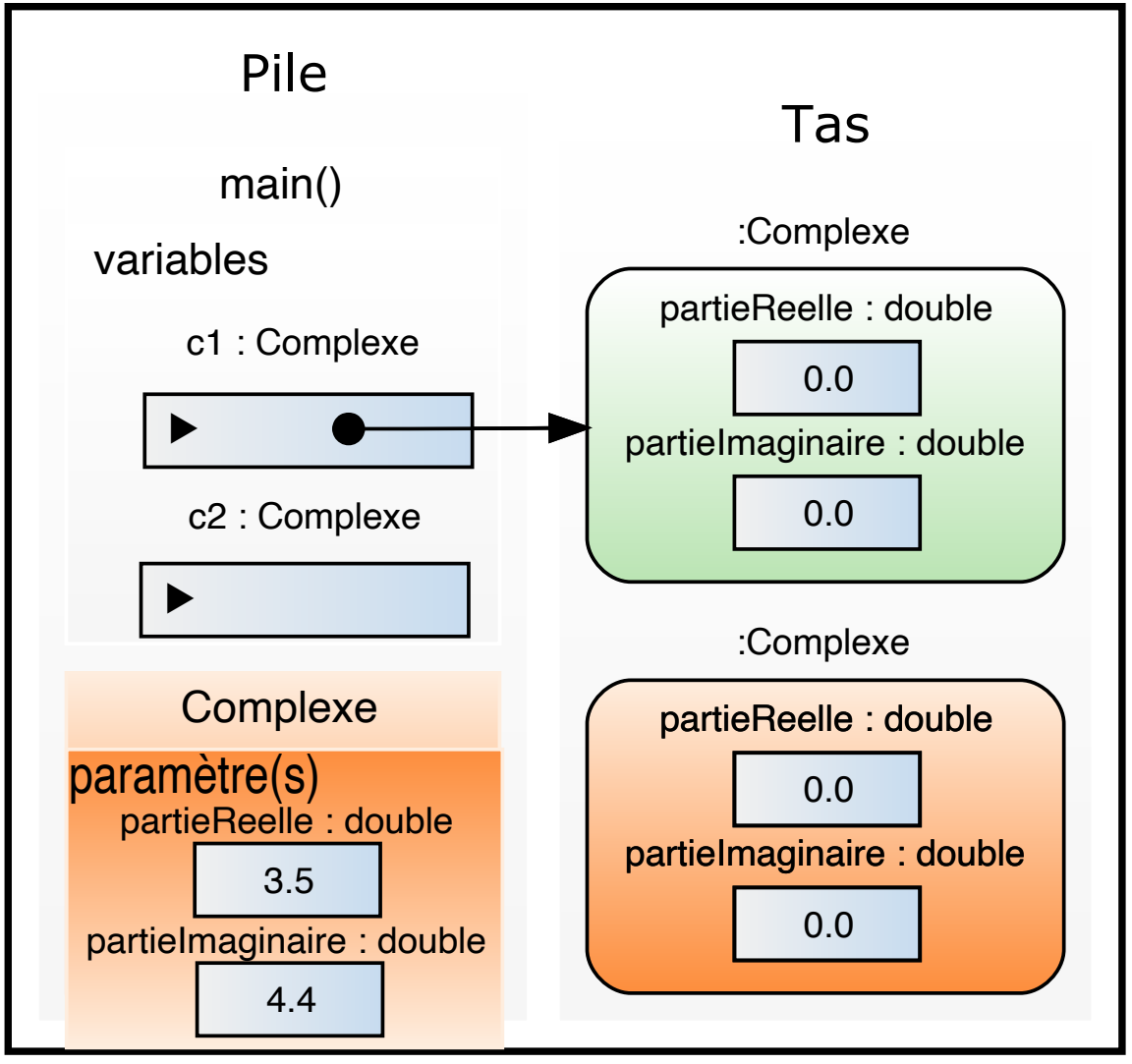
    - `Complexe c2 ;`



Ici, dans

- la méthode `main` de la classe `Test`,
  - 1ère étape de l'instanciation
    - Appel à l'opérateur `new` (création de l'instance)

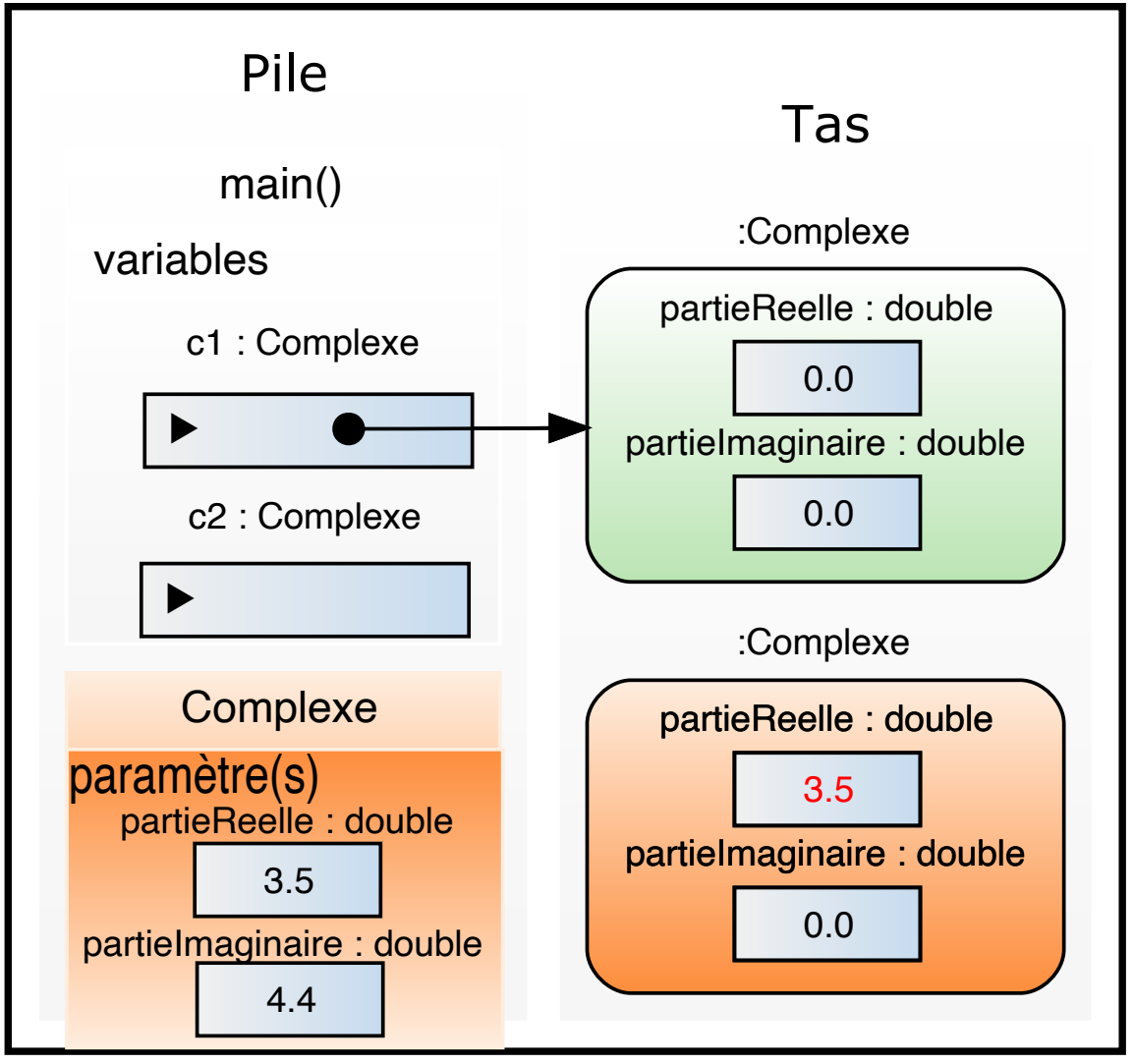
      - `c2 = new Complexe(...) ;`



Ici, dans

- la méthode `main` de la classe `Test`,
  - 2ème étape de l'instanciation
    - Appel au constructeur Complexe() avec 2 paramètres de valeur 3.5 et 4.4
      - `c2 = new Complexe(3.5, 4.4) ;`
- Dans la classe Complexe
  - Création d'un constructeur qui prend 2 paramètres de type double

```
Complexe(double partieReelle, double
partieImaginaire) {


}
```

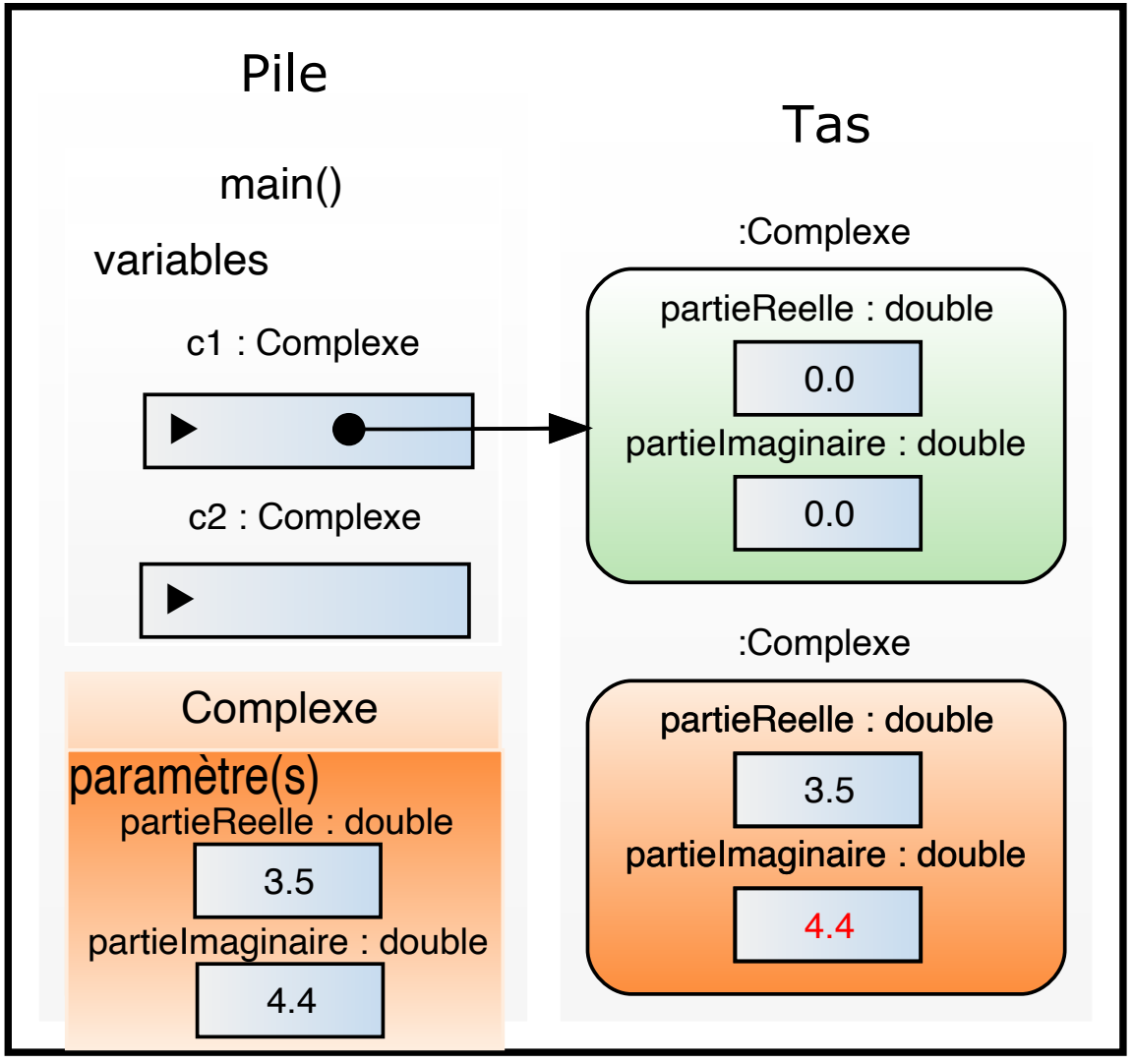


Ici, dans

- la méthode `main` de la classe `Test`,
  - 2ème étape de l'instanciation
    - Appel au constructeur Complexe() avec 2 paramètres de valeur 3.5 et 4.4
      - `c2 = new Complexe(3.5, 4.4) ;`

- Dans la classe Complexe
  - l'attribut partieReelle prend la valeur du paramètre partieReelle

```
Complexe(double partieReelle, double
partieImaginaire) {
   this.partieReelle = partieReelle ;

}
```

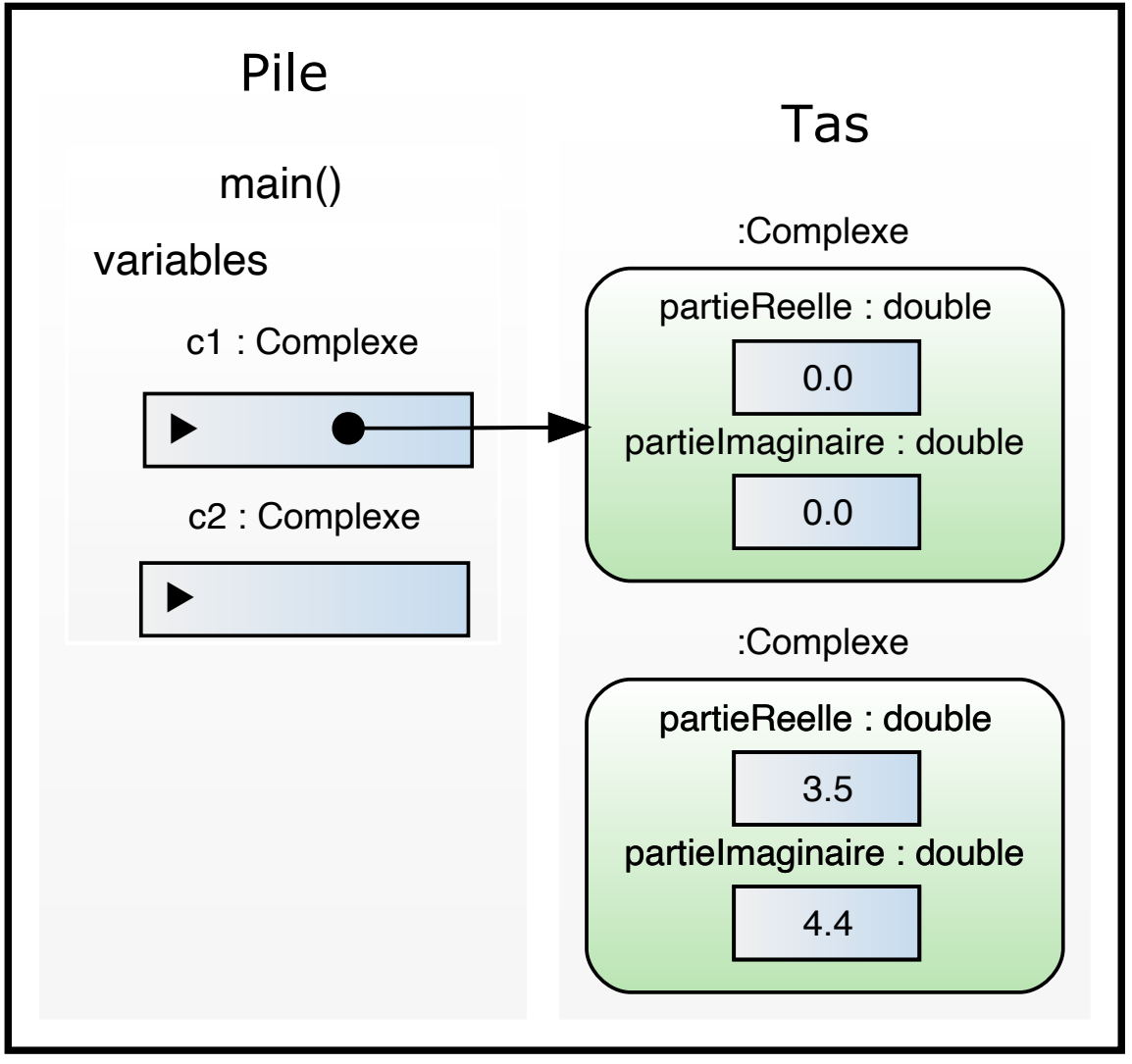


Ici, dans

- la méthode `main` de la classe `Test`,
  - 2ème étape de l'instanciation
    - Appel au constructeur Complexe() avec 2 paramètres de valeur 3.5 et 4.4
      - `c2 = new Complexe(3.5, 4.4) ;`

- Dans la classe Complexe
  - l'attribut partieImaginaire prend la valeur du paramètre partieImaginaire

```
Complexe(double partieReelle, double
partieImaginaire) {
   this.partieReelle = partieReelle ;
   this.partieImaginaire = partieImaginaire ;

}
```

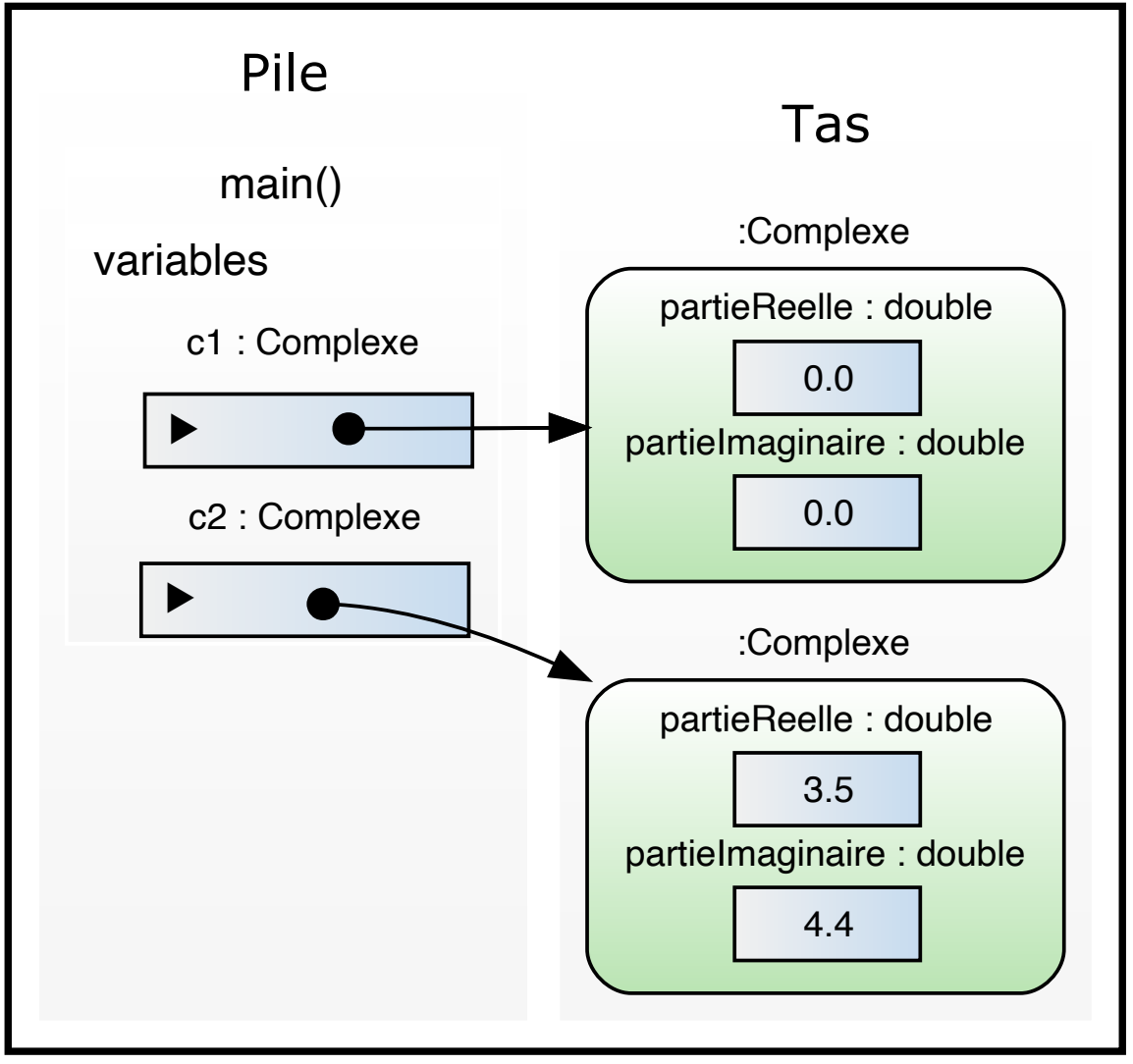


Ici, dans

- la méthode `main` de la classe `Test`,
  - 2ème étape de l'instanciation
    - Appel au constructeur Complexe() avec 2 paramètres de valeur 3.5 et 4.4
    - `c2 = new Complexe(3.5, 4.4) ;`

- Dans la classe Complexe
  - On sort du constructeur

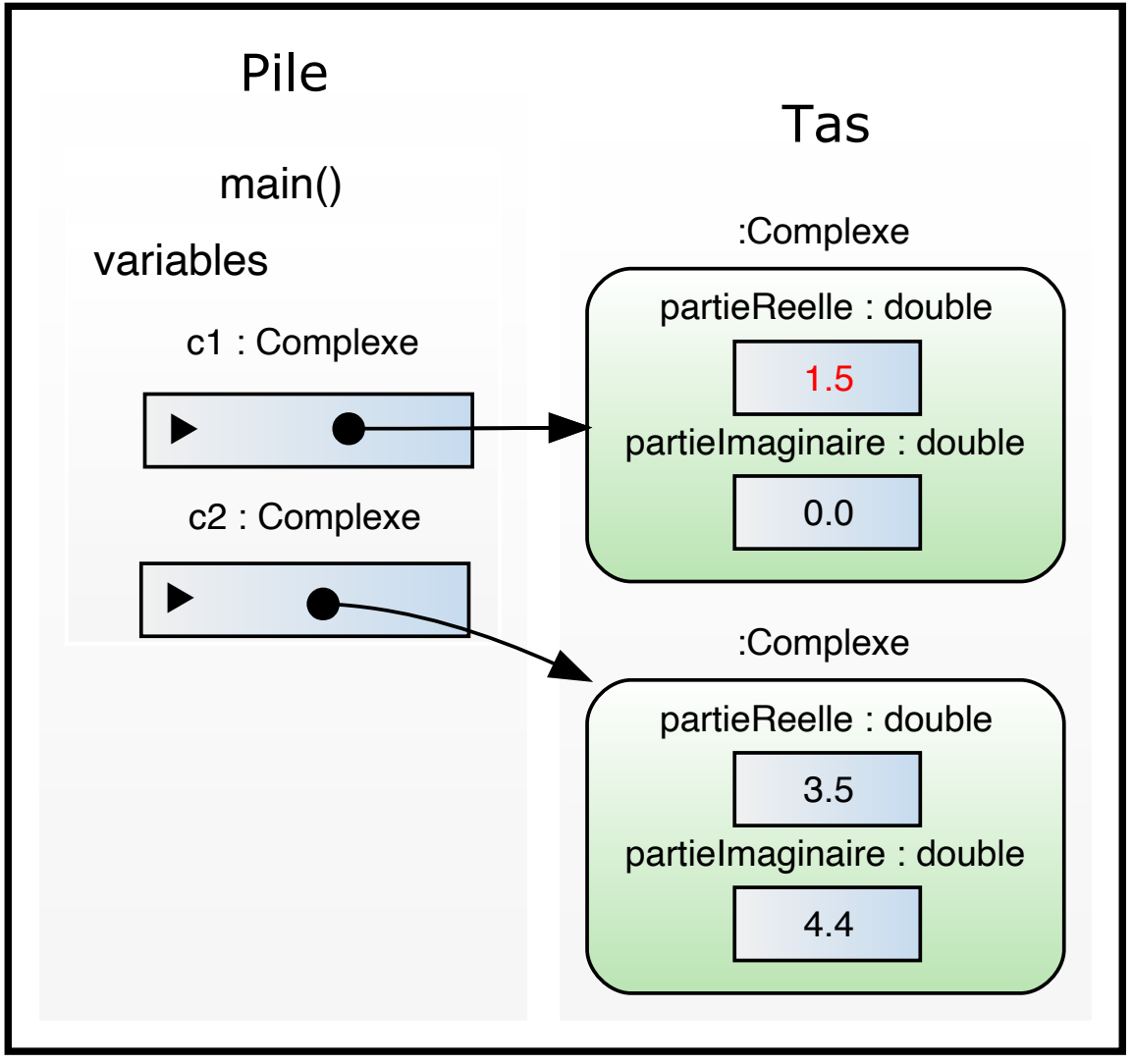
```
Complexe(double partieReelle, double
partieImaginaire) {
   this.partieReelle = partieReelle ;
   this.partieImaginaire = partieImaginaire ;

}
```



Ici, dans

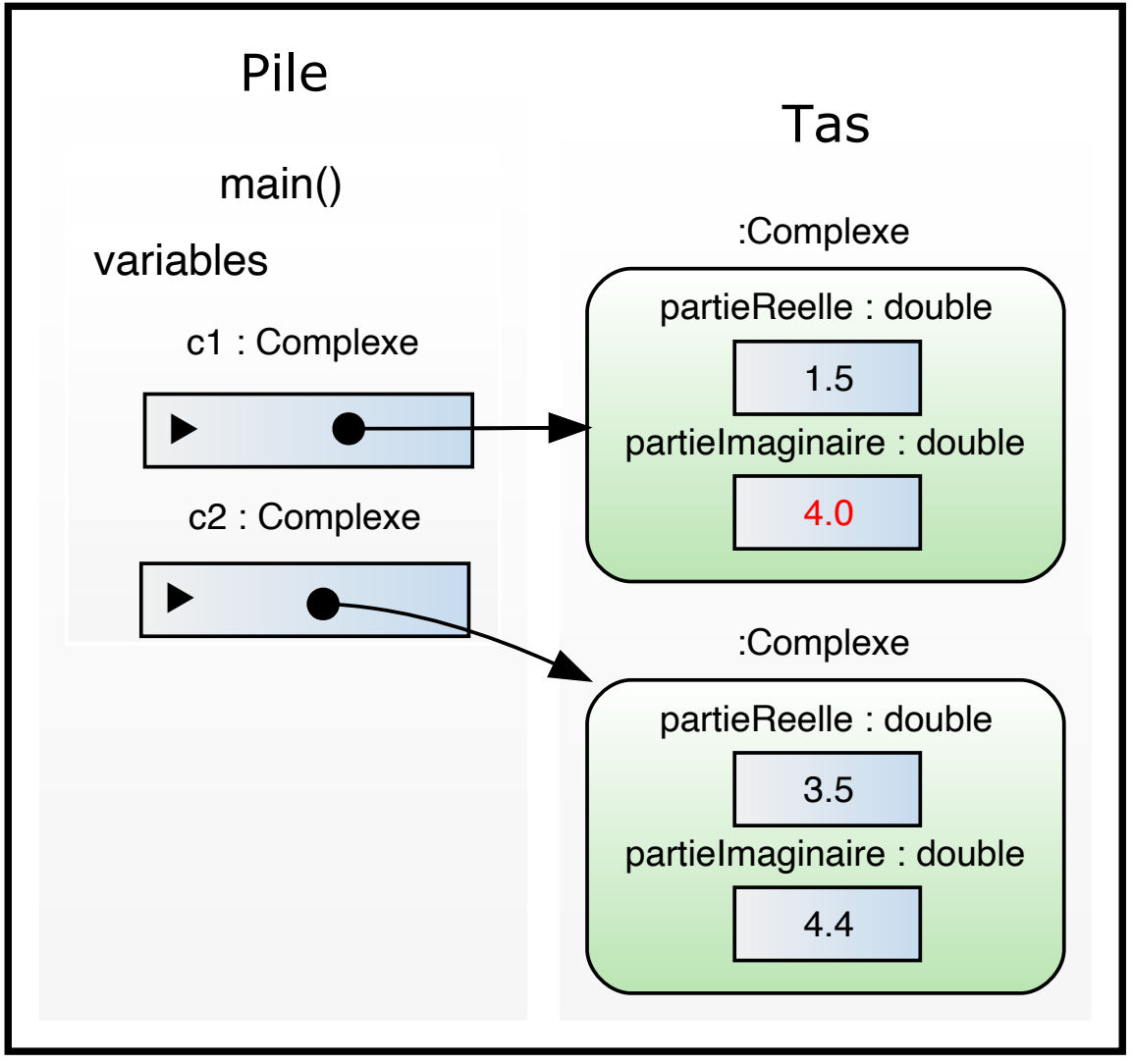
- la méthode `main` de la classe `Test`,
  - 3ème étape de l'instanciation
    - Affectation de la variable c1 à l'instance.

      - `c2 = new Complexe(3.5, 4.4) ;`



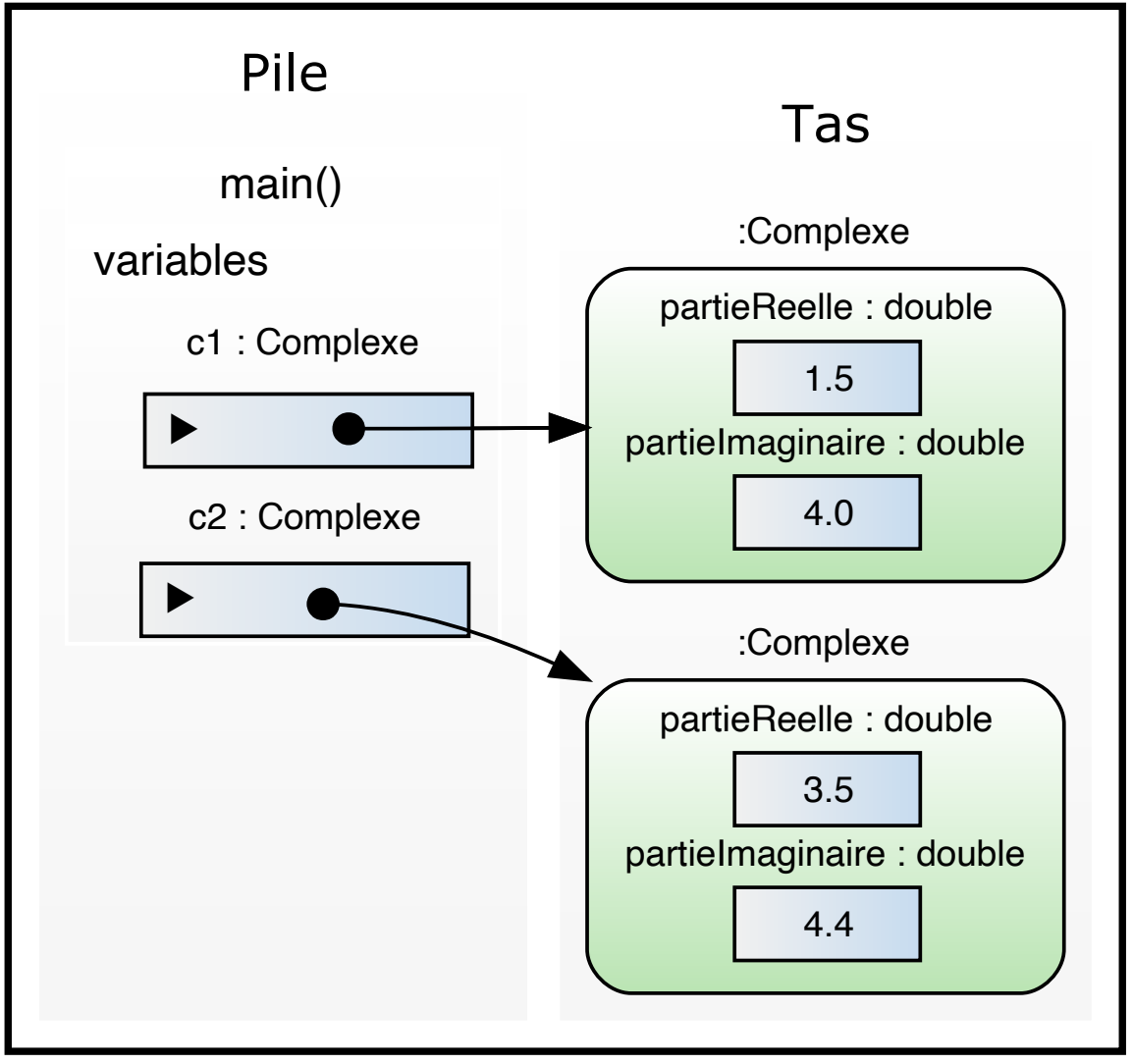
Ici, dans

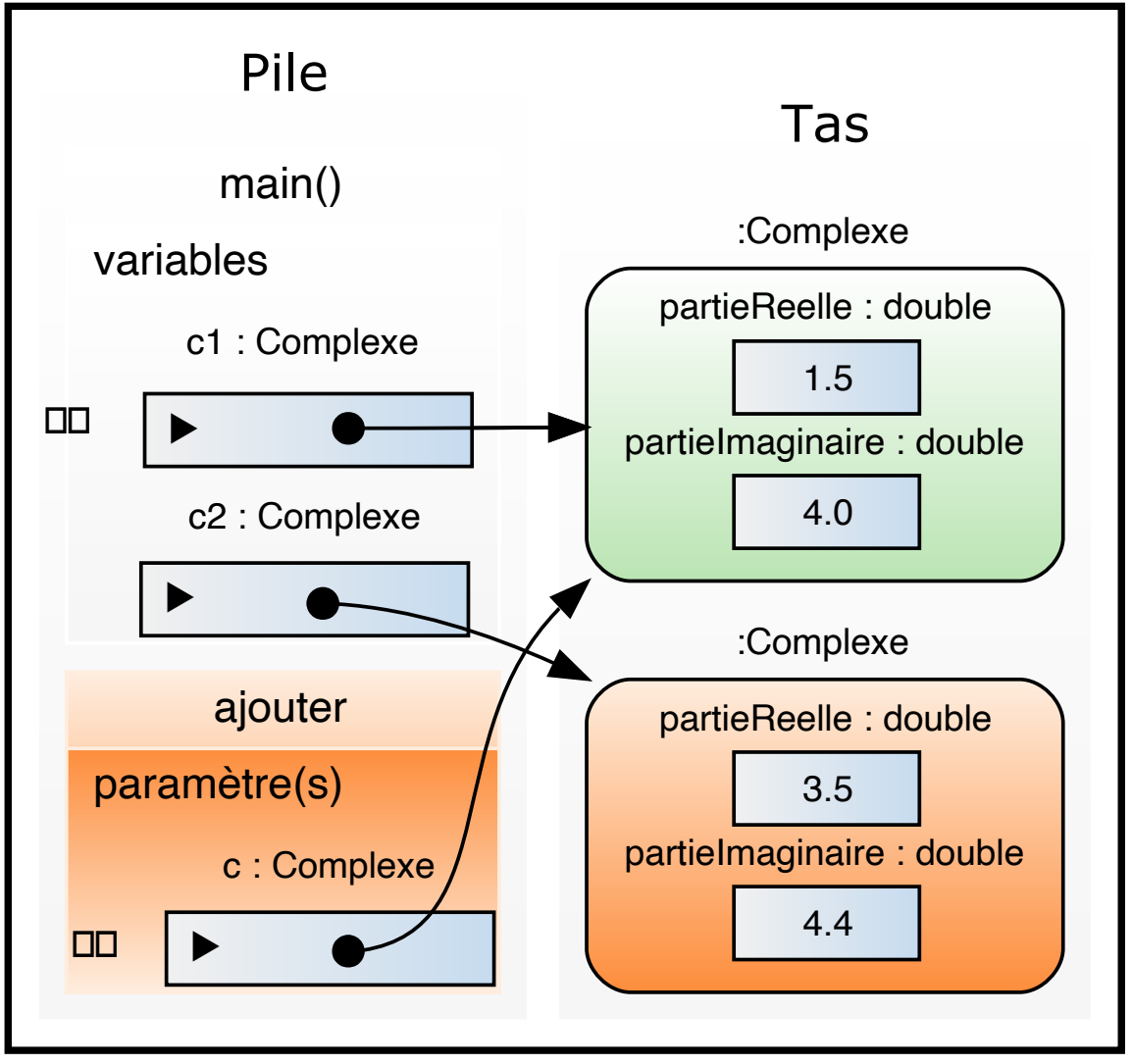
- la méthode `main` de la classe `Test`,
  - On change la valeur de l'attribut `partieReelle` de l'instance de Complexe référencée par `c1` (`c1.partieReelle`).
    -
      - `c1.partieReelle = 1.5 ;`



Ici, dans

- la méthode `main` de la classe `Test`,
  - On change la valeur de l'attribut `partieImaginaire` de l'instance de Complexe référencée par `c1` (`c1.partieImaginaire`).
  -
    - `c1.partieImaginaire = 4.0 ;`


Pile
Tas
main()
variables
c1 : Complexe
c2 : Complexe
:Complexe
partieReelle : double
1.5
partieImaginaire : double
4.0
:Complexe
partieReelle : double
3.5
partieImaginaire : double
4.4



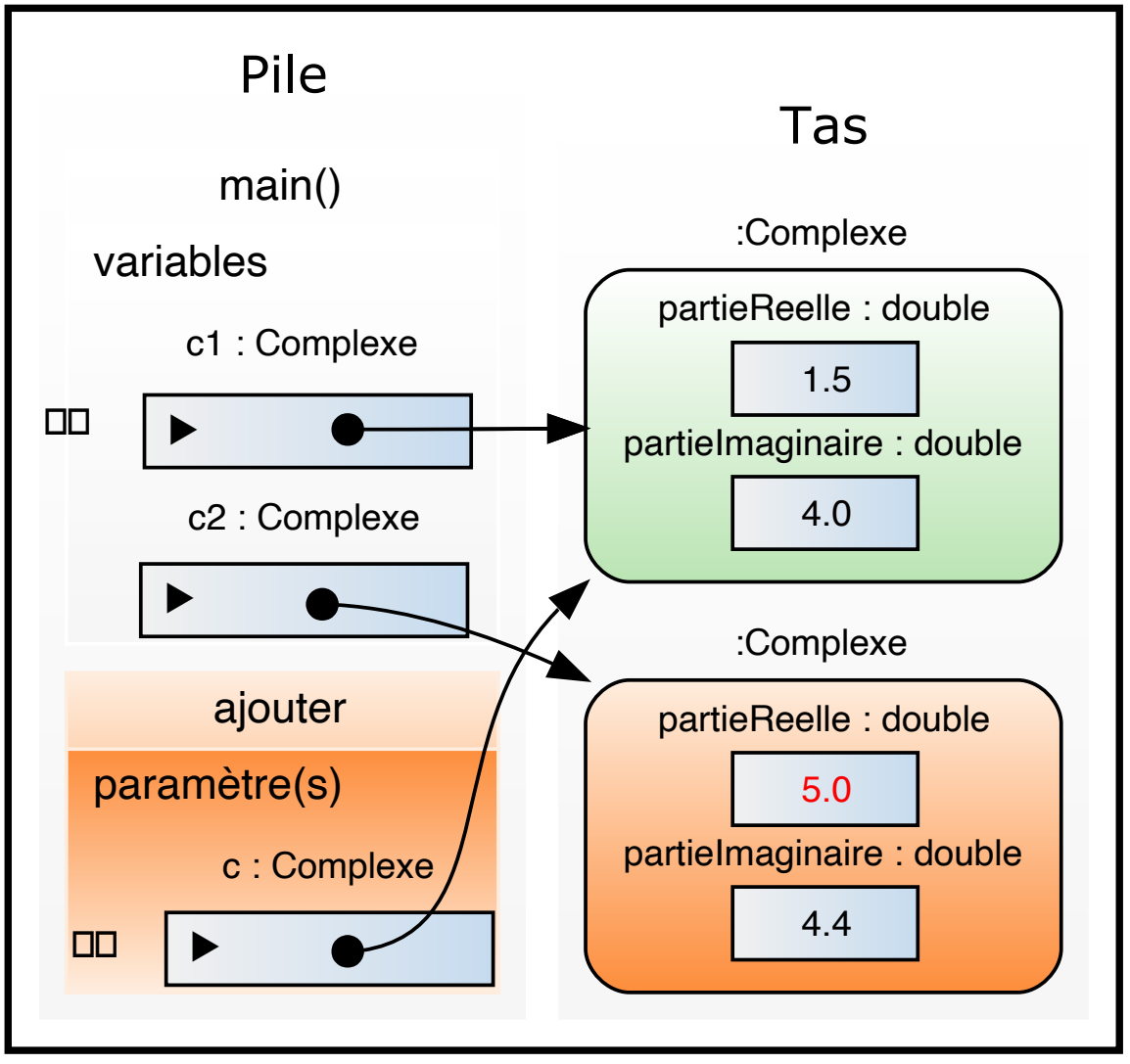
Ici, dans

- la méthode `main` de la classe `Test`,
  - On applique la méthode `ajouter` à l'instance référencée par `c2`. On passe en paramètre la référence vers l'instance verte référencée par `c1` (i.e. on passe `c1` en paramètre).

    ```
    c2.ajouter(c1);
    ```

- Dans la classe Complexe
  - Création d'une méthode `ajouter` qui prend un paramètre `c` de type référence vers un `Complexe`.

```
ajouter(Complexe c) {


}
```



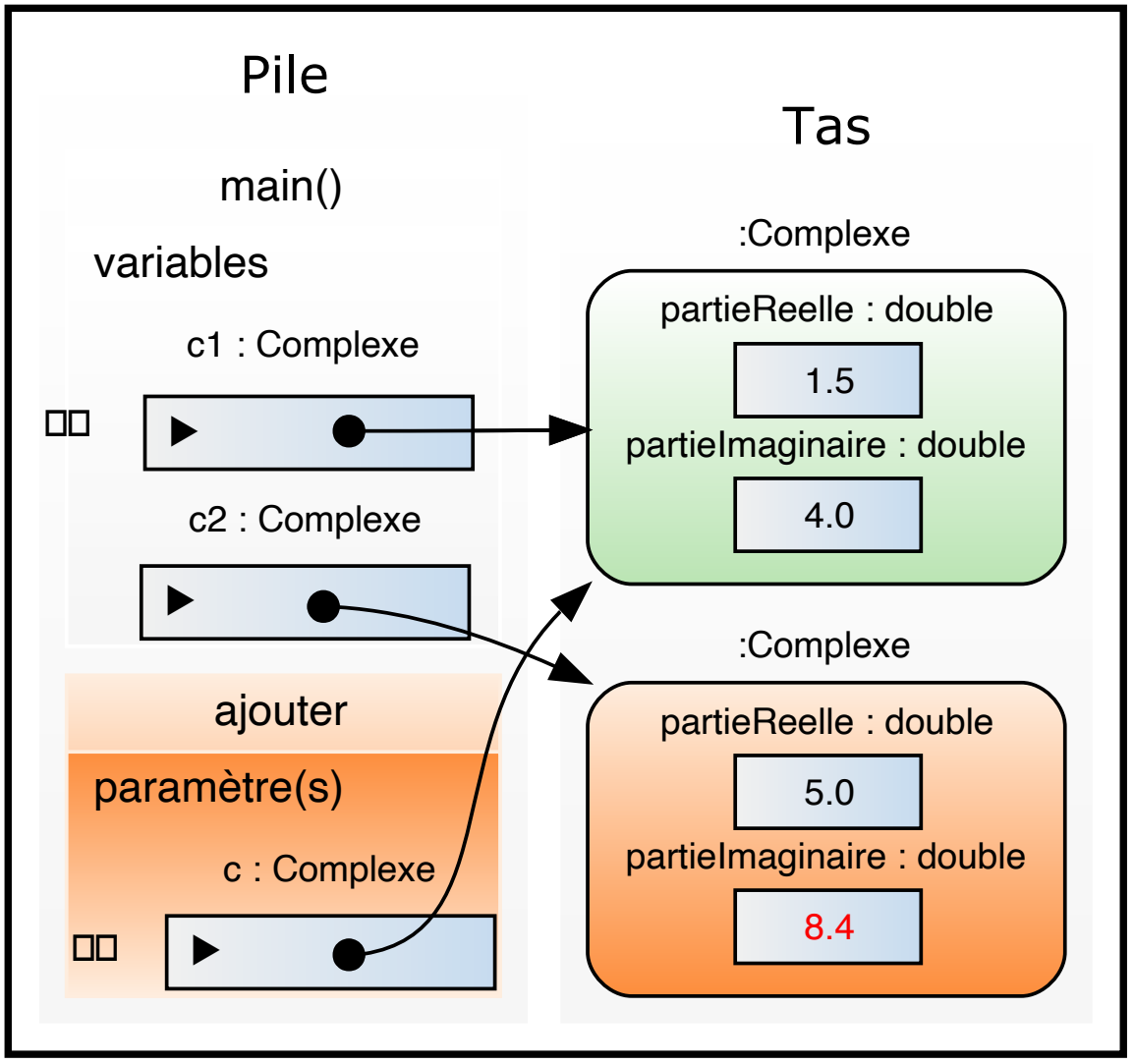
Ici, dans

- la méthode `main` de la classe `Test`,
  - On applique la méthode `ajouter` à l'instance référencée par `c2`. On passe en paramètre la référence vers l'instance verte référencée par `c1` (i.e. on passe `c1` en paramètre).

    - `c2.ajouter(c1);`

- Dans la classe Complexe
  - l'atribut `partieReelle` de l'instance courante pour la méthode (`this.partieReelle`) prend comme valeur 5.0 = 3.5 + 1.5. Or 3.5 est l'ancienne valeur de `this.partieReelle` et 1.5 est la valeur de `c.partieReelle`

```
ajouter(Complexe c) {
   this.partieReelle = this.partieReelle +
c.partieReelle ;

}
```



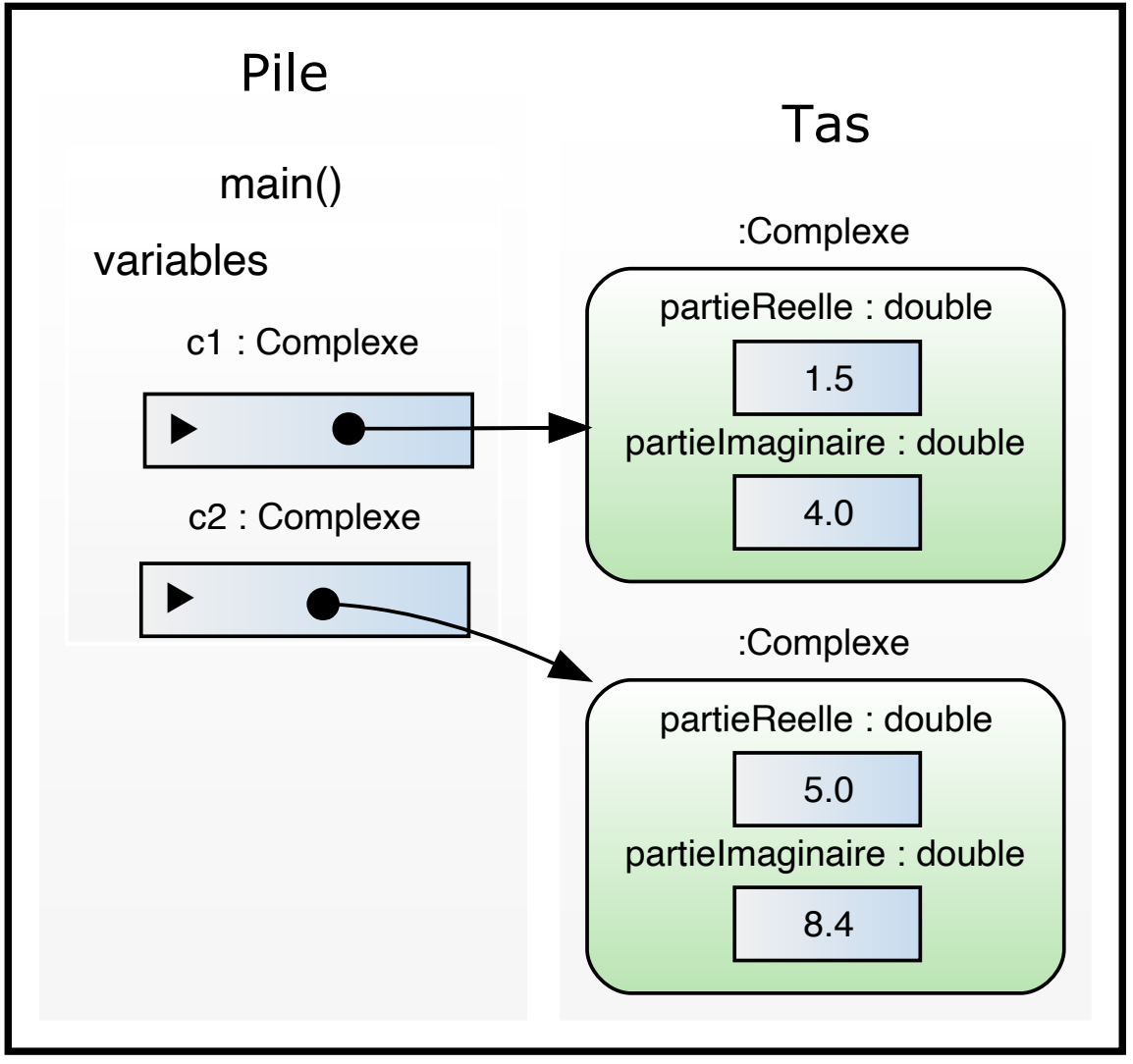
Ici, dans

- la méthode `main` de la classe `Test`,
  - On applique la méthode `ajouter` à l'instance référencée par `c2`. On passe en paramètre la référence vers l'instance verte référencée par `c1` (i.e. on passe `c1` en paramètre).

    - `c2.ajouter(c1);`

- Dans la classe Complexe
  - l'atribut `partieImaginaire` de l'instance courante pour la méthode (`this.partieImaginaire`) prend comme valeur 8.4 = 4.4 + 4.0. Or 4.4 est l'ancienne valeur de `this.partieImaginaire` et 4.0 est la valeur de `c.partieImaginaire`

```
ajouter(Complexe c) {
   this.partieReelle = this.partieReelle +
c.partieReelle ;
    this.partieImaginaire =
this.partieImaginaire + c.partieImaginaire ;

}
```


Pile
Tas
main()
variables
c1 : Complexe
c2 : Complexe
:Complexe
partieReelle : double
1.5
partieImaginaire : double
4.0
:Complexe
partieReelle : double
5.0
partieImaginaire : double
8.4