

TP - JAVA AVANCE

source de code :

TP-EXCEPTION :

https://gitlab.com/JaitiMohammed/java_advanced/-/tree/master/TP%20Exception

TP-JDBC :

https://github.com/JaitiMohammed/Gestion_cabinet_medical

TP-EXCEPTION :

Exercice 1 :



new commit

Jaiti Mohammed authored 17 hours ago

EntNaT.java 237 Bytes

```
1
2 public class EntNaT {
3
4     private int value;
5     public EntNaT(int value) throws ErrConst{
6
7         if(value < 0) {
8             throw new ErrConst("la valeur ne conviendra pas");
9         }
10        this.value=value;
11
12    }
13
14    public int getN() {
15        return value;
16    }
17
18 }
```



new commit

Jaiti Mohammed authored 17 hours ago



ErrConst.java 102 Bytes



```
1
2 public class ErrConst extends Exception{
3
4     public ErrConst(String message) {
5         super(message);
6     }
7
8 }
```



new commit

Jaiti Mohammed authored 17 hours ago

Test.java 397 Bytes

```
1
2 public class Test {
3
4     public Test() {
5         // TODO Auto-generated constructor stub
6     }
7
8     public static void main(String[] args) {
9         try {
10             EntNaT n1 = new EntNaT(20);
11             System.out.println("n1 = " + n1.getN());
12             EntNaT n2 = new EntNaT(-30);
13             System.out.println("n2 = " + n2.getN());
14
15
16         } catch (ErrConst e) {
17             System.out.println("Erreur construction");
18             System.exit(-1);
19         }
20
21
22     }
23
24
25 }
```

Exercice 2 :



new commit

Jaiti Mohammed authored 17 hours ago

 **EntNaT2.java** 217 Bytes 

```
1
2 public class EntNaT2 {
3
4     private int value;
5     public EntNaT2(int value) throws ErrConst2{
6
7         if(value < 0) {
8             throw new ErrConst2(value);
9         }
10        this.value=value;
11
12    }
13
14    public int getN() {
15        return value;
16    }
17
18 }
```



new commit

Jaiti Mohammed authored 17 hours ago

 **ErrConst2.java** 173 Bytes 

```
1
2 public class ErrConst2 extends Exception {
3     private int valeur;
4     public ErrConst2(int value) {
5         this.valeur=value;
6
7     }
8
9     public int getValeur() {
10        return valeur;
11    }
12 }
```



new commit

Jaiti Mohammed authored 17 hours ago



TestStart.java 353 Bytes



```
1 public class TestStart {  
2  
3     public static void main(String[] args) {  
4  
5         try {  
6             EntNaT2 n1 = new EntNaT2(20);  
7             EntNaT2 n2 = new EntNaT2(-20);  
8             System.out.println("n1 = " + n1.getN());  
9             System.out.println("n2 = " + n2.getN());  
10        } catch (ErrConst2 e) {  
11            System.out.println("Erreur de constructeur " + e.getValeur());  
12            System.exit(-1);  
13        }  
14    }  
15 }  
16 }
```

TP-JDBC

Description :

L'architecture de ce projet comprend : dao / application / entities se basant sur le design pattern MVC

dao contient :

- Interface qui englobe tout les méthodes nécessaires pour l'application (IMetier.java)
- Une classe qui met en pratique les méthodes déclarées dans l'interface IMetier (IMetierImpl.java)
- Une classe avec le design pattern Singleton qui permet de charger la connexion vers la base de donnée

application contient :

ce package comprend des classes de JavaFX :

- window : est la classe de start
- add : comprend des méthodes utilisés dans le stage avec différentes composants pratiques

entities :

ce package comprend les classes lié avec la base de donnée

- rv / creneaux
- clients / medecins / personne