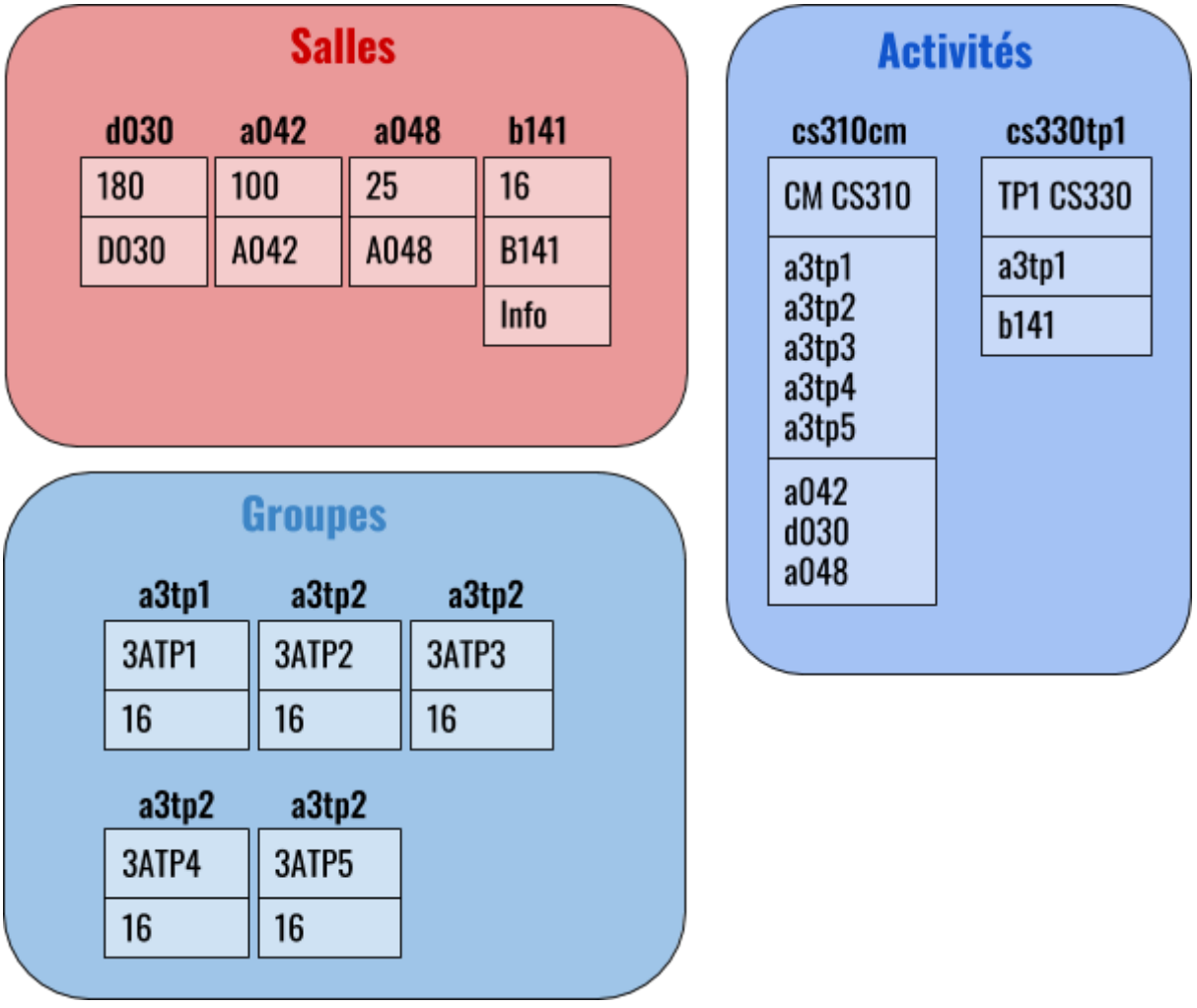
TP2 CS312 - Java

Adrian Bonnet et Gauthier Ros 04/03/2019

1. Inclusions et définition

Les Activités CM, TD et TP sont des classes définies dans les fichiers CM.java, TD.java et TP.java. On trouve dans ces définitions l'ajout des salles correspondant aux activités. D'après ces définitions, on peut affecter une salle de TP à une activité de TD, car il n'y a pas de vérification lors de la saisie.

2. Compréhension : salles, activités et groupes



3. Méthode `verifDuree()` dans `Creneau.java`

```
private boolean verifDuree()
{
      if (this.heure < 8 || this.heure > 19) return false;
      else return true;
}
```

4. Méthode `verifSalle()` dans `Creneau.java`

```
private boolean verifSalle()
{
	for (Salle s : activite.getSalles())
		if (s.equals(salle))
			return true;
	return false;
}
```

5. Méthode `verifCapacite()` dans `Creneau.java`

```
private boolean verifCapacite()
{
	int somme = 0;
	for (Groupe g : activite.getGroupes())
		somme = somme + g.getEffectif();
	if (salle.getCapacite() > somme)	return true;
	else return false;
}
```

6. Méthode `intersection()` dans `Creneau.java`

```
public boolean intersection(Creneau c)
{
	if (annee == c.annee && mois == c.mois && jour == c.jour)
	{
		if (((this.heure * 60 + this.miutes) < (c.heure * 60 +
c.miutes + c.duree)) && ((c.heure * 60 + c.miutes) < (this.heure *
60 + this.miutes + this.duree)))
		{
			if (this.salle == c.salle)		return false;
			else
				for (Groupe g1 : this.activite.groupes)
					for (Groupe g2 : c.activite.groupes)
						if (g1 == g2)
							return false;
		}
	}
	return true;
}
```

7. Méthode `planningGroupe (Creneau c)` dans `Planning.java`

```
public List<Creneau> planningGroupe(Groupe groupe)
{
      ArrayList<Creneau> liste = new ArrayList<Creneau>();
      for (Creneau c : edt)
            for (Groupe g : c.getActivite().getGroupes())
                  if (g == groupe)      liste.add(c);
      return liste;
}
```

8. Méthode `totalHeuresGroupe (Creneau c)` dans `Planning.java`

```
public float totalHeuresGroupe(Groupe groupe)
{
      int total_heures = 0;
      for (Creneau c : this.edt)
            for ( Groupe g : c.getActivite().getGroupes() )
                  if ( g == groupe )    total_heures += c.getDuree();
      return total_heures;
}
```

9. Exécution de la méthode `main` dans `Planning.java`

Lors de cette exécution, nous n'obtenons rien d'affiché à l'écran. Pourtant, la pré-compilation et l'exécution se passent sans erreur. Nous n'avons pas trouvé d'où venait ce léger contre-temps.

10. Disciplines pour les salles de TP

Les salles de TP possèdent l'attribut `Discipline`, qui peut être `Automatique`, `Informatique`, `Electronique` ou `Reseau`, définis dans `Discipline.java`. Ce n'est pas le cas des instances de la classe `TP`. Il suffirait donc, afin qu'un TP d'Automatique ne puisse pas se dérouler dans une salle d'Informatique, ajouter un attribut `Discipline` aux instances de la classe `TP`. Il faudrait alors, à l'ajout d'un créneau, vérifier, pour les TPs, que la discipline et la salle correspondent.

11. Héritage des groupes d'étudiants

Pour représenter plus fidèlement les instances de la classe `Groupe`, il faudrait créer des sous-classes `groupeTD` et groupeTP. Ces sous-classes hériteraient de la classe `Groupe`.

Lors de la vérification de la méthode `intersection()`, il faudrait vérifier qu'aucune des instances des deux sous-classes ne soit incompatible avec aucune autre instance des sous-classes ou classe de l'autre groupe.