

```

public 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();
}

```

```

public 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) {
        return anneeActuelle - getAnneeFabrication()-50;
    }
}

```

```

public 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 Math.PI * rayon * rayon;
    }

    @Override
    public double calculerValeur(int anneeActuelle) {
        return anneeActuelle - getAnneeFabrication()-50;
    }
}

```

```

public 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) {
        return anneeActuelle - getAnneeFabrication()-50;
    }
}

```

```

public class Collection {
    public static void main(String[] args) {

        Ustensile[] ustensiles = new Ustensile[5];
        ustensiles[0] = new AssietteRonde(1970, 15.0);
        ustensiles[1] = new AssietteCarree(1980, 30.0);
        ustensiles[2] = new Cuillere(1960, 25.0);
        ustensiles[3] = new AssietteRonde(2000, 30.0);
        ustensiles[4] = new Cuillere(1950, 30.0);

        int anneeActuelle = 2024;

        afficherSurfaceAssiettes(ustensiles);
    }
}

```

```

        afficherCuilleres(ustensiles);

        for (Ustensile u : ustensiles) {
            System.out.println("Valeur de l'ustensile : " +
u.calculerValeur(anneeActuelle));
        }
    }

    public static void afficherCuilleres(Ustensile[] ustensiles) {
        int comp = 0;
        for (Ustensile u : ustensiles) {
            if (u instanceof Cuillere) {
                comp++;
            }
        }
        System.out.println("Le nombre de cuillères est: " + comp);
    }

    public static void afficherSurfaceAssiettes(Ustensile[] ustensiles) {
        for (Ustensile u : ustensiles) {
            if (u instanceof Assiette) {
                Assiette a = (Assiette) u;
                System.out.println("Surface de l'assiette : " +
a.calculerSurface());
            }
        }
    }
}

```