R2.01 : Développement Orienté Objets C++

cours 2

François Delobel, Anaïs Durand, Abdel Hasbani, Laurent Provot, Carine Simon

IUT CLERMONT AUVERGNE
Aurillac - Clermont-Ferrand - Le Puy-en-Velay
Montluçon - Moulins - Vichy

Mémoire : la pile/le tas

2 zones mémoire : la pile et le tas.

- la pile :
  - pas très grande
  - on y empile toutes les informations des fonctions en cours d'exécution (paramètres, variables statiques,...)
  - on dépile/libére à la fin de l'exécution de la fonction.
- le tas :
  - plus vaste
  - on y entasse les instances allouées dynamiquement
  - libération explicite de la mémoire nécessaire
    ATTENTION aux fuites mémoires !!!

Schéma mémoire

schéma mémoire :

- permet de représenter l'état de la mémoire à un instant donné;
- est composé de 3 parties : contexte, pile, tas.

Réalisation du schéma mémoire :

1. on part de la 1ère ligne du main;
2. on effectue les instructions une à une (évolution du schéma);
3. on s'arrête à la ligne de code correspondant à l'instant voulu.

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
  pta2 = uneFonction(3);
  int  retour = a1.uneMethodeDeA(2);
>-----------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

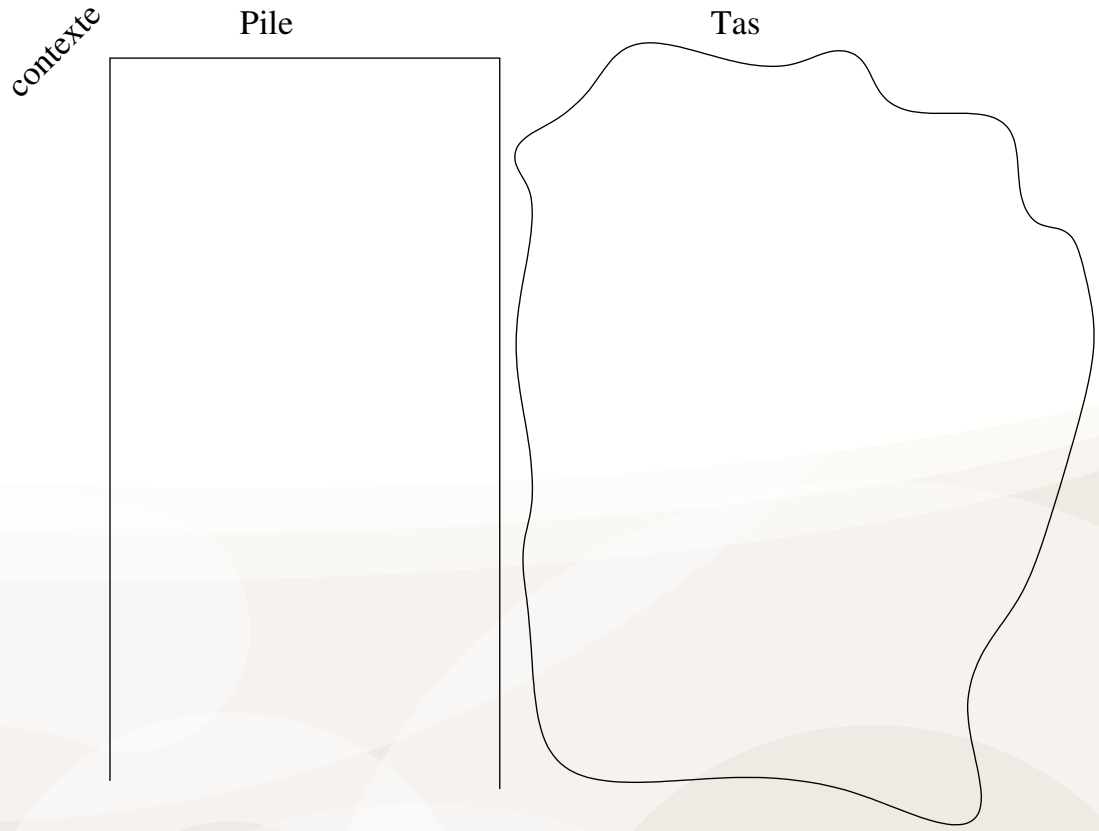

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------
int main(){
>  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
  pta2 = uneFonction(3);
  int  retour = a1.uneMethodeDeA(2);
>-----------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

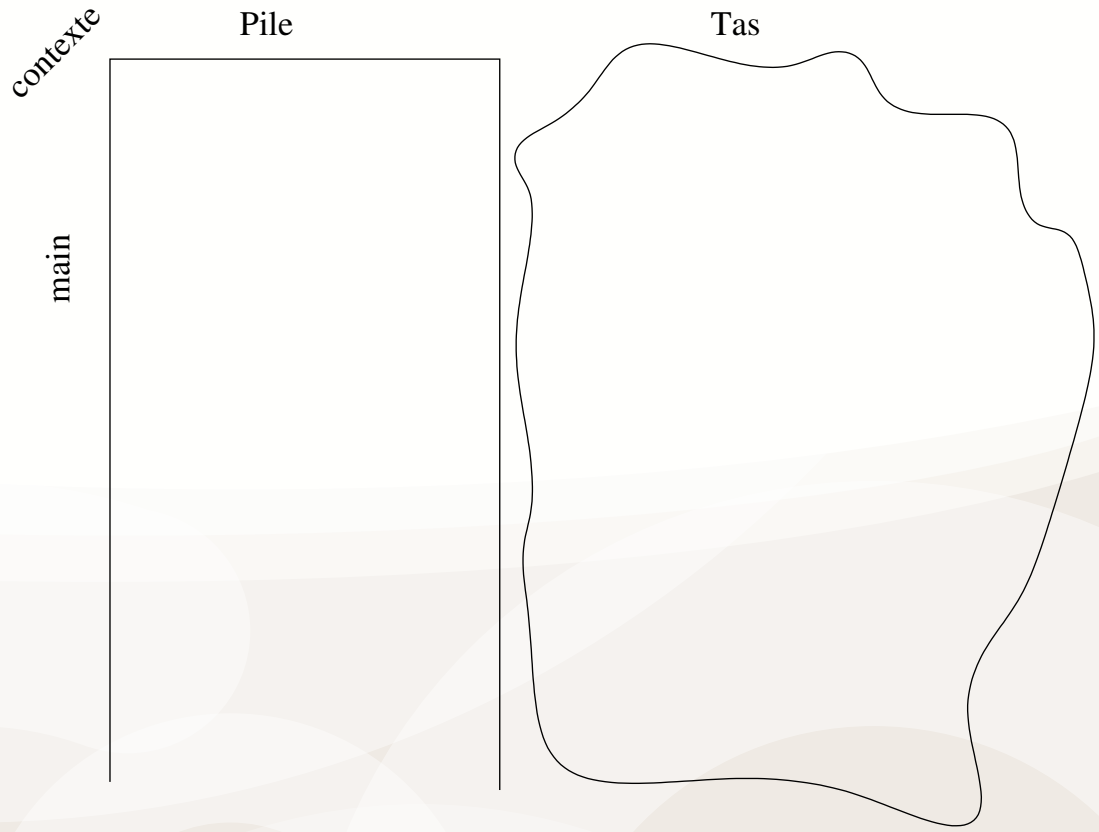

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------
int main(){
>  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
  pta2 = uneFonction(3);
  int  retour = a1.uneMethodeDeA(2);
>-----------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

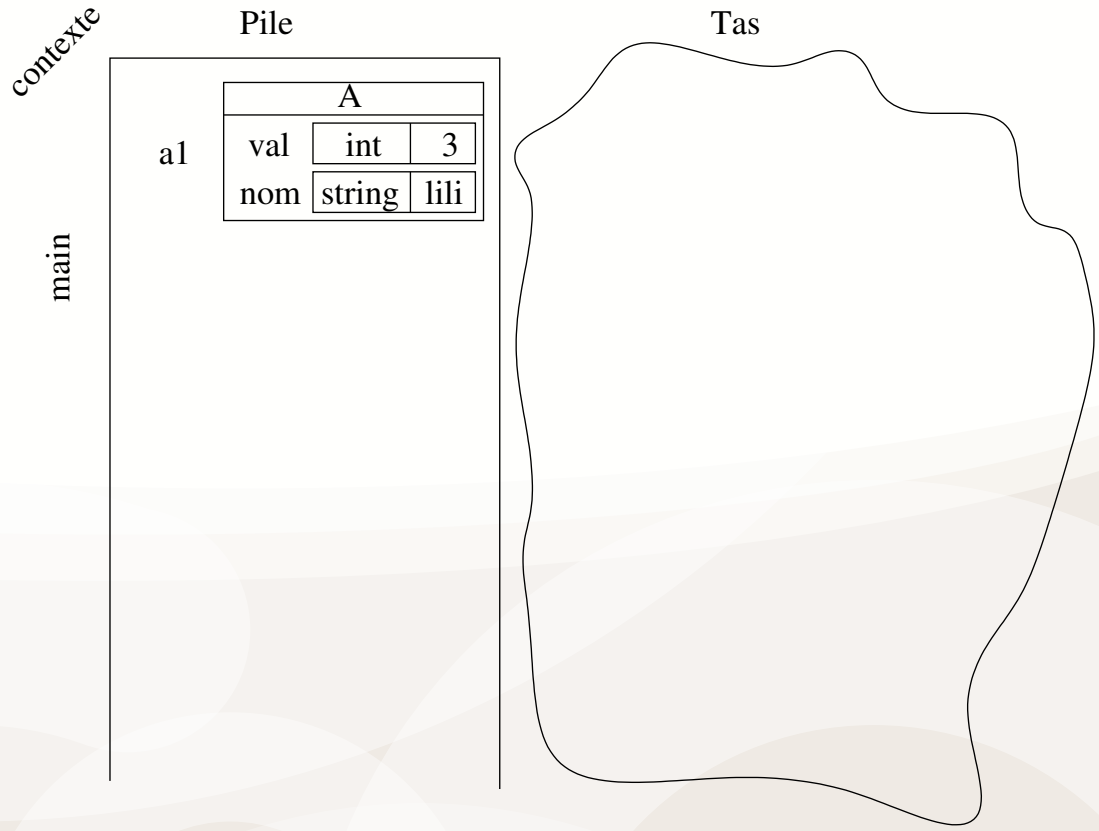

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------------
int main(){
  A a1{3,"lili"};
>  A *pta1, *pta2;
  pta1 = new A[2];
  pta2 = uneFonction(3);
  int  retour = a1.uneMethodeDeA(2);
>-----------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

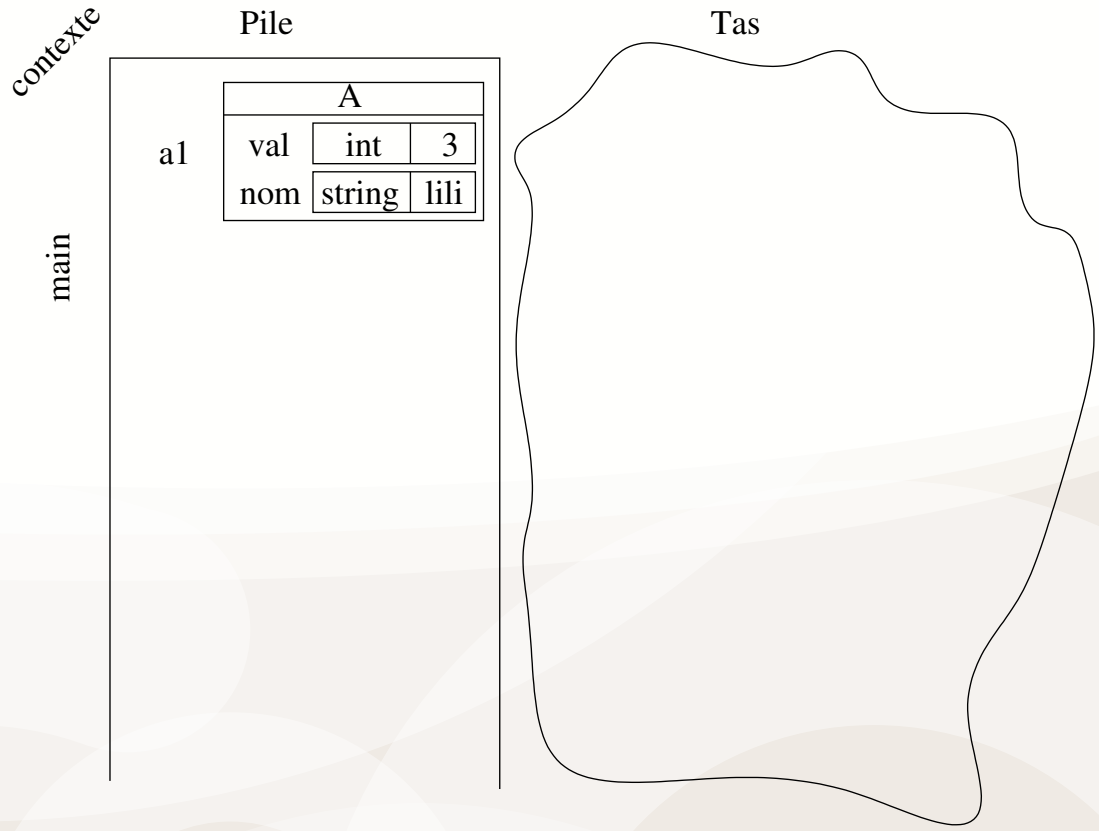

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------------
int main(){
  A a1{3,"lili"};
>  A *pta1, *pta2;
  pta1 = new A[2];
  pta2 = uneFonction(3);
  int  retour = a1.uneMethodeDeA(2);
>----------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

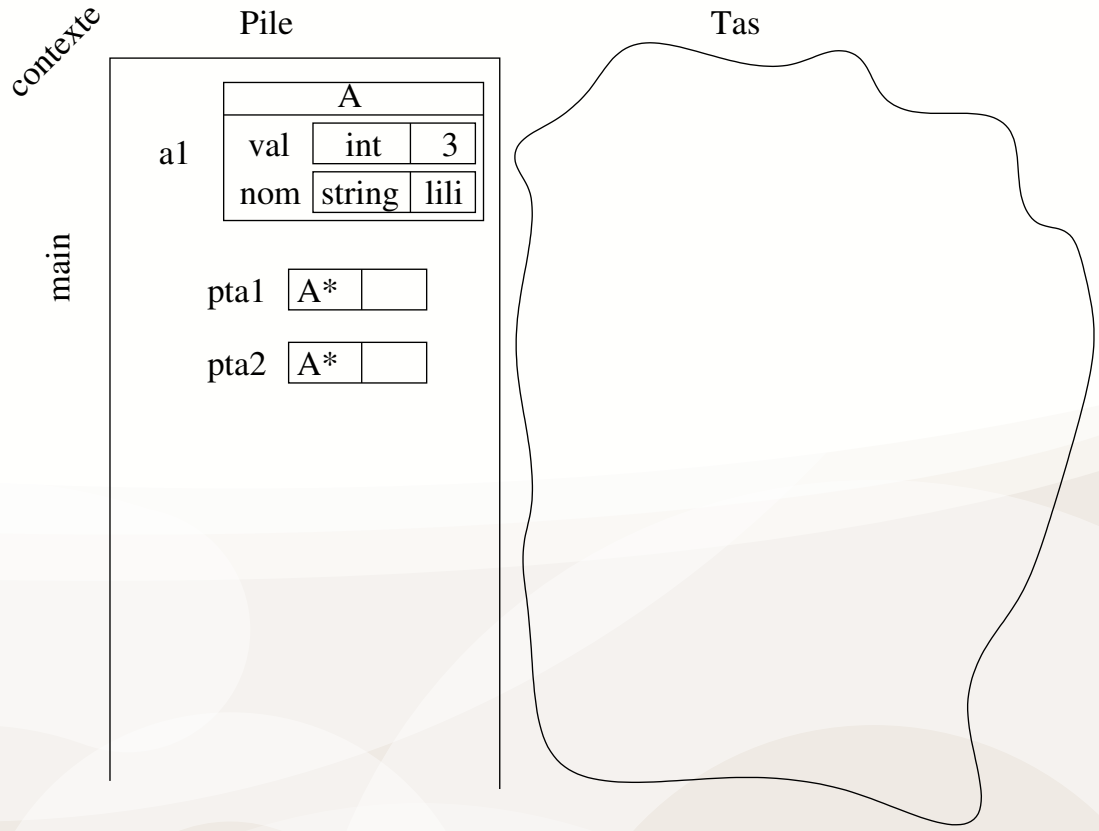

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
>  pta1 = new A[2];
  pta2 = uneFonction(3);
  int  retour = a1.uneMethodeDeA(2);
>-----------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

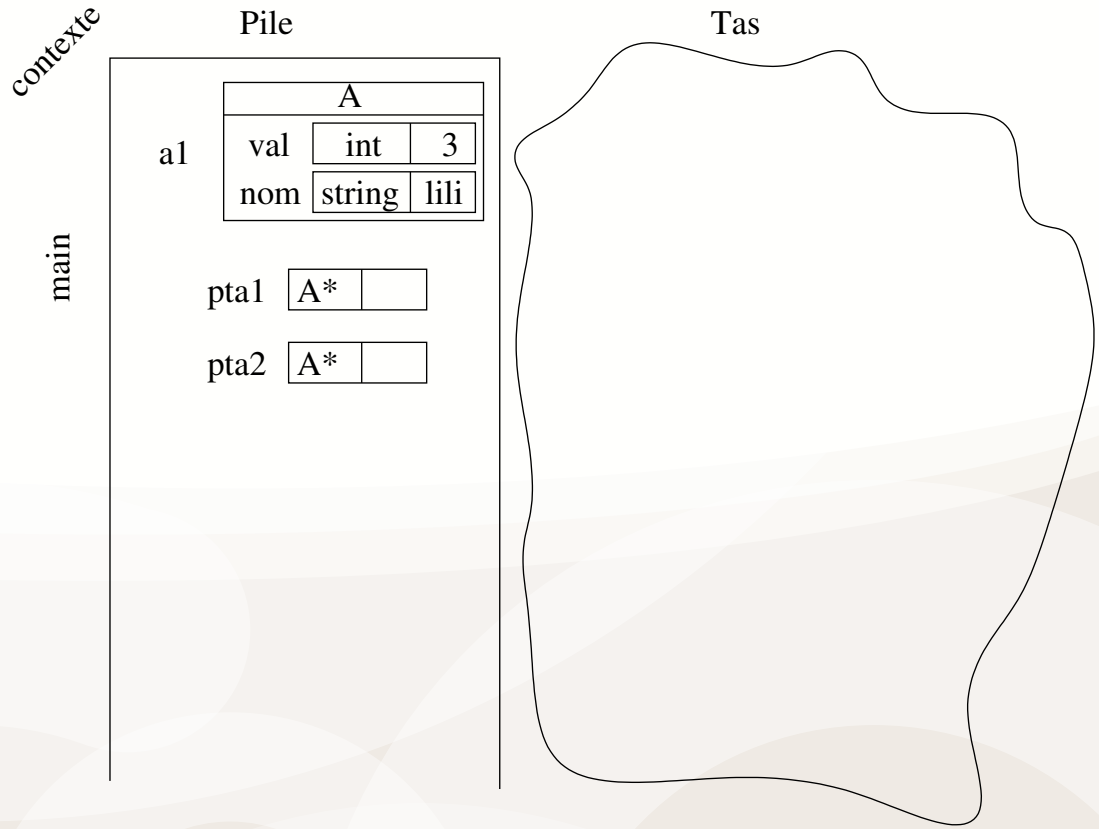

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
>  pta1 = new A[2];
  pta2 = uneFonction(3);
  int  retour = a1.uneMethodeDeA(2);
>-----------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

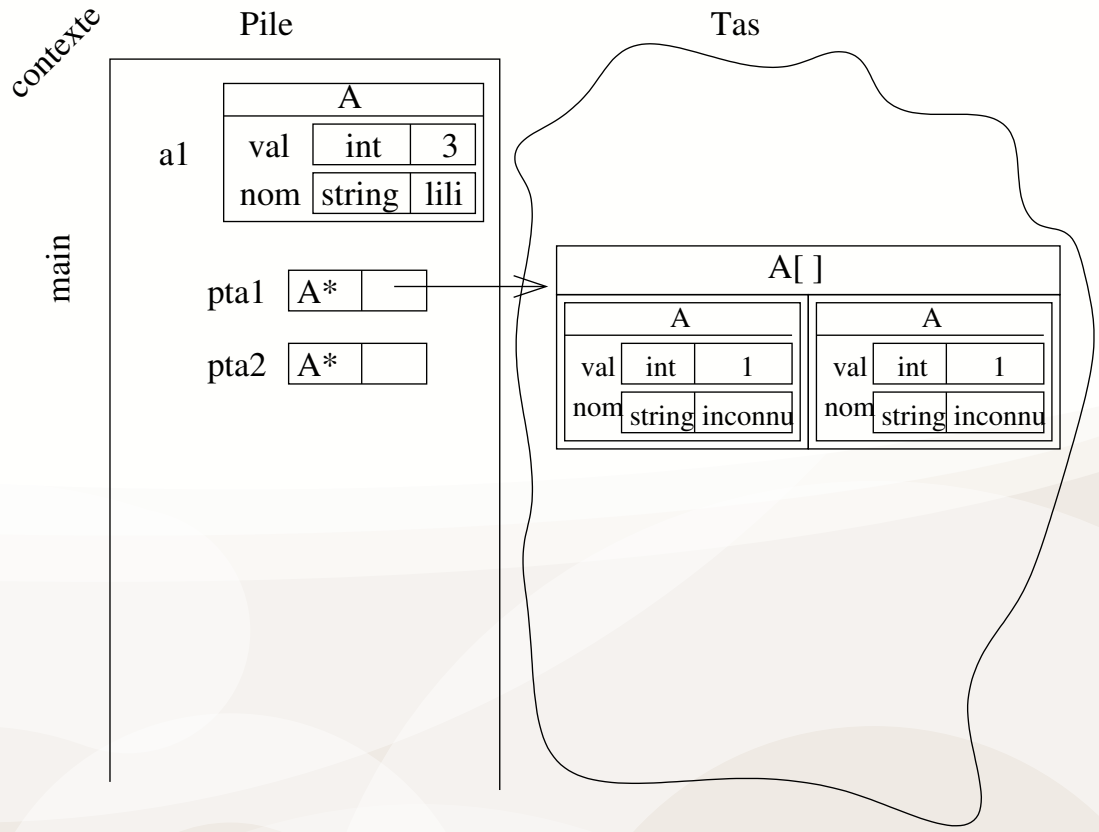

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
>  pta2 = uneFonction(3);
  int  retour = a1.uneMethodeDeA(2);
>-----------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

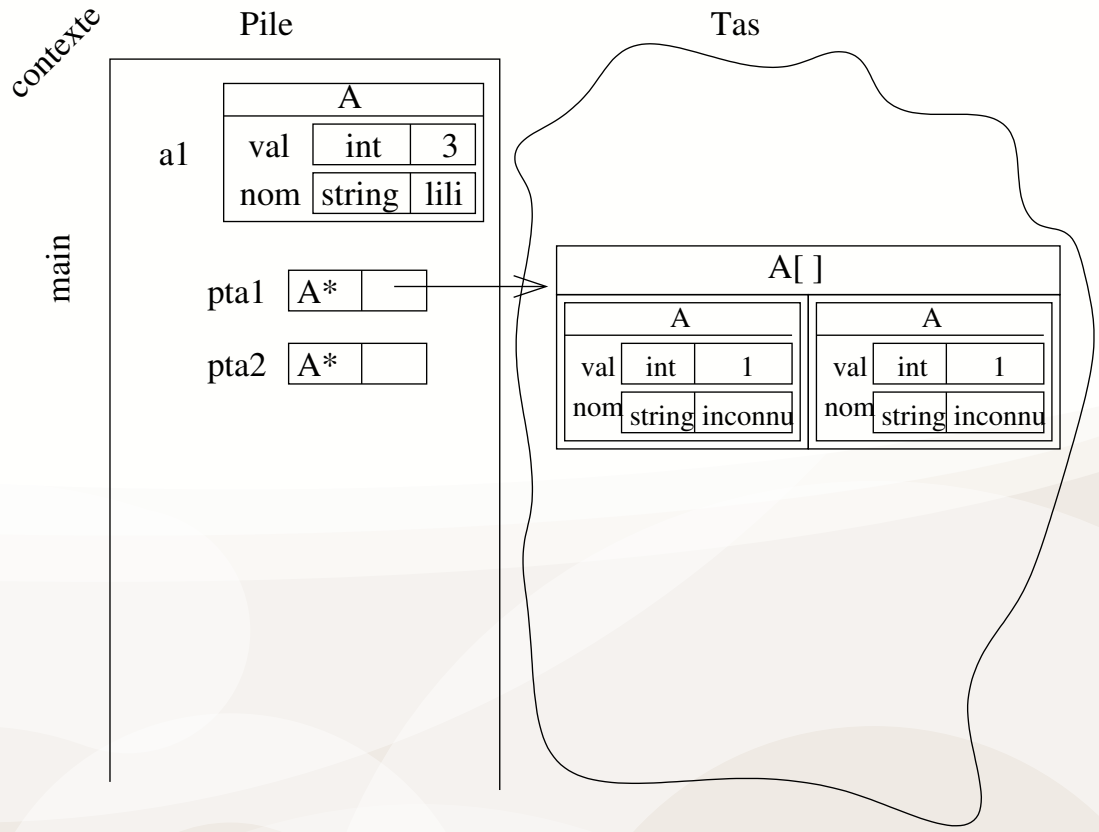

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------
>A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
>  pta2 = uneFonction(3);
  int  retour = a1.uneMethodeDeA(2);
>----------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

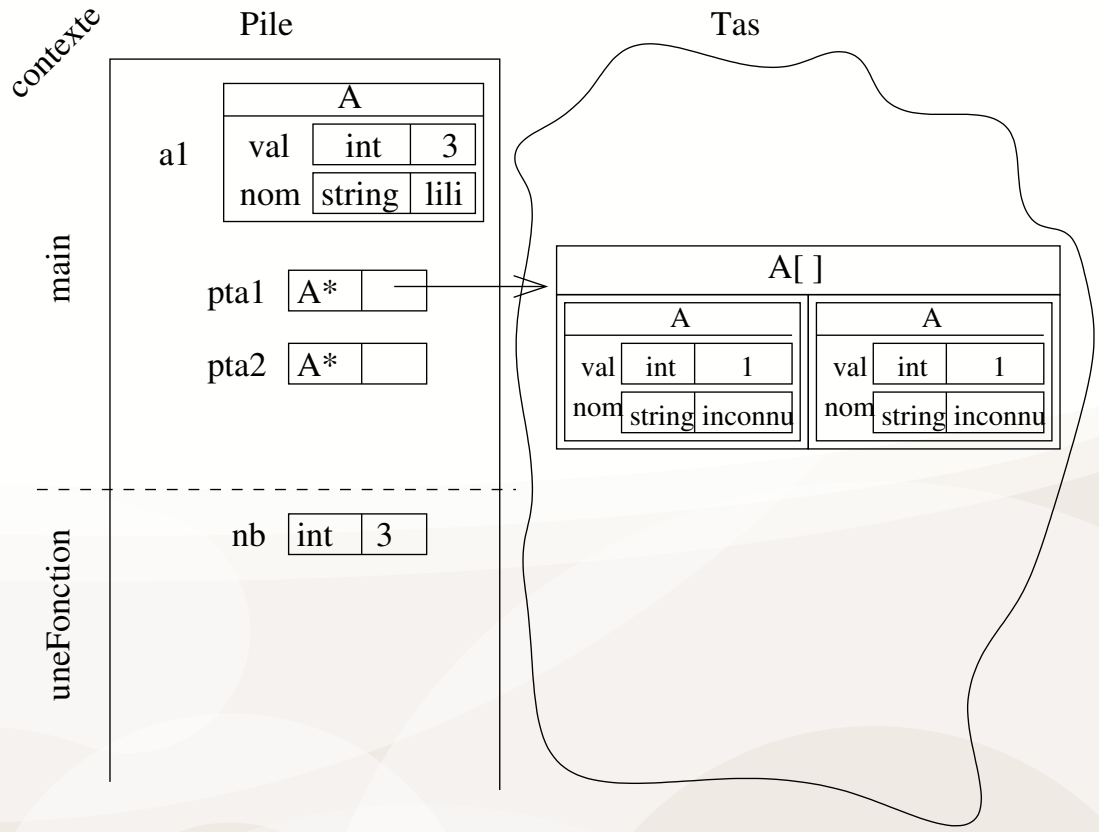

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------
A* uneFonction(int nb){
>  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
>  pta2 = uneFonction(3);
  int  retour = a1.uneMethodeDeA(2);
>-----------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

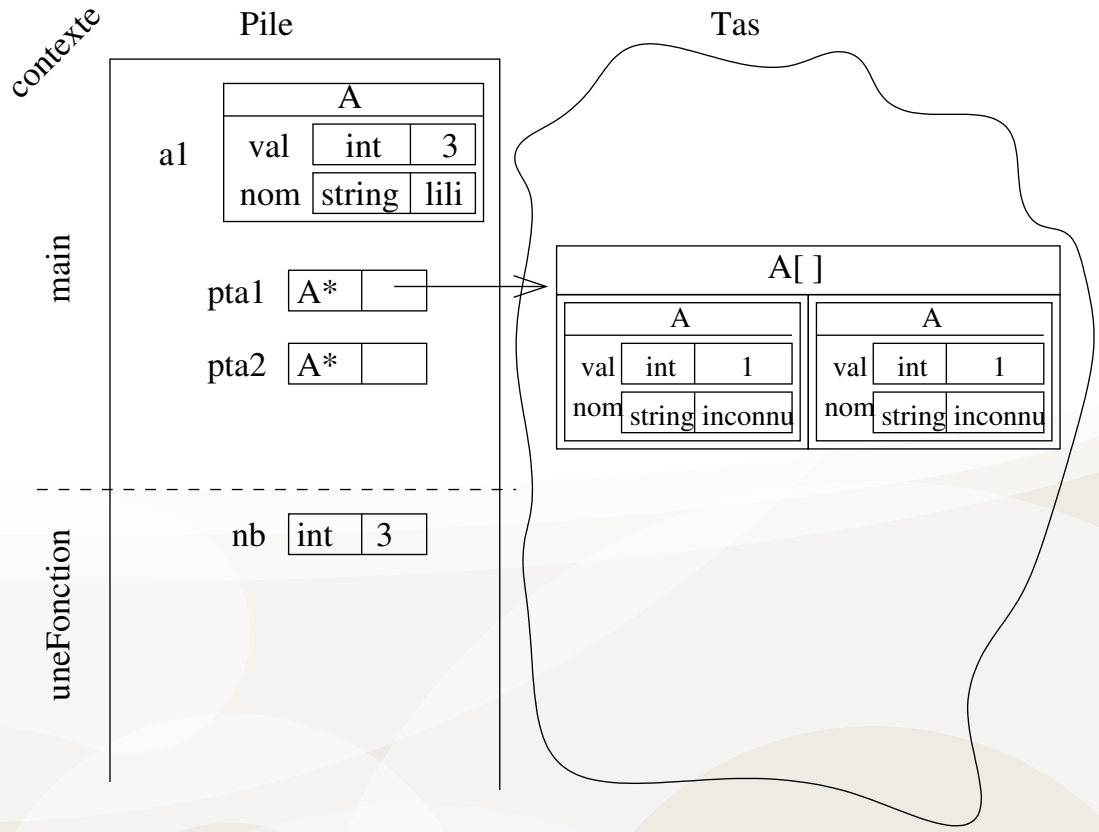

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------
A* uneFonction(int nb){
>  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
>  pta2 = uneFonction(3);
  int  retour = a1.uneMethodeDeA(2);
>-----------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

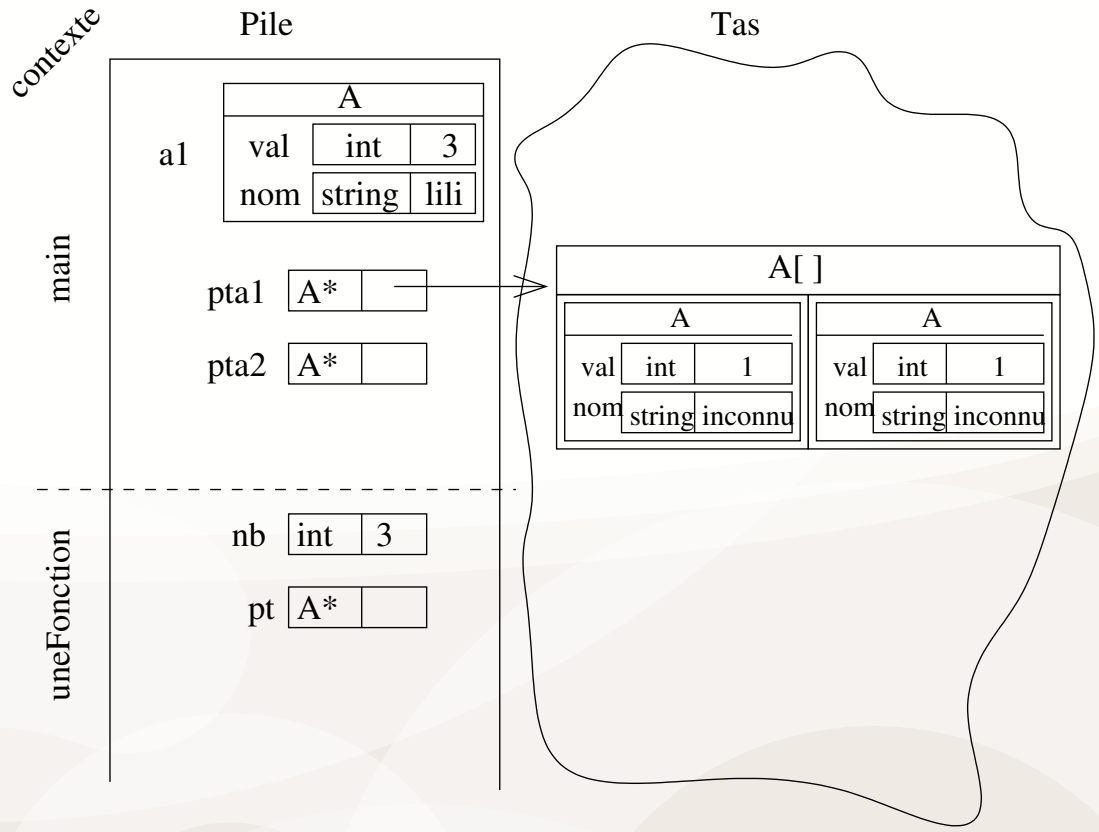

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------
A* uneFonction(int nb){
>  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
>  pta2 = uneFonction(3);
  int  retour = a1.uneMethodeDeA(2);
>-----------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

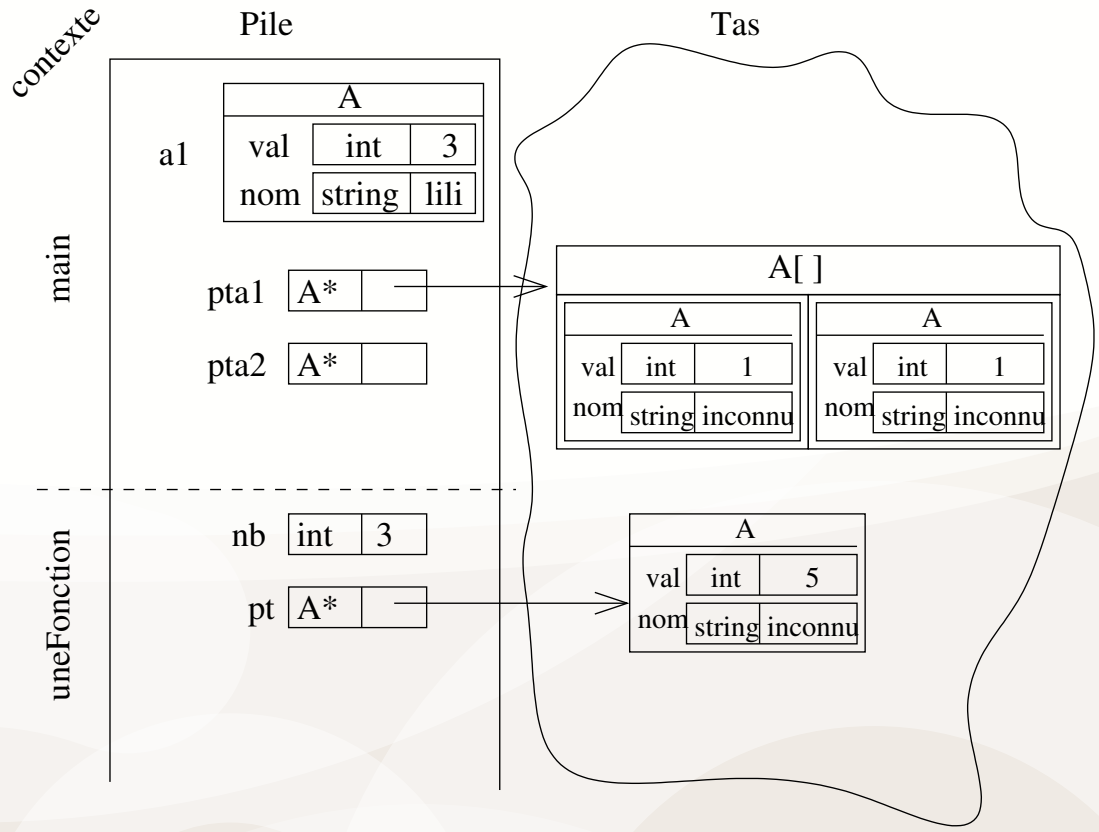

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
>  return pt;
}
//-------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
>  pta2 = uneFonction(3);
  int  retour = a1.uneMethodeDeA(2);
>-----------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

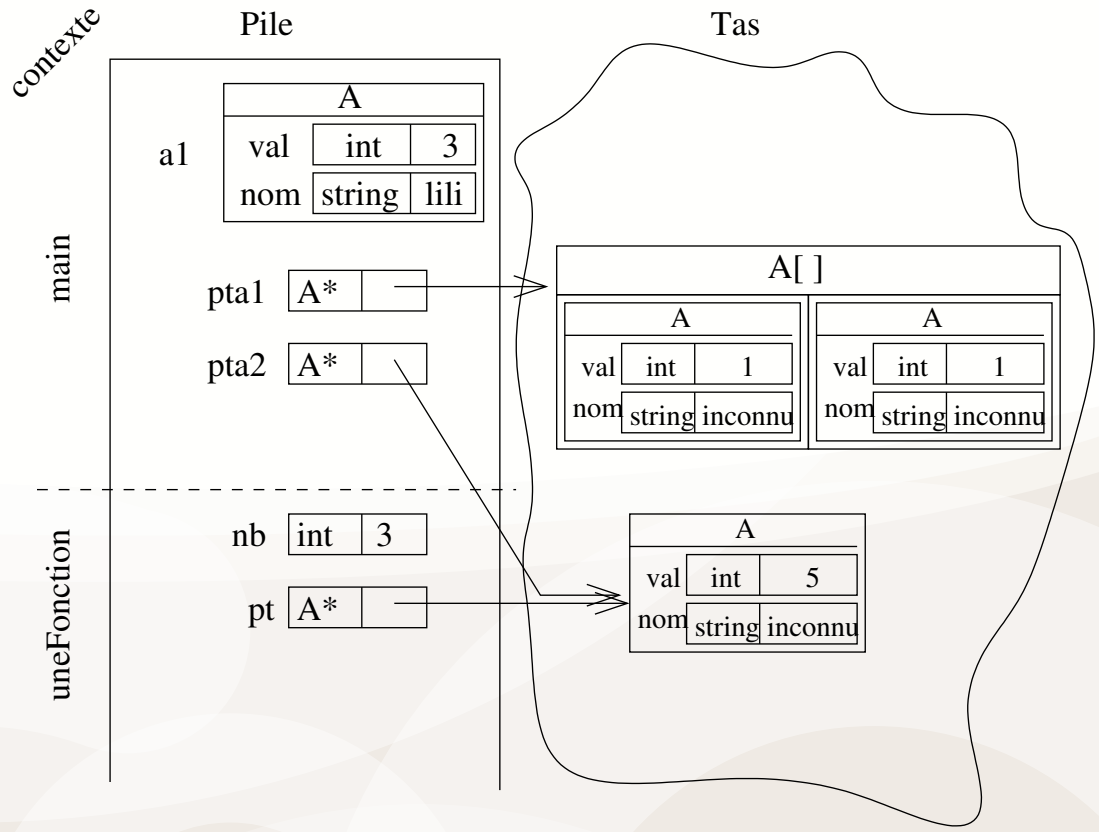

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
>}
//-------------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
_  pta2 = uneFonction(3);
  int  retour = a1.uneMethodeDeA(2);
>----------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

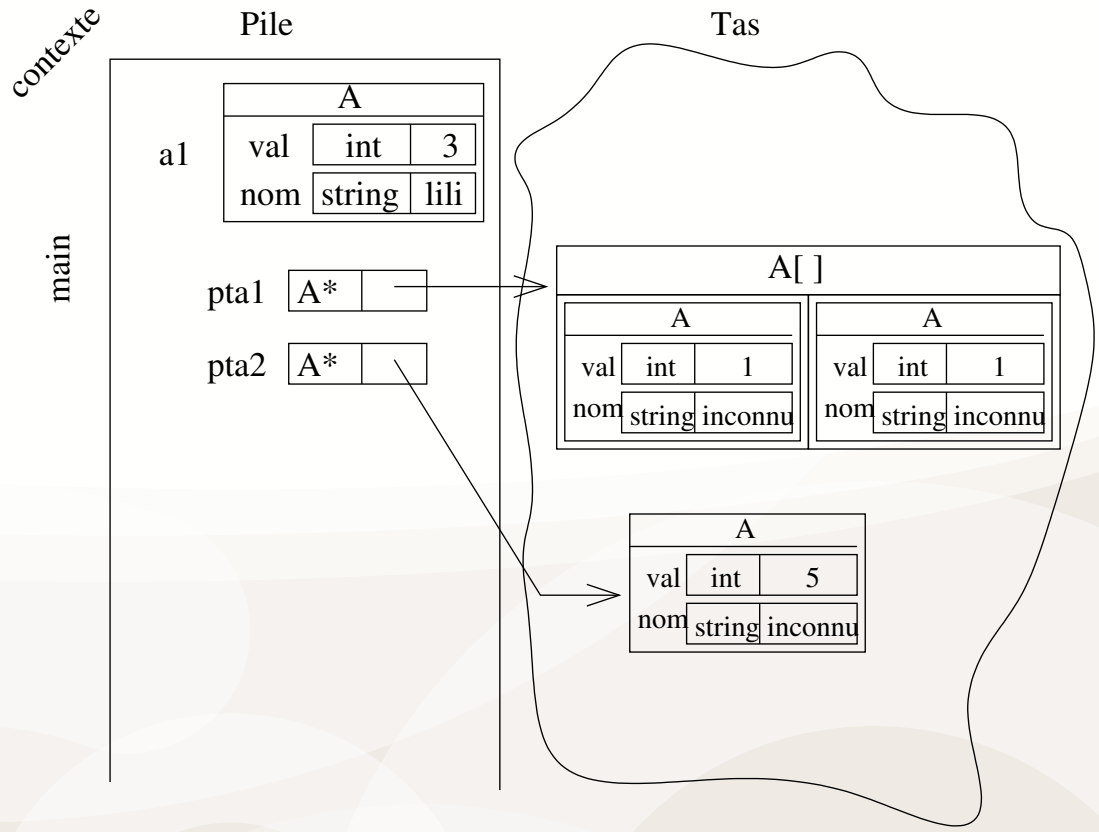

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
  pta2 = uneFonction(3);
>  int  retour = a1.uneMethodeDeA(2);
>-----------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

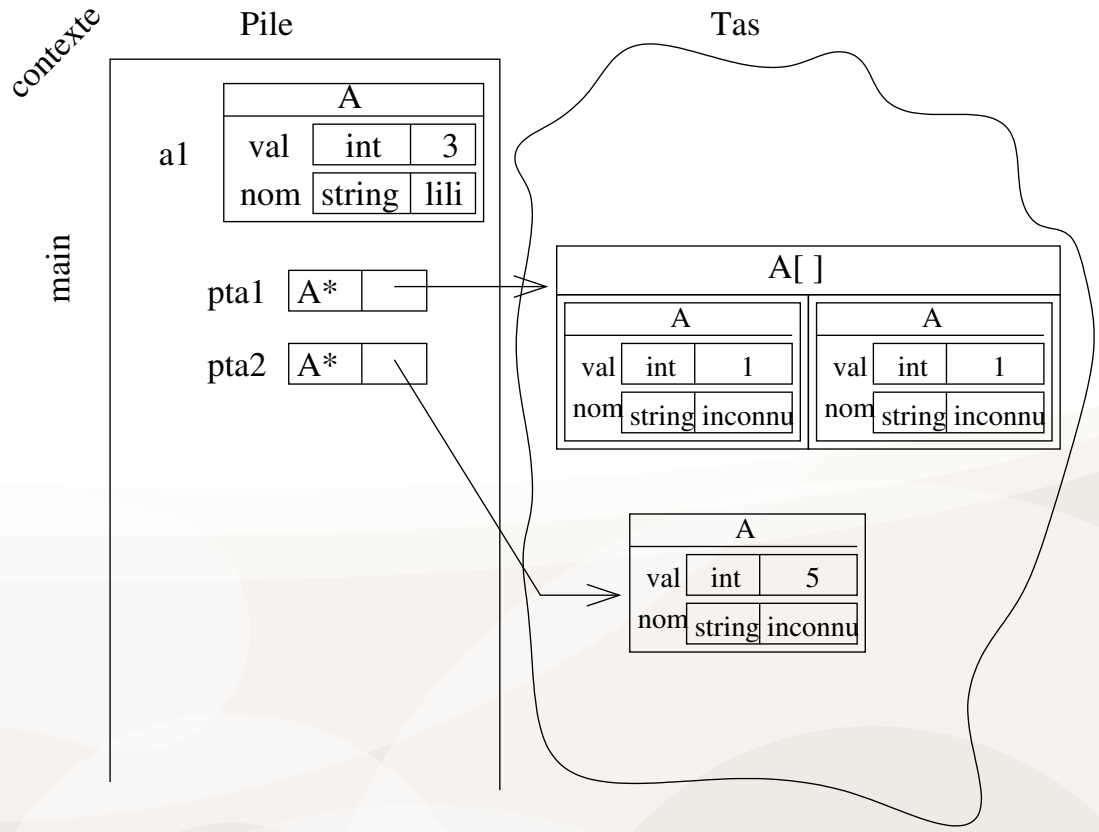

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
  pta2 = uneFonction(3);
>  int  retour = a1.uneMethodeDeA(2);
>-----------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

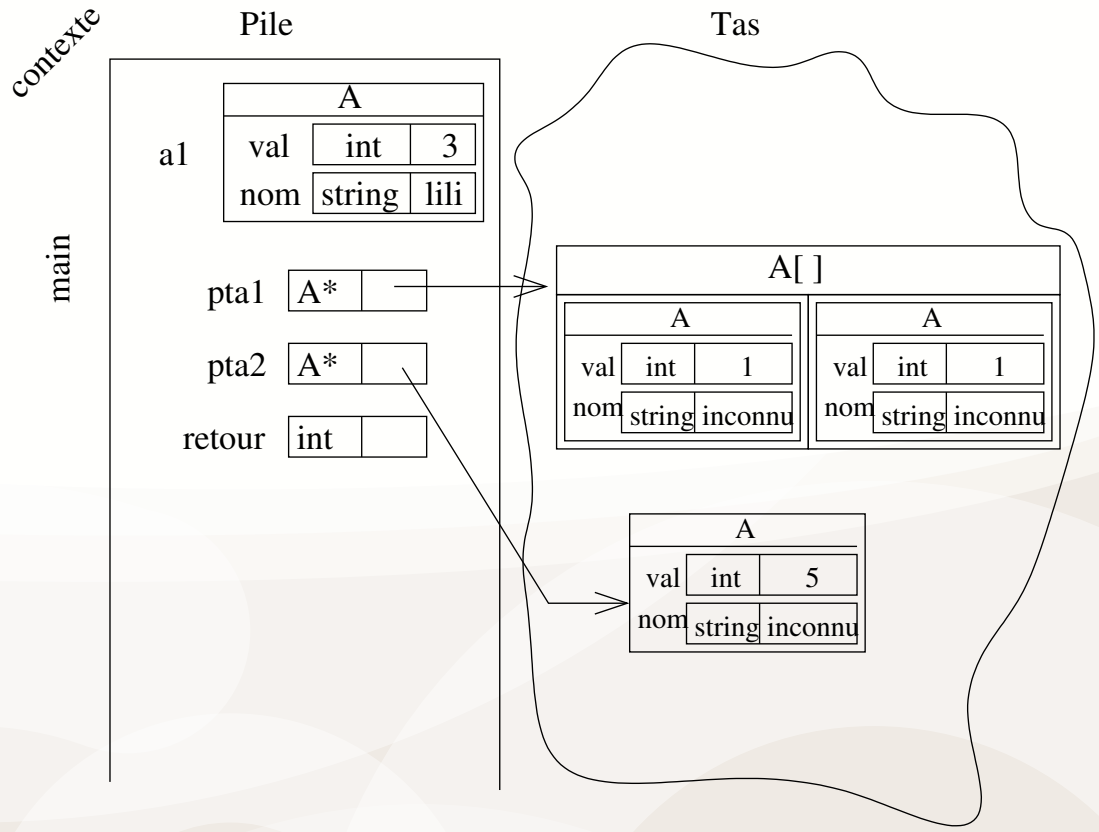

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
>int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
  pta2 = uneFonction(3);
>  int  retour = a1.uneMethodeDeA(2);
>---------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

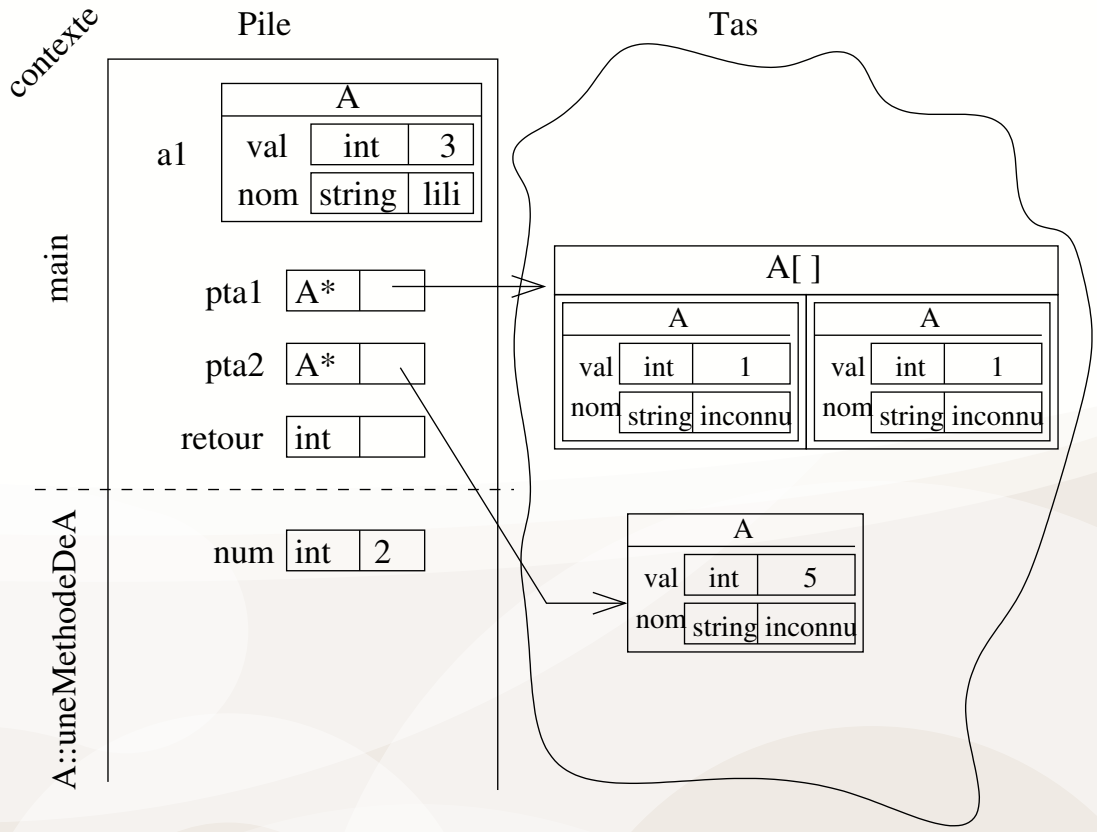

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
>   int res = val*num;
  return res;
}
//-------------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
  pta2 = uneFonction(3);
>   int   retour = a1.uneMethodeDeA(2);
>-----------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

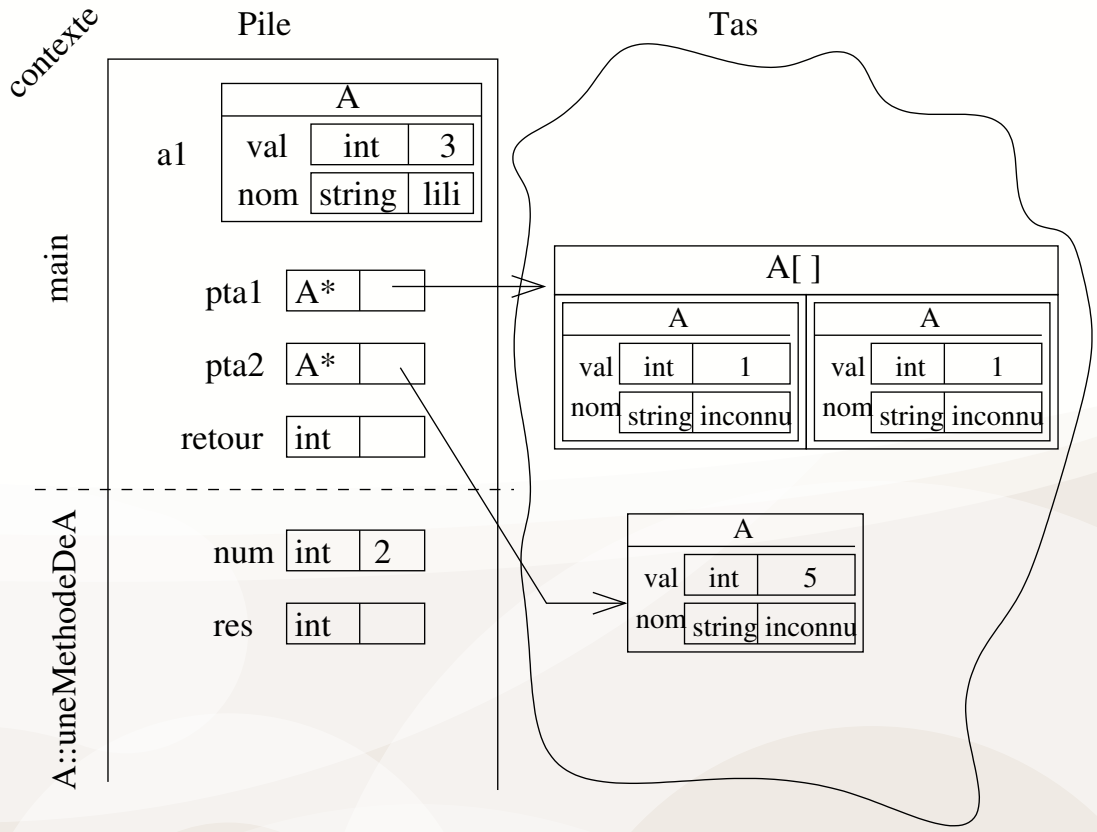

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
>   int res = val*num;
  return res;
}
//-------------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
  pta2 = uneFonction(3);
>   int   retour = a1.uneMethodeDeA(2);
>-----------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

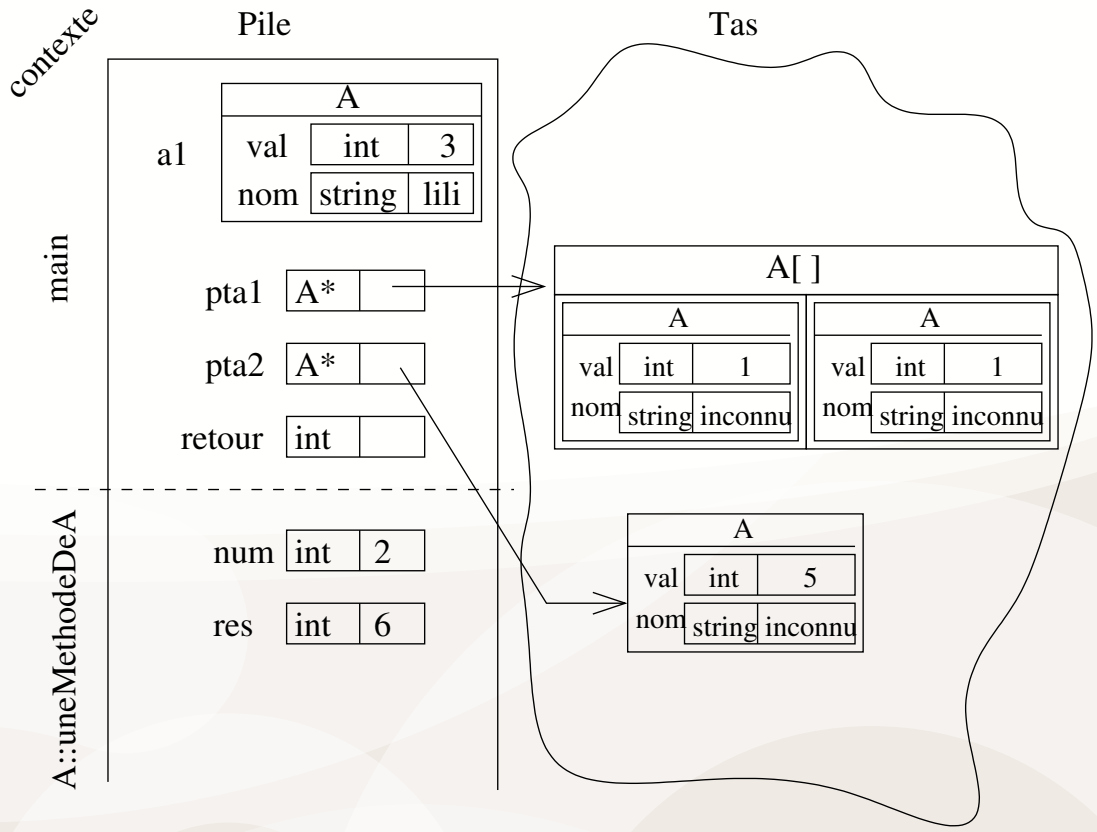

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
>  return res;
}
//-------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
  pta2 = uneFonction(3);
>  int  retour = a1.uneMethodeDeA(2);
>-----------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

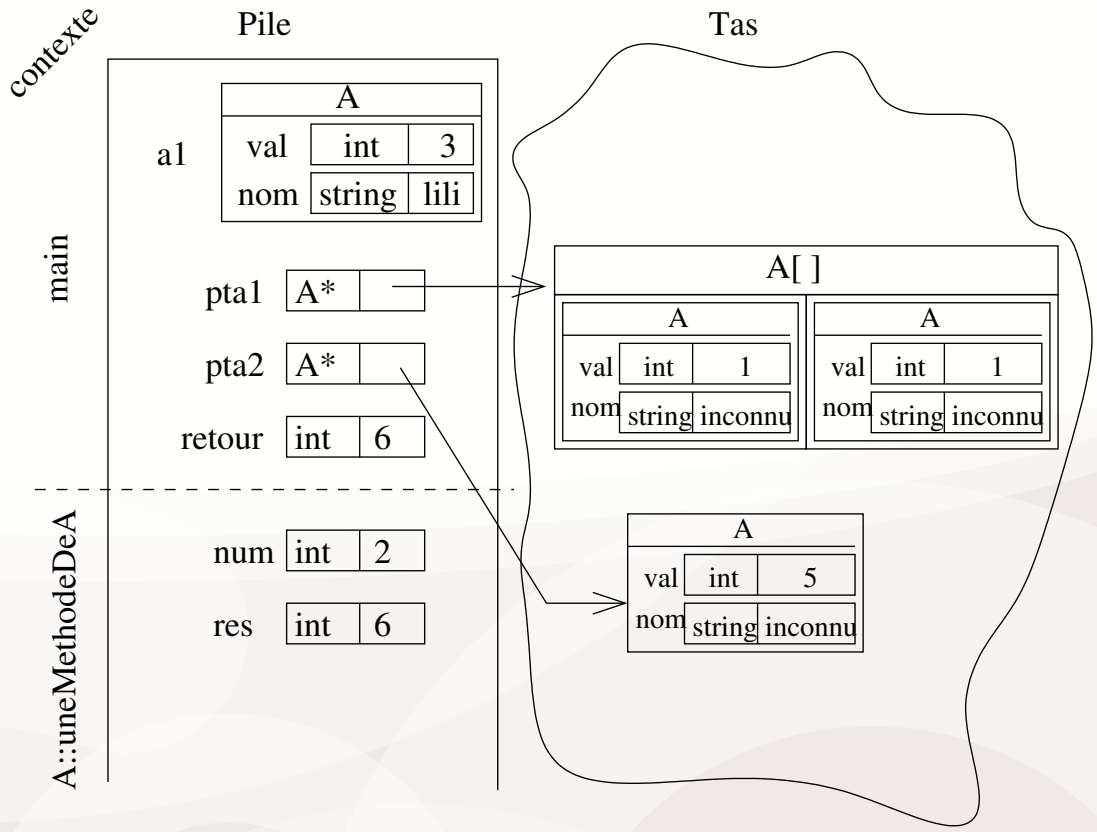

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
>}
//-------------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
  pta2 = uneFonction(3);
  int  retour = a1.uneMethodeDeA(2);
>----------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

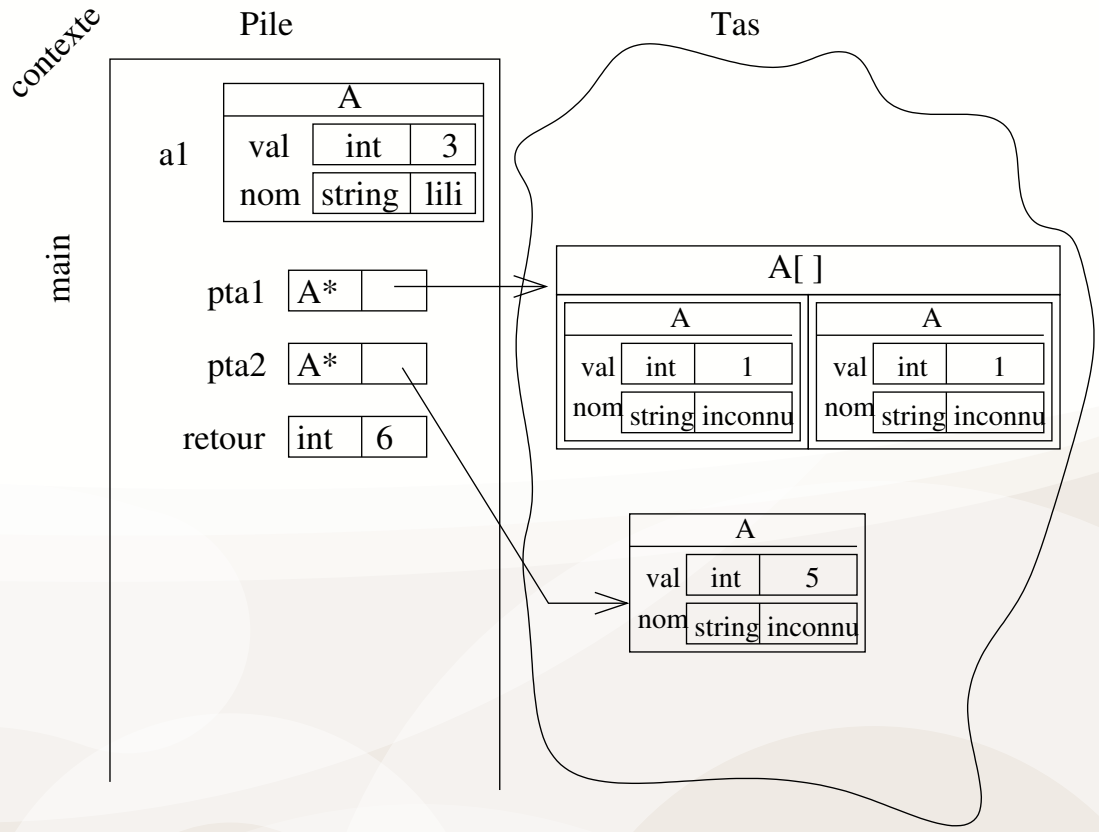

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
  pta2 = uneFonction(3);
  int  retour = a1.uneMethodeDeA(2);
//-------------------------------
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
```

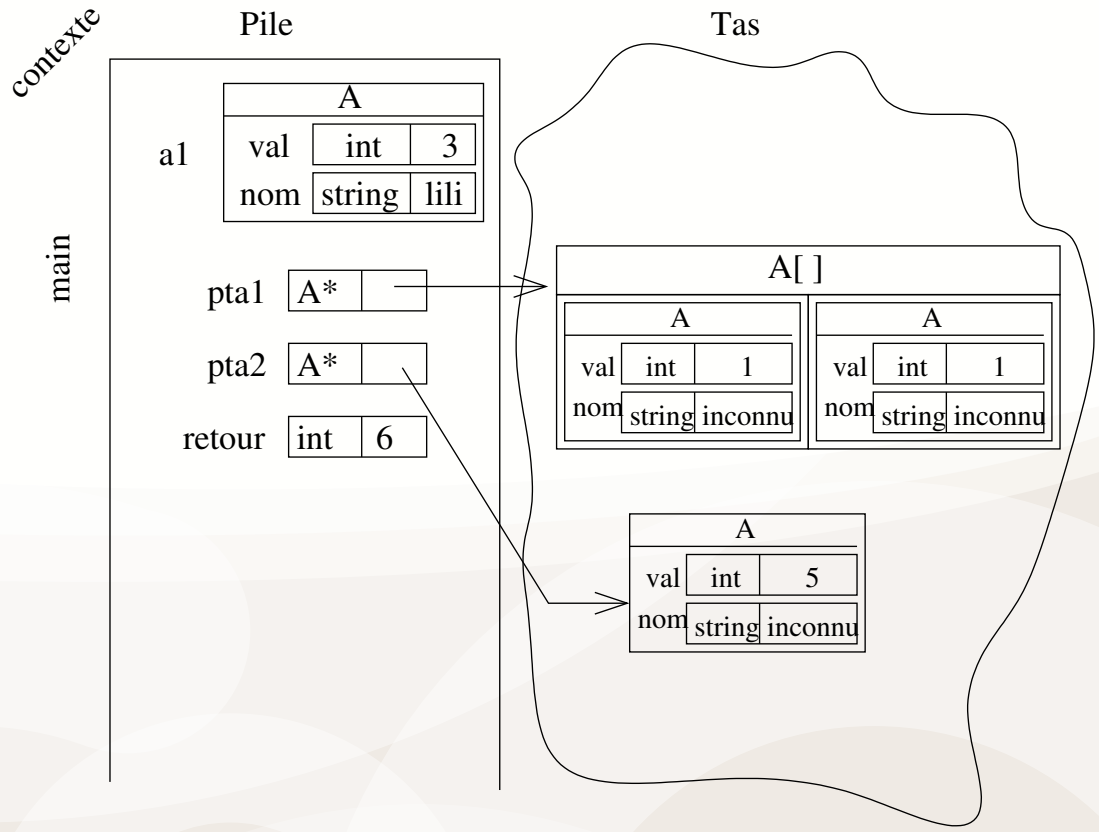

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
  pta2 = uneFonction(3);
  int  retour = a1.uneMethodeDeA(2);
  cout << retour << endl;
>-----------------
  delete[] pta1;
  delete pta2;
}
```

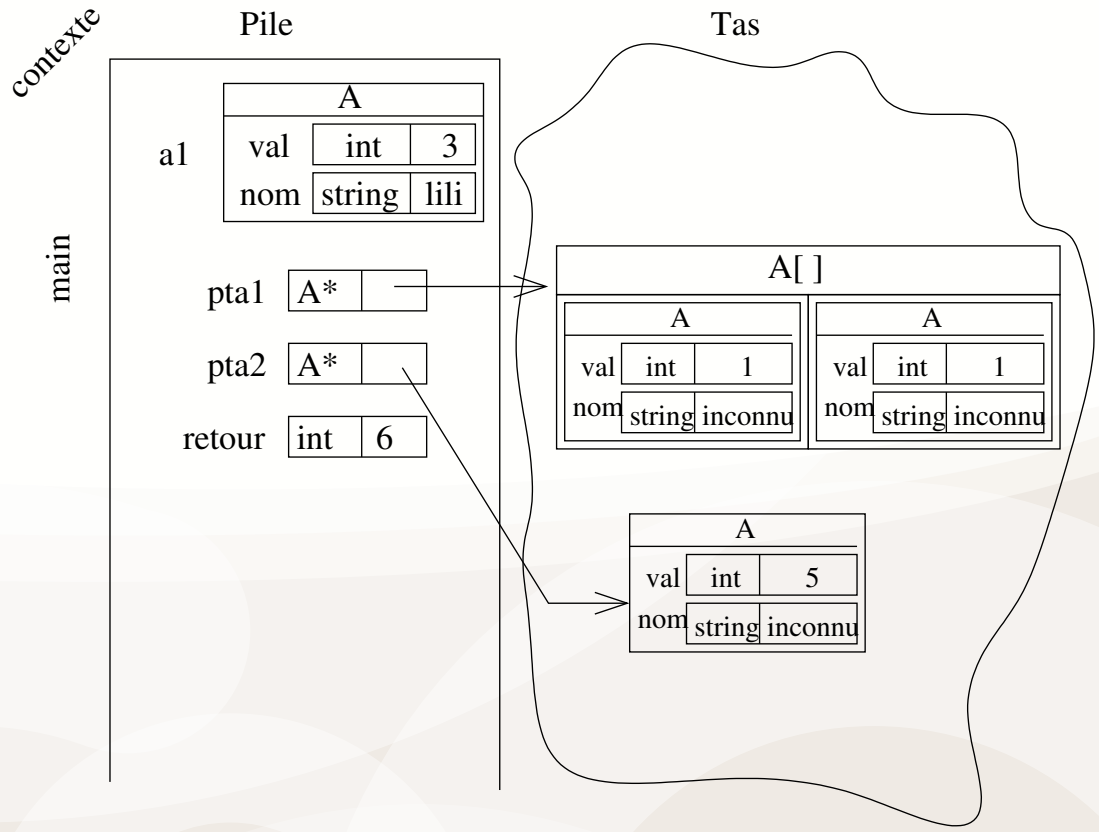

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
  pta2 = uneFonction(3);
  int  retour = a1.uneMethodeDeA(2);
  cout << retour << endl;
  delete[] pta1;
>-----------------
  delete pta2;
}
```

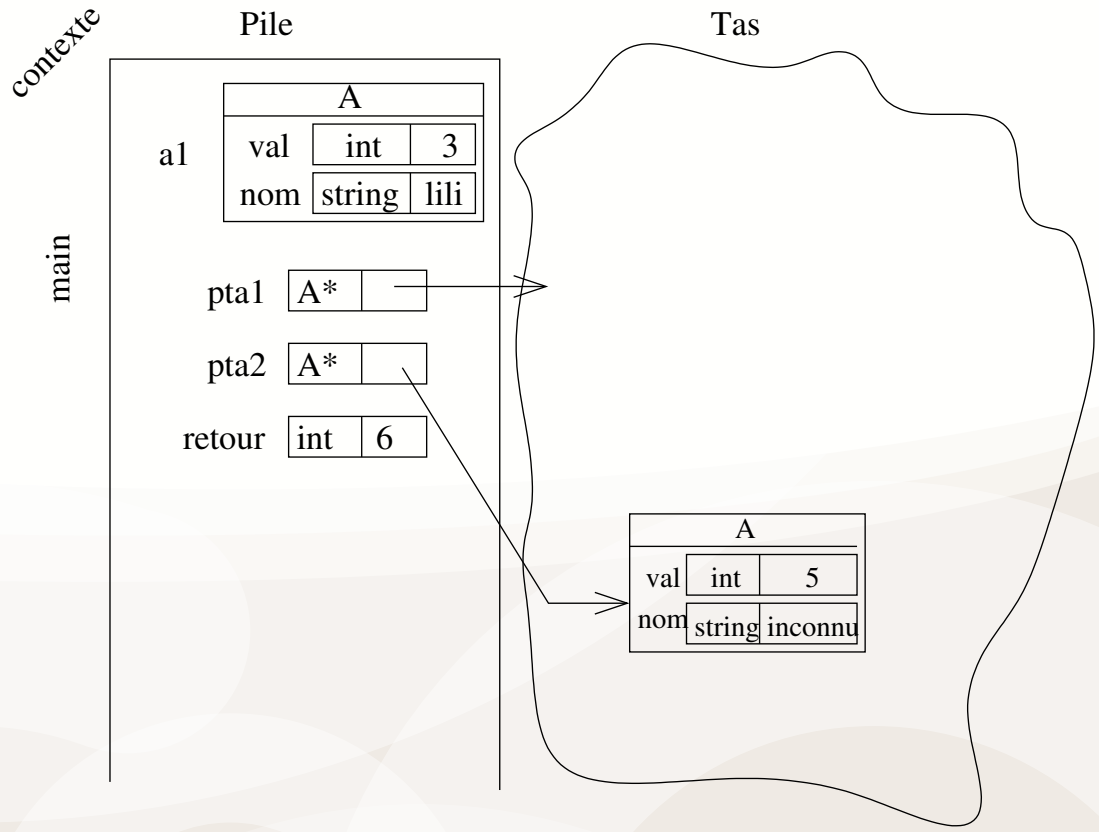

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
  pta2 = uneFonction(3);
  int  retour = a1.uneMethodeDeA(2);
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
>-----------------
}
```

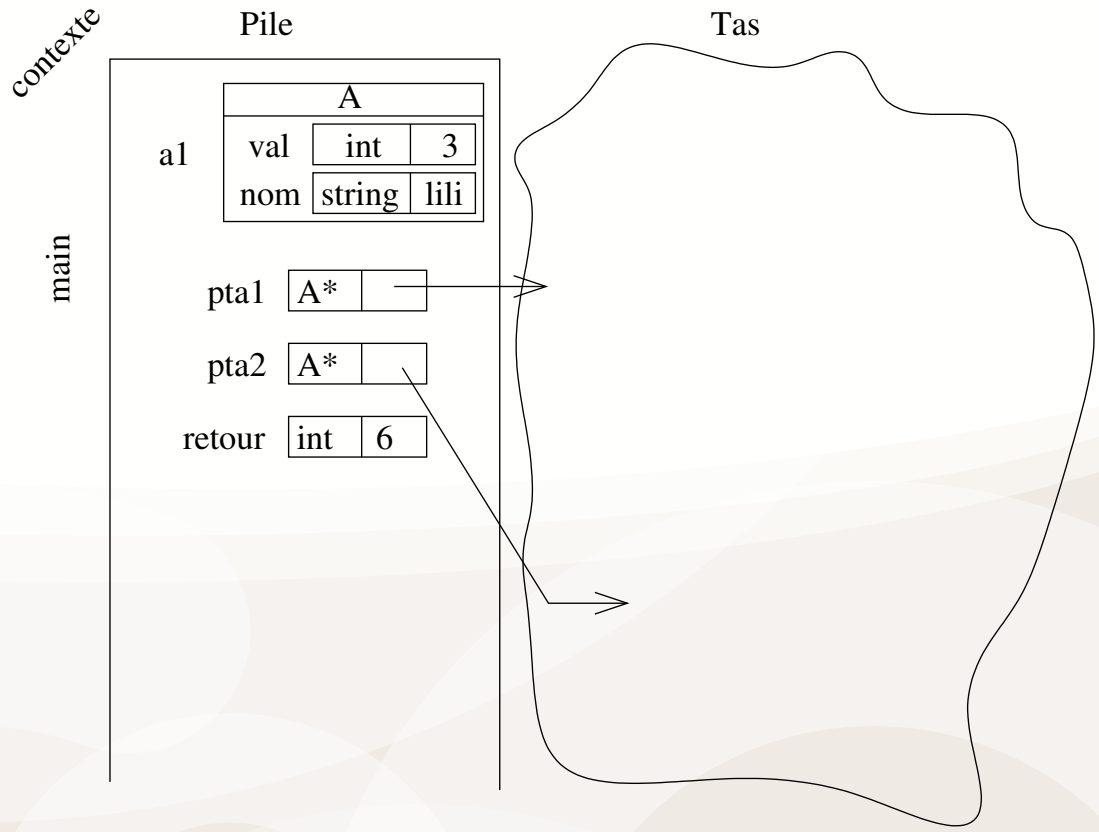

Schéma mémoire : exemple

```
class A{
  int val;
  string nom;
public :
  A(int val=1, string nom = "inconnu");
  ~A(){}
  int uneMethodeDeA(int num);
};
//-------------------------------------------------
A::A(int val, string nom):val{val},nom{nom}
{ }
int A::uneMethodeDeA(int num){
  int res = val*num;
  return res;
}
//-------------------------------------------------
A* uneFonction(int nb){
  A * pt = new A{nb+2};
  return pt;
}
//-------------------------------------------------
int main(){
  A a1{3,"lili"};
  A *pta1, *pta2;
  pta1 = new A[2];
  pta2 = uneFonction(3);
  int  retour = a1.uneMethodeDeA(2);
  cout << retour << endl;
  delete[] pta1;
  delete pta2;
}
>-----------------
```