

tp devObj 1

Aujourd'hui le programme du tp et de voir comment créer un code Java et un UML à partir d'une phrase

Exercice 1

code de la classe utilisateur

```
public class utilisateur {
    public void appuyer (){
        interuteur interuteur = new interuteur()
        interuteur.appuyer();
    }
}
```

code de la classe interuteur :

```
public class interuteur {
    private boolean position = false;

    public interuteur (){
        this.position = false;
    }
    public void appuyer() {
        if (this.position = false) {
            this.position = true ;
            resau resau = new resau();
            resau.cirucler();
        }
        else {
            this.position = false ;
            resau resau = new resau();
            resau.couper();
        }
    }
}
```

code de la class resau

```
public class resau extends Lampe {
    private boolean courant = false ;

    public void cirucler() {
        this.courant = true;
        Lampe lampe = new Lampe();
        lampe.allumer();
    }
    public void couper(){
        this.courant = false;
        Lampe lampe = new Lampe();
        lampe.ettindre();
    }
}
```

code de la classe lampe :

```
public class Lampe {  
    private boolean power = false;  
  
    public void allumer () {  
        this.power = true;  
    }  
  
    public void etteindre () {  
        this.power = false;  
    }  
}
```

code de la classe main :

```
public class main {  
  
    public static void main(String[] args) {  
        utilisateur utilisateur = new utilisateur();  
        utilisateur.appuyer();  
    }  
}
```

UML de la lampe :  Exercice 2 : a)

main

```
public class main {  
    public static void main(String[] args) {  
        developpeur jean = new developpeur("jean");  
        nourriture n = new nourriture();  
        jean.manger(n);  
        n.produire("burger");  
        jean.manger(n);  
        System.out.println(jean.language);  
    }  
}
```

Personne :

```
public class Personne {  
    private String nom = "";  
  
    public Personne(String nom) {  
        this.nom = nom;  
    }  
  
    public void manger(nourriture n) {  
        System.out.println(this.nom + " mange " + n.alliment);  
    }  
    public void program() {  
        System.out.println(this.nom + " developpe en " + this);  
    }  
}
```

Developpeur

```

public class developpeur extends Personne{
    public String language ="java";

    public developpeur(String nom) {
        super(nom);
    }
    public void programmer(String langue){
        this.language = langue;
    }
}

```

Nouriture:

```

public class nourriture {
    public String alliment ="pizza";

    public void produire(String alli){
        this.alliment = alli;
    }
}

```

diagramme jean pizza ☐ b) code de la question 2 Philosophe :

```

public class Philosophe extends  Personne{
    public String dicide;

    public Philosophe(String nom){
        super(nom);
    }

    public void disciple(String q){
        if (this.dicide.length() > 0)
            this.dicide = ", "+q;
        else {
            this.dicide = q;
        }
    }
}

```

Personne :

```

public class Personne {
    public String nom ="";

    public Personne(String nom){
        this.nom = nom;
    }
}

```

Disciple :

```

public class Disciple extends Personne{
    public String maitre = "";

    public Disciple(String n){
        super(n);
    }

    public void master(String m){
        this.maitre = m;
    }
}

```

main :

```

public class main {
    public static void main(String[] args) {
        Disciple platon = new Disciple("platon");
        Philosophe socrate = new Philosophe("socrate");
        Disciple Xenophon = new Disciple("Xenophon");
        Disciple aristote = new Disciple("aristote");
        Disciple toto = new Disciple("toto");
        platon.master(socrate.nom);
        socrate.disciple(platon.nom);
        Xenophon.master(socrate.nom);
        socrate.disciple(Xenophon.nom);
        aristote.master(platon.nom);
        toto.master(platon.nom);
        System.out.println(platon.nom+" est un disciple de " + platon.maitre + ", "
            + Xenophon.nom + " est un disciple de " + Xenophon.maitre + ", "
            + aristote.nom + " est un disciple de " + aristote.maitre + ", "
            + toto.nom + "est un disciple de " + toto.maitre);
    }
}

```

uml philosophe

ce tp maura donc appris comment me servir de intellij, els base de ce logielle, et comment passer de phrase simple a un uml et coder ce derniere.