

Attention : ceci est un exemple. Les réponses sont évidemment fausses.

Les cadres en rouge sont à supprimer :-)

Ce TP a été réalisé suite au chapitre sur les sockets.

Sachez utiliser ce texte pour vous.
Vous avez droit à vos rapports en CT.

1 Exercices de compréhension

1.1 Donnez la suite d'instructions pour rechercher la page <http://m3101.merciol.fr/debut> sans proxy

J'ai utilisé gimp et j'ai saisi dans la console de texte suivant :

```
$ java-jar toto.jar
java-jar : commande introuvable
$ java -jar toto.jar
```

J'ai également créé un répertoire pour ce TP

1.2 Donnez la suite d'instructions pour rechercher la page <http://m3101.merciol.fr/debut> avec le proxy squidva.univ-ubs.fr:3128

après plusieurs tentatives j'ai utilisé le paramètre réseau de la calculatrice

```
menu => préférence => pluton
```

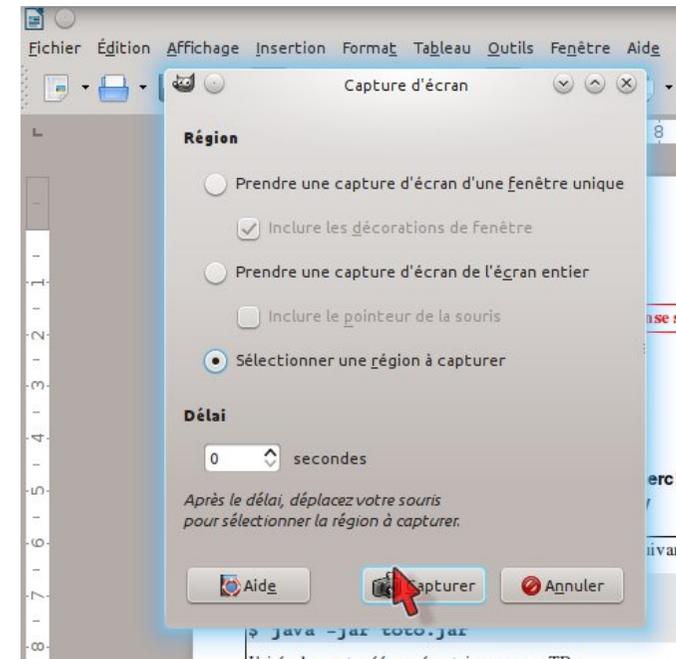
Là j'ai sélectionné l'option : « café »

2 Validation des outils

2.1 Téléchargez le programme recvcmd.java et testez le avec le port 8080. Comment vous connectez-vous à lui ?

J'ai lancé libreoffice est j'ai obtenu le résultat suivant :

Nom du module



Ne comptez pas sur la couleur dans des graphiques.
Vous imprimez les rapports en noir et blanc.

3 Réalisation du logiciel

3.1 Indiquez le code permettant d'initialiser le client (avec son nom)

L'année 2015/2016 permet à l'enseignant de mettre le papier à recyclé le moment venu.

La date de rendu sera rempli par l'enseignant.

Le n° de TP ET le nom du TP vont permettre de ne pas mélanger les rendus et de contrôler que les rendus sont faits.

Mettez votre nom ! (1 % des rendus n'ont pas d'auteur et ce ne sont jamais les mêmes).

n° de page ET nombre de page

Nom de famille en majuscule en cas de confusion possible.

3.2 Indiquez le code permettant au client d'attendre un message

Le nom du module est indispensable, car votre enseignant assure plusieurs cours.

Le numéro de page est indispensable, car vous ne savez pas si vous pourrez les agraffer.

3.3 Indiquez le code permettant d'envoyer le message aux autres clients

Ici, on choisit la de redimensionner en fonction de la taille la plus grande.

```

5 public ImageIcon getSide (int max) {
    int width = original.getWidth ();
    int height = original.getHeight ();
    return get (width>height ? new Dimension (max,
    (height*max)/width) : new Dimension ((width*max)/height, max),
10 0);
}

```

3.4 Indiquez le code permettant au serveur d'attendre un client

Même remarque que précédemment.

```

15 public ImageIcon get (DimensionDouble real, double scale,
    double theta) {
    return get (new Dimension ((int)(real.getWidth ()*scale),
    (int)(real.getHeight ()*scale)), 0);
}

```

3.5 Indiquez le code permettant de gérer les clients

Nous sommes obligés d'utiliser une matrice de transformation pour la rotation centrée sur l'icône.

```

public AffineTransform getAffineTransform (Dimension2D
size, double theta) {

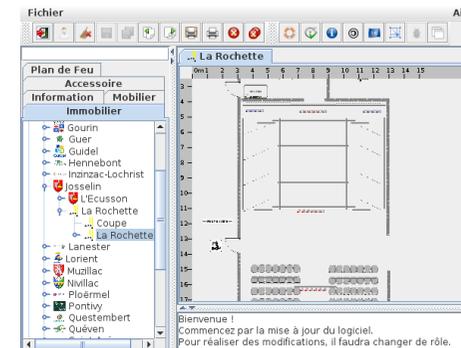
```

```

int width = original.getWidth ();
int height = original.getHeight ();
double scaleX = size.getWidth () / width;
double scaleY = size.getHeight () / height;
if (scaleX == 0 || scaleY == 0)
    return null;
if (horizontalSpin)
    scaleX = -scaleX;
if (verticalSpin)
    scaleY = -scaleY;
AffineTransform at = AffineTransform.getRotateInstance
(Math.toRadians ((int) Math.toDegrees (theta)));
at.scale (scaleX, scaleY);
at.translate (-width/2., -height/2.);
return at;
}

```

3.6 Donnez des exemples de fonctionnement



3.7 Quelles sont les limites ? Quelles sont les améliorations à apporter ?

Dépassé les bornes, il n'y a plus de limite.

Malgré sa perfection, il est possible d'ajouter un distributeur de café.