

E4D : ÉTUDE DE CAS

Durée : 5 heures

Coefficient : 5

CAS CREDAUTO

Éléments de correction

Barème

| | | |
|--|-----|--------|
| Dossier 1 : Recherche de solutions techniques | 18 | points |
| Dossier 2 : Suivi du recouvrement amiable | 22 | points |
| Dossier 3 : Gestion des frais | 30 | points |
| Dossier 4 : Gestion des avenants par région | 30 | points |
| Total | 100 | points |

1.1 Parmi la liste des documents constituant un dossier de prêt, indiquer ceux qui devront être numérisés à l'aide d'un scanner, en justifiant les réponses.

D'après la description du fonctionnement, il existe un document constitué de données, le contrat et des documents à numériser.

Ces derniers documents peuvent être :

- les documents d'identité,
- les documents justifiant le domicile,
- les documents personnels bancaires (RIB),
- les courriers échangés avec le client.

On acceptera dans la réponse la référence à la demande de prêt.

On évaluera principalement l'absence de documents issus de traitements dans la liste des documents numérisés. La présence du RIB n'est pas exigée. En cas de présence du contrat, celle-ci devra être plus particulièrement justifiée.

1.2 Définir succinctement les principales étapes nécessaires pour qu'un document puisse être intégré dans une GED.

La gestion électronique des documents (GED) consiste essentiellement à dématérialiser la documentation papier afin de l'archiver et de la classer selon une nomenclature propre à l'entreprise. Ce système permet d'optimiser la recherche et la réutilisation des documents par les différents services de l'entreprise. La GED appliquée correctement comprend :

- 1- La numérisation des sources.
- 2- La reconnaissance des textes et des images.
- 3- Le stockage et l'indexation des documents numérisés.
- 4- La diffusion et l'exploitation des données grâce à des logiciels spécialisés.

Note : la définition de la GED n'est pas demandée.

1.3 Donner le critère principal à prendre en compte pour évaluer la qualité de numérisation d'un document par un scanner.

La **résolution optique** d'un scanner constitue le critère principal à prendre en compte. En effet, la résolution d'un scanner se compte en dpi. Cela correspond au nombre de points différents que peut distinguer un scanner sur une ligne de 1 pouce, soit 2,54 cm. Plus une image comporte de points, plus sa résolution est grande et meilleure est sa qualité de reproduction.

1.4 Dresser un tableau comparatif entre les formats « bitmap » et « vectoriel » concernant les caractéristiques suivantes :

- La technique d'enregistrement dans le fichier,
- Les possibilités de changement de taille et de modification,
- Le poids obtenu (taille en octets) de l'image.

| | BITMAP | VECTORIEL |
|--|--|--|
| Technique d'enregistrement | L'image est sauvegardée sous la forme d'une matrice de points appelés pixels (contraction de l'anglais <i>picture element</i>). | L'image est sauvegardée sous la forme d'un ensemble de formules mathématiques. |
| Possibilités de changement de taille et de modification | Si l'on change la taille d'un élément (en particulier si on tente de l'agrandir), il apparaît un "effet d'escalier" fort désagréable. Le format bitmap ne permet donc pas d'agrandir l'image sans déperdition de qualité (zoom délicat, même impossible) | Le format vectoriel permet de garder la précision du tracé quelle que soit la taille de l'image (ex : zoom illimité, car nouveau "calcul" du logiciel) |
| | Modification très délicate : modifier point par point (!) ou tous les points (!). Ce format peut être compressé. | Modification aisée par "élément" (seul ou "groupé") qui peut être déplacé, agrandi, modifié... |
| Poids obtenus | Fichiers volumineux (chaque point est codé et mémorisé) | Fichiers légers (des formules mathématiques sont codées) |

1.5 Donner la taille en octets d'un fichier au format « bitmap » obtenu en numérisant un document 25 cm x 25 cm avec une précision de 200 dpi (avec 1 inch = 2,5 cm) et une définition de 24 bits par pixel. Justifier le résultat en fournissant le calcul détaillé.

Taille du fichier en bits = nombre de points * nombre de bits par points
 Nombre de points = (nombre de points longueur côté)²
 = (nombre de points par cm * longueur côté en cm)²
 = ((200 / 2,5) * 25)²
 = 2000²
 Taille du fichier = 4 000 000 * 24 = 96 000 000 bits, soit **12 000 000 octets**

1.6 a) Expliquer pourquoi l'adresse IP de cette station cliente ne lui permet pas de communiquer avec le serveur.

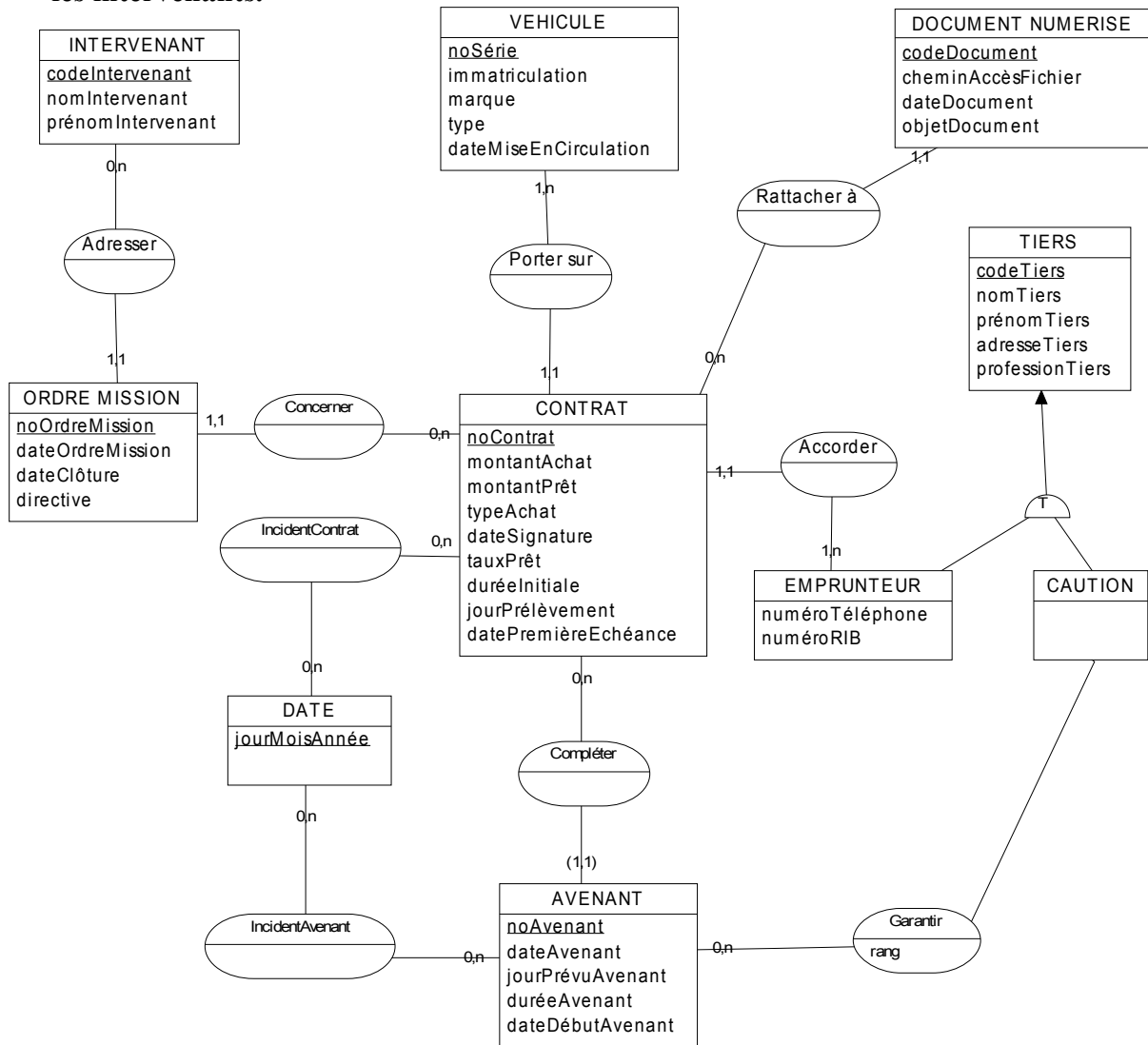
L'adresse de réseau du serveur est 192.168.10.0 (adresse 192.168.10.10 et masque 255.255.255.0). Or la station cliente possède l'adresse de réseau 192.168.20.0 (adresse 192.168.20.11 et masque 255.255.255.0). Ils sont donc sur deux réseaux différents et ne peuvent communiquer ensemble.

b) Indiquer, parmi les six adresses IP suivantes, celles qui peuvent être affectées à la station de numérisation. Justifier la réponse.

Compte tenu du masque, la plage d'adresses utilisables est [192.168.10.1, 192.168.10.254]. D'une part, une adresse IP doit être unique, on ne peut donc affecter l'adresse 192.168.10.10 utilisée par le serveur. D'autre part, 192.168.10.0 correspond à l'adresse réseau et 192.168.10.255 correspond à l'adresse de diffusion, ces 2 adresses ne pouvant être affectées à un poste. On peut donc affecter à la station de numérisation une de ces trois adresses 192.168.10.1, 192.168.10.11 et 192.168.10.254.

DOSSIER 2 | **SUIVI DU RECOUVREMENT AMIABLE**

2.1 Présenter un schéma entité-association du domaine « recouvrement amiable » prenant en compte les besoins exprimés par le chef du service « Contentieux » et par les intervenants.



Bloc 1 : Description des tiers (entité générique TIERS, entités spécialisées EMPRUNTEUR et CAUTION et contrainte de totalité)

Bloc 2 : Description du contrat (entités CONTRAT, VEHICULE ; associations Porter sur, Accorder)

Bloc 3 : Description des incidents de paiement (entité DATE ; associations IncidentContrat, IncidentAvenant)

Bloc 4 : Description des intervenants et de leurs missions (entités INTERVENANT, ORDRE MISSION ; associations Adresser, Concerner)

Bloc 5 : Description des avenants et des cautions (entité AVENANT – l’identifiant relatif est attendu – ; associations Compléter, Garantir)

Bloc 6 : Description des documents associés à un contrat (entité DOCUMENT NUMERISE ; association Rattacher à)

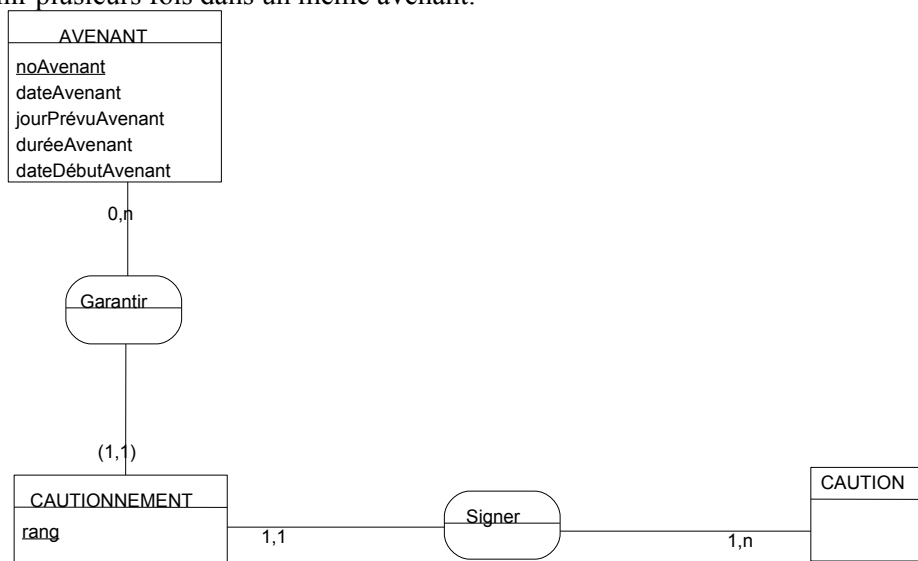
Commentaires :

- (1) Une propriété *mensualité* peut figurer dans CONTRAT même si elle peut être calculée à partir des propriétés *montantPrêt*, *tauxPrêt* et *duréeInitiale*.
Idem pour une propriété *montantÉchéanceAvenant* dans AVENANT.
Les propriétés *jourPrélèvement* et *jourPrévuAvenant* peuvent ne pas figurer sur le schéma car elles peuvent être calculées à partir respectivement des propriétés *datePremièreÉchéance* et *dateDébutAvenant*.
On acceptera la présence des propriétés *immatriculation*, *typeAchat*, soit dans l'entité VEHICULE, soit dans l'entité CONTRAT.
On acceptera une solution basée sur la notion d'échéancier ou d'engagement (*jour prélèvement*, *montant*, *durée*, *date début*), considérant qu'un avenant constitue un nouvel échéancier du contrat.
Une solution consistant à généraliser les entités CONTRAT et AVENANT en une entité DOSSIER (identifiée par une propriété spécifique) est acceptée mais l'identification relative de l'entité AVENANT n'est pas possible, une association entre les entités CONTRAT et AVENANT est nécessaire, et les entités CONTRAT et AVENANT ne doivent pas avoir d'identifiant.

- (2) On acceptera les cardinalités 1,n sur l'association « Rattacher à » côté entité DOCUMENT NUMERISE : en effet, on peut considérer qu'un même document numérisé soit repris pour des contrats différents s'ils concernent le même emprunteur.

- (3) Une contrainte peut être mentionnée sur le schéma : à une date donnée, on ne peut pas connaître à la fois un « incident contrat » et un « incident avenant » qui concerne ce contrat. La mention de cette contrainte ne sera pas exigée.
La solution liant uniquement DATE et CONTRAT par une association Incident sera admise, ainsi que la solution reposant sur deux entités INCIDENT CONTRAT et INCIDENT AVENANT, identifiées relativement à CONTRAT et AVENANT.

- (4) En choisissant une représentation basée sur une association, on garantit l'unicité du couple AVENANT-CAUTION. Mais il faudrait mentionner par ailleurs la contrainte qui concerne le rang : dans un avenant, le rang est unique, il ne saurait être partagé par plusieurs cautions liées au même avenant.
La solution suivante sera admise : ici, l'unicité du rang dans un avenant est garantie. Mais il faudrait mentionner par ailleurs la contrainte qui concerne la caution : une caution ne peut intervenir plusieurs fois dans un même avenant.

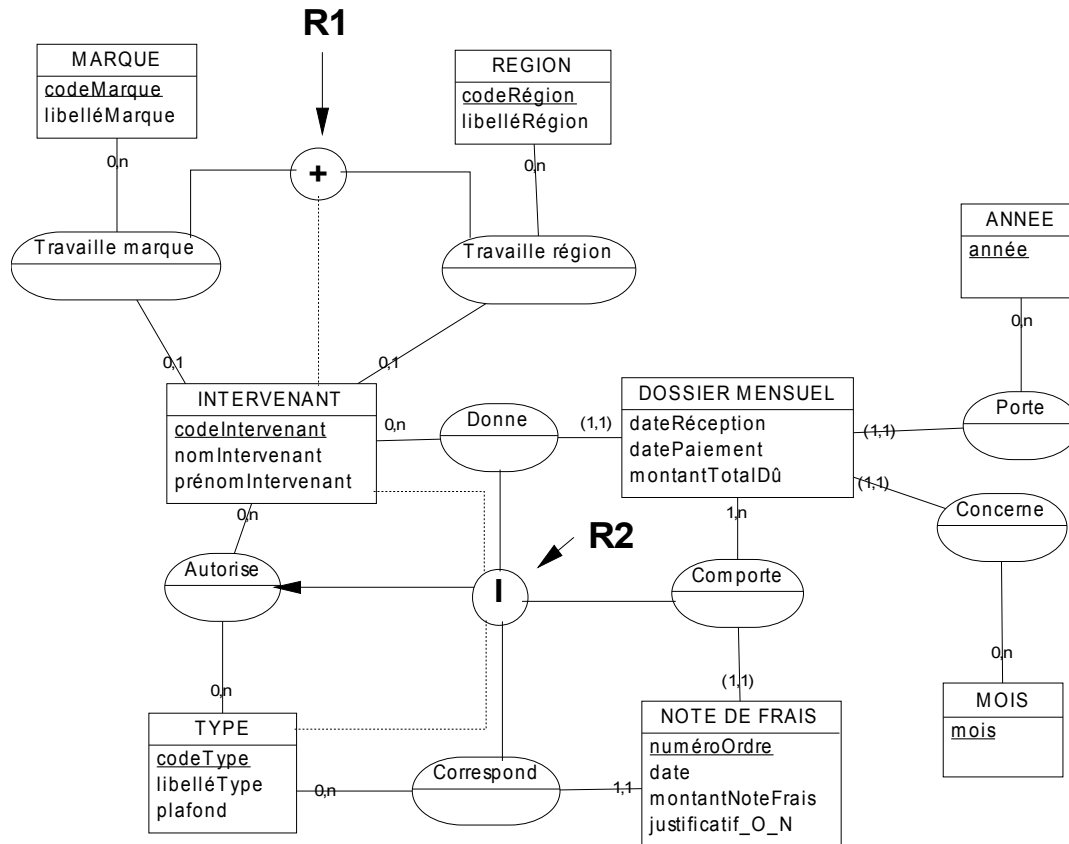


Dans les deux cas, la mention de la contrainte ne sera pas exigée.

- (5) On ne pénalisera pas le candidat qui aura présenté une association entre ORDRE MISSION et AVENANT. La mission concerne avant tout un contrat et précède l'avenant (c'est le rôle de la mission que de définir s'il doit y avoir un avenant et quelles en seront les modalités).

DOSSIER 3 | **GESTION DES FRAIS**

3.1 En utilisant l'annexe 3A, compléter le schéma entité-association présenté pour prendre en compte les règles de gestion R1 et R2.



La règle R1, exprimée par le symbole « + » peut être remplacée par les symboles « XT ». De par l'imbrication des identifications relatives entre les entités NOTE DE FRAIS, DOSSIER MENSUEL et INTERVENANT, on peut admettre une seule association *source* Correspond.

La présence des pivots ne sera pas exigée, ni pour la règle R1, ni pour la règle R2.

On acceptera une description textuelle des contraintes à condition que les noms des associations mises en jeu soient indiqués ainsi que la nature des contraintes (partition pour R1 et inclusion pour R2).

L'expression algébrique des contraintes sera également admise :

Pour R1 :

$$\text{TravailleMarque}[\text{codeIntervenant}] \cap \text{travailleRegion}[\text{codeIntervenant}] = \emptyset$$

ET

$$\text{TravailleMarque}[\text{codeIntervenant}] \cup \text{travailleRegion}[\text{codeIntervenant}] = \text{Intervenant}$$

Pour R2:

$$\text{Correspond}[\text{codeType}, \text{codeIntervenant}] \subseteq \text{Autorise}$$

3.2 En se basant sur le schéma relationnel fourni en annexe 3B, écrire en SQL les requêtes permettant de réaliser les opérations suivantes :

A. Afficher le montant total des remboursements dus à l'intervenant de code "980045" au titre de l'année 2006.

```
SELECT SUM(montantTotalDu)
FROM DOSSIERMENSUEL
WHERE codeIntervenant = '980045'
AND année = 2006
```

Remarque : On acceptera une restriction sur les champs *dateRéception* ou *datePaiement*. On n'acceptera pas un cumul sur le champ *montant* de la table NOTEDEFRAIS qui représente les montants des frais engagés qui peuvent être différents des montants dus (plafonnement). On acceptera toute expression de sélection de la date.

B. Afficher le nombre de notes de frais sans justificatif par code intervenant et par mois pour l'année 2006, liste triée par ordre croissant des codes d'intervenant et par nombre décroissant de notes de frais.

```
SELECT codeIntervenant, mois, count(*)
FROM NOTEDEFRAIS
WHERE justificatif_O_N= false
AND année = 2006
GROUP BY codeIntervenant, mois
ORDER BY codeIntervenant, 3 DESC
```

Remarque : On peut écrire ORDER BY codeIntervenant, count(*) DESC
On acceptera que la restriction sur l'année 2006 soit réalisée à partir de la colonne *date*.

C. Afficher les codes des intervenants qui n'ont déposé aucune demande de remboursement de frais en avril 2006.

```
SELECT code
FROM INTERVENANT
WHERE code NOT IN
(SELECT codeIntervenant
FROM DOSSIERMENSUEL
WHERE année = 2006
AND mois = 'avril')
```

Remarque : On ne pénalisera pas le candidat qui utilise la table NOTEDEFRAIS même s'il est précisé dans l'énoncé que le dossier est créé lors de la première note de frais. Le mois pourra apparaître sous forme numérique (4).

Autre solution avec une jointure externe gauche :

```
SELECT I.code
FROM INTERVENANT I
LEFT OUTER JOIN DOSSIERMENSUEL D
ON D.codeIntervenant = I.Code AND année = 2006 AND mois = 'avril'
WHERE D.codeIntervenant IS NULL
```

Par une sous-requête existentielle :

```
SELECT code
FROM INTERVENANT I
WHERE NOT EXISTS (SELECT *
                  FROM DOSSIERMENSUEL D
                  WHERE année = 2006 AND mois = 'avril'
                  AND D.codeIntervenant = I.code)
```

Par l'opérateur ensembliste MINUS ou EXCEPT :

```
SELECT code
FROM INTERVENANT
MINUS
SELECT codeIntervenant
FROM DOSSIERMENSUEL
WHERE année = 2006 AND mois = 'avril'
```

D. Affecter l'intervenant de code '980045' non plus à la région de code "PACA" mais à la marque de code 15, en respectant la règle R1.

```
UPDATE INTERVENANT
SET codeMarque = 15, codeRégion = Null
WHERE code = '980045'
```

3.3 Dans le cas d'un incident survenant suite à l'exécution de la troisième requête INSERT :

- **Indiquer quelles seront les lignes ajoutées dans les tables DOSSIERMENSUEL et NOTEDEFRAIS pour l'intervenant de code "980045" en avril 2006. Justifier la réponse.**

La ligne ('980045', 2006, 'avril', '02/04/2006', null, null) est ajoutée dans la table DOSSIERMENSUEL. En revanche aucune ligne ne sera ajoutée dans la table NOTEDEFRAIS ; en effet, seule l'instruction validée par le premier *commit* est effective.

- **Indiquer, en utilisant l'annexe 3A, la règle de gestion qui ne sera pas respectée dans ce cas.**

Dans le dossier, on précise que « *Un dossier mensuel de notes de frais est alors créé par intervenant lors de la saisie de la première note de frais du mois pour l'intervenant* » et sur le schéma conceptuel des données, le rôle joué par l'entité DOSSIERMENSUEL dans l'association Comporte possède une cardinalité minimale égale à 1 : pour un dossier, il y a toujours au moins une note de frais ; c'est cette règle de gestion qui ne sera donc pas respectée.

- **Proposer un réaménagement de la séquence d'ordres SQL figurant ci-dessus afin de résoudre ce problème.**

Gérer une seule transaction : elle débutera avant l'insertion du dossier mensuel et sera validée après l'insertion de toutes les notes de frais. Il faut donc supprimer les instructions COMMIT et BEGIN se trouvant après l'insertion de la ligne dans DOSSIER.

3.4 Écrire l'algorithme de la fonction *totalJour()* en utilisant les fonctions disponibles.

Fonction totalJour (mtNuit : Réel, mtRepas1 : Réel, mtRepas2 : Réel) : Réel

VAR vAPayerJ, plafNuit, plafRepas : Réel

Début

plafNuit ← getPlafond ("N")

plafRepas ← getPlafond ("R")

si mtRepas2 <> 0

alors vAPayerJ ← minimum(mtRepas1 + mtRepas2 , 2 * plafRepas * 1.1)

sinon vAPayerJ ← minimum(mtRepas1, plafRepas)

fsi

vAPayerJ ← vAPayerJ + minimum(mtNuit, plafNuit)

retourner vAPayerJ // ou totalJour ← vAPayerJ

Fin

Remarque : d'autres solutions et d'autres modes de représentation algorithmique sont possibles.

DOSSIER 4

GESTION DES RECOUVREMENTS PAR RÉGION

4.1 Écrire la méthode *getNbMaxRecouvrements* de la classe *Région*.

Fonction Région::getNbMaxRecouvrements() : Entier // D'autres écritures de la déclaration de la fonction sont possibles

VAR i, nb, max : Entier

Début

 nb ← mesStats.cardinal()

 max ← 0

 Pour i de 1 JQA nb

 Si mesStats.extraireObjet(i).getNbRecouvrements() > max

 Alors max ← mesStats.extraireObjet(i).getNbRecouvrements()

 Finsi

 FinPour

 retourner max

Fin

On acceptera l'initialisation suivante de max :

 max ← mesStats.extraireObjet(1).getNbRecouvrements()

 bien que cette initialisation ne prenne pas en compte le cas de la collection *mesStats* vide.

On acceptera le parcours de la collection par une instruction type "*For each*".

4.2 Écrire la méthode *addStat* de la classe *Région*.

Procédure Région::addStat(unLibModèle : Chaîne, unLibMarque : Chaîne, unNbPrêts : Entier, unNbRecouv : Entier)

VAR laStat : Stat

Début

 laStat.init(unLibModèle, unLibMarque, unNbPrêts, unNbRecouv)

 mesStats.ajouter(laStat)

Fin

4.3 En utilisant la classe *Passerelle*, écrire l'instruction d'affectation permettant de valoriser un objet *uneRégion* de la classe *Région*, correspondant à la région de code "PACA".

```
uneRégion ← Passerelle.donneLaRégion("PACA")
```

Remarque : On accepte une *syntaxe C++* : `Passerelle::donneLaRegion("PACA") ;`

4.4 Écrire la méthode *chargeLesStats* de la classe *Passerelle*.

```
Procédure Passerelle::chargeLesStats(uneRégion : Région)
VAR  jeuStats : JeuEnregistrements
     requête : Chaîne
Début
    requête ← "SELECT V_libMod, V_libMarq, V_nbPrêt, V_nbRec
              FROM vRecouvrement
              WHERE V_codeRég = "
              + uneRégion.getCode() + ""
    jeuStats.init(requête)
    tant que non jeuStats.fin()
        uneRégion.addStat(jeuStats.getValeur("V_libMod"),
                          jeuStats.getValeur("V_libMarq"),
                          jeuStats.getValeur("V_nbPrêt"),
                          jeuStats.getValeur("V_nbRec"))
        jeuStats.lireSuivant()
    ftq
    jeuStats.fermer()
Fin
```

4.5 Écrire la méthode *chargeCaractéristiquesRégion* de la classe *Passerelle*.

```
Fonction Passerelle::chargeCaractéristiquesRégion(code : Chaîne) : Région
VAR  jeuRégion : JeuEnregistrements
     requête : Chaîne
     maRégion : Région
Début
    maRégion ← null
    requête ← "SELECT distinct V_codeRég, V_libRég FROM vRecouvrement
              WHERE V_codeRég = " + code + ""
    jeuRégion.init(requête)
    si non jeuRégion.fin()
        alors // 1 enregistrement présent dans le résultat
            maRégion.init(jeuRégion.getValeur("V_codeRég"),
                          jeuRégion.getValeur("V_libRég"))
    fsi
    jeuRégion.fermer()
    retourner maRégion
Fin
```

Remarque : On acceptera d'autres syntaxes concernant la concaténation de chaînes.

- La clause DISTINCT de la requête SQL n'est pas obligatoire. En son absence, le jeu résultat obtenu pourra contenir plusieurs lignes identiques.