

# Primitives haut niveau des types Graphe et DiGraphe

---

Type Primitives

*Graphe*

```
.créerGraphe()
.ajouterSommet() : Sommet
.ajouterArête(E u : Sommet, E v : Sommet) : Arête
.supprimerSommet(E u : Sommet)
.supprimerArête(E e : Arête)

.nombreSommets() : N
.nombreArêtes() : N
.sommets() : Itérateur de Sommet
.arêtes() : Itérateur de Arête
.voisins(E u : Sommet) : Itérateur de Sommet
.arêtesIncidentes(E u : Sommet) : Itérateur de Arête
.extrémités(E e : Arête) : (Sommet, Sommet)
.degré(E u : Sommet) : N
.existeArête(E u : Sommet, E v : Sommet) : Booléen
```

---

*DiGraphe*

```
.créerDiGraphe()
.ajouterSommet() : Sommet
.ajouterArc(E u : Sommet, E v : Sommet) : Arc
.supprimerSommet(E u : Sommet)
.supprimerArc(E e : Arc)

.nombreSommets() : N
.nombreArcs() : N
.sommets() : Itérateur de Sommet
.arcs() : Itérateur de Arc
.voisinsSortants(E u : Sommet) : Itérateur de Sommet
.voisinsEntrants(E u : Sommet) : Itérateur de Sommet
.voisins(E u : Sommet) : Itérateur de Sommet
.arcsEntrants(E u : Sommet) : Itérateur de Arc
.arcsSortants(E u : Sommet) : Itérateur de Arc
.arcsIncidents(E u : Sommet) : Itérateur de Arc
.source(E e : Arc) : Sommet
.destination(E e : Arc) : Sommet
.degréEntrant(E u : Sommet) : N
.degréSortant(E u : Sommet) : N
.degré(E u : Sommet) : N
.existeArc(E u : Sommet, E v : Sommet) : Booléen
```

---

---

---

**Action** *afficheExtrémitésArêtes* (**E** *g* : **Graphe**)

**Variable(s)**

    src : **Sommet**  
    dest : **Sommet**  
└ e : **Arête**

**Début**

**Pour chaque** *e* **de** *g.arêtes()* **Faire**  
        (src, dest)  $\leftarrow$  *g.extrémités*(*e*)  
        **Écrire** ("*source :*", *src*, "*et destination :*", *dest*)  
└

---