

Main.java

```
package magasin;
```

```
public class Main {
```

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

```
        System.out.println("Hello world!");
```

```
    }
```

```
}
```

## Client.java

```
package magasin.metier;

import java.time.LocalDate;
import java.util.ArrayList;
import java.util.List;
import java.util.Objects;

/**
 * classe métier de gestion d'un client
 *
 * @author Michel Poriaux
 * @version 1.0
 * @see ComFact
 */

public class Client {
    /**
     * identifiant du client
     */
    protected int idclient;
    /**
     * nom du client
     */
    protected String nom;
    /**
     * prénom du client
     */
    protected String prenom;
    /**
     * code postal de la localité
     */
    protected Integer cp;
    /**
     * localité
     */
    protected String localite;
    /**
     * rue
     */
    protected String rue;
    /**
     * numéro de rue
     */
    protected String num;
    /**
     * numéro de téléphone du client
     */
    protected String tel;
```

```

/**
 * ensemble des commandes du client
 */
protected List<ComFact> comFacts = new ArrayList<>();

/**
 * constructeur paramÃ©trÃ©
 *
 * @param idclient identifiant unique du client, affectÃ© par la
base de
 * donnÃ©es
 * @param nom nom du client
 * @param prenom prÃ©nom du client
 * @param cp code postal de l'adresse
 * @param localite localitÃ© de l'adresse
 * @param rue rue de l'adresse
 * @param num numÃ©ro de l'adresse
 * @param tel tÃ©lÃ©phone du client
 */
public Client(int idclient, String nom, String prenom, Integer cp,
String localite, String rue, String num, String tel) {
    this.idclient = idclient;
    this.nom = nom;
    this.prenom = prenom;
    this.cp = cp;
    this.localite = localite;
    this.rue = rue;
    this.num = num;
    this.tel = tel;
}

/**
 * getter idclient
 *
 * @return identifiant du client
 */
public int getIdclient() {
    return idclient;
}

/**
 * setter idclient
 *
 * @param idclient identifiant du client
 */
public void setIdclient(int idclient) {
    this.idclient = idclient;
}

```

```

}

/**
 * getter code postal
 *
 * @return code postal
 */
public Integer getCp() {
    return cp;
}

/**
 * setter code postal
 *
 * @param cp code postal
 */
public void setCp(Integer cp) {
    this.cp = cp;
}

/**
 * getter localite
 *
 * @return localite
 */
public String getLocalite() {
    return localite;
}

/**
 * setter localite
 *
 * @param localite localitÃ©
 */
public void setLocalite(String localite) {
    this.localite = localite;
}

/**
 * getter nom
 *
 * @return nom du client
 */
public String getNom() {
    return nom;
}

/**
 * setter nom du client

```

```

*
* @param nom nom du client
*/
public void setNom(String nom) {
    this.nom = nom;
}

/**
 * getter numÃ©ro de rue
 *
 * @return numÃ©ro de rue
 */
public String getNum() {
    return num;
}

/**
 * setter numÃ©ro de rue
 *
 * @param num numÃ©ro de rue
 */
public void setNum(String num) {
    this.num = num;
}

/**
 * getter prÃ©nom du client
 *
 * @return prÃ©nom du client
 */
public String getPrenom() {
    return prenom;
}

/**
 * setter nom du client
 *
 * @param prenom prÃ©nom du client
 */
public void setPrenom(String prenom) {
    this.prenom = prenom;
}

/**
 * getter rue
 *
 * @return rue
 */
public String getRue() {

```

```

        return rue;
    }

    /**
     * setter rue
     *
     * @param rue rue de l'adresse
     */
    public void setRue(String rue) {
        this.rue = rue;
    }

    /**
     * getter tÃ©lÃ©phone
     *
     * @return tÃ©lÃ©phone du client
     */
    public String getTel() {
        return tel;
    }

    /**
     * setter tÃ©lÃ©phone
     *
     * @param tel tÃ©lÃ©phone du client
     */
    public void setTel(String tel) {
        this.tel = tel;
    }

    /**
     * getter comFacts
     * @return liste des commandes du client
     */
    public List<ComFact> getComFacts() {
        return comFacts;
    }

    /**
     * setter comFacts
     * @param comFacts liste des commandes du client
     */
    public void setComFacts(List<ComFact> comFacts) {
        this.comFacts = comFacts;
    }

    /**
     * mÃ©thode de vÃ©rification d'Ã©galitÃ© de deux clients

```

```

    * @param o  autre client
    * @return Ã©galitÃ© ou pas
    */
@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;
    Client client = (Client) o;
    return idclient == client.idclient;
}

/**
 * calcul du hashCode du client
 * @return hashCode
 */

@Override
public int hashCode() {
    return Objects.hash(idclient);
}

/**
 * ajout d'une commande
 * @param cf comfact
 */
public void addComFact(ComFact cf){
    comFacts.add(cf);
    cf.setClient(this);
}

/**
 * suppression d'une commande
 * @param cf comfact
 */
public void supComFact(ComFact cf){
    comFacts.remove(cf);
    cf.setClient(null);
}

/**
 * Recherche des factures payÃ©es
 * @return liste des factures payÃ©es
 */
public List<ComFact> factPayees() {
    List<ComFact> lcf = new ArrayList<>();
    for(ComFact cf : comFacts){
        if(cf.getEtat()=='p') lcf.add(cf);
    }
}

```

```

        }
        return lcf;
    }

    /**
     * Recherche des factures en retard
     * @return liste des factures en retard
     */

    public List<ComFact> factRetard() {
        List<ComFact> lcf = new ArrayList<>();
        for(ComFact cf : comFacts){
            if(cf.estEnRetard())lcf.add(cf);
        }
        return lcf;
    }

    /**
     * Recherche des factures non pay  es
     * @return liste des factures non pay  es
     */
    public List<ComFact> factNonPayees() {
        List<ComFact> lcf = new ArrayList<>();
        for (ComFact cf : comFacts) {
            if (cf.estNonPayee()) lcf.add(cf);
        }
        return lcf;
    }

    /**
     * m  thode toString
     *
     * @return informations compl  tes
     */
    @Override
    public String toString() {
        return "Client" + "idclient=" + idclient + ", nom=" + nom + ",
prenom=" + prenom + ", cp=" + cp + ", localite=" + localite + ", rue=" +
rue + ", num=" + num + ", tel=" + tel + ' ';
    }
}

```



## ComFact.java

```
package magasin.metier;

import java.math.BigDecimal;
import java.math.RoundingMode;
import java.time.LocalDate;
import java.util.ArrayList;
import java.util.List;

/**
 * classe métier de gestion d'une commande-facture
 *
 * @author Michel Poriaux
 * @version 1.0
 * @see Client
 * @see Ligne
 */
public class ComFact {

    /**
     * identifiant unique-numéro de commande
     */
    protected int idcommande;
    /**
     * numéro de facture - unique
     */
    protected Integer numfact;

    /**
     * date de la commande
     */
    protected LocalDate datecom;

    /**
     * date de la facture
     */
    protected LocalDate dateFacturation;
    /**
     * date du paiement
     */
    protected LocalDate datePaiement;
    /**
     * État de la facture :c:en commande-f:facturé-p:payé
     */
    protected char etat;
    /**
     * montant total de la commande
     */
}
```

```

protected BigDecimal montant;
/**
 * client de la commande
 */
protected Client client;

/**
 * liste des lignes de commande
 */
protected List<Ligne> lignes = new ArrayList<>();

/**
 * constructeur paramÃ©trÃ©
 *
 * @param idcommande numÃ©ro de commande
 * @param numfact numÃ©ro de facture
 * @param datecom date de commande
 * @param etat Ã©tat de la commande C,F,P
 * @param montant montant total de la commande
 */
public ComFact(int idcommande, Integer numfact, LocalDate datecom,
char etat, BigDecimal montant, Client client) {
    this.idcommande = idcommande;
    this.numfact = numfact;
    this.datecom = datecom;
    this.etat = etat;
    this.montant = montant.setScale(2, RoundingMode.HALF_UP);
    this.client = client;
}

/**
 * getter client de la commande
 * @return client de la commande
 */
public Client getClient() {
    return client;
}

/**
 * setter client de la commande
 * @param client client de la commande
 */
public void setClient(Client client) {
    this.client = client;
}

/**

```

```

    * getter date de commande
    *
    * @return date de la commande
    */
    public LocalDate getDatecom() {
        return datecom;
    }

    /**
     * setter date de commande
     *
     * @param datecom date de la commande
     */
    public void setDatecom(LocalDate datecom) {
        this.datecom = datecom;
    }

    /**
     * getter date de commande
     *
     * @return date de facturation
     */
    public LocalDate getDateFacturation() {
        return dateFacturation;
    }

    /**
     * setter date de commande
     *
     * @param dateFacturation date de facturation
     */
    public void setDateFacturation(LocalDate dateFacturation) {
        this.dateFacturation = dateFacturation;
    }

    /**
     * getter date de paiement
     *
     * @return date de paiement
     */
    public LocalDate getDatePayment() {
        return datePayment;
    }

    /**
     * setter date de paiement
     *
     * @param datePayment date de paiement

```

```

    */
    public void setDatePayment(LocalDate datePayment) {
        this.datePayment = datePayment;
    }

    /**
     * getter État de la commande
     *
     * @return État de la commande (C:en commande,F:facturé,P:payé)
     */
    public char getEtat() {
        return etat;
    }

    /**
     * setter État de la commande
     *
     * @param etat État de la commande (C:en
    commande,F:facturé,P:payé)
     */
    public void setEtat(char etat) {
        this.etat = etat;
    }

    /**
     * getter montant
     *
     * @return montant total de la commande
     */
    public BigDecimal getMontant() {
        return montant;
    }

    /**
     * setter du montant
     *
     * @param montant montant total de la commande
     */
    public void setMontant(BigDecimal montant) {
        this.montant = montant.setScale(2, RoundingMode.HALF_UP);
    }

    /**
     * getter numéro de commande
     *
     * @return numéro de la commande
     */
    public int getIdcommande() {
        return idcommande;
    }

```

```

}

/**
 * setter numÃ©ro de commande
 *
 * @param idcommande numÃ©ro de la commande
 */
public void setIdcommande(int idcommande) {
    this.idcommande = idcommande;
}

/**
 * getter numÃ©ro de facture
 *
 * @return nulÃ©ro de la facture
 */
public Integer getNumfact() {
    return numfact;
}

/**
 * setter numÃ©ro de facture
 *
 * @param numfact numÃ©ro de facture
 */
public void setNumfact(Integer numfact) {
    this.numfact = numfact;
}

/**
 * getter lignes de commandes
 *
 * @return listes des lignes de commande
 */
public List<Ligne> getLignes() {
    return lignes;
}

/**
 * setter lignes
 *
 * @param lignes liste des lignes de commande
 */
public void setLignes(List<Ligne> lignes) {
    this.lignes = lignes;
    majMontant();
}

/**

```

```

    * méthode toString
    * @return informations complètes de la commande
    */

/**
 * ajout d'une ligne à la commande
 * @param ligne ligne à ajouter
 * @return ajout effectué ou pas
 */
public boolean addLigne(Ligne ligne){
    for(Ligne l : lignes){
        if(l.getProduit().equals(ligne.getProduit())) return false;
    }
    lignes.add(ligne);
    majMontant();
    return true;
}

/**
 * suppression d'une ligne à la commande
 * @param ligne ligne à supprimer
 */
public void supLigne(Ligne ligne){
    lignes.remove(ligne);
    majMontant();
}

@Override
public String toString() {
    return "ComFact" +
        "idcommande=" + idcommande +
        ", numfact=" + numfact +
        ", datecom=" + datecom +
        ", dateFacturation=" + dateFacturation +
        ", datePayement=" + datePayement +
        ", etat=" + etat +
        ", montant=" + montant +
        ", client=" + client +
        '\n';
}

/**
 * mise à jour du montant de la commande sur base des lignes de
commande
 */
private void majMontant(){
    montant = new BigDecimal(0).setScale(0,RoundingMode.HALF_UP);
    for(Ligne ligne : lignes){
        montant.add(ligne.valLigne());
    }
}

```

```

    }

    /**
     * vÃ©rification du retard d'une facture
     * @return en retard ou pas
     */
    public boolean estEnRetard(){
        return etat=='f' && datePayement==null &&
dateFacturation.plusDays(30).isBefore(LocalDate.now());
    }

    public boolean estNonPayee(){
        return etat=='f' && datePayement == null;
    }

    /**
     * calcul du hashcode de la commande
     * @return hashcode de la commande
     */

    @Override
    public int hashCode() {
        int hash = 3;
        hash = 89 * hash + this.idcommande;
        return hash;
    }

    /**
     * vÃ©rification de l'Ã©galitÃ© de deux commandes basÃ©e sur l'id de
la commande
     * @param obj autre commande
     * @return Ã©galitÃ© ou pas
     */

    @Override
    public boolean equals(Object obj) {
        if (this == obj) {
            return true;
        }
        if (obj == null) {
            return false;
        }
        if (getClass() != obj.getClass()) {
            return false;
        }
        final ComFact other = (ComFact) obj;
        if (this.idcommande != other.idcommande) {
            return false;
        }
    }

```

```
return true;
```

```
}
```

```
}
```



## Ligne.java

```
/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package magasin.metier;

import java.math.BigDecimal;
import java.math.RoundingMode;
import java.util.Objects;

/**
 * classe représentant les informations d'une ligne de commande
 * @see ComFact
 * @see Produit
 * @author Michel
 */
public class Ligne {
    /**
     * id unique de la ligne de commande
     */
    protected int idligne;
    /**
     * produit relatif à la ligne de commande
     */
    protected Produit produit;
    /**
     * quantité achetée
     */
    protected int quantite;
    /**
     * prix d'achat du produit au moment de la commande
     */
    protected BigDecimal prixAchat;

    /**
     * constructeur paramétré
     * @param idligne identifiant de la ligne
     * @param produit produit acheté
     * @param quantite quantité achetée
     * @param prixAchat prix d'achat au moment de l'achat
     */
    public Ligne(int idligne, Produit produit, int quantite, BigDecimal
prixAchat) {
        this.idligne=idligne;
        this.produit = produit;
    }
}
```

```

        this.quantite = quantite;
        this.prixAchat = prixAchat;
        prixAchat.setScale(2, RoundingMode.HALF_UP);
    }

    /**
     * getter idligne
     * @return idligne
     */
    public int getIdligne() {
        return idligne;
    }

    /**
     * setter idligne
     * @param idligne identifiant de la ligne
     */
    public void setIdligne(int idligne) {
        this.idligne = idligne;
    }

    /**
     * getter produit
     * @return produit achet  
     */
    public Produit getProduit() {
        return produit;
    }

    /**
     * setter produit
     * @param produit produit achet  
     */
    public void setProduit(Produit produit) {
        this.produit = produit;
    }

    /**
     * getter quantit  
     * @return quantit   achet  e
     */
    public int getQuantite() {
        return quantite;
    }

    /**
     * setter quantit  

```

```

    * @param quantite quantite à acheter
    */
    public void setQuantite(int quantite) {
        this.quantite = quantite;
    }

    /**
     * getter prix d'achat
     * @return prix d'achat
     */
    public BigDecimal getPrixAchat() {
        return prixAchat;
    }

    /**
     * setter prix d'achat
     * @param prixAchat prix d'achat du produit
     */
    public void setPrixAchat(BigDecimal prixAchat) {
        this.prixAchat = prixAchat;
    }

    /**
     * calcul de la valeur d'une ligne basée sur le prix d'achat du
    produit et la quantite achetée
     * @return valeur de la ligne
     */
    public BigDecimal valLigne() {
        return prixAchat.multiply(new
    BigDecimal(quantite)).setScale(2, RoundingMode.HALF_UP);
    }

    /**
     * méthode toString
     * @return informations complètes
     */
    @Override
    public String toString() {
        return "Ligne" + "produit=" + produit + ", quantite=" +
    quantite + ", prixAchat=" + prixAchat + '\n';
    }

    /**
     * méthode de vérification d'égalité de deux lignes
     * @param o autre ligne
     * @return égalité ou pas
     */
    @Override
    public boolean equals(Object o) {
        if (this == o) return true;

```

```
    if (o == null || getClass() != o.getClass()) return false;
    Ligne ligne = (Ligne) o;
    return idligne == ligne.idligne;
}
```

```
/**
 * calcul du hashcode de la ligne
 * @return hashcode de la ligne
 */
@Override
public int hashCode() {
    return Objects.hash(idligne);
}
```

```
}
```

## Produit.java

```
package magasin.metier;

import java.math.BigDecimal;
import java.math.RoundingMode;
import java.util.Objects;

/**
 * classe métier de gestion d'un produit
 * @author Michel Poriaux
 * @version 1.0
 */
public class Produit {

    /**
     * id unique du produit
     */
    protected int idproduit;

    /**
     * numéro unique du produit
     */
    protected String numprod;
    /**
     * description du produit
     */
    protected String description;
    /**
     * prix hors tva
     */
    protected BigDecimal phtva;
    /**
     * stock en cours
     */
    protected int stock;

    /**
     * stock minimum en cours
     */
    protected int stockMin;

    /**
     * constructeur par défaut
     */
    public Produit() {}

    /**
     * constructeur paramétré
```

```

* @param idproduit id numÃ©rique unique du produit
* @param numprod numÃ©ro unique du produit
* @param description description du produit
* @param phtva prix hors tva du produit
* @param stock stock en cours du produit/**
* constructeur paramÃ©trÃ©
*/

    public Produit(int idproduit,String numprod, String description,
BigDecimal phtva, int stock,int stockMin) {
        this.idproduit=idproduit;
        this.numprod = numprod;
        this.description = description;
        this.phtva = phtva;
        phtva.setScale(2,RoundingMode.HALF_UP);
        this.stock = stock;
        this.stockMin=stockMin;
    }

/**
 * getter phtva
 * @return prix hors tva du produit
 */
    public BigDecimal getPhtva() {
        return phtva;
    }

/**
 * setter phtva
 * @param phtva prix hors tva du produit
 */
    public void setPhtva(BigDecimal phtva) {
        this.phtva = phtva;
    }

/**
 * getter stock
 * @return stock en cours
 */

    public int getStock() {
        return stock;
    }

/**
 * setter stock
 * @param stock nouveau stock
 */

```

```

public void setStock(int stock) {
    this.stock = stock;
}

/**
 * getter stockMin
 * @return stock en cours
 */

public int getStockMin() {
    return stockMin;
}

/**
 * setter stock
 * @param stockMin nouveau stock
 */
public void setStockMin(int stockMin) {
    this.stockMin = stockMin;
}

/**
 * getter description
 * @return description du produit
 */
public String getDescription() {
    return description;
}

/**
 * setter description
 * @param description nouvelle description
 */
public void setDescription(String description) {
    this.description = description;
}

/**
 * getter numéro de produit
 * @return numéro de produit
 */
public String getNumprod() {
    return numprod;
}

@Override
public String toString() {
    return "Produit" +

```

```

        "idproduit=" + idproduit +
        ", numprod='" + numprod + '\'' +
        ", description='" + description + '\'' +
        ", phtva=" + phtva +
        ", stock=" + stock +
        ", stockMin=" + stockMin +
        ' ';

    }

    /**
     * affichage des infos du produit
     * @return description compl te du produit
     */

    /**
     * v rification si le stock est suffisant
     * @return stock suffisant ou pas
     */
    public boolean stockSuffisant(){
        return stock>=stockMin;
    }

    /**
     * calcul de la quantit    recommander pour r alimenter le stock
    du produit
     * @return quantit    recommander
     */
    public int quantiteACommander(){
        int stockMax = (int)(stockMin*1.5);
        if (stockMax>stock) return stockMax-stock;
        else return 0;
    }

    /**
     * R approvisionnement du stock
     * @param q quantit  du r approvisionnement
     */
    public void reapprovisionner(int q){
        stock+=q;
    }

    /**
     * Demande de diminution du stock
     * @param q valeur de la diminution
     * @return diminution effectu e ou pas selon la valeur en stock
     */
    public boolean retirer(int q){

```



```

        if(q>stock) return false;
        stock-=q;
        return true;
    }

    /**
     * calcul de la valeur monétaire en stock
     * @return valeur monétaire du stock
     */
    public BigDecimal valeurStock() {
        return phtva.multiply(new BigDecimal(stock));
    }

    /**
     * vérification de l'égalité de deux produits basée sur le
    numéro du produit
     * @param o
     * @return égalité ou pas
     */
    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Produit produit = (Produit) o;
        return numprod.equals(produit.numprod);
    }

    /**
     * calcul du hashcode du produit
     * @return valeur du hashcode du produit
     */
    @Override
    public int hashCode() {
        return Objects.hash(numprod);
    }
}

```

