**Rapport de Projet P.O.O**

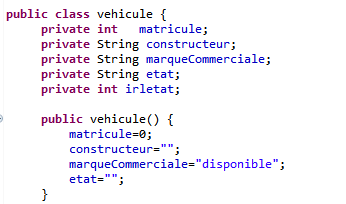
Réalisateurs :

-Ferchichi Oubaid Allah

On a le package car qui regroupe tous nous class

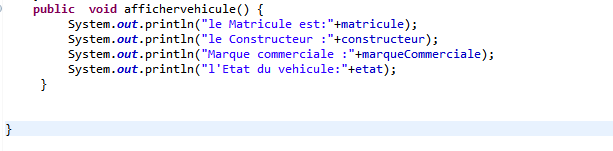


Class Vehicule :

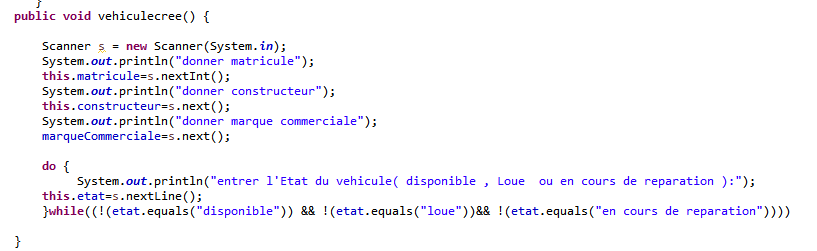


-La classe véhicule représente les véhicule.

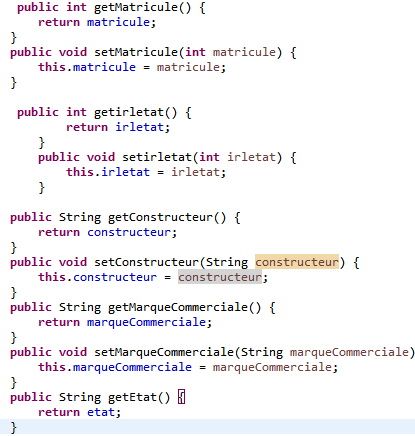
L’attribut irletat est un attribut qui présente l’état d’une véhicule (bonne état ou non)



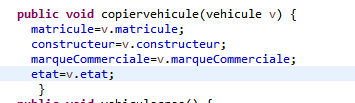
-affichervehicule() :C’est une méthode pour l’affichage de véhicule



vehiculecreer() : méthode permet au utilisateur de la création et la saisie d’une nouvelle véhicule .

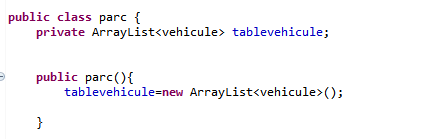


Les Méthode get et set correspond au class véhicule .

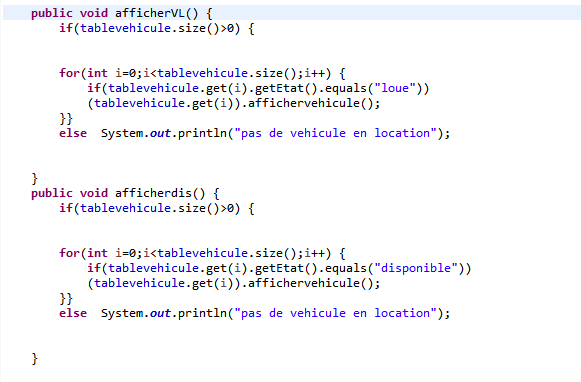


Copiervehicule() : nous permet de copier une véhicule

Class Park :

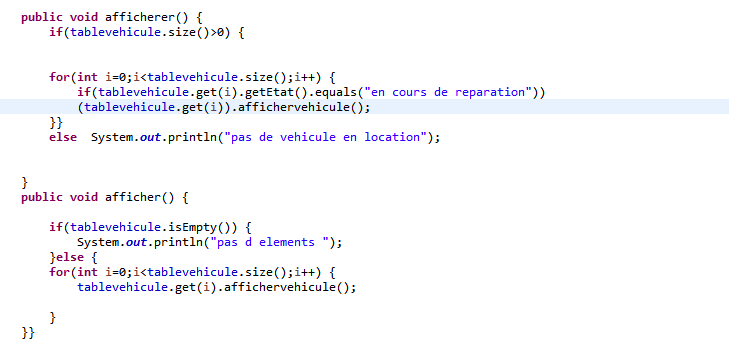


La class parc est constituer d’un tableu dynamique de type vehicule.



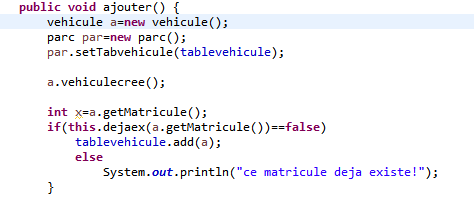
AfficherVl() : Afficher les vehicule dans l’etat alloue.

Afficherdis() :Afficher les vehicule dans l’etat disponible .

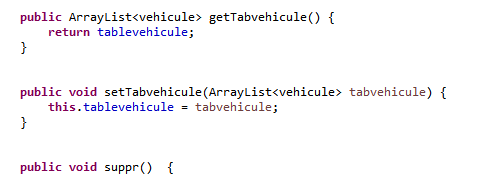


Afficherer() : permet de l’affichage des vehicule dans l’etat est en cour de reparation.

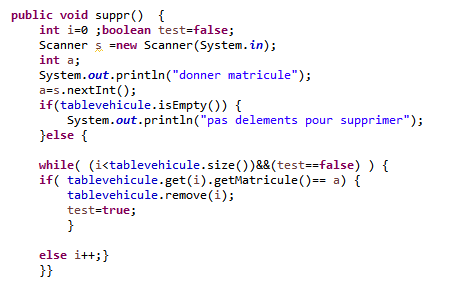
Afficher() : permet l’affichage de tableau tablevehicule.

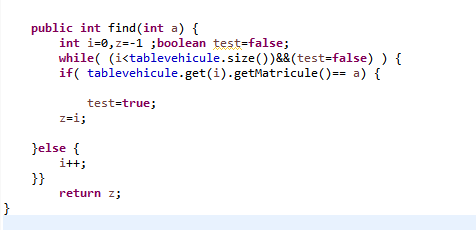


Ajouter() : permet l’ajout d’une vehicule dans la table.

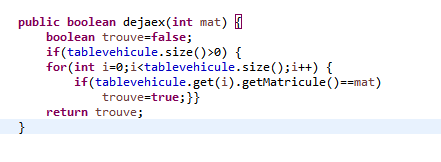


Les methode get et set de class parc

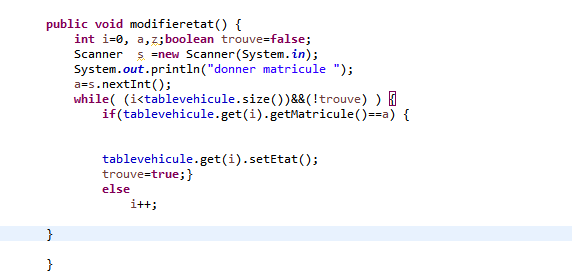


Suppr() : permet de la supression d’une voiture . 

Find(a) : permet de chercher une voiture d’apres la matricule a et qui retourne l’indice ou se trouve .

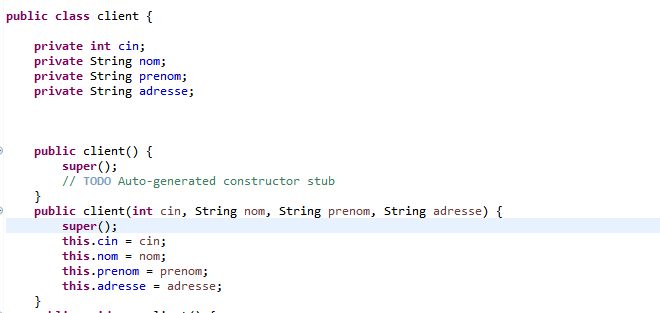


Dejaex() : teste si la vehicule de matricule mat existe ou non .

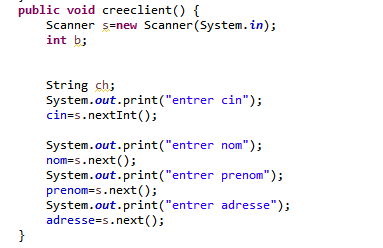


Modifieretat() :permet la modification l’état d’une véhicule dans le tableau

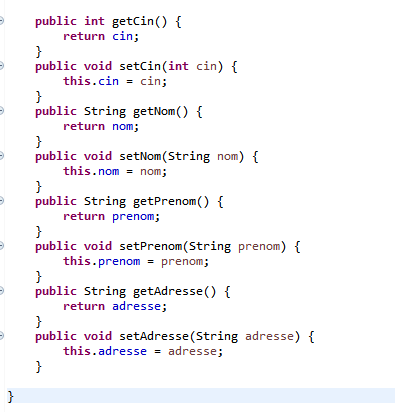
La Classe Client :



Un Constructeur paramétrer et non paramétrer .

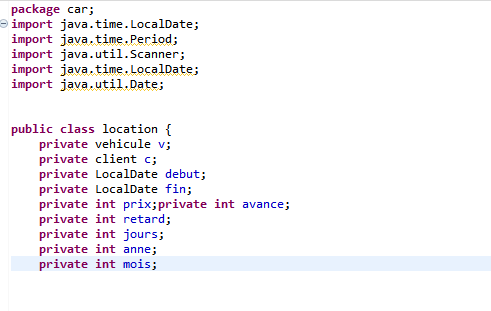


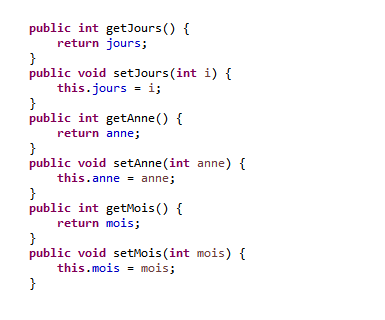
Creeclient() : permet de la création d’un client .



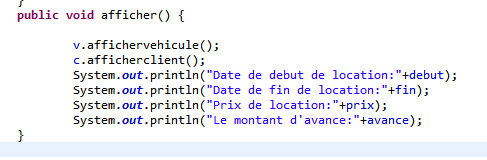
Les Méthode get et set correspond a la classe client .

La classe location :

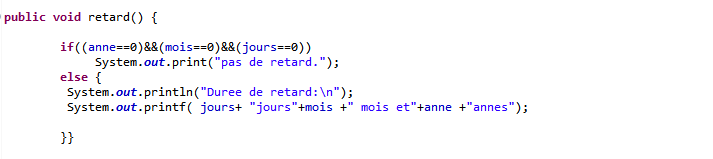




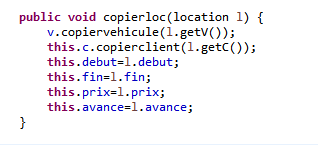
Les Méthode get et set correspond a les date.



Affircher() :permet l’affichage d’une location.



Retard() : permet l’affichage de retard de retour d’une véhicule s’il existe .



Copierloc(l) : permet de copier la location (l) dans la location .

**public** **void** cree(vehicule v, client c) {;

**int** a,b,cc;

**this**.v=v;

**this**.c=c;

Scanner s = **new** Scanner(System.***in***);

System.***out***.println("entrer la date de debut");

**do** {

System.***out***.println("anne:");

a=s.nextInt();}**while**((a<1000)||(a>9999));

**do** {

System.***out***.println("mois:");

b=s.nextInt();

}**while**((b<1 )|| (b>12));

**if**((b==2)) {

**do** {

System.***out***.println("jour");

cc=s.nextInt();

}**while**((cc<1 )|| (cc>28)); }

**else** **if**((b==4)|| (b==6 ) ||(b==11 )|| (b==9 ) ){

**do** {

System.***out***.println("jour");

cc=s.nextInt();

}**while**((cc<1 )|| (cc>30)); }

**else** {

System.***out***.println("jour");

cc=s.nextInt();

}**while**((cc<1 )|| (cc>31));

**this**.debut= LocalDate.*of*(a, b, cc);

System.***out***.println("entrer la date dez fin:");

System.***out***.println("anne:");

a=s.nextInt();

**do** {

System.***out***.println("mois:");

b=s.nextInt();

}**while**((b<1 )|| (b>12));

**if**((b==2)) {

**do** {

System.***out***.println("jour");

cc=s.nextInt();

}**while**((cc<1 )|| (cc>28)); }

**else** **if**((b==4)|| (b==6 ) ||(b==11 )|| (b==9 ) ){

**do** {

System.***out***.println("jour");

cc=s.nextInt();

}**while**((cc<1 )|| (cc>30)); }

**else** {

System.***out***.println("jour");

cc=s.nextInt();

}**while**((cc<1 )|| (cc>31));

**this**.fin= LocalDate.*of*(a, b, cc);

System.***out***.println("entrer le prix:");

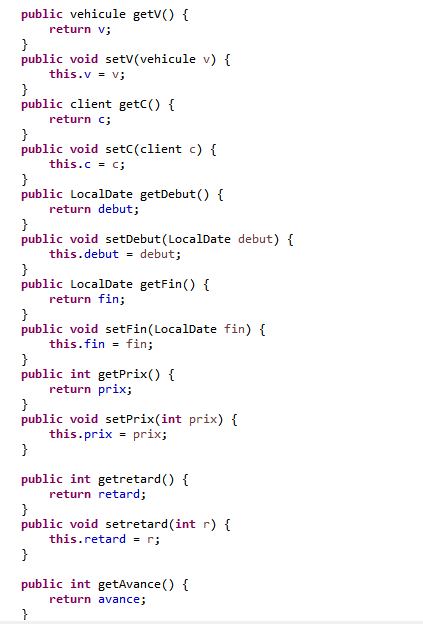
prix=s.nextInt();

System.***out***.println("entrer l'avance:) :");

avance=s.nextInt();

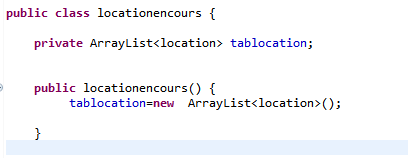
}

Créer(v,c):permet de créer une location en donnant la véhicule et le client comme paramètre et puis la saisi de date de début et fin et le prix et l’avance .

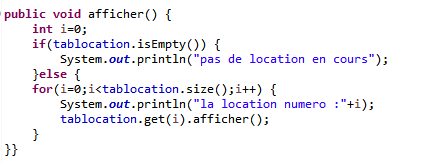


Les get et set des attributs de la classe location .

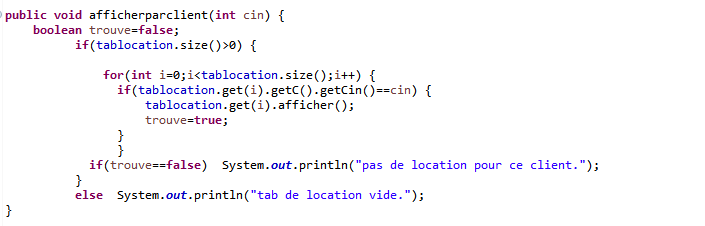
Class locationencours :



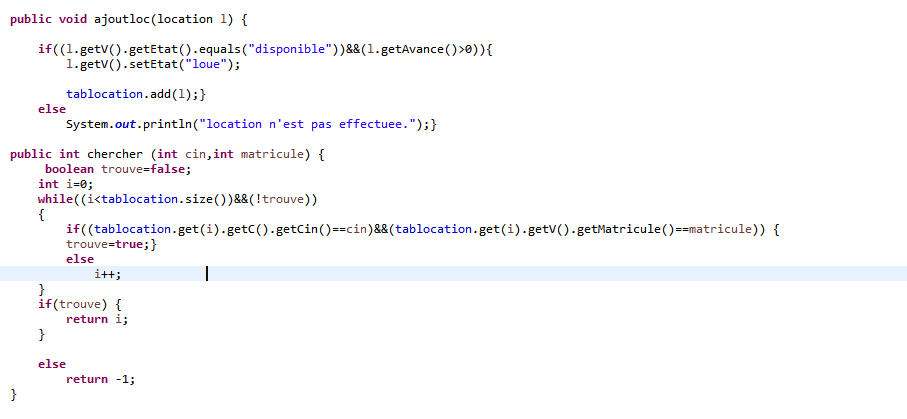
La class locationencours et constituer d’un tableau dynamique de type location avec un constructeur non paramétrer.



Afficher() : permet l’affichage de tableau tablocation .



Afficherparclient(cin) : permet de donner le cin d’un client et chercher et afficher les location le correspond s’elle existe



1-Ajoutloc(l) : permet d’ajouter une location le tableau tablocation si l’état de véhicule est disponible on donnant une avance (avance>0)

2-chercher(cin,matricule) :permet de chercher une location d’après cin de client et une matricule de véhicule donnée comme paramètre et retourne l’indice et s’il n’existe pas retourne -1.

**public** location suppression() {

location kaka=**new** location();

**if**(tablocation.isEmpty()) {

System.***out***.println("lite vide no ");

**return** kaka;

}**else** {

location l=**new** location();

Scanner s=**new** Scanner(System.***in***);

System.***out***.println("donner matricule");

**int** mat,ncin;

mat=s.nextInt();

System.***out***.println("donner cin");

ncin=s.nextInt();

**int** in = 0, i=0;**boolean** trouve=**false**;

**if**(**this**.chercher(ncin, mat)!=-1) {

LocalDate n = n=tablocation.get(**this**.chercher(ncin, mat)).getFin();

**if**(LocalDate.*now*().compareTo(n)>0) {

tablocation.get(**this**.chercher(ncin, mat)).setretard(1);

System.***out***.println("date de retour non respecter (il y a un retard).");

Period difference = Period.*between*(n, LocalDate.*now*());

System.***out***.print("Duree de retard:\n");

System.***out***.printf( "%d days, %d months and %d years",difference.getDays(),difference.getMonths(), difference.getYears());

tablocation.get(**this**.chercher(ncin, mat)).setJours(difference.getDays());

tablocation.get(**this**.chercher(ncin, mat)).setMois(difference.getMonths());

tablocation.get(**this**.chercher(ncin, mat)).setAnne(difference.getYears());

}

**int** a;

**do** {

System.***out***.print("donner l'etat de vehicule (1 si en bon etat 0 si non)."); a=s.nextInt();

}**while**((a!=1)&&(a!=0));

tablocation.get(**this**.chercher(ncin, mat)).getV().setirletat(a);

kaka=tablocation.get(**this**.chercher(ncin, mat));

tablocation.remove(**this**.chercher(ncin, mat)); }

**else** {

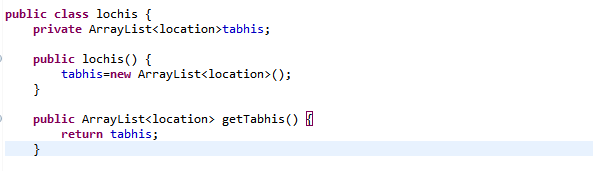
System.***out***.println("location inexistant");

}

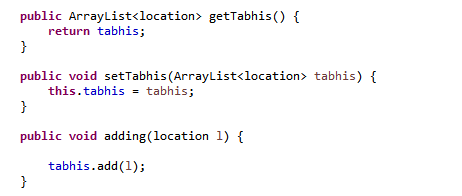
**return** kaka;}}

suppression(): c’est une méthode qui permet de supprimer une location d’après un tableau de locationencours et après la suppression elle retourne la location déjà supprimer pour les ajouter dans le tableau de location historique.

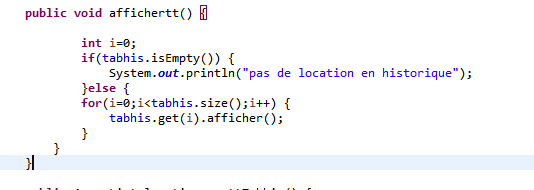
class lochis :



La class lochis est constituer d’un tableau dynamique de type location.



Les méthode get et set et add de tabhis (tableau historique).



Affichertt() :permet l’affichage de tableau.

**public** **void** nblocpourperiode () {

**boolean** trouve=**false**;

**int** i=0;

LocalDate dd1,dd2;

lochis l=**new** lochis();

location lH=**new** location();

Scanner s=**new** Scanner(System.***in***);

l.settabhis(tabhis);

**int** a,b,cc;

LocalDate deb,fin;

System.***out***.println("entrer la date de debut");

**do** {

System.***out***.println("anne:");

a=s.nextInt();}**while**((a<2020)||(a>2022));

**do** {

System.***out***.println("mois:");

b=s.nextInt();

}**while**((b<1 )|| (b>12));

**if**((b==2)) {

**do** {

System.***out***.println("jour");

cc=s.nextInt();

}**while**((cc<1 )|| (cc>28)); }

**else** **if**((b==4)|| (b==6 ) ||(b==11 )|| (b==9 ) ){

**do** {

System.***out***.println("jour");

cc=s.nextInt();

}**while**((cc<1 )|| (cc>30)); }

**else** {

System.***out***.println("jour");

cc=s.nextInt();

}**while**((cc<1 )|| (cc>31));

deb= LocalDate.*of*(a, b, cc);

System.***out***.println("entrer la date de fin:");

System.***out***.println("anne:");

a=s.nextInt();

**do** {

System.***out***.println("mois:");

b=s.nextInt();

}**while**((b<1 )|| (b>12));

**if**((b==2)) {

**do** {

System.***out***.println("jour");

cc=s.nextInt();

}**while**((cc<1 )|| (cc>28)); }

**else** **if**((b==4)|| (b==6 ) ||(b==11 )|| (b==9 ) ){

**do** {

System.***out***.println("jour");

cc=s.nextInt();

}**while**((cc<1 )|| (cc>30)); }

**else** {

System.***out***.println("jour");

cc=s.nextInt();

}**while**((cc<1 )|| (cc>31));

fin= LocalDate.*of*(a, b, cc);

**if**(tabhis.size()>0) {

**while**(i<tabhis.size()) {

dd1=l.get(i).getDebut();

dd2=l.get(i).getFin();

**if**(((dd1.compareTo(deb))>=0)&&((dd2.compareTo(fin))<0)) {

trouve=**true**;

lH=l.get(i);

lH.affichehis();

}

i++;

}

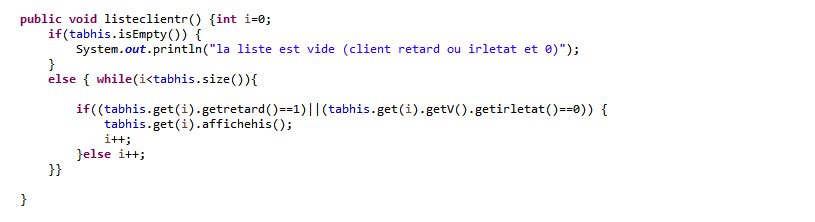
**if**(trouve==**false**) System.***out***.println("pas de location disponible dans cette periode");

}

**else** System.***out***.println("Liste de historique vide!");

}

nblocpourperiode():qui permet d’afficher les location de une période saisie a partir de l’utilisateur.



C’est une méthode qui permet d’afficher ensemble de location qui ont un retard ou la véhicule n’est pas en bonne état.

**public** **void** totrevenue() {

Scanner s=**new** Scanner (System.***in***);

LocalDate comp1,comp2;

**int** i=0;

lochis l=**new** lochis();

location lo=**new** location();

**int** rev=0;

l.setTabhis(tabhis);

**if**(!(tabhis.isEmpty())) {

**int** x,y,z,a1,b1,cc1;

LocalDate deb,fin;

System.***out***.println("entrer la date de debut");

**do** {

System.***out***.println("anne:");

x=s.nextInt();}**while**((x<2000)||(x>2023));

**do** {

System.***out***.println("mois:");

y=s.nextInt();

}**while**((y<1 )|| (y>12));

**if**((y==2)) {

**do** {

System.***out***.println("jour");

z=s.nextInt();

}**while**((z<1 )|| (z>28)); }

**else** **if**((y==4)|| (y==6 ) ||(y==11 )|| (y==9 ) ){

**do** {

System.***out***.println("jour");

z=s.nextInt();

}**while**((z<1 )|| (z>30)); }

**else** {

System.***out***.println("jour");

z=s.nextInt();

}**while**((z<1 )|| (z>31));

deb= LocalDate.*of*(x, y, z);

System.***out***.println("entrer la date de fin:");

**do** {

System.***out***.println("anne:");

a1=s.nextInt();}**while**(((a1<1000)||(a1>9999))||(a1<x ));

**if**(a1==x) {

**do** {

System.***out***.println("mois:");

b1=s.nextInt();

}**while**(((b1<1 )|| (b1>12))||(b1<y ));

}

**else**

**do** {

System.***out***.println("mois:");

b1=s.nextInt();

}**while**((b1<1 )|| (b1>12));

**if**((b1==y)&&(a1==x)){

**if**((b1==2)) {

**do** {

System.***out***.println("jour");

cc1=s.nextInt();

}**while**(((cc1<1 )|| (cc1>28))||(cc1<z )); }

**else** **if**((b1==4)|| (b1==6 ) ||(b1==11 )|| (b1==9 ) ){

**do** {

System.***out***.println("jour");

cc1=s.nextInt();

}**while**(((cc1<1 )|| (cc1>30))||(cc1<z )); }

**else** {

**do** {

System.***out***.println("jour");

cc1=s.nextInt();

}**while**(((cc1<1 )|| (cc1>31))||(cc1<z ));

}

}

**else** {

**if**((b1==2)) {

**do** {

System.***out***.println("jour");

cc1=s.nextInt();

}**while**((cc1<1 )|| (cc1>28)); }

**else** **if**((b1==4)|| (b1==6 ) ||(b1==11 )|| (b1==9 ) ){

**do** {

System.***out***.println("jour");

cc1=s.nextInt();

}**while**((cc1<1 )|| (cc1>30)); }

**else** {

**do** {

System.***out***.println("jour");

cc1=s.nextInt();

}**while**((cc1<1 )|| (cc1>31));

}}

fin= LocalDate.*of*(a1, b1, cc1);

**boolean** tt=**false**;**int** revenue=0;

comp1=tabhis.get(i).getDebut();

comp2=tabhis.get(i).getFin();

**while**( i<tabhis.size()){

**if**(((comp1.compareTo(deb))>0)&&((comp2.compareTo(fin))<0)) {

tt=**true**;

revenue+=l.get(i).getAvance();

revenue+=l.get(i).getPrix();

}

i++;

}

**if**(tt==**true**) System.***out***.println("le totale des revenue dans cette periode est:"+revenue+"dtn");

**if**(tt==**false**) System.***out***.println("pas de location disponible dans cette periode");

}

**else** {

System.***out***.println("historique est vide");}

}

}

La méthode totrevenue() :qui fait un parcoure dans le tableau historique(tabhis) puis l’ajout des revenue de tous location existe entre la date début et date fin saisie a partir de l’utilisateur .

Class Main :

**public** **static** **void** main(String[] args) {

lochis boost=**new** lochis();

parc pi=**new** parc();

locationencours pa=**new** locationencours();

**int** choix,choixx;

Scanner s=**new** Scanner(System.***in***);

**do** {

**do** {

System.***out***.println("1-Gestion du Parc de l�Agence: ");

System.***out***.println("2-Gestion Locations: ");

System.***out***.println("3-Statistiques: ");

System.***out***.println("4-quitter ");

choix=s.nextInt();

}**while** ((choix!=4)&&(choix!=3)&&(choix!=2)&&(choix!=1));

**if**(choix==1) {

**do** {

**do** {

System.***out***.println("1. Nouvelle acquisition (Ajout d’un véhicule)");

System.***out***.println("2. Modification de l’Etat d’un Véhicule ");

System.***out***.println("3. Suppression d’un Véhicule");

System.***out***.println("4. Consultation Parc (tous les véhicules)");

System.***out***.println("5. Liste des véhicules loués") ;

System.***out***.println("6. Liste des véhicules en cours de réparation");

System.***out***.println("7. Liste des véhicules disponibles");

System.***out***.println("8. retour au menu principal");

choixx=s.nextInt();

}**while**((choixx!=1)&&(choixx!=2)&&(choixx!=3)&&(choixx!=4)&&(choixx!=5)&&(choixx!=6)&&(choixx!=7)&&(choixx!=8));

**switch**(choixx) {

**case** 1:

pi.ajouter();

**break**;

**case** 2:

pi.modifieretat();

**break**;

**case** 3 :

pi.suppr();

**break** ;

**case** 4:

pi.afficher();

**break**;

**case** 5:

pi.afficherVL();

**break**;

**case** 6:

pi.afficherer();

**break** ;

**case** 7 :

pi.afficherdis();

**break**;

}

}**while**(choixx!=8);

}

**else** **if**(choix==2) {

**do** {

**do** {

System.***out***.println("1. Nouvelle Location (Ajout d’une location)");

System.***out***.println("2. Fin de Location (Suppression) ");

System.***out***.println("3. Consultation des locations en cours");

System.***out***.println("4. Consultation des Véhicules Loués par un client");

System.***out***.println("5. retour au menu principal");

choixx=s.nextInt();

}**while**((choixx!=4)&&(choixx!=3)&&(choixx!=2)&&(choixx!=1)&&(choixx!=5));

**switch**(choixx) {

**case** 1:

location lo=**new** location();

Scanner r=**new** Scanner(System.***in***);**int** a;

client l =**new** client();

l.creeclient();

System.***out***.println("donner le matricule ");

a=r.nextInt();

**if**(pi.vexist(a)!=-1) {

lo.cree(pi.get(pi.vexist(a)),l);

pa.ajoutloc(lo);}

**else** System.***out***.println("voiture inexistant dans parking ");

**break**;

**case** 2:

location his=**new** location();

his=pa.suppression();

boost.adding(his);

**break** ;

**case** 3:

pa.afficher();

**break**;

**case** 4:

System.***out***.println("donner le cin");**int** gh;

Scanner ls=**new** Scanner(System.***in***);

gh=ls.nextInt();

pa.afficherparclient(gh);

**break**;

}

}**while**(choixx!=5); }

**else** **if**(choix==3) {

**do** {

**do** { System.***out***.println("1. Nombre de locations pour une période déterminée");

System.***out***.println("2. Total des revenus pour une période déterminé");

System.***out***.println("3. Liste des clients à risque (Retard / mauvais état de véhicules rendus)");

System.***out***.println("4. retour menu principal");

choixx=s.nextInt();

}**while**((choixx!=4)&&(choixx!=3)&&(choixx!=2)&&(choixx!=1));

**switch**(choixx) {

**case** 1:

boost.nblocpourperiode();

**break** ;

**case** 2:boost.totrevenue();

**break**;

**case** 3:

boost.listeclientr();

}

}**while**(choixx!=4);

}

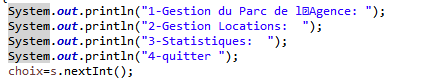
}**while**(choix!=4);

}

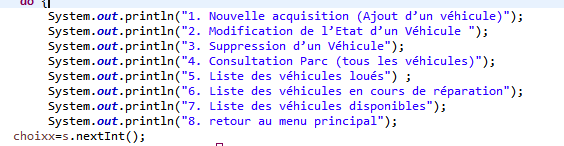
}

Dans le main on a créer un menu interactif qui permet au utilisateur de communiquer avec le programme a partir de 1 menu principale et 3 sous menu

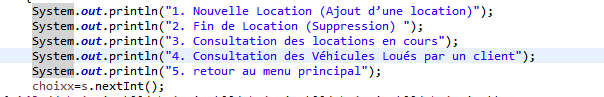
-Menu Principale :



1-sous menu de « Gestion du parc de l’agence: » qui permet de :



2-sous menu de «Gestion locations:» qui permet de :



3-sous menu de «Statistique:» qui permet de :

