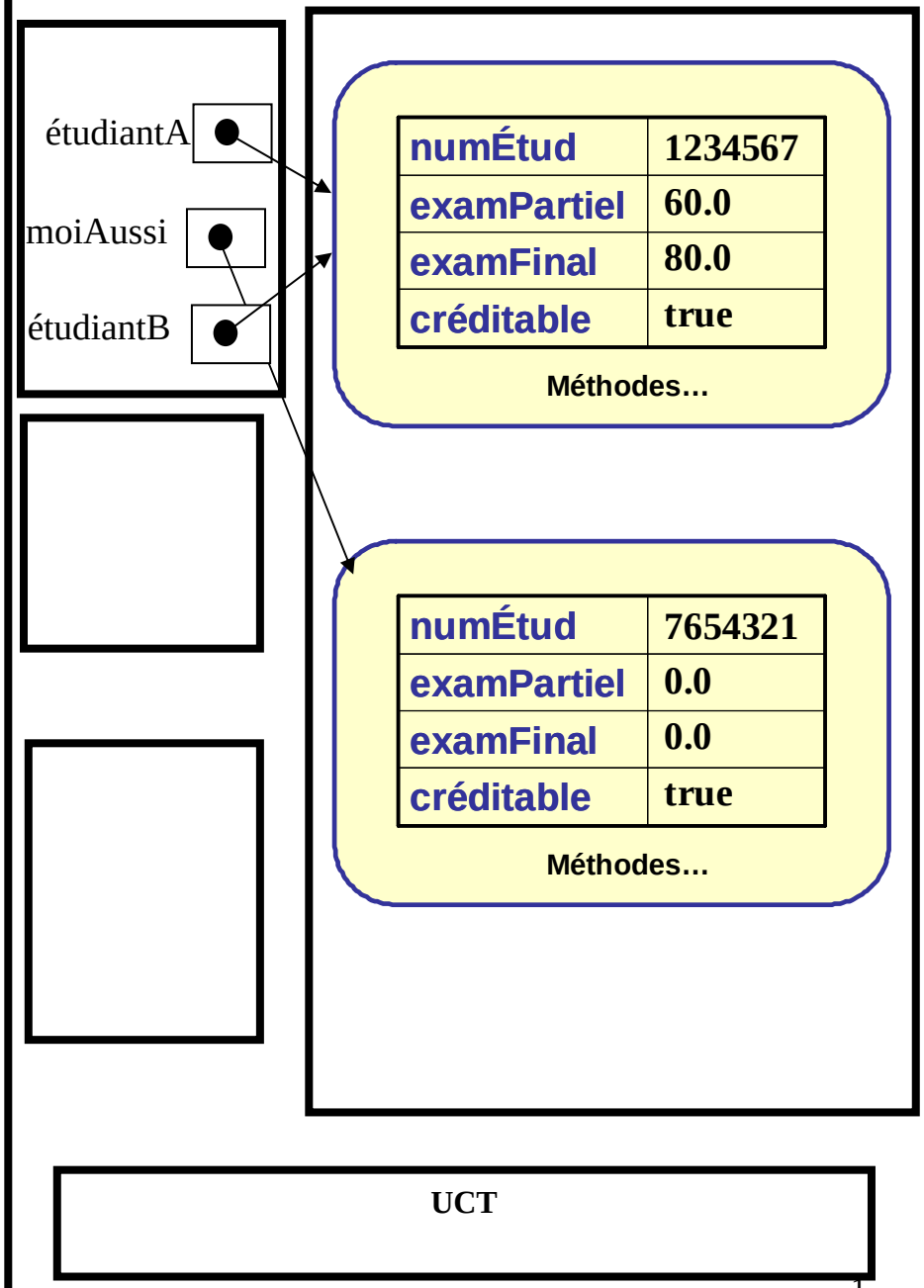
Mémoire de programme — Exercice 11-1 – Utilisation des Constructeurs — Mém de travail — Mémoire globale

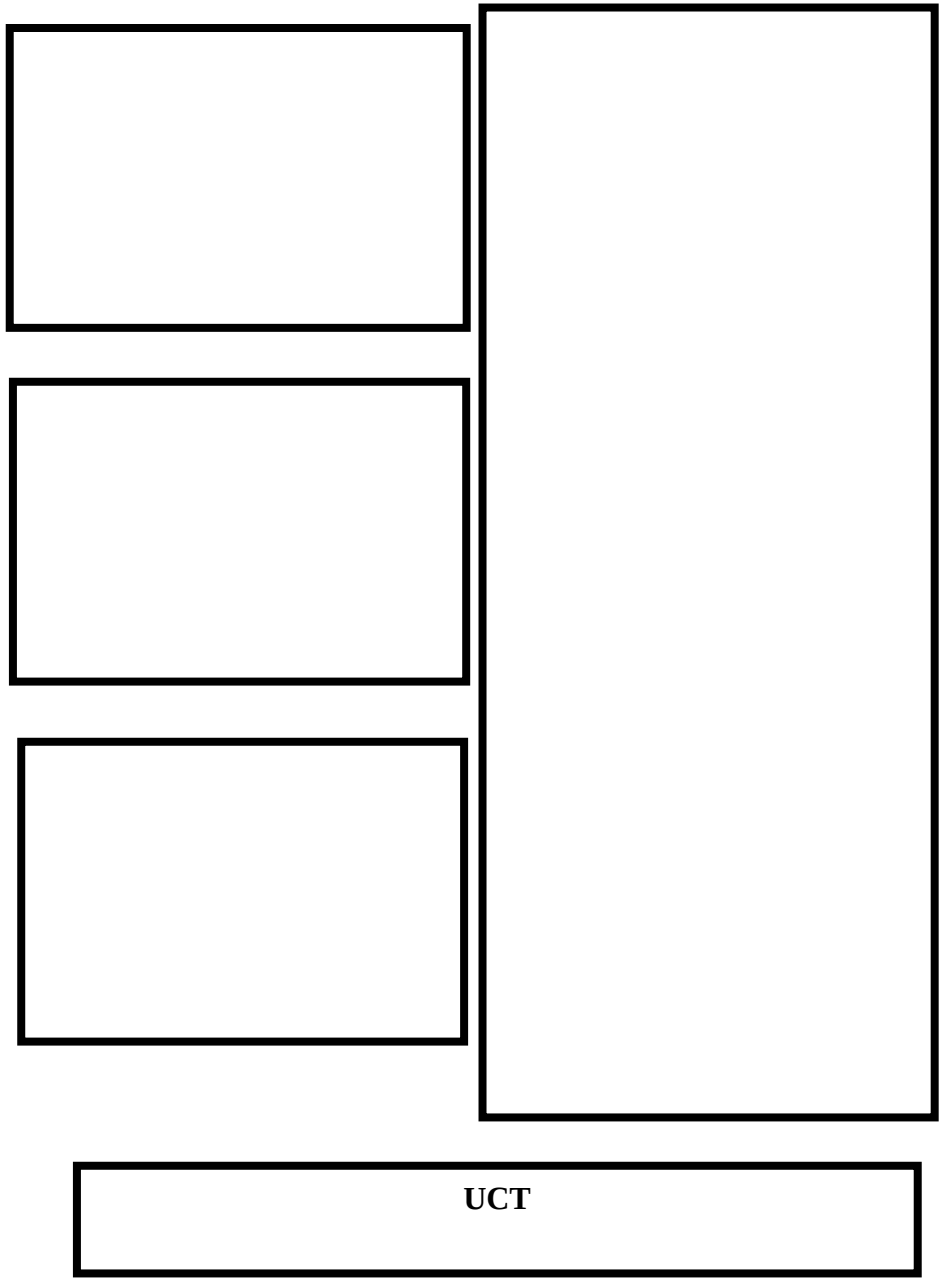
```
public class Section11
{
   public static void main(String [] args)
   {
     Étudiant étudiantA; // variable de référence
     Étudiant moiAussi;  // autre variable de réf.
     Étudiant étudiantB; // une 3ième variable de réf
               •
               •
     étudiantA = new Étudiant(1234567,60.0,80.0,true);
     moiAussi = new Étudiant(7654321,true);
     étudiantB = étudiantA;
               •
               •
               •
   }
}

class Étudiant
{
   // ... Les champs seraient définis ici ...
   public Étudiant(int n, double ePartiel,
                   double eFinal, boolean crédit)
   {
      numÉtud = n;
      examPartiel = ePartiel;
      examFinal = eFinal;
      créditable = crédit;
   }
   public Étudiant(int n, boolean crédit )
   {
      numÉtud = n;
      examPartiel = 0.0;    // une valeur « sûre »
      examFinal = 0.0;      // une valeur « sûre »
      créditable = crédit;
   }
   // ... Autres méthodes ...
```



```
public class Student
{
   // Attributes
   public int numÉtud;
   public double examPartiel;
   public double examFinal;
   public boolean créditable;
   private double [] devoirs;
   // Methods
   private double calcMoyDevoirs()
   {
      double sommeDevoirs; //INT: somme devoirs
      int nbDevoirs = 5; //INT CONSTANTE: #devoir
      int index; // INT: index pour boucle
      double moyDevoirs; // RÉS: Moyenne devoirs
      sommeDevoirs = 0.0;
      for ( index = 0; index < nbDevoirs;
            index = index + 1 )
      {
         sommeDevoirs = sommeDevoirs +
                        getDevoirs( index );
      }
      moyDevoirs = sommeDevoirs /
                   (double) nbDevoirs;
      return moyDevoirs;
   }

   public double getNoteFinale()
   {
      double moyDevoirs; //INT: Moyenne des devs
      double noteFinale; //RÉS: Note finale
      moyDevoirs = calcMoyDevoirs( );
      noteFinale =   0.55 *  getExamFinal( )
                   + 0.20 * getExamPartiel( )
                   + 0.25 * moyDevoirs;
      return noteFinale;
   }
} // fin de la classe Étudiant
```



UCT

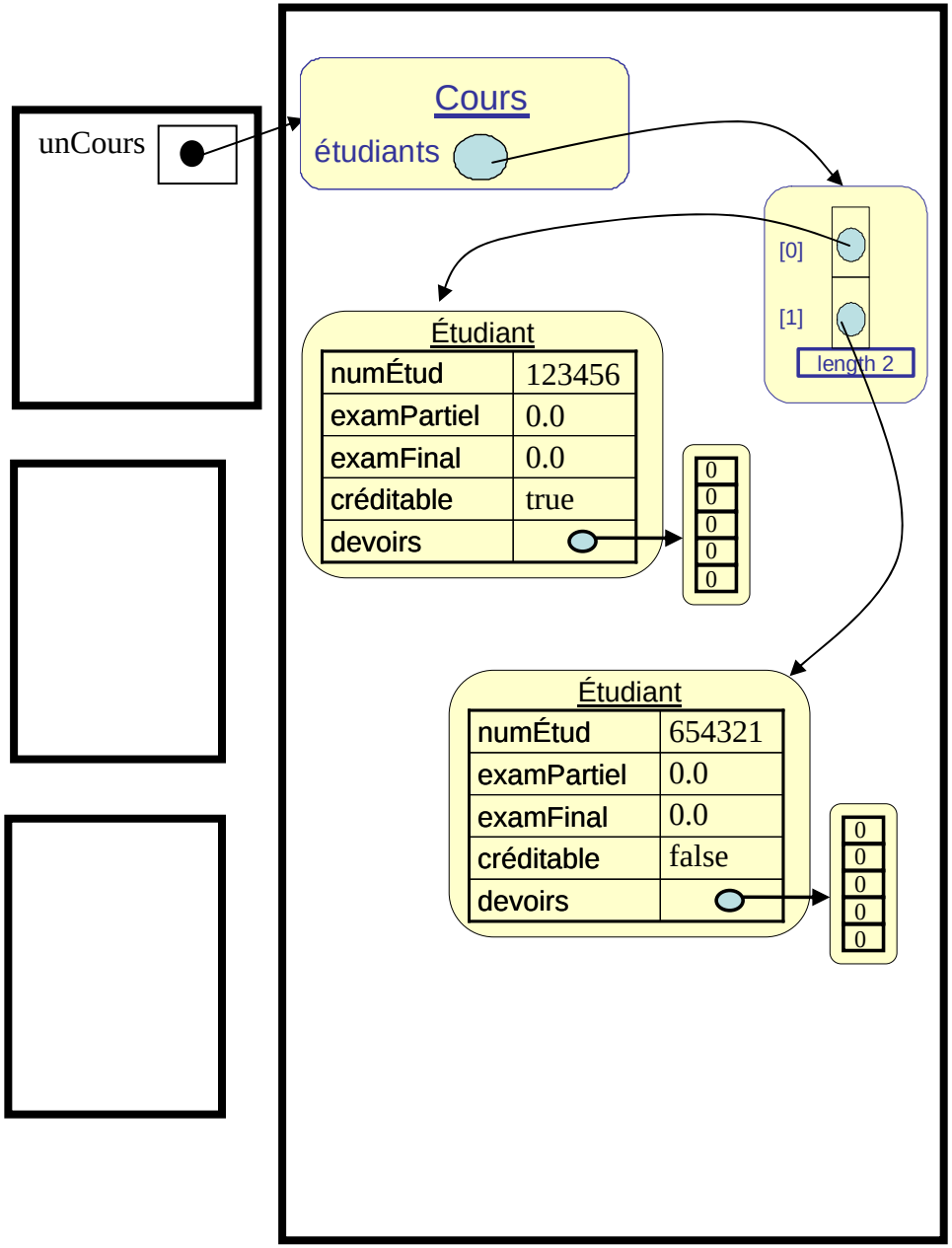
Mémoire de programme **Exercice 11-3 – Tableau d'objets** **Mémoire de travail** **Mémoire globale**

```
public class Section11
{
   public static void main(String [] args)
   {
     Cours unCours;
     unCours = new Cours("ITI1520",
                         "Intro. à l'info.");
     unCours.ajoutÉtudiant(123456,true);
     unCours.ajoutÉtudiant(654321,false);
   }
}
```

Cours

– cote : String
– titre : String
– étudiants: Etudiant [ ]
…

+ Cours(cote : String, titre : String)
+ ajoutÉtudiant(numÉtud : int, créditable : boolean)



UCT

Mémoire de programme **Exercice 11-4 – Utilisation des variables et méthodes de classe** **Mémoire globale**

```
public class Section11
{
   public static void main(String [] args)
   {
     int numDev;
     Étudiant étudiantA; // variable de référence
     Étudiant moiAussi;  // autre variable de réf.
     étudiantA = new Étudiant(1234567,60.0,79.0,true);
     moiAussi = new Étudiant(7654321,54.5,83.4,true);
     for(numDev=0 ; numDev<5 , numDev=i+1)
     {
        étudiantA.setDevoirs(numDev, 60.0);
        moiAussi.setDevoirs(numDev, 65.0);
     }
     System.out.println("La note pour étudiant "
                        +étudiantA.getNumÉtud()+" est "
                        + étudiantA.getNoteFinale();
     Étudiant.setPoidsPartiel(0.30);
     Étudiant.setPoidsDevoirs(0.15);
     System.out.println("La note pour étudiant "
                        +moiAussi.getNumÉtud()+" est "+
                         moiAussi.getNoteFinale();
     System.out.println("La note pour étudiant "
                        +étudiantA.getNumÉtud()+" est "
                        + étudiantA.getNoteFinale();
   }
}
```

Mémoire de travail

étudiantA

moiAussi

Étudiant (variables de classe)

poidsDevoirs	~~0.25~~ 0.15
poidsPartiel	~~0.20~~ 0.30
poidsFinal	0.55

Étudiant

numÉtud	1234567
examPartiel	60.0
examFinal	79.0
créditable	true
devoirs	

80 80 80 80 80

Étudiant

numÉtud	654321
examPartiel	54.5
examFinal	83.4
créditable	true
devoirs	

65 65 65 65 65

Fenêtre de console

La note pour étudiant 1234567 est 75.45
La note pour étudiant 7654321 est 71.97
La note pour étudiant 1234567 est 73.45

(80*0.25)+(60*0.20)+(79*0.55)

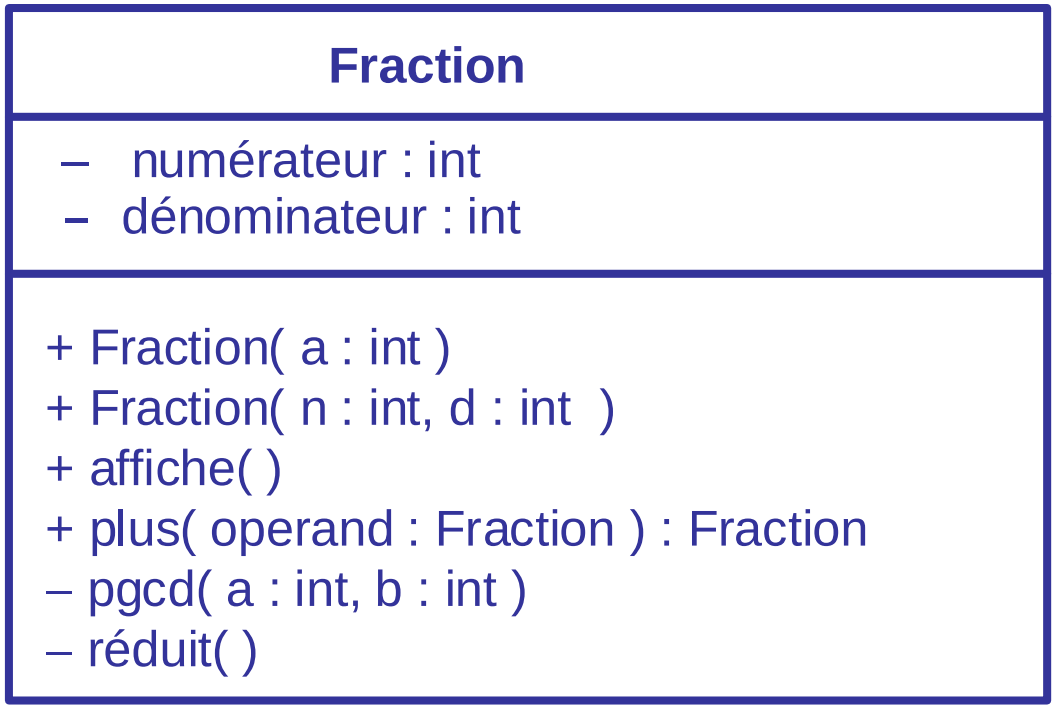
(65*0.15)+(54.5*0.30)+(83.4*0.55)

(80*0.15)+(60*0.30)+(79*0.55)

UCT

Exercice 11-5 – Conception de la classe Fraction

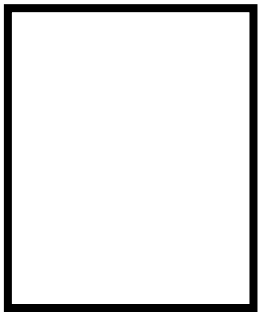
- Quelle information avons-nous besoin d'emmagasiner dans une Fraction?
  - **Numérateur (entier)**
  - **Dénominateur (entier > 0)**
- De quelles opérations avons-nous besoin?
  - Nous nous limiterons à l'addition comme opération mathématique
  - **Création (avec numérateur, avec numérateur et dénominateur)**
  - **Somme**
  - **Affichage**
  - **PGCD (méthode de support)**
  - **Réduction (méthode de support)**

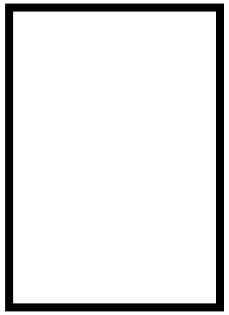


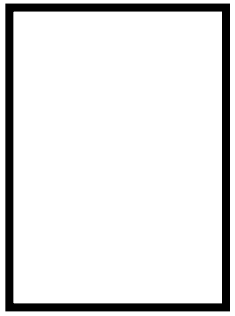
Mémoire de programme **Exercice 11-6 – Réduction d'une Fraction** **Mémoire globale**

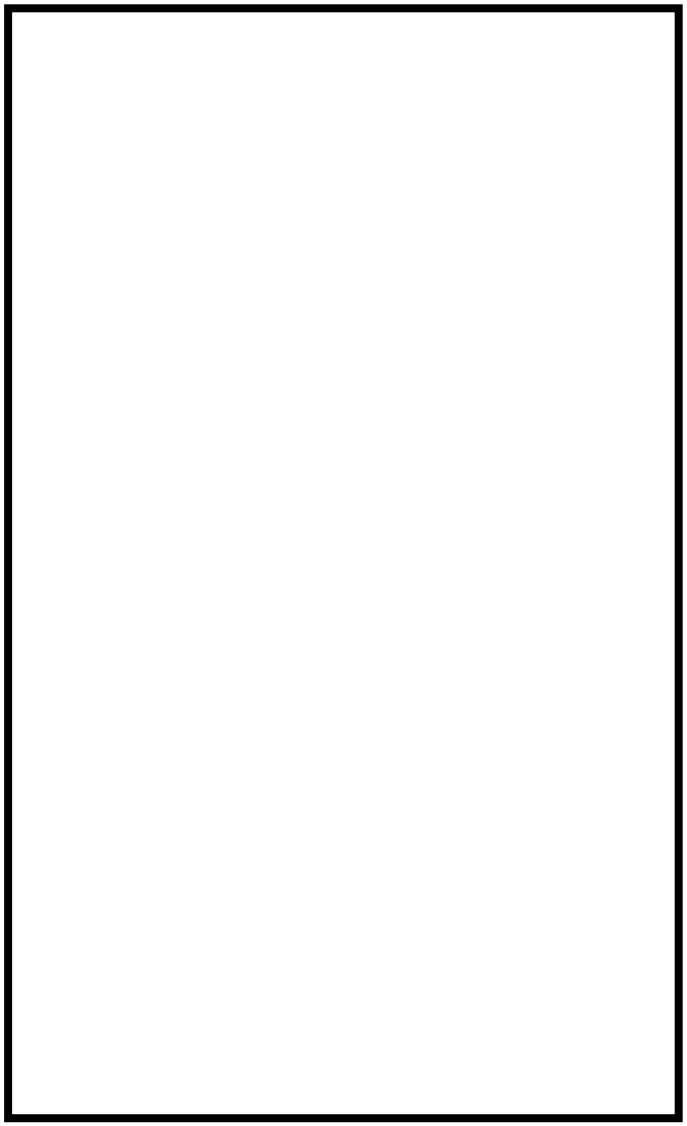
```
// méthode d'instance
private void réduit( )
{
  int r = pgcd(numérateur,
               dénominateur);
  if (r != 0)
  {
      numérateur = numérateur / r;
      dénominateur = dénominateur / r;
  }
  else
  {
    /* ne rien faire */ ;
  }
  if (dénominateur < 0)
   {
     numérateur = -numérateur;
     dénominateur = -dénominateur;
   }
   else
   {
     /* ne rien faire */ ;
   }
}
```

Mémoire de travail







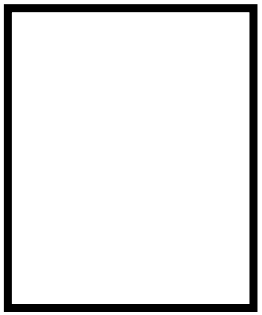


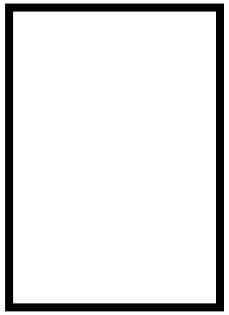
UCT

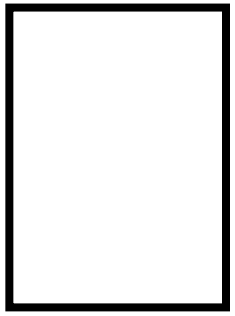
Mémoire de programme **Exercice 11-7 – Méthode pour PGCD** **Mémoire globale**

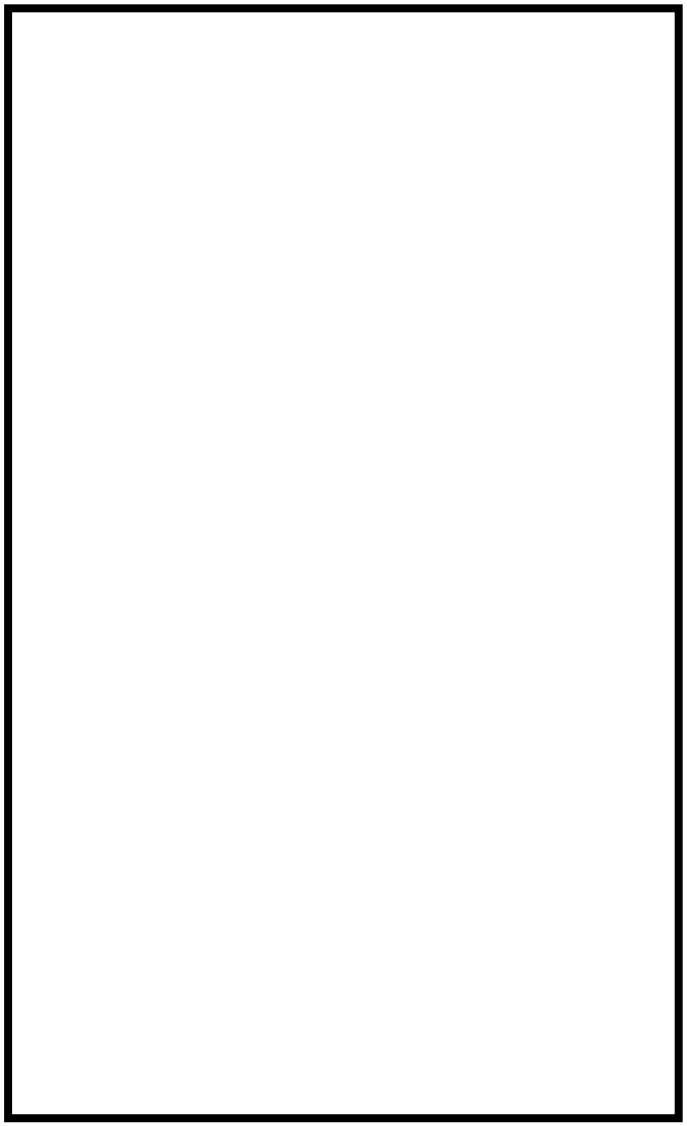
```
// méthode de classe
private static int pgcd (int a, int b)
{
   int résultat;
   int restant;
   if(b == 0)
   {
     restant = 0;
   }
   else
   {
     restant = a % b;
   }
   if (restant == 0)
   {
      résultat =  b;
   }
   else
   {  // appel récursif
      résultat = pgcd(b, restant);
   }
} // notez que pgcd(0, 0) == 0
```

Mémoire de travail









Mémoire de programme

Mémoire globale

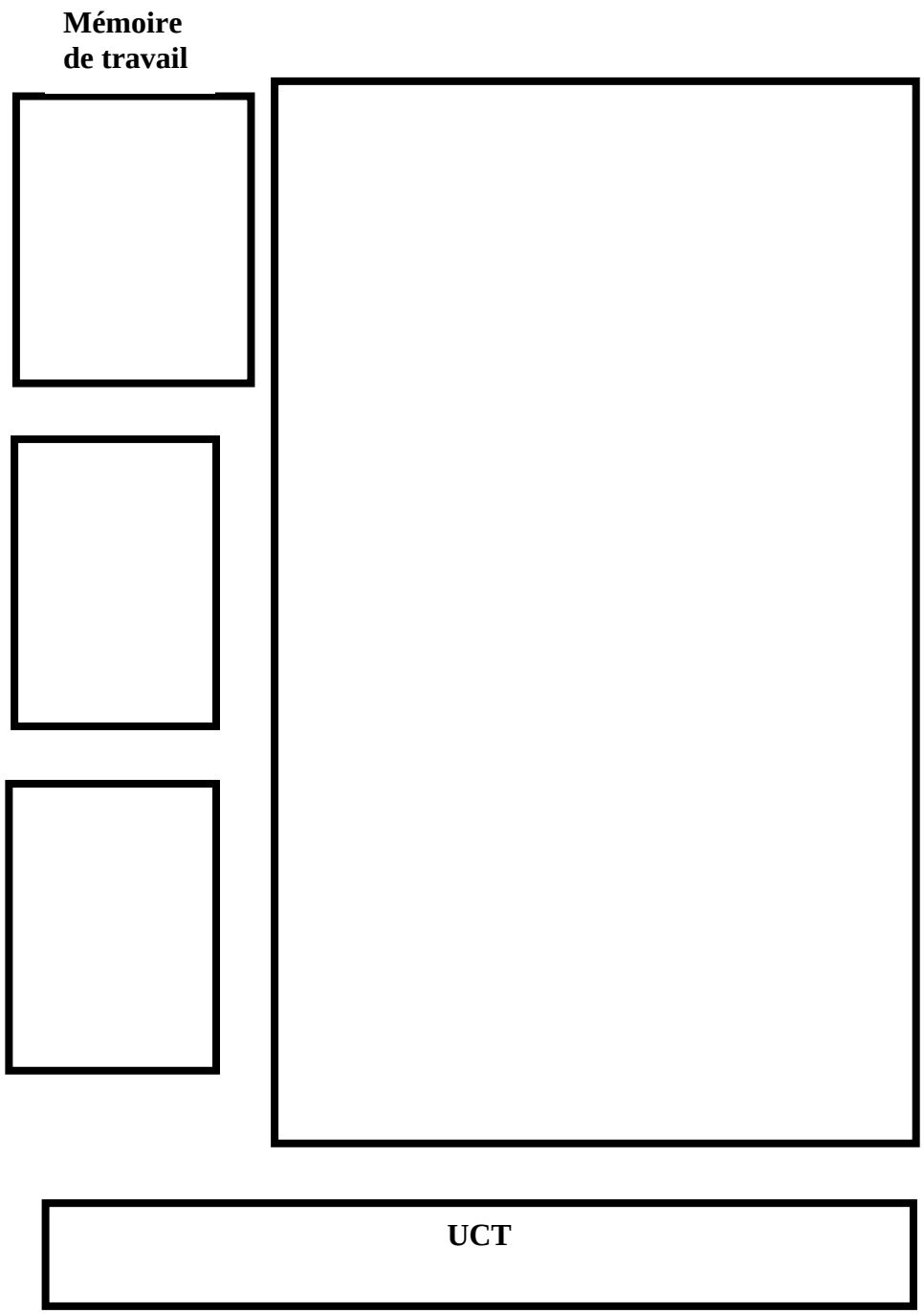
Exercice 11-8 – Constructeurs de Fraction

```
public Fraction(int n, int d)
{
       numérateur = n;
       dénominateur = d;
       réduit( );
}

public Fraction(int entier)
{
       numérateur = entier;
       dénominateur = 1;
}
```

Exercice 11-9 Affichage des Fractions

```
public void affiche( )
{
   if (dénominateur != 1)
   {
      System.out.print(numérateur + "/"
                       + dénominateur);
   }
   else
   {
      System.out.print(numérateur);
                      // nombre entier
   }
}
```



Mémoire de programme **Mémoire globale**

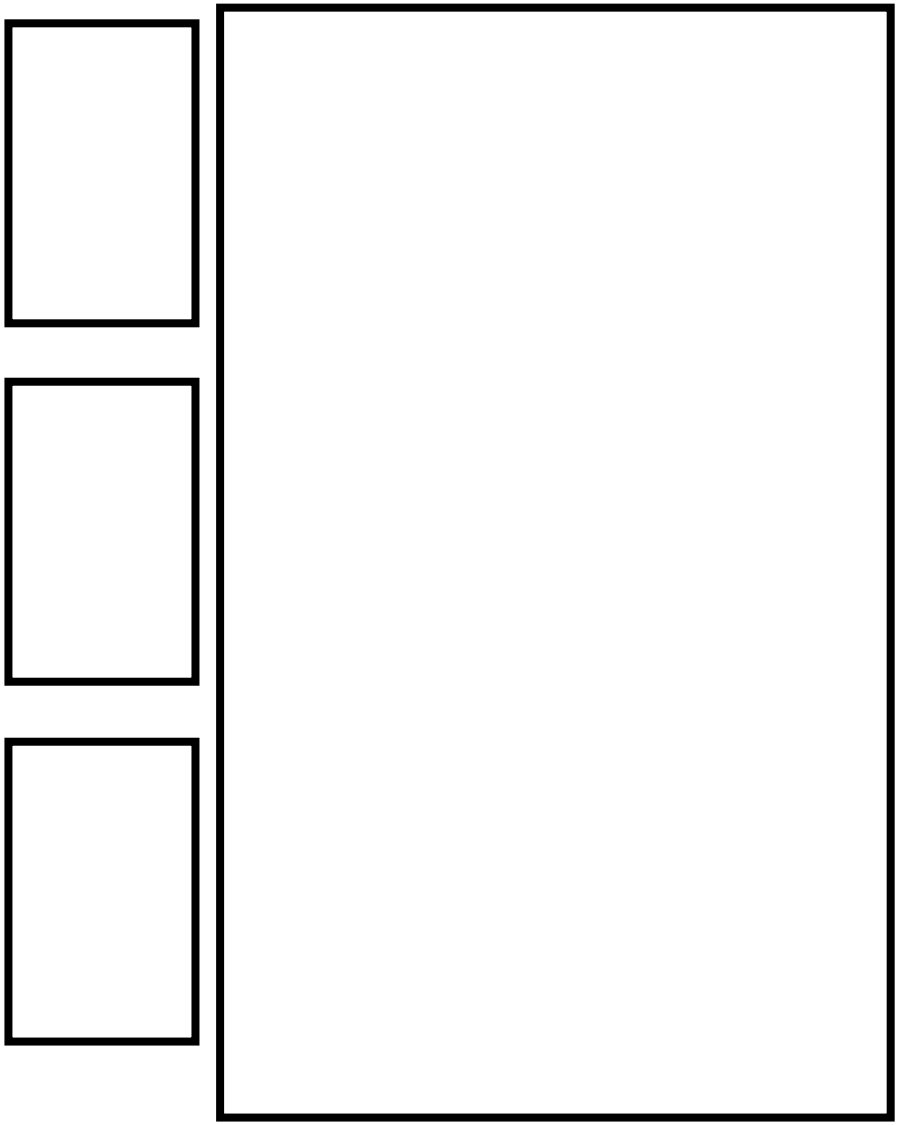
Exercice 11-10 – La somme de Fractions

```
public Fraction plus(Fraction opérande)
{
   int n = numérateur * opérande.dénominateur
           + dénominateur * opérande.numérateur;
   int d = dénominateur * opérande.dénominateur;
   return new Fraction(n, d); // Réduit automat.!
}
```

Exercice 11-11: Ajouter un entier à une fraction

```
public Fraction plus(int entier)
{
     return this.plus(new Fraction(entier));
    // this est optionnel ici
}
```

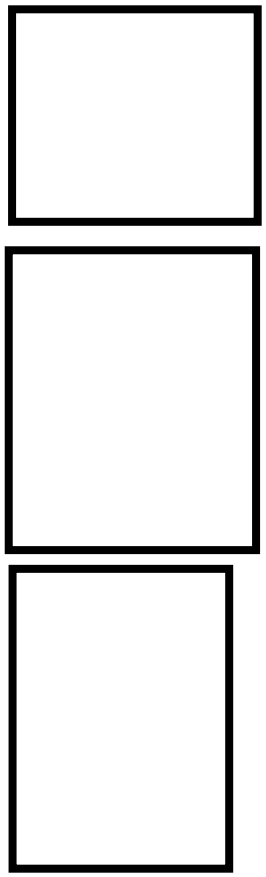
Mémoire de travail

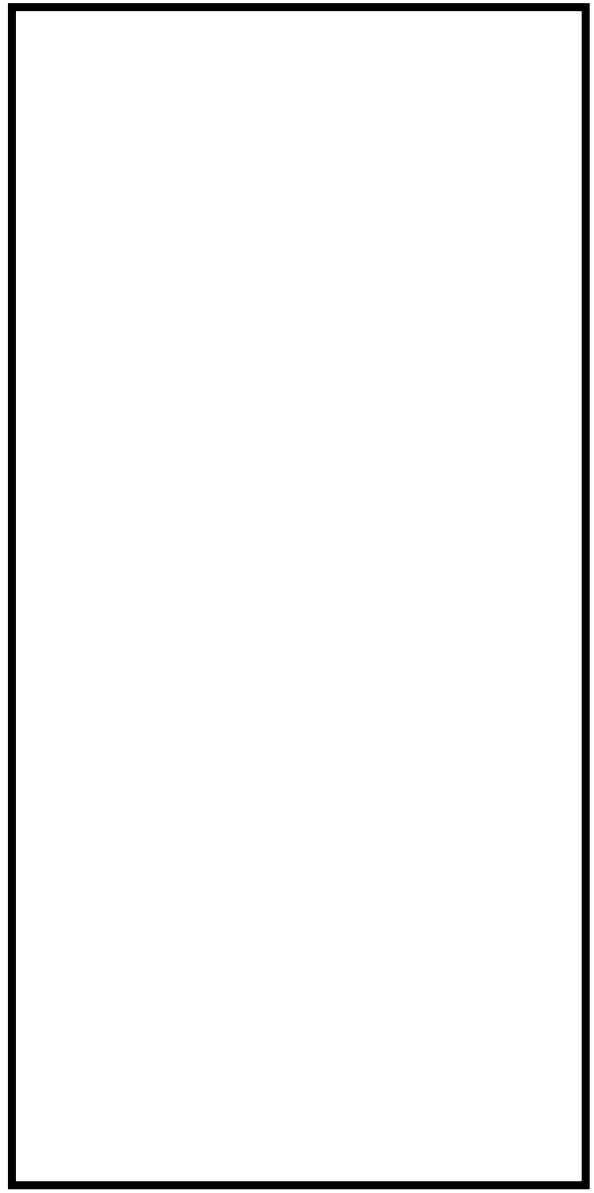


Mémoire de programme **Exercice- L'utilisation de Fraction** **Mémoire globale**

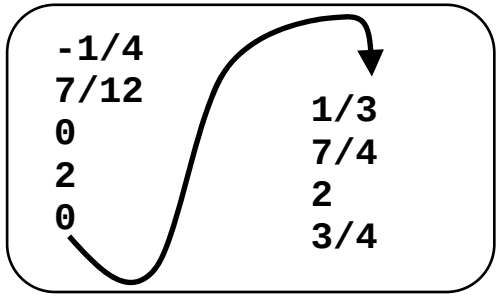
```
// Cette classe teste notre réalisation de la classe Fraction
public class TestFraction
{
  public static void main(String [ ] args)
  {
    // teste les constructeurs et l'affichage
    Fraction a = new Fraction (1, -4);
    Fraction b = new Fraction (-14, -24);
    Fraction c = new Fraction (0, -3);
    Fraction d = new Fraction (2);
    Fraction e = new Fraction (0);
    int i = 2;
    a.affiche( ); System.out.println( ); //-1/4
    b.affiche( ); System.out.println( ); //7/12
    c.affiche( ); System.out.println( ); //0
    d.affiche( ); System.out.println( ); //2
    e.affiche( ); System.out.println( ); //0
    // teste somme
    a.plus(b).affiche( ); System.out.println(); //1/3
    a.plus(d).affiche( ); System.out.println(); //7/4
    c.plus(d).affiche( ); System.out.println(); //2
    e=Fraction.somme(-1, a.plus(i));
    e.affiche( ); System.out.println( );       //3/4
  }
}
```

Mémoire de travail





Fenêtre console



UCT