

Rappels et compléments

V. Héritage

1. Mécanisme d'héritage
 - i. Définition générale
 - ii. Relation entre classes
 - iii. Formes d'héritage en Java
2. Sous-typage Java
 - i. Sous-type Java direct
 - ii. Sous-type Java
 - iii. Sous-types tableaux
3. Expressions
 - i. Définitions
 - ii. Valeur d'une expression
 - iii. Types d'une expression
 - iv. Transtypage
 - a. Définition
 - b. Extensibilité potentielle
4. Masquage d'attribut
5. Chaînage des constructeurs
6. Redéfinition de méthode
 - i. Redéf° et liaison dynamique
 - ii. Modification d'entête
 - iii. @Override
 - iv. Redéfinition et surcharge
7. Invocation de méthode
 - i. Principe
 - a. Méthode virtuelle
 - b. Méthode de classe
 - c. Méthode privée
 - d. Méthode avec super
 - ii. Invocation et surcharge
 - a. Résolution d'appel
 - b. Surcharge et héritage
 - c. Piège de la surcharge
 - d. Ambiguïté
8. Accessibilité
 - i. Paquetage
 - ii. Accessibilité des types
 - iii. Accessibilité des caract.

Le corps d'un constructeur **commence toujours** par un appel à un autre constructeur (sauf pour la classe `Object`).

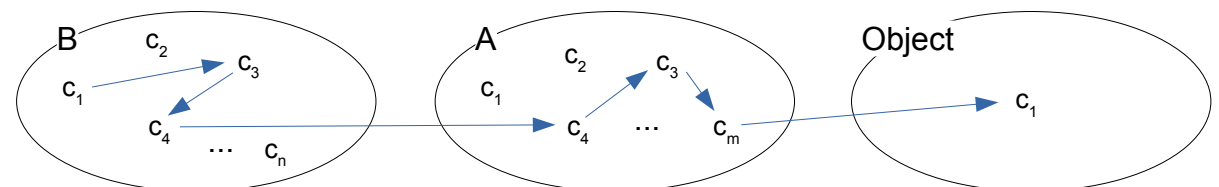
ni `this(...)` ni `super(...)`
le compilateur ajoute `super();`
(invocation de `Object()`)

```
class A {
    private int i;
    A(int k) {
        i = k;
    }
}
```

```
class B extends A {
    B() {
        this(1);
    }
    B(int k) {
        super(k);
    }
}
```

compilation

```
class A {
    A(int);
    Code:
        0: aload_0
        1: invokespecial #10 // Method java/lang/Object."<init>":()V
        4: aload_0
        5: iload_1
        6: putfield     #13 // Field i:I
        9: return
    ...
}
```



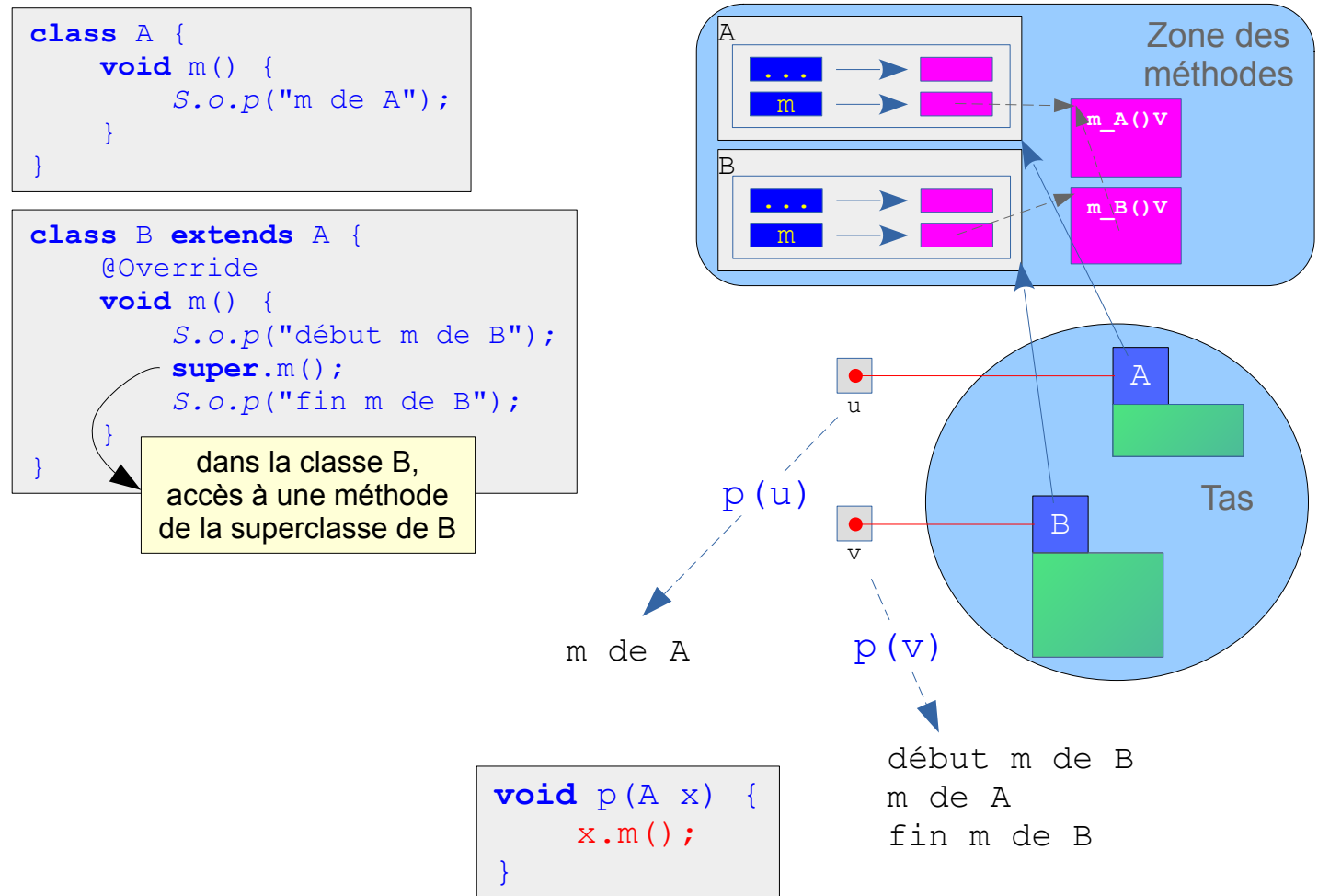
Rappels et compléments

V. Héritage

1. Mécanisme d'héritage
 - i. Définition générale
 - ii. Relation entre classes
 - iii. Formes d'héritage en Java
2. Sous-typage Java
 - i. Sous-type Java direct
 - ii. Sous-type Java
 - iii. Sous-types tableaux
3. Expressions
 - i. Définitions
 - ii. Valeur d'une expression
 - iii. Types d'une expression
 - iv. Transtypage
 - a. Définition
 - b. Extensibilité potentielle
4. Masquage d'attribut
5. Chaînage des constructeurs
6. Redéfinition de méthode
 - i. Redéf^o et liaison dynamique
 - ii. Modification d'entête
 - iii. @Override
 - iv. Redéfinition et surcharge
7. Invocation de méthode
 - i. Principe
 - a. Méthode virtuelle
 - b. Méthode de classe
 - c. Méthode privée
 - d. Méthode avec super
 - ii. Invocation et surcharge
 - a. Résolution d'appel
 - b. Surcharge et héritage
 - c. Piège de la surcharge
 - d. Ambiguïté
8. Accessibilité
 - i. Paquetage
 - ii. Accessibilité des types
 - iii. Accessibilité des caract.

Redéfinition de méthode : changement du corps d'une méthode (concrète) héritée.

Liaison dynamique : mécanisme qui retarde à l'exécution la sélection du code d'une méthode.
Cette sélection se fait en fonction du type de l'objet cible.

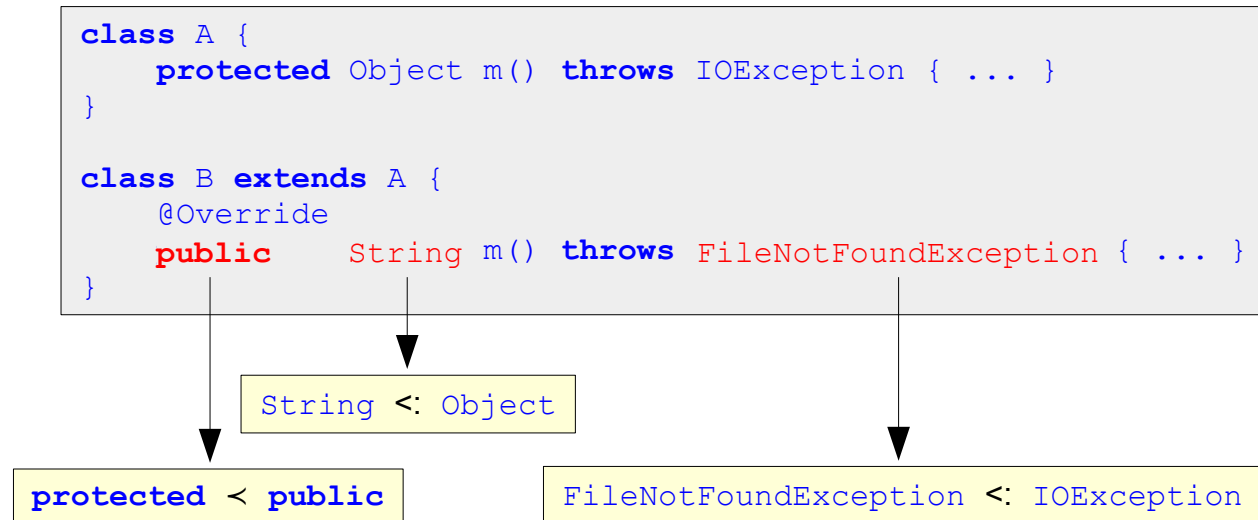


Rappels et compléments

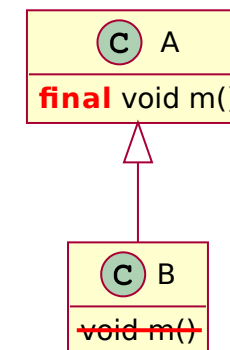
V. Héritage

1. Mécanisme d'héritage
 - i. Définition générale
 - ii. Relation entre classes
 - iii. Formes d'héritage en Java
2. Sous-typage Java
 - i. Sous-type Java direct
 - ii. Sous-type Java
 - iii. Sous-types tableaux
3. Expressions
 - i. Définitions
 - ii. Valeur d'une expression
 - iii. Types d'une expression
 - iv. Transtypage
 - a. Définition
 - b. Extensibilité potentielle
4. Masquage d'attribut
5. Chaînage des constructeurs
6. Redéfinition de méthode
 - i. Redéf° et liaison dynamique
 - ii. Modification d'entête
 - iii. `@Override`
 - iv. Redéfinition et surcharge
7. Invocation de méthode
 - i. Principe
 - a. Méthode virtuelle
 - b. Méthode de classe
 - c. Méthode privée
 - d. Méthode avec super
 - ii. Invocation et surcharge
 - a. Résolution d'appel
 - b. Surcharge et héritage
 - c. Piège de la surcharge
 - d. Ambiguïté
8. Accessibilité
 - i. Paquetage
 - ii. Accessibilité des types
 - iii. Accessibilité des caract.

Modifications d'entête acceptées lors de la redéfinition



Il est possible d'interdire la redéfinition



Rappels et compléments

V. Héritage

1. Mécanisme d'héritage
 - i. Définition générale
 - ii. Relation entre classes
 - iii. Formes d'héritage en Java
2. Sous-typage Java
 - i. Sous-type Java direct
 - ii. Sous-type Java
 - iii. Sous-types tableaux
3. Expressions
 - i. Définitions
 - ii. Valeur d'une expression
 - iii. Types d'une expression
 - iv. Transtypage
 - a. Définition
 - b. Extensibilité potentielle
4. Masquage d'attribut
5. Chaînage des constructeurs
6. Redéfinition de méthode
 - i. Redéf° et liaison dynamique
 - ii. Modification d'entête
 - iii. `@Override`
 - iv. Redéfinition et surcharge
7. Invocation de méthode
 - i. Principe
 - a. Méthode virtuelle
 - b. Méthode de classe
 - c. Méthode privée
 - d. Méthode avec super
 - ii. Invocation et surcharge
 - a. Résolution d'appel
 - b. Surcharge et héritage
 - c. Piège de la surcharge
 - d. Ambiguïté
8. Accessibilité
 - i. Paquetage
 - ii. Accessibilité des types
 - iii. Accessibilité des caract.

je veux redéfinir dans B la méthode héritée de A



```
class A {
    void nomSuperDurAEcrirePourCeuxQuiNeVoientPasClair() { ... }
}

class B extends A {
    void nomSuperDurAEcrirePourCeuxQuiNeVoientPasClair() { ... }
}
```

compilation : OK !



```
class A {
    void nomSuperDurAEcrirePourCeuxQuiNeVoientPasClair() { ... }
}

class B extends A {
    @Override
    void nomSuperDurAEcrirePourCeuxQuiNeVoientPasClair() { ... }
}
```

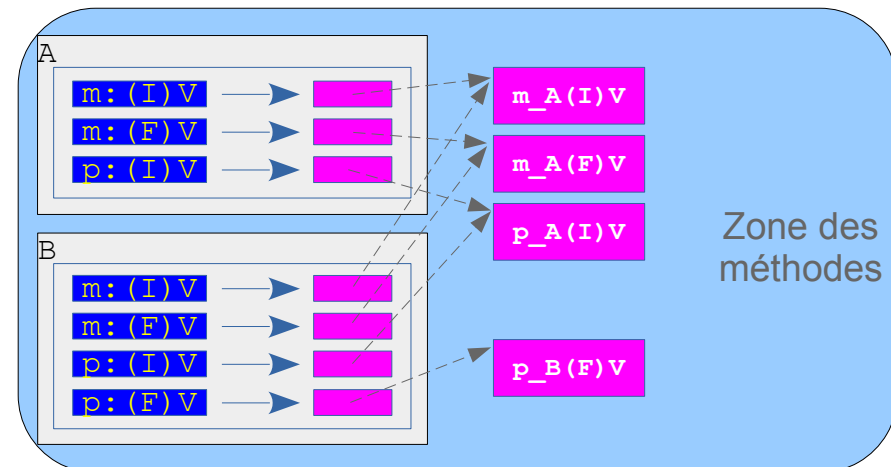
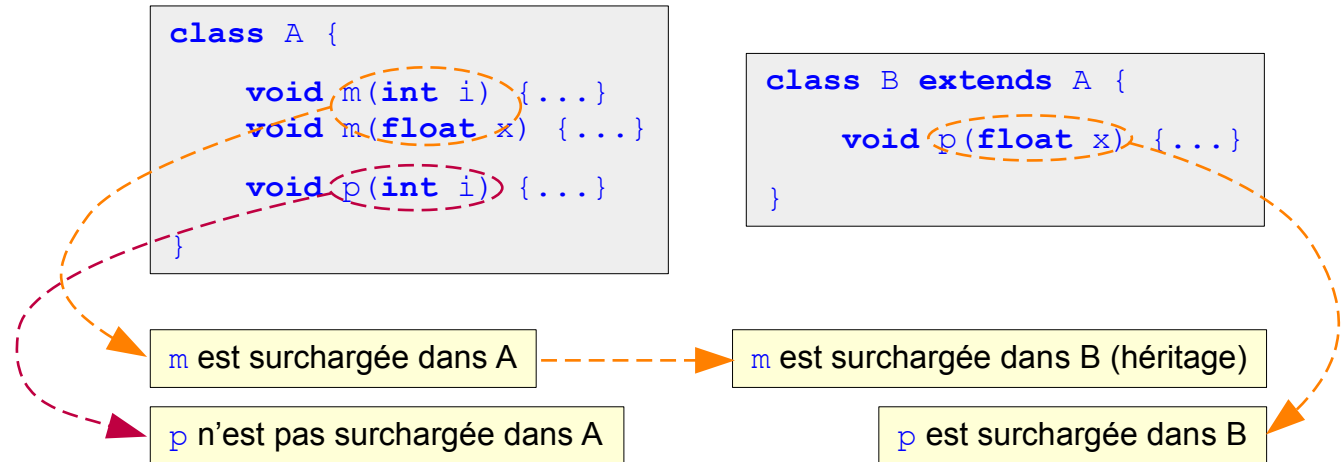
compilation : erreur !

Rappels et compléments

V. Héritage

1. Mécanisme d'héritage
 - i. Définition générale
 - ii. Relation entre classes
 - iii. Formes d'héritage en Java
2. Sous-typage Java
 - i. Sous-type Java direct
 - ii. Sous-type Java
 - iii. Sous-types tableaux
3. Expressions
 - i. Définitions
 - ii. Valeur d'une expression
 - iii. Types d'une expression
 - iv. Transtypage
 - a. Définition
 - b. Extensibilité potentielle
4. Masquage d'attribut
5. Chaînage des constructeurs
6. Redéfinition de méthode
 - i. Redéf. et liaison dynamique
 - ii. Modification d'entête
 - iii. @Override
 - iv. Redéfinition et surcharge
7. Invocation de méthode
 - i. Principe
 - a. Méthode virtuelle
 - b. Méthode de classe
 - c. Méthode privée
 - d. Méthode avec super
 - ii. Invocation et surcharge
 - a. Résolution d'appel
 - b. Surcharge et héritage
 - c. Piège de la surcharge
 - d. Ambiguïté
8. Accessibilité
 - i. Paquetage
 - ii. Accessibilité des types
 - iii. Accessibilité des caract.

Surcharge de méthode (appelée aussi polymorphisme ad'hoc)
Déclaration de plusieurs méthodes ayant même nom mais des signatures différentes.



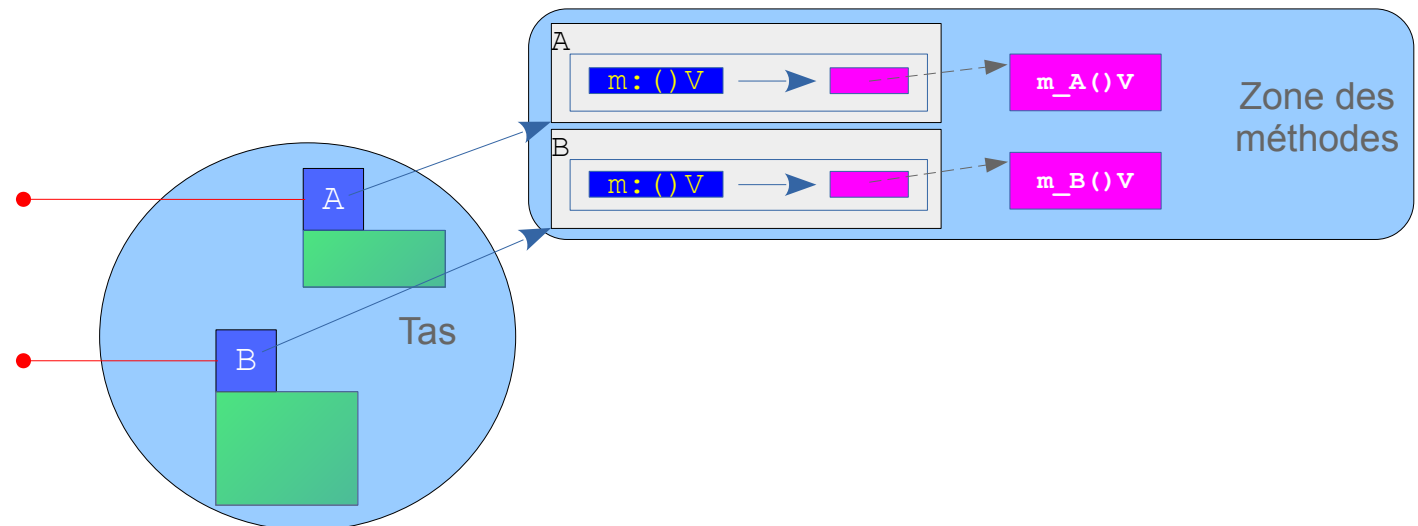
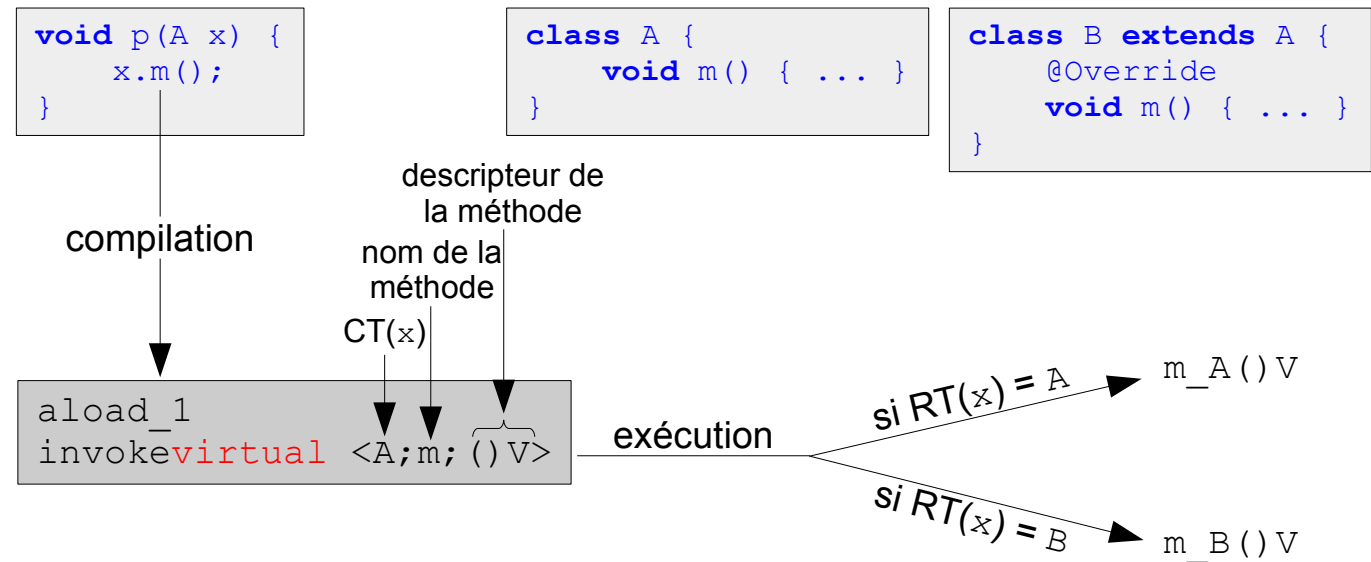
Rappels et compléments

V. Héritage

1. Mécanisme d'héritage
 - i. Définition générale
 - ii. Relation entre classes
 - iii. Formes d'héritage en Java
2. Sous-typage Java
 - i. Sous-type Java direct
 - ii. Sous-type Java
 - iii. Sous-types tableaux
3. Expressions
 - i. Définitions
 - ii. Valeur d'une expression
 - iii. Types d'une expression
 - iv. Transtypage
 - a. Définition
 - b. Extensibilité potentielle
4. Masquage d'attribut
5. Chaînage des constructeurs
6. Redéfinition de méthode
 - i. Redéf° et liaison dynamique
 - ii. Modification d'entête
 - iii. @Override
 - iv. Redéfinition et surcharge
7. Invocation de méthode
 - i. Principe
 - a. Méthode virtuelle
 - b. Méthode de classe
 - c. Méthode privée
 - d. Méthode avec super
 - ii. Invocation et surcharge
 - a. Résolution d'appel
 - b. Surcharge et héritage
 - c. Piège de la surcharge
 - d. Ambiguïté
8. Accessibilité
 - i. Paquetage
 - ii. Accessibilité des types
 - iii. Accessibilité des caract.

Invocation de méthode : 4 cas possibles

- méthode d'instance, virtuelle (cas classique, liaison dynamique)
- méthode de classe (**static**, liaison statique)
- méthode d'instance, privée (**private**, liaison statique)
- méthode d'instance, appelée avec **super** (liaison statique)



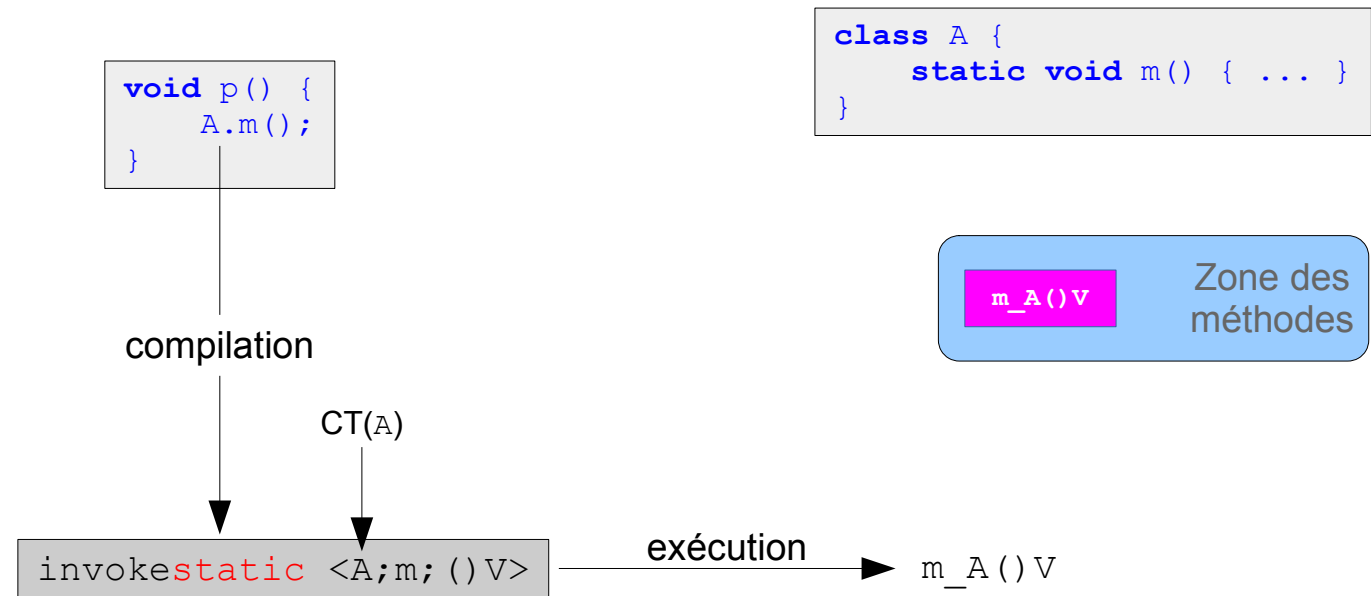
Rappels et compléments

V. Héritage

1. Mécanisme d'héritage
 - i. Définition générale
 - ii. Relation entre classes
 - iii. Formes d'héritage en Java
2. Sous-typage Java
 - i. Sous-type Java direct
 - ii. Sous-type Java
 - iii. Sous-types tableaux
3. Expressions
 - i. Définitions
 - ii. Valeur d'une expression
 - iii. Types d'une expression
 - iv. Transtypage
 - a. Définition
 - b. Extensibilité potentielle
4. Masquage d'attribut
5. Chaînage des constructeurs
6. Redéfinition de méthode
 - i. Redéf° et liaison dynamique
 - ii. Modification d'entête
 - iii. @Override
 - iv. Redéfinition et surcharge
7. Invocation de méthode
 - i. Principe
 - a. Méthode virtuelle
 - b. Méthode de classe
 - c. Méthode privée
 - d. Méthode avec super
 - ii. Invocation et surcharge
 - a. Résolution d'appel
 - b. Surcharge et héritage
 - c. Piège de la surcharge
 - d. Ambiguïté
8. Accessibilité
 - i. Paquetage
 - ii. Accessibilité des types
 - iii. Accessibilité des caract.

Invocation de méthode : 4 cas possibles

- méthode d'instance, virtuelle (cas classique, liaison dynamique)
- méthode de classe (**static**, liaison statique)
- méthode d'instance, privée (**private**, liaison statique)
- méthode d'instance, appelée avec **super** (liaison statique)



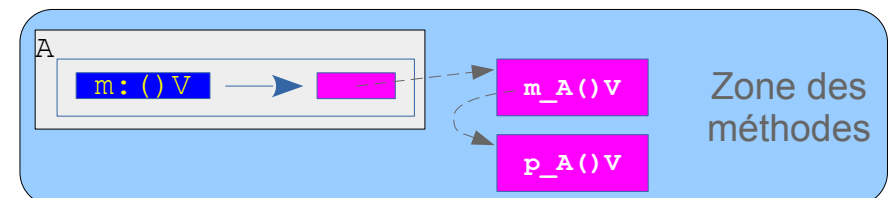
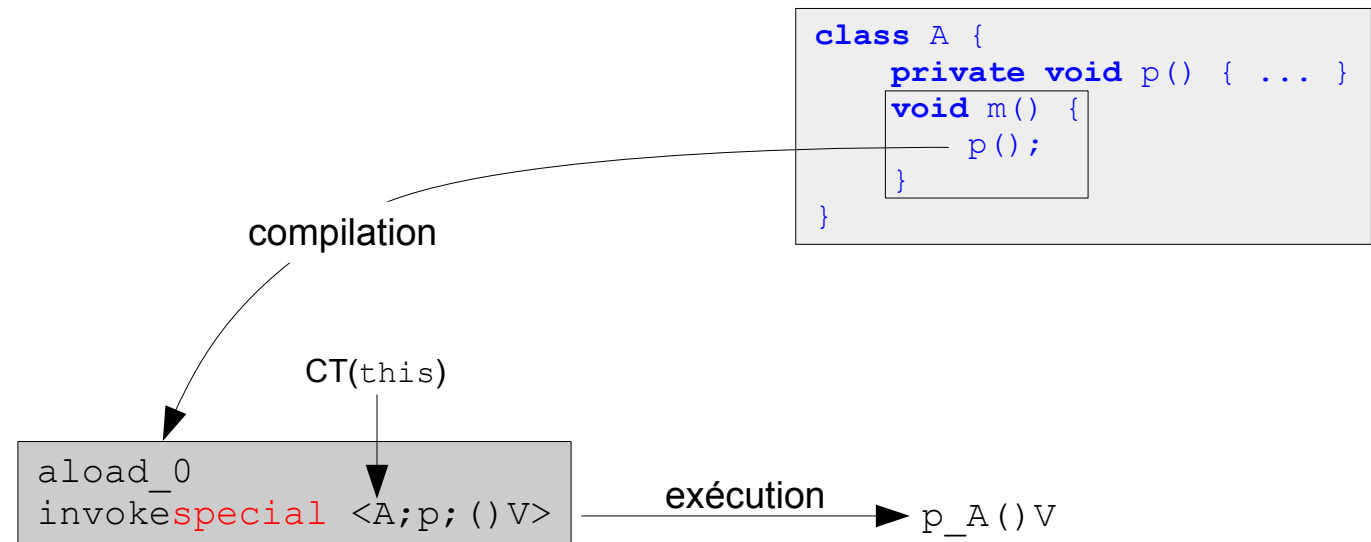
Rappels et compléments

V. Héritage

1. Mécanisme d'héritage
 - i. Définition générale
 - ii. Relation entre classes
 - iii. Formes d'héritage en Java
2. Sous-typage Java
 - i. Sous-type Java direct
 - ii. Sous-type Java
 - iii. Sous-types tableaux
3. Expressions
 - i. Définitions
 - ii. Valeur d'une expression
 - iii. Types d'une expression
 - iv. Transtypage
 - a. Définition
 - b. Extensibilité potentielle
4. Masquage d'attribut
5. Chaînage des constructeurs
6. Redéfinition de méthode
 - i. Redéf° et liaison dynamique
 - ii. Modification d'entête
 - iii. @Override
 - iv. Redéfinition et surcharge
7. Invocation de méthode
 - i. Principe
 - a. Méthode virtuelle
 - b. Méthode de classe
 - c. **Méthode privée**
 - d. Méthode avec super
 - ii. Invocation et surcharge
 - a. Résolution d'appel
 - b. Surcharge et héritage
 - c. Piège de la surcharge
 - d. Ambiguïté
8. Accessibilité
 - i. Paquetage
 - ii. Accessibilité des types
 - iii. Accessibilité des caract.

Invocation de méthode : 4 cas possibles

- méthode d'instance, virtuelle (cas classique, liaison dynamique)
- méthode de classe (**static**, liaison statique)
- méthode d'instance, privée (**private**, liaison statique)
- méthode d'instance, appelée avec **super** (liaison statique)



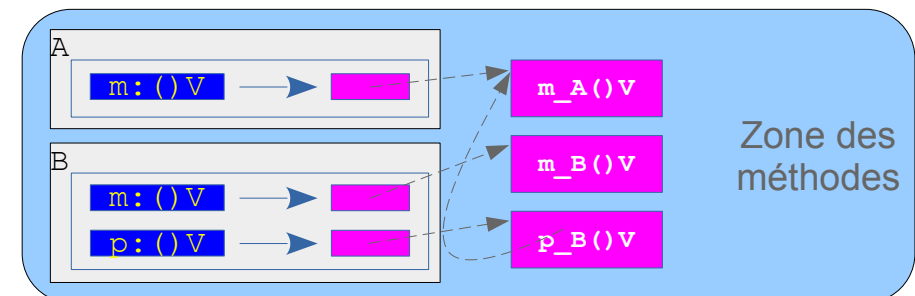
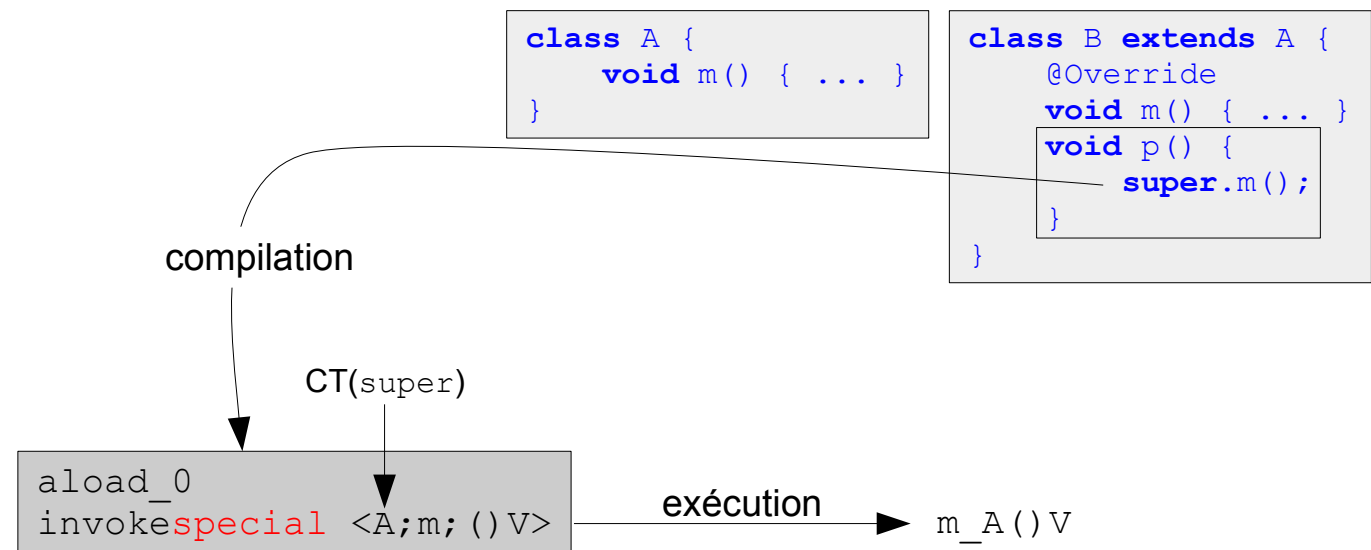
Rappels et compléments

V. Héritage

1. Mécanisme d'héritage
 - i. Définition générale
 - ii. Relation entre classes
 - iii. Formes d'héritage en Java
2. Sous-typage Java
 - i. Sous-type Java direct
 - ii. Sous-type Java
 - iii. Sous-types tableaux
3. Expressions
 - i. Définitions
 - ii. Valeur d'une expression
 - iii. Types d'une expression
 - iv. Transtypage
 - a. Définition
 - b. Extensibilité potentielle
4. Masquage d'attribut
5. Chaînage des constructeurs
6. Redéfinition de méthode
 - i. Redéf° et liaison dynamique
 - ii. Modification d'entête
 - iii. `@Override`
 - iv. Redéfinition et surcharge
7. Invocation de méthode
 - i. Principe
 - a. Méthode virtuelle
 - b. Méthode de classe
 - c. Méthode privée
 - d. Méthode avec `super`
 - ii. Invocation et surcharge
 - a. Résolution d'appel
 - b. Surcharge et héritage
 - c. Piège de la surcharge
 - d. Ambiguïté
8. Accessibilité
 - i. Paquetage
 - ii. Accessibilité des types
 - iii. Accessibilité des caract.

Invocation de méthode : 4 cas possibles

- méthode d'instance, virtuelle (cas classique, liaison dynamique)
- méthode de classe (`static`, liaison statique)
- méthode d'instance, privée (`private`, liaison statique)
- méthode d'instance, appelée avec `super` (liaison statique)

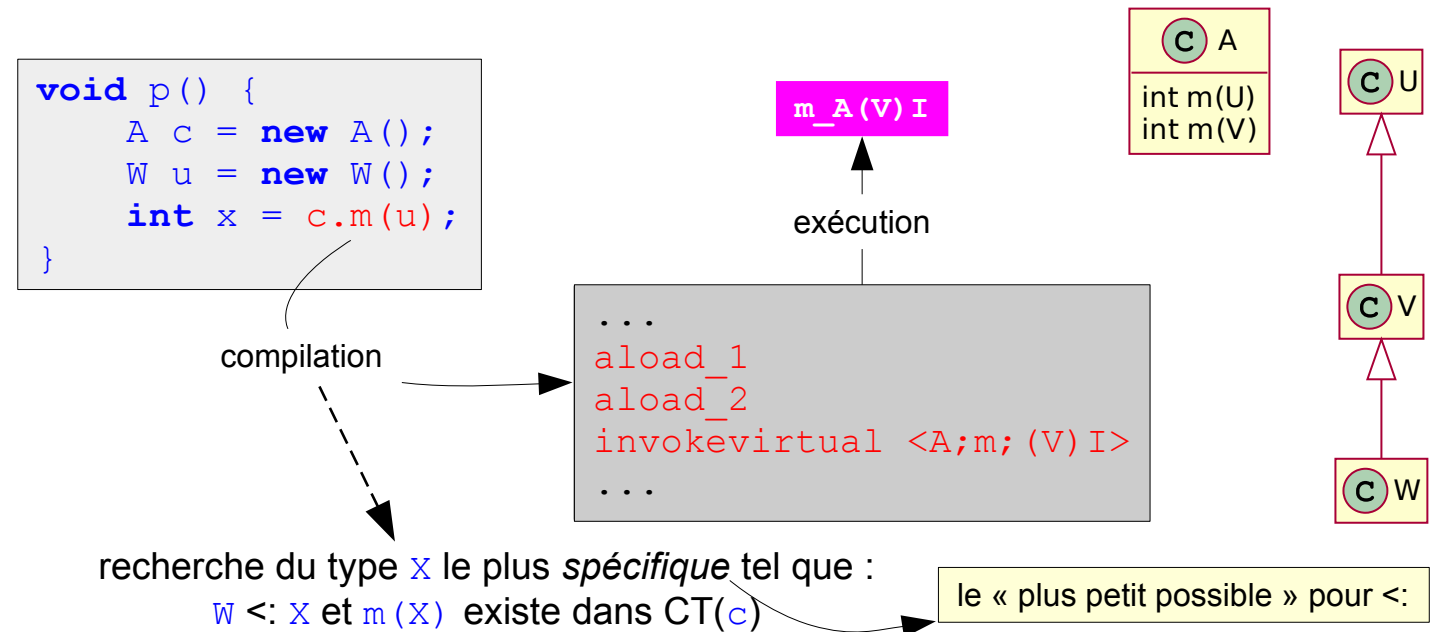


Rappels et compléments

V. Héritage

1. Mécanisme d'héritage
 - i. Définition générale
 - ii. Relation entre classes
 - iii. Formes d'héritage en Java
2. Sous-typage Java
 - i. Sous-type Java direct
 - ii. Sous-type Java
 - iii. Sous-types tableaux
3. Expressions
 - i. Définitions
 - ii. Valeur d'une expression
 - iii. Types d'une expression
 - iv. Transtypage
 - a. Définition
 - b. Extensibilité potentielle
4. Masquage d'attribut
5. Chaînage des constructeurs
6. Redéfinition de méthode
 - i. Redéf° et liaison dynamique
 - ii. Modification d'entête
 - iii. @Override
 - iv. Redéfinition et surcharge
7. Invocation de méthode
 - i. Principe
 - a. Méthode virtuelle
 - b. Méthode de classe
 - c. Méthode privée
 - d. Méthode avec super
 - ii. Invocation et surcharge
 - a. Résolution d'appel
 - b. Surcharge et héritage
 - c. Piège de la surcharge
 - d. Ambiguïté
8. Accessibilité
 - i. Paquetage
 - ii. Accessibilité des types
 - iii. Accessibilité des caract.

Invocation de méthode en présence de surcharge



Appel `c.m(u1, ..., un)`

Compilation

données :

$T = CT(c), T1 = CT(u1), \dots, Tn = CT(un)$

calcul :

$(S1, \dots, Sn) = \min\{(X1, \dots, Xn) \mid T1 <: X1, \dots, Tn <: Xn, m(X1, \dots, Xn) \in T\}$

résultat : **SIG** = `m(S1, ..., Sn)`

Exécution

si `m` est **static** ou **private**, ou si `c` est **super**

liaison statique : exécution de SIG de `CT(c)`

sinon

liaison dynamique : exécution de SIG de `RT(c)`

$T = A$
 $n = 1$
 $T1 = W$
 $S1 = V$