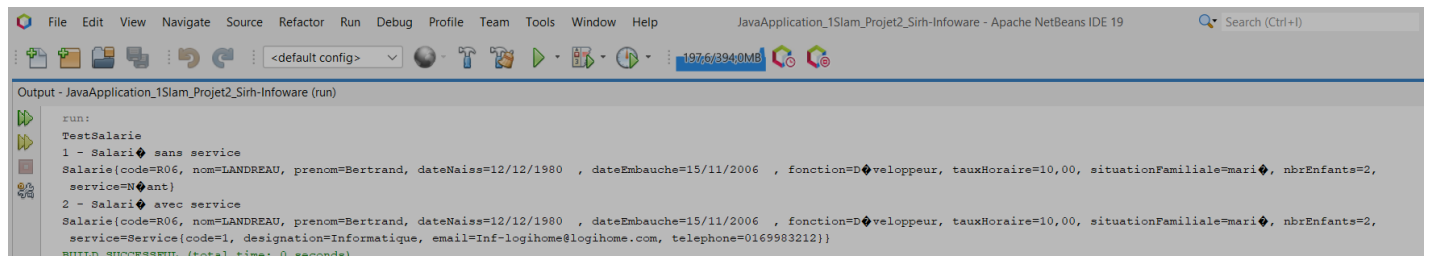


2023/2024	Projet JAVA – Etape 02
BTS SIO	Auteur : Loric Worms / Stefen Rialland
1SIOB - SLAM	Date de rédaction : 16/04/2024

Compte-rendu étape 02

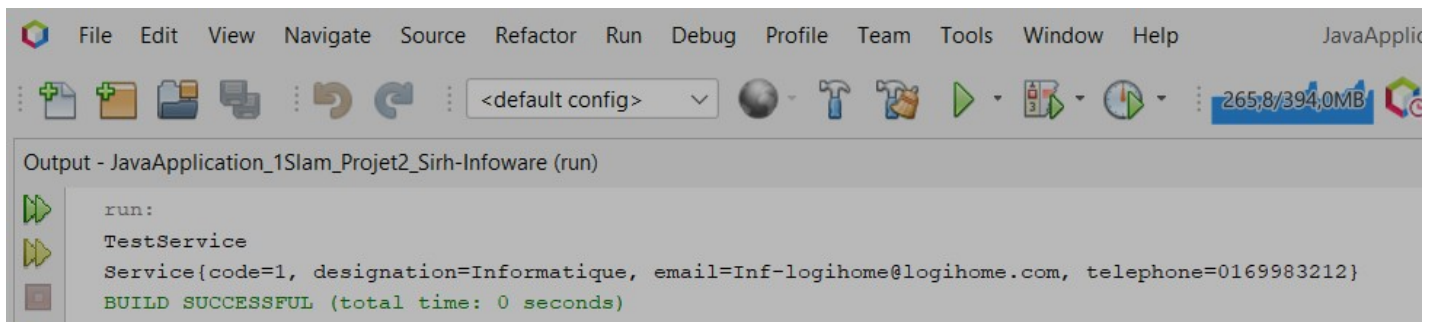


```

run:
TestSalarie
1 - Salarie sans service
Salarie{code=R06, nom=LANDREAU, prenom=Bertrand, dateNaiss=12/12/1980, dateEmbauche=15/11/2006, fonction=Dveloppeur, tauxHoraire=10,00, situationFamiliale=mari, nbrEnfants=2, service=Nant}
2 - Salarie avec service
Salarie{code=R06, nom=LANDREAU, prenom=Bertrand, dateNaiss=12/12/1980, dateEmbauche=15/11/2006, fonction=Dveloppeur, tauxHoraire=10,00, situationFamiliale=mari, nbrEnfants=2, service=Service{code=1, designation=Informatique, email=Inf-logihome@logihome.com, telephone=0169983212}}
BUILD SUCCESSFUL (total time: 0 seconds)

```

Exécution de « TestSalarie.java »



```

run:
TestService
Service{code=1, designation=Informatique, email=Inf-logihome@logihome.com, telephone=0169983212}
BUILD SUCCESSFUL (total time: 0 seconds)

```

Exécution de « testService.java »

création de la classe « Categorie »

```

public class Categorie {

    private String code;
    private String libelle;
    private double salaireMini;
    private String caisseRetraite;
    private int prime;

    public Categorie(String code, String libelle, double salaireMini, String caisseRetraite, int prime) {
        this.code = code;
        this.libelle = libelle;
        this.salaireMini = salaireMini;
        this.caisseRetraite = caisseRetraite;
        this.prime = prime;
    }
}

```

Ajout du constructeur

2023/2024	Projet JAVA – Etape 02
BTS SIO	Auteur : Loric Worms / Stefen Rialland
1SIOB - SLAM	Date de rédaction : 16/04/2024

```

public String getCode() {
    return code;
}

public void setCode(String code) {
    this.code = code;
}

public String getLibelle() {
    return libelle;
}

public void setLibelle(String libelle) {
    this.libelle = libelle;
}

public double getSalaireMini() {
    return salaireMini;
}

public void setSalaireMini(double salaireMini) {
    this.salaireMini = salaireMini;
}

public String getCaisseRetraite() {
    return caisseRetraite;
}

public void setCaisseRetraite(String caisseRetraite) {
    this.caisseRetraite = caisseRetraite;
}

public int getPrime() {
    return prime;
}

```

← Ajout des accesseurs et mutateurs

```

@Override
public String toString() {
    return "Catégorie{" + "code=" + code + ", libelle=" + libelle + ", salaire minimum="
        + salaireMini + ", caisse retraite=" + caisseRetraite + ", prime=" + prime + '}';
}

```

Ajout de la méthode redéfinie « toString »

Création de la classe TestCategorie

```

package test.metier;

import modele.metier.Categorie;

/**
 * @author loric
 */
public class TestCategorie {

    public static void main(String[] args) {

        System.out.println(x: "TestCategorie");

        Categorie uneCategorie = new Categorie (code: "C1", libelle: "Cadre moyen", salaireMini: 2500, caisseRetraite: "AGIRC", prime: 1);

        System.out.println(x: uneCategorie.toString());

    }

}

```

2023/2024	Projet JAVA – Etape 02
BTS SIO	Auteur : Loric Worms / Stefen Rialland
1SIOB - SLAM	Date de rédaction : 16/04/2024

Output - JavaApplication_1Slam_Projet2_Sirh-Infoware (run)

```
run:
TestCategorie
Catégorie{code=C1, libelle=Cadre moyen, salaire minimum=2500.0, caisse retraite=AGIRC, prime=1}
BUILD SUCCESSFUL (total time: 0 seconds)
```

Exécution de la classe testCategorie

Modification de la classe « Salarie »

```
public class Salarie {

    private String code = "";
    private String nom = "";
    private String prenom = "";
    private Date dateNaiss;
    private Date dateEmbauche;
    private String fonction;
    private double tauxHoraire;
    private String situationFamilliale;
    private int nbrEnfants;
    private Service service;
    private Categorie categorie;
```

← Ajout de la catégorie dans les attributs.

```
public Salarie(String code, String nom, String prenom, Date
    this.code = code;
    this.nom = nom;
    this.prenom = prenom;
    this.dateNaiss = dateNaiss;
    this.dateEmbauche = dateEmbauche;
    this.fonction = fonction;
    this.tauxHoraire = tauxHoraire;
    this.situationFamilliale = situationFamilliale;
    this.nbrEnfants = nbrEnfants;
    this.service = null;
    this.categorie = null;
}
```

← Prise en compte de la catégorie dans les constructeur

```
public Salarie(String code, String nom, String prenom, Date dateNaiss, Date dateEmbauche, String fonction, double tauxHoraire,
    this( code, nom, prenom, dateNaiss, dateEmbauche, fonction, tauxHoraire, situationFamilliale, nbrEnfants);
    this.service = serv;
    this.categorie = categorie;
}
```

```
@Override
public String toString() {
    String dateNaissFmt = String.format(format:"%1$td/%1$tm/%1$tY ", args: dateNaiss);
    String dateEmbFmt = String.format(format:"%1$td/%1$tm/%1$tY ", args: dateEmbauche);
    String txHoraireFmt = String.format(1: Locale.FRANCE, format:"%1$5.2f", args: tauxHoraire);
    String serviceToString = (service == null ? "Néant" : service.toStringEtat());
    String categorieToString = (categorie == null ? "Néant" : categorie.toString());
    return "Salarie{" + "code=" + code + ", nom=" + nom + ", prenom=" + prenom + ", dateNaiss"
        + ", fonction=" + fonction + ", tauxHoraire=" + txHoraireFmt + ", situationFamilliale"
        + ",\n service=" + serviceToString
        + ",\n categorie=" + categorieToString + '}';
}
```

2023/2024	Projet JAVA – Etape 02
BTS SIO	Auteur : Loric Worms / Stefen Rialland
1SIOB - SLAM	Date de rédaction : 16/04/2024

← Modification de la méthode « toString »

```
public Kategorie getKategorie() {
    return categorie;
}

public void setKategorie(Kategorie categorie) {
    this.categorie = categorie;
}
```

← Ajout de l'accesseur et du mutateur de la catégorie.

Modification de la classe « testSalarie ».

```
public class TestSalarie {

    public static void main(String[] args) {
        SimpleDateFormat sdf = new SimpleDateFormat(pattern: "dd/MM/yy");
        Salarie unSalarie = null;

        System.out.println(x: "TestSalarie");

        System.out.println(x: "1 - Salarie sans service");
        try {
            unSalarie = new Salarie(code: "R06", nom: "LANDREAU", prenom: "Bertrand", dateNaiss: sdf.parse(source: "12/12/1980"), dateEmbauche: sdf.parse(source: "15/11/2006")
        } catch (ParseException ex) {
            ex.printStackTrace();
        }
        System.out.println(x: unSalarie.toString());

        System.out.println(x: "2 - Salarie avec service");
        try {
            Service serv = new Service(code: 1, designation: "Informatique", email: "Inf-logihome@logihome.com", telephone: "0169983212");
            Kategorie cat = new Kategorie (code: "C1", libelle: "Cadre moyen", salaireMini: 2500, caisseRetraite: "AGIRC", prime: 1);
            unSalarie = new Salarie(code: "R06", nom: "LANDREAU", prenom: "Bertrand", dateNaiss: sdf.parse(source: "12/12/1980"), dateEmbauche: sdf.parse(source: "15/11/2006")
        } catch (ParseException ex) {
            ex.printStackTrace();
        }
        System.out.println(x: unSalarie.toString());
    }
}
```

```
run:
TestSalarie
1 - Salarie sans service
Salarie(code=R06, nom=LANDREAU, prenom=Bertrand, dateNaiss=12/12/1980 , dateEmbauche=15/11/2006 , fonction=Dveloppeur, tauxHoraire=10,00, situationFamiliale=mari, nbrEnfants=2,
service=Nant,
categorie=Nant)
2 - Salarie avec service
Salarie(code=R06, nom=LANDREAU, prenom=Bertrand, dateNaiss=12/12/1980 , dateEmbauche=15/11/2006 , fonction=Dveloppeur, tauxHoraire=10,00, situationFamiliale=mari, nbrEnfants=2,
service=Service(code=1, designation=Informatique, email=Inf-logihome@logihome.com, telephone=0169983212),
categorie=Categorie(code=C1, libelle=Cadre moyen, salaire minimum=2500.0, caisse retraite=AGIRC, prime=1))
BUILD SUCCESSFUL (total time: 0 seconds)
```

Exécution de la classe « TestSalarie »