

```
abstract class Ustensile {  
    private int anneeFabrication;  
  
    public Ustensile(int anneeFabrication) {  
        this.anneeFabrication = anneeFabrication;  
    }  
  
    public int getAnneeFabrication() {  
        return anneeFabrication;  
    }  
  
    public void setAnneeFabrication(int anneeFabrication) {  
        this.anneeFabrication = anneeFabrication;  
    }  
  
    public abstract double calculerValeur(int anneeActuelle);  
}  
  
abstract class Assiette extends Ustensile {  
    public Assiette(int anneeFabrication) {  
        super(anneeFabrication);  
    }  
  
    public abstract double calculerSurface();  
}  
  
class AssietteRonde extends Assiette {  
    private double rayon;  
  
    public AssietteRonde(int anneeFabrication, double rayon) {  
        super(anneeFabrication);  
    }  
}
```

```
        this.rayon = rayon;
    }
```

```
public double getRayon() {
    return rayon;
}
```

```
public void setRayon(double rayon) {
    this.rayon = rayon;
}
```

```
@Override
public double calculerSurface() {
    return 3.14 * rayon * rayon;
}
```

```
@Override
public double calculerValeur(int anneeActuelle) {
    int age = anneeActuelle - getAnneeFabrication();
    return (age > 50) ? (age - 50) : 0;
}
}
```

```
class AssietteCarree extends Assiette {
    private double cote;

    public AssietteCarree(int anneeFabrication, double cote) {
        super(anneeFabrication);
        this.cote = cote;
    }
}
```

```
public double getCote() {  
    return cote;  
}
```

```
public void setCote(double cote) {  
    this.cote = cote;  
}
```

```
@Override  
public double calculerSurface() {  
    return cote * cote;  
}
```

```
@Override  
public double calculerValeur(int anneeActuelle) {  
    int age = anneeActuelle - getAnneeFabrication();  
    double valeurRonde = (age > 50) ? (age - 50) : 0;  
    return valeurRonde * 5;  
}  
}
```

```
class Cuillere extends Ustensile {  
    private double longueur;  
  
    public Cuillere(int anneeFabrication, double longueur) {  
        super(anneeFabrication);  
        this.longueur = longueur;  
    }
```

```
public double getLongueur() {  
    return longueur;  
}
```

```

    }

    public void setLongueur(double longueur) {
        this.longueur = longueur;
    }

    @Override
    public double calculerValeur(int anneeActuelle) {
        int age = anneeActuelle - getAnneeFabrication();
        return (age > 50) ? (age - 50) : 0;
    }
}

```

Main

```

public class Collection {

    public static void main(String[] args) {
        int anneeActuelle = 2024;

        // Création des objets
        Ustensile[] us = new Ustensile[5];
        us[0] = new AssietteRonde(1943, 10);
        us[1] = new AssietteCarree(1943, 8);
        us[2] = new Cuillere(1950, 15);
        us[3] = new AssietteRonde(2000, 5);
        us[4] = new Cuillere(1960, 12);

        afficherCuilleres(us);
        afficherSurfaceAssiettes(us);
    }
}

```

```

        afficherValeurTotale(us, anneeActuelle);
    }

    public static void afficherCuilleres(Ustensile[] us) {
        int count = 0;
        for (Ustensile u : us) {
            if (u instanceof Cuillere) {
                count++;
            }
        }
        System.out.println("Il y a " + count + " cuillères.");
    }

    public static void afficherSurfaceAssiettes(Ustensile[] us) {
        double surfaceTotale = 0;
        for (Ustensile u : us) {
            if (u instanceof Assiette) {
                surfaceTotale += ((Assiette) u).calculerSurface();
            }
        }
        System.out.println("Surface totale des assiettes : " + surfaceTotale);
    }

    public static void afficherValeurTotale(Ustensile[] us, int anneeActuelle) {
        double valeurTotale = 0;
        for (Ustensile u : us) {
            valeurTotale += u.calculerValeur(anneeActuelle);
        }
        System.out.println("Valeur totale de la collection : " + valeurTotale);
    }
}

```