Exercice 1

```
/**
 *
 */
public void dft(Image image, Image module, Image phase) {
    int N = image.width;
    int M = image.height;
    double fRe = 0;
    double fIm = 0;
    double valPha = 0.0;
    for (int u = 0; u < N; u++) {
        for (int v = 0; v < M; v++) {
            fRe = 0;
            fIm = 0;
            for (int x = 0; x < N; x++) {
                for (int y = 0; y < M; y++) {
                    fRe += (double) (image.getValue(x, y)) * Math.cos(
                            2.0 * Math.PI * ((double) (x * u) / (double) (N) + (double) (y * v) /
(double) (M)));
                    fIm += (double) (image.getValue(x, y)) * Math.sin(
                            2.0 * Math.PI * ((double) (x * u) / (double) (N) + (double) (y * v) /
(double) (M)));
                }
            }
            fIm = 0 - fIm;
            valPha = Math.atan((double) (fRe / fIm));
            module.setValue(u, v, (int) Math.sqrt((Math.pow(fRe, 2) + Math.pow(fIm,
2))));
            phase.setValue(u, v, (int) valPha);
        }
    }
}
```

Exercice 2

```
public Image getReImage() {
    int N = this.width;
    int M = this.height;
    Image ret = new Image(N,M);
    double fRe = 0;
    for (int u = 0; u < N; u++) {
        for (int v = 0; v < M; v++) {
            fRe = 0;
            for (int x = 0; x < N; x++) {
                for (int y = 0; y < M; y++) {
                    fRe += (double) (this.getValue(x, y)) * Math.cos(
                            2.0 * Math.PI * ((double) (x * u) / (double) (N) + (double) (y * v) /
(double) (M)));
                }
            }
            ret.setValue(u, v, (int)fRe);
        }
    }
    return ret;
}

public Image getImImage() {
    int N = this.width;
    int M = this.height;
    Image ret = new Image(N,M);
    double fIm = 0;
    double valPha = 0.0;
    for (int u = 0; u < N; u++) {
        for (int v = 0; v < M; v++) {
```

```
            fIm = 0;
            for (int x = 0; x < N; x++) {
               for (int y = 0; y < M; y++) {
                  fIm += (double) (this.getValue(x, y)) * Math.sin(
                     2.0 * Math.PI * ((double) (x * u) / (double) (N) + (double) (y * v) /
(double) (M)));
               }
            }
            ret.setValue(u, v, (int)-fIm);
         }
      }
      return ret;
   }
```

Exercice 3

```
   public Image dwtHaar(int n){
      Image ret = new Image(this);
      for(int i=0; i<n; i++){
         Image ilines = new Image(this.getWidth(), this.getHeight());

         for(int y=0; y<this.height/(i+1); y++){
            for (int k=0; k<(int)(this.width/2)/(i+1)-1; k++){
               int s = (int)((ret.getValue(2*k, y) + ret.getValue(2*k+1, y))/2);
               int d = (int)((ret.getValue(2*k, y) - ret.getValue(2*k+1, y))/2);
               ilines.setValue(k, y, s);
               ilines.setValue((int)(this.width/2)/(i+1)+k, y, d);
            }
         }

         for(int k=0; k<(int)(this.height/2)/(i+1)-1; k++){
            for (int x=0; x<this.width/(i+1); x++){
               int s = (int)((ilines.getValue(x,2*k) + ilines.getValue(x, 2*k+1))/2);
               int d = (int)((ilines.getValue(x, 2*k) - ilines.getValue(x, 2*k+1))/2);
               ret.setValue(x, k, s);
               ret.setValue(x, (int)(this.height/2)/(i+1)+k, d);
            }
         }
      }
      return ret;
   }
```