Exo1:

1/ La classe A lit un fichier texte contenant des symptômes, compte combien de fois chaque symptôme apparaît, et stocke ces informations dans une HashMap.

Exo2:

1/ ImageIO: permet de lire et d'écrire des images dans différents formats de fichiers, ce qui est utile pour de nombreuses applications graphiques ou de traitement d'images.

2/

- **read(File input) :** Cette méthode lit une image à partir d'un fichier. Elle prend en entrée un objet File représentant le fichier image à lire, et retourne un objet BufferedImage contenant les données de l'image.
- **write(RenderedImage im, String formatName, OutputStream output) :** Cette méthode écrit une image dans un flux de sortie. Le premier argument **im** est l'objet RenderedImage (par exemple, BufferedImage) à écrire. Le deuxième argument **formatName** est une chaîne représentant le format de fichier à utiliser (par exemple, "jpg", "png", etc.). Le troisième argument **output** est un OutputStream dans lequel écrire les données de l'image.
- **createImageInputStream(Object input) :** Cette méthode crée un ImageInputStream à partir d'une source d'entrée spécifiée. L'argument **input** peut être un File, un InputStream, une URL, etc. Un ImageInputStream est utilisé pour lire les données d'une image de manière efficace.
- **read(InputStream input) :** Cette méthode lit une image à partir d'un flux d'entrée (InputStream). Elle prend en entrée un objet **InputStream** représentant le flux à partir duquel lire les données de l'image, et retourne un objet BufferedImage contenant les données de l'image. C'est la méthode utilisée dans votre code initial.
- **write(RenderedImage im, String formatName, File output) :** Cette méthode écrit une image dans un fichier. Le premier argument **im** est l'objet RenderedImage (par exemple, BufferedImage) à écrire. Le deuxième argument **formatName** est une chaîne représentant le format de fichier à utiliser (par exemple, "jpg", "png", etc.). Le troisième argument **output** est un objet File représentant le fichier dans lequel écrire les données de l'image. C'est la méthode utilisée dans votre code initial.

3/ Cette application permet de capturer une région de l'écran côté client, d'envoyer cette capture au serveur via un socket, et d'afficher la capture d'écran reçue dans une fenêtre côté serveur.

4/

- La classe **Robot** peut également être utilisée pour capturer une partie ou la totalité de l'écran.

- Dans le code fourni, un objet Rectangle est créé avec les coordonnées (0, 0) pour le coin supérieur gauche, une largeur de 500 pixels et une hauteur de 300 pixels. Cet objet Rectangle est passé à la méthode createScreenCapture de la classe Robot pour spécifier la région de l'écran à capturer.

- la classe **Rectangle** est utilisée pour définir la zone rectangulaire à capturer.

Exo3:

1/ Cette méthode copie le contenu d'un flux d'entrée (InputStream) vers un flux de sortie (OutputStream) par blocs de 1024 bytes en utilisant un tampon (buf) pour stocker temporairement les données lues.

2/ Ce code permet à un client d'envoyer un fichier "test1.txt" au serveur, qui le reçoit et écrit son contenu dans un fichier "test2.txt"