

Compte rendu Getap

Creation classe main instanciant une dvctap

```
Classe classeSio22 = new Classe(1,"sio22");
```

instanciation d'une nouvelle classe sio22

```
User eleve1 = new User((long)
1,"Roy","Baptiste",classeSio22,"eleve","broy","broy","baptiste.r45@laposte.net");
User professeur1 = new User((long)
2,"Capuozzo","Olivier",classeSio22,"professeur","Ocapu","Ocapu","Ocapu@laposte.net
");
```

instanciation élève / professeur d'après la classé instanciée précédemment
déclarée par classeSio22 , problème de variable correspondant à l'id crée en
integer => cast en variable long correspondant au constructeur.

```
AccPersonalise accPerso = new AccPersonalise(1,"Salon du libre",1,(long) 1);
```

instanciation de l'accompagnement spécialisé

```
Date dateacc = Date.valueOf("07-10-2012");
```

```
DemandeValidationConsoTempsAccPers dvctap = new
DemandeValidationConsoTempsAccPers((long) 1, "2012-2013", dateacc, 20,
professeur1,accPerso,eleve1, 0);
```

Création d'une dctap

```
System.out.println(dvctap);
dvctap.setEtat(4);
System.out.println(dvctap);
```

```
dvctap.setEtat(32);
System.out.println(dvctap);
```

Différentes modification des états d'une dctap

Refactoring avec tests unitaires

```
public class DemandeValidationConsoTempsAccPers {
    Classe classeSio22 = new Classe(1,"sio22");

    User eleve1 = new User((long)
1,"Roy","Baptiste",classeSio22,"eleve","broy","broy","baptiste.r45@laposte.net");
    User professeur1 = new User((long)
2,"Capuozzo","Olivier",classeSio22,"professeur","Ocapu","Ocapu","Ocapu@laposte.net
");

    AccPersonalise accPerso = new AccPersonalise(1,"Salon du libre",1,(long) 1);

    Date dateacc = Date.valueOf("07-10-2012");

    DemandeValidationConsoTempsAccPers dvctap = new
DemandeValidationConsoTempsAccPers((long) 1, "2012-2013", dateacc, 20,
professeur1,accPerso,eleve1, 0);
```

on ré-instancie afin de réobtenir une dctap de test.

Dans cette partie ,nous mettons en place les différents tests unitaires:

```
@Test
public void testEtatInitial() {
    assertTrue("Etat initial", dvctap.isEtatInitial());
}
```

celui ci correspond a un état initial
définition d'un test unitaire via @Test si message renvoyé correspond
à la méthode isEtatinitial() définit.

```

@Test
    public void test() {
        try{
            dvctap.setEtat(4);
            assertTrue("Etat 4", dvctap.setEtat(4));
        } catch (DVCTAPEException e){
            try{
                dvctap.setEtat(1024);
                assertTrue("Etat 4", dvctap.setEtat(1024));
            } catch (DVCTAPEException e){
                try{
                    dvctap.setEtat(1024);
                    assertTrue("Etat 8", dvctap.setEtat(8));
                } catch (DVCTAPEException e)
                {}
            }
        }
    }
}

```

Ici utilisation de gestion des exceptions (try/catch) permettant de faire des tests.

try => opérations de test sur différents états en retournant via le assertTrue les possibles erreurs.

catch => dans le cas d'une erreur dans bloc try , possibilité d'émettre une exception pour la dvctap instanciée DVCTAPEException e .

Si le try n'est pas adéquate , la clause catch associée sera exécutée.