

Exemple de code corrigé – Saly Ndiaye

Module Java 250

1) Présence de code commenté :

```
@RequestMapping(value="/{idTechnicien}/manager/add", params = "matricule", method = RequestMethod.GET)
public String addManager(@PathVariable(name="idTechnicien") Long id, @RequestParam(name="matricule") String matricule,
    Map<String, Object> model) {
    //On recherche le manager désiré
    //Manager manager = (Manager) employeService.findByMatricule(matricule);
    //On set le manager a ce technicien
    //technicien.setManager(manager);
    //On persiste dans la base cette nouvelle info
    //technicien = employeService.updateEmploye(id, technicien);
    //On renvoie dans la vue les données
    Technicien technicien = technicienService.addManager(id, matricule);
    model.put("employe",technicien);
    return "employes/detail";
}
```

Correction:

```
@RequestMapping(value="/{idTechnicien}/manager/add", params = "matricule", method = RequestMethod.GET)
public String addManager(@PathVariable(name="idTechnicien") Long id, @RequestParam(name="matricule") String matricule,
    Map<String, Object> model) {
    Technicien technicien = technicienService.addManager(id, matricule);
    model.put("employe",technicien);
    return "employes/detail";
}
```

2) Ne respecte pas les conventions d'écriture camelCase

```
@Override
public Integer getNbConges() {
    Integer CongesRestants = Entreprise.NB_CONGES_BASE - this.getNbCongesConsommées();
    return CongesRestants;
}
```

Correction :

```
@Override
public Integer getNbConges() {
    Integer congesRestants = Entreprise.NB_CONGES_BASE - this.getNbCongesConsommées();
    return congesRestants;
}
```

3) Distinctions non significatives

```
public interface CadreRepository extends CrudRepository<Employee,String> {

    String save(String employee, Long id);

    HashSet<Employee> findAll (Integer grade);
    HashSet<Employee> findAll(Sort sort);
    ListType<Employee> findAll(Double salaire);

    Employee save(String nom, String prenom, String matricule, LocalDate dateEmbauche, Double salaire);
}
```

Correction

```
public interface CadreRepository extends CrudRepository<Employee,String> {

    String save(String employee, Long id);
```

```

HashSet<Employee> findAllSortByGrade (Integer grade);
HashSet<Employee> findAllSortById(Sort sort);
ListType<Employee> findAllBySalaire(Double salaire);

```

```

Employee save(String nom, String prenom, String matricule, LocalDate dateEmbauche, Double salaire);
}

```

4) Fonction retournant null

```

public Note equivalenceNote(){
    if(performance == null){
        return null ;
    }

    switch (performance) {
        case 0:
        case 50:
            return Note.INSUFFISANT;
        case 100:
            return Note.PASSABLE;
        case 150:
            return Note.BIEN;
        case 200:
            return Note.TRES_BIEN;
        default:
            return null;
    }
}

```

Correction:

```

public Note equivalenceNote() throws Exception {
    if(performance == null){
        throw new Exception("La performance ne peut être nulle");
    }
}

```

```
switch (performance) {  
    case 0:  
    case 50:  
        return Note.INSUFFISANT;  
    case 100:  
        return Note.PASSABLE;  
    case 150:  
        return Note.BIEN;  
    case 200:  
        return Note.TRES_BIEN;  
    default:  
        return Note.NON_EVALUE;  
}  
}
```