**CORRIGÉ (LES RÉPONSES APPARAISSENT AVEC CE FOND DE CARACTÈRE)**

U.B.S. - I.U.T. de Vannes

## Département Informatique

Contrôle de 2e année
Date : 11/03/2015

M4102C - Programmation répartie

**Devoir surveillé**

Responsable(s) et auteur(s) : F. Merciol – D. Lesage

Documents personnels et calculatrice autorisés
Téléphone interdit — Durée : 2h
Les barèmes sont donnés à titre indicatif — 12 page(s) de texte



---

**Conseils :**

- En cas de doute sur un énoncé, ne perdez pas de temps à demander aux surveillants, **faîtes une hypothèse** pour lever le doute et indiquez la dans votre copie.
- Indiquez **votre nom** sur chaque feuille à rendre **dès qu'elles vous sont données**.
- Afin d'éviter la copie, toute réponse non justifiée sera considérée comme nulle.
- **Les parties sont indépendantes** (commencez par la plus simple pour vous).
- Il est demandé des réponses à la fois claires et concises.
- **Lisez en entier** le contrôle avant de commencer à répondre.
- **Ne restez pas bloqué**, vous pourrez revenir sur une question difficile par la suite.
- **Ne brûlez pas toute votre énergie**, des calculs longs peuvent vous rapporter moins de points que la réponse à des questions de réflexion.
- **Conservez du temps pour chaque partie**
  (120 min / 20 pts => pas plus de 6 minutes par point)

---

Le devoir se compose de **3 parties à rendre séparément** (avec votre nom en tête) :

1) [4 pts] questions de cours (QCM : Questions à Choix Multiple)
2) [8 pts] problème 1
3) [8 pts] utilisation des sockets (en Java)

Ce corrigé est donné à titre indicatif. D'autres réponses peuvent être considérés comme justes.

**CORRIGÉ (LES RÉPONSES APPARAISSENT AVEC CE FOND DE CARACTÈRE)**

**NOM : _____ PRENOM : _____ GROUPE TD : ___ Note :**

## 1.) [4 pts] question de cours (QCM : Questions à Choix Multiple)

✗ Cochez la case réponse

| Question | A | B | C | D |
|---|---|---|---|---|
| Par quoi est encadré une définition d'un objet ?<br>A) ' ' B) [ ] C) { } D) " " | | | ○ | |
| Qu'est-ce qui n'est pas une méthode HTML ?<br>A) PUT B) POST C) LIST D) GET | | | ○ | |
| Lequel de ces logiciels est un serveur web ?<br>A) firefox B) apache C) coyote D) panda roux | | ○ | | |
| Quelle boucle n'existe pas en Javascript ?<br>A) while B) repeat C) do D) for | | ○ | | |
| Quelles informations doit-on fournir pour se connecter à un socket distant ?<br>A) service seulement B) port seulement C) adresse IP et port D) routeur et service | | | ○ | |
| Que signifie AJAX ?<br>A) Asynchronous JavaScript and XML eXchange B) logiciel aéronautique C) A JavaScript Asynchronous D) A JavaScript Analyser for XML | ○ | | | |
| Comment se nomme, en javascript, l'attribut class gérant les CSS ?<br>A) className B) class C) css D) cssClasses | ○ | | | |
| Où ne peut on pas trouver de code javascript en cours d'exécution ?<br>A) un fichier JS séparé B) un attribut HTML C) une balise HTML dédiée D) une image | | | | ○ |
| Que faut-il utiliser en Java pour pouvoir gérer plusieurs clients simultanément ?<br>A) un portefeuille B) des priorités C) un protocole adapté D) des tâches | | | | ○ |
| Qu'est-ce qui n'est pas contenu dans une requête HTTP d'un client ?<br>A) des données B) le protocole utilisé C) un code d'erreur D) un contexte | | | ○ | |
| Quel est l'opérateur pour réaliser un test d'égalité sans conversion implicite ?<br>A) = B) == C) === D) =?= | | | ○ | |
| Qu'est-ce qui ne peut être utilisé comme index de tableaux ?<br>A) entier positif B) des chaînes de caractères C) mot clef du langage D) entier négatif | | | ○ | |
| Que signifie RMI ?<br>A) Rappelle des Méthodes Interactives B) Remote Method Invocation C) Informaticiens mal payés D) Recall Method Interface | | ○ | | |
| Quel programme connecte un socket à l'écran et au clavier ?<br>A) mail B) route C) telnet D) ifconfig | | | ○ | |
| Comment considère-t-on une variable sans déclaration dans une fonction ?<br>A) attribut B) local C) global D) erreur | ○ | | | |
| Pour communiquer via l'Internet, qu'utilisent les applications ?<br>A) des pushs B) des chaussettes C) des sockets D) des pulls | | | ○ | |
| Que permet de parcourir l'instruction for (x in y) ?<br>A) entier dans un intervalle B) attributs d'un objet C) éléments d'un tableau D) les caractères d'une chaîne | | ○ | | |
| Quelle méthode permet d'attendre un client ?<br>A) sudoku B) wait C) accept D) suspend | | | ○ | |
| Que signifie DOM ?<br>A) Document Object Model B) Document Oriented Method C) Dominique (prénom de l'inventeur) D) Data Or Method | ○ | | | |
| Comment se nomme l'élément suivant au même niveau dans un DOM ?<br>A) Sister B) Brother C) Sibling D) Child | | | ○ | |

**Corrigé (Les réponses apparaissent avec ce fond de caractère)**

**CORRIGÉ (LES RÉPONSES APPARAISSENT AVEC CE FOND DE CARACTÈRE)**

**2.) [8 pts] problème 1**

La méthode Java suivante est utilisée par un « servlet » permettant de retourner éventuellement à un client HTTP des « évènements » sous la forme d'un tableau d'objets au format JSON.

```
private void getEvents(HttpResponse resp) throws IOException {
   AppEvent appEvent;
   JsonObjectBuilder event;
   JsonArrayBuilder events;
   long timeout;
   boolean again;
   int nEvents;

   events=Json.createArrayBuilder();
   timeout=0L;
   appEvent=null;
   again=true;
   nEvents=0;
   do {
      try {
         if ((appEvent=eventsQueue.poll(timeout,TimeUnit.MILLISECONDS))!=null) {
            event=Json.createObjectBuilder();
            event.add("evt",String.valueOf(appEvent.getCode()));
            event.add("dateTime",appEvent.getDateTimeAsString());
            event.add("data",appEvent.getData());
            event.add("error",appEvent.isError());
            events.add(event);
            nEvents++;
            timeout=0L;
         }
         else {
            if (timeout==0L) {
               if (nEvents==0) timeout=SERVLET_TIMEOUT;
               else again=false;
            } else again=false;
         }
      }
      catch (InterruptedException e) { }
   } while (again);

   resp.setHeader("Content-Type","application/json");
   resp.setEntity(new StringEntity(events.build().toString()));
}
```

**CORRIGÉ (LES RÉPONSES APPARAISSENT AVEC CE FOND DE CARACTÈRE)**

**NOM : _____ PRENOM : _____ GROUPE TD : ___ Note :**

**2.A) [1 pt]** Décrivez comment la méthode « getEvents » gère la présence (ou l'absence) d'évènements dans la queue « eventQueue » (la méthode « poll » retourne un évènement disponible dans la queue, ou « null » si le hors temps passé en paramètre s'est écoulé avant qu'un évènement ne soit disponible dans la queue).

Dans la méthode présentée, une première tentative de chargement d'évènement depuis la queue est réalisée, sans que celle-ci ne soit bloquante (hors temps nul).

Si un premier évènement a été immédiatement chargé, la prochaine tentative de chargement se fera de nouveau en utilisant un hors temps nul, sinon, une nouvelle tentative de chargement est réalisée en utilisant un hors temps prédéfini.

Dans tous les cas, une nouvelle tentative de chargement d'évènement est réalisée avec un hors temps nul si un évènement a été chargé précédemment, quelque soit la valeur de hors temps utilisée.

**2.B) [1 pt]** Quelle stratégie le programmeur de cette méthode a-t-il voulu mettre en œuvre en ce qui concerne le temps de réponse de son « servlet » ? Quelles conséquences ce choix a-t-il sur le programme appelant ?

Le programmeur a choisi de faire répondre son « servlet » dès que des évènements sont disponibles ou que le hors temps SERVLET_TIMEOUT est écoulé. Aussi, le logiciel client interrogeant le « servlet » doit être en capacité de gérer de « longs » temps de réponse.

**CORRIGÉ (LES RÉPONSES APPARAISSENT AVEC CE FOND DE CARACTÈRE)**

**2.C) [3 pt]** Votre chef de projet vous demande de faire évoluer cette méthode afin qu'elle retourne au plus un évènement, et ceci sans que le client n'ait à patienter ... Ecrivez cette nouvelle version.

Si on considère que l'on va retourner un tableau contenant 0 ou 1 élément …

```
private void getEvents(HttpResponse resp) throws IOException
{
   AppEvent appEvent;
   JsonObjectBuilder event;
   JsonArrayBuilder events;

   events=Json.createArrayBuilder();
   try {
      if ((appEvent=eventsQueue.poll(0L,TimeUnit.MILLISECONDS))!=null) {
         event=Json.createObjectBuilder();
         event.add("evt",String.valueOf(appEvent.getCode()));
         event.add("dateTime",appEvent.getDateTimeAsString()));
         event.add("data",appEvent.getData());
         event.add("error",appEvent.isError());
         events.add(event);
      }
   }
   catch (InterruptedException e) { }

   resp.setHeader("Content-Type","application/json");
   resp.setEntity(new StringEntity(events.build().toString()));
}
```

**Si on considère que l'on va retourner un élément ou rien …**

```
private void getEvents(HttpResponse resp) throws IOException
{
   AppEvent appEvent;
   JsonObjectBuilder event;

   try {
      if ((appEvent=eventsQueue.poll(0L,TimeUnit.MILLISECONDS))!=null) {
         event=Json.createObjectBuilder();
         event.add("evt",String.valueOf(appEvent.getCode()));
         event.add("dateTime",appEvent.getDateTimeAsString()));
         event.add("data",appEvent.getData());
         event.add("error",appEvent.isError());

         resp.setHeader("Content-Type","application/json");
         resp.setEntity(new StringEntity(event.build().toString()));
      }
   }
   catch (InterruptedException e) { }
}
```

**Corrigé (Les réponses apparaissent avec ce fond de caractère)**

**2.D) [1 pt]** Que pensez-vous de cette consigne du chef de projet ? Quelles conséquences peut-elle avoir sur le fonctionnement du logiciel client interrogeant cette nouvelle version du « servlet » depuis une page HTML ?

La mise en œuvre de cette nouvelle version entraîne, côté client, de considérer des temps de réponse immédiats. La succession d'appels au « servlet » peut entraîner une surcharge côté client (comme côté serveur d'ailleurs) et gêner l'exécution d'autres processus (surcharge CPU). Typiquement, si le « servlet » est interrogé « en boucle » depuis un navigateur (depuis une page HTML), les fonctionnalités interactives éventuellement disponibles dans cette même page pourraient ne plus être très réactives …

**Corrigé (Les réponses apparaissent avec ce fond de caractère)**

**2.E) [1 pt]** Le « servlet » est donc appelé depuis une page HTML proposant un affichage des évènements reçus. Les appels successifs au « servlet » et l'affichage des données associées aux évènements récupérés sont réalisés sans que la page n'ait besoin d'être rechargée, à l'aide de la technique « Ajax ». Précisez la construction d'un appel « Ajax » en Javascript en utilisant une instance de l'objet XMLHttpRequest (vous indiquerez en particulier comment traiter l'issue de la requête HTTP exécutée dans le cas de l'appel à notre « servlet »).

**[1 pt]** Vous précisez ce que peut apporter l'utilisation de la librairie « jQuery » dans la gestion d'appels « Ajax » et plus particulièrement dans notre cas de figure.

En utilisant une instance d'objet XMLHttpRequest …

Création de l'instance de l'instance
Initialisation de l'attribut onreadystatechange permettant l'exécution de code spécifique (gestion des différents états de l'automate de gestion de requête, une attention particulière étant réservée à l'état 4)
Appel de la fonction membre open (définition du type de requête, précision de l'URL à utiliser, mode de gestion de la requête – synchrone ou asynchrone)
Emission de la requête (appel de la fonction membre send)

Lors de la gestion de l'état 4 (valeur contenue dans l'attribut readyState de l'objet XMLHttpRequest), il conviendra de vérifier que le serveur a retourné un code HTTP 200 et il sera nécessaire de convertir les données reçues (au format JSON) afin de faciliter leur manipulation (utilisation de la fonction JSON.parse).

En utilisant la librairie jQuery, il est possible d'utiliser la fonction $.ajax qui décharge le programmeur de la création d'une instance d'objet XMLHttpRequest et qui simplifie le suivi d'exécution de la requête. De même, en précisant (lors de l'appel à la fonction $.ajax) que des données au format JSON sont attendue, il n'est plus nécessaire d'assurer de conversion, les données reçues pouvant être immédiatement manipulées comme des objets Javascript.

CORRIGÉ (LES RÉPONSES APPARAISSENT AVEC CE FOND DE CARACTÈRE)

**NOM : _____ PRENOM : _____ GROUPE TD : ___ Note :**

**3.) [8 pts] utilisation des sockets (en Java)**

On se situe dans le contexte d'un groupe d'étudiants qui sont réunis pour une activité en réseau (développent de projet de synthèse, jeu en réseau, ...). Ils ont déjà rencontré le besoin de mettre en place une messagerie instantanée et ont développé le TP1 intitulé « Clavardage ». Aujourd'hui, ils sont confrontés à un nouveau défi : se mettre d'accord ensemble quand il y a des choix à faire (quel schéma UML parmi 5 propositions ? Quelles armées lancer contre ? …). C'est but du logiciel à résoudre ici.

**Compréhension du logiciel**

On veut réaliser un logiciel de sondage instantané. Le logiciel résultant doit avoir les propriétés suivantes :

- Le programme est écrit en Java
- Il ouvre un port en mode serveur
- Il accepte de nombreux clients
- Les clients sont anonymes (ils ne donnent pas de nom)
- N'importe quel client peut demander un sondage en indiquant le nombre maximum de choix
- Chaque client ne peut voter qu'une seule fois
- A chaque vote, tous les clients reçoivent un état du sondage ( % de votant et décompte des votes par choix)
- Lorsqu'un vote est en cours, personne ne peut en ouvrir un nouveau
- N'importe qui peut demander à clore le sondage. Tous les clients reçoivent alors un résultat du vote (indice le plus bas ayant reçut le maximum de suffrage).

**3.A) [½ pt]** Combien de tâches doivent être mise en œuvre ?

```
- Une tâche devra attendre la connexion de nouveau client.
- Une tâche sera dédiée à chaque client pour attendre une commande
Il n'est pas utile d'avoir une tâche pour la transmission les résultats.
La tâche du client qui émet une commande est suffisante.
```

**3.B) [1 pt]** Décrivez en pseudo-code le comportement des différentes tâches.

```
Tâche serveur :
pour toujours
s = acceptUnClient
(in, out) = trouveLesEntreeSortie (s)
client = creeClient (in , out )
ajouteClient ( client )
lanceTacheCLient
```

```
Tâche client :
pour toujours
ligne = attendreLigne (in)
traiteCommande (ligne)
```

```
serveur.envoiEffetAuxAutres (moi, ligne)
```

**CORRIGÉ (LES RÉPONSES APPARAISSENT AVEC CE FOND DE CARACTÈRE)**

Le serveur possède 3 attributs :

```
    public Vector<Client> clients = new Vector<Client> ();
    ArrayList<Client> expressed;
    int[] opinion;
```

**3.C) [½ pt]** Indiquez le rôle de chaque attribut ?

clients : l'ensemble des utilisateurs connecté avec les entrées/sorties

expressed : les clients ayent exprimé un vote pour le sondage en cour

opinion : l'ensemble des résultats par option

**3.D) [1 pt]** Pourquoi ces attributs sont-ils suffisants pour :

- savoir si un sondage est en cours,
- connaître le pourcentage de vote exprimé et
- déterminer le résultat ?

Un sondage est en cours si il exsite un tableau de résultat

c'est le nombre de votants (expressed.size ()) sur le nombre de participants (clients.size ())

Il faut chercher le rang dans opinion qui a la plus grande valeur

Voici la méthode d'analyse du client :

```
   public boolean analyse (String line)
      throws IOException{
      if (line == null || line.equals ("sort")) {
         removeClient (this);
         out.close ();
         in.close ();
         return true;
      } else if (line.equals ("fin")) {
         System.exit (0);
      } else if (line.startsWith ("question ")) {
         open (Integer.parseInt (line.substring (9)));
      } else if (line.startsWith ("opte ")) {
         choose (Integer.parseInt (line.substring (8)));
      } else if (line.startsWith ("ferme")) {
         close ();
      }
      return false;
   }
```

On considère que les méthodes « open » et « close » sont dans la classe du serveur et que la méthode « choose » est dans celle du client.

**CORRIGÉ (LES RÉPONSES APPARAISSENT AVEC CE FOND DE CARACTÈRE)**

**3.E) [2 pt]** Ecrivez la classe du client dont le constructeur prend un « socket » en argument (n'écrivez pas les attributs et méthodes déjà donnés).

```
       public class Client extends Thread {
              PrintStream out;
              BufferedReader in;

              public Client (Socket call) {
                     try {
                            in = new BufferedReader (new InputStreamReader
(call.getInputStream ()));
                            out = new PrintStream (call.getOutputStream ());
                            start ();
                     } catch (IOException e) {
                     }
              }

              public void run () {
                     try {
                            addClient (this);
                            for (;;)
                                   if (analyse (in.readLine ()))
                                          break;
                     } catch (IOException e) {
                     }
              }
              public void choose (int choice) {
                     synchronized (expressed) {
                            if (expressed == null || expressed.contains (this))
                                   return;
                            opinion[choice]++;
                            expressed.add (this);
                            showCurrent ();
                            if (expressed.size () == clients.size ())
                                   close ();
                     }
              }
    }
}
```

Voici des méthodes de la classe du serveur :

```
   public synchronized void broadcast (String line) {
       String date = dateFormat.format (new Date ());
       for (Client listener : clients)
          listener.out.println (date+": " + line);
   }
   public void showResult () {
       int maxVal = 0, maxPos = 0;
       for (int i = 0; i < opinion.length; i++)
       if (opinion [i] > maxVal) {
          maxPos = i;
          maxVal = opinion [maxPos];
       }
       broadcast ("Sondage fini. Le choix: "+maxPos);
   }
   public void showCurrent () {
       broadcast (""+((expressed.size ()*100)/clients.size ())+"% "+
                  Arrays.toString (opinion));
   }
```

**Corrigé (Les réponses apparaissent avec ce fond de caractère)**

**3.F) [3 pt]** Ecrivez la classe du serveur avec un lanceur qui prend un numéro de port en argument (n'écrivez pas les attributs et méthodes déjà donnés).

```
    public static void main (String [] arg) {
            if (arg.length != 1) {
                System.err.println ("Usage: java Sondage port");
                System.exit (1);
            }
            new Sondage (Integer.parseInt (arg [0]));
    }
    public synchronized void addClient (Client client) {
            clients.add (client);
    }

    public synchronized void removeClient (Client client) {
            clients.remove (client);
    }
    public Sondage (int port) {
            try {
                ServerSocket server = new ServerSocket (port);
                for (;;)
                    new Client (server.accept ());
            } catch (IOException e) {
            }
    }
       public void open (int nbChoice) {
            synchronized (expressed) {
                if (expressed != null)
                    return;
                expressed = new ArrayList<Client> ();
                opinion = new int [nbChoice];
                broadcast ("Votre avis parmit [0.."+(nbChoice-1)+"]");
            }
       }

       public void close () {
            synchronized (expressed) {
                if (expressed == null)
                    return;
                showResult ();
                expressed = null;
                opinion = null;
            }
       }
```