

E0345MS20060327

EJ05

JAVA PROGRAMMATION - EXERCICES

Exo01 Outils du JDK en mode commande.....	3
Exo02 Récupération des arguments de la ligne de commande. Manipulation de tableau. Utilisation d'un objet.....	4
Exo03 Implémentation de la classe Personne. Constructeur, getter, setter, redéfinition de méthodes. Tableau d'objets. Méthode statique.....	5
Exo04 Package et encapsulation.....	7
Exo05 Héritage et polymorphisme.....	9
Exo06 Interface et classe interne.....	11
Exo07 Exception.....	13
Exo08 Map, casting et java.IO.....	15
Exo09 AWT , Frame , Listener et Event.....	17
Exo10 Applet.....	18
Exo11 java.IO, FilterReader et pattern décorateur.....	19
Exo12 Thread, java.net , javax.swing , java.io.Serializable.....	21

Préalable .

Pour chaque exercice consulter les commentaires dans les sources et plus particulièrement ceux marqués `//=>`

Exo01 Outils du JDK en mode commande.

Par rapport au répertoire racine du projet EJ05Modele, les sources java sont dans `./src`, les classes dans `./bin`

Dans la suite, on nomme ce répertoire : "racine"

Classes en jeu :



A réaliser :

1. Ouvrir une fenêtre DOS
2. Définir le PATH afin d'utiliser le bon JDK
3. Se positionner dans le répertoire "racine" du projet EJ05Modele
4. Compiler `exo01/HelloWorld.java` de sorte que la classe `exo01.HelloWorld` se retrouve dans le répertoire "racine"\`bin`
5. Exécuter `exo01.HelloWorld`
6. Examiner `exo01.HelloWorld` (avec `javap`)
7. Exécuter une classe inexistante (`exo01.helloworld` par exemple) et examiner le résultat.

Sources :

```
/**
 * Exo01
 * class exo01.HelloWorld
 */
package exo01;

public class HelloWorld {

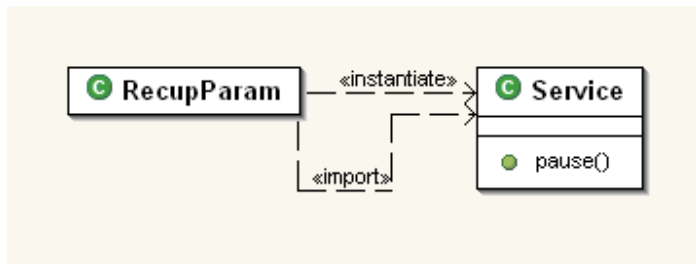
    public static void main(String[] args) {
        System.out.println("Ceci est un Hello World Java !");
    }
}
```

Exo02 Récupération des arguments de la ligne de commande.

Manipulation de tableau.

Utilisation d'un objet.

Classes en jeu :



A réaliser :

1. Compléter la méthode main de RecupParam.
Voir les commentaires dans le source.
2. Dans Eclipse, faire 2 "run configuration".
Recuparam(1) sans paramètre.
Recuparam(2) avec 3 paramètres: paul gazon 1234
3. Tester avec Recuparam(1).

```

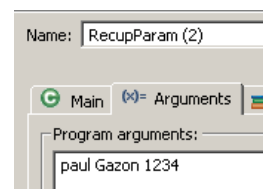
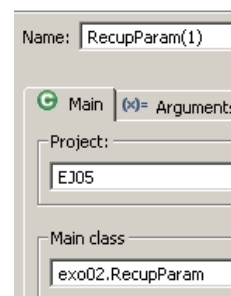
Sur System.out
Vous n'avez pas passé d'argument
Exo2 terminé !
  
```

4. Tester avec Recuparam(2).

```

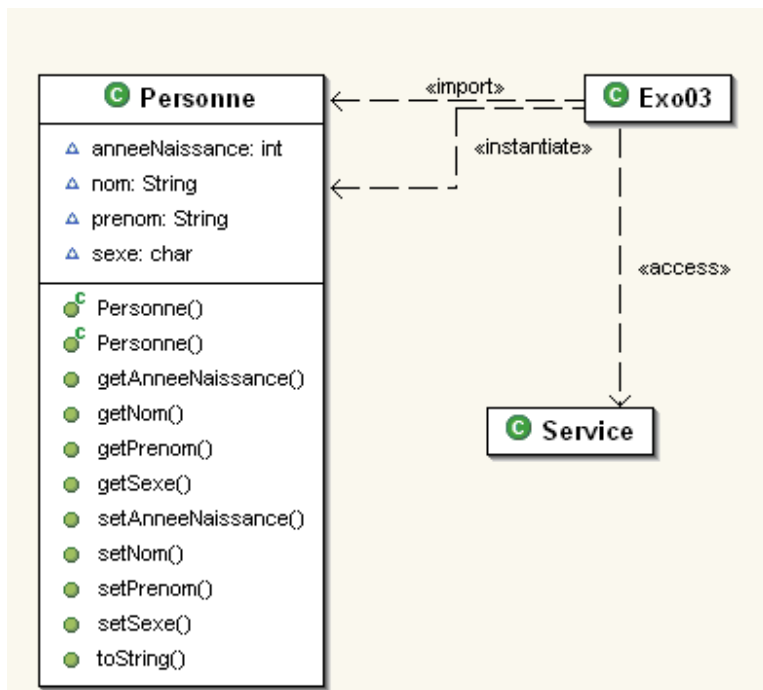
Sur System.out
Vous avez passé 3 argument(s)
Frappé la touche <entree> pour continuer

argument[0]: paul
argument[1]: Gazon
argument[2]: 1234
Exo2 terminé !
  
```



Exo03 Implémentation de la classe Personne.
Constructeur, getter, setter, redéfinition de méthodes.
Tableau d'objets.
Méthode statique.

Classes en jeu :



A réaliser :

1. Compléter la classe Personne. Voir diagramme et commentaires dans le source.
2. Modifier la classe Service. Rendre la méthode pause "static".
3. Compléter la méthode main de la classe Exo03.
4. Faire un "run configuration" Exo03 avec 4 arguments et tester.

```

Sur System .out
EX03
====
tab_p[0]==> exo03.Personne@1add2dd|victor|zoe|2000|f
tab_p[1]==> exo03.Personne@eee36c|Burma|Nestor|1950|M
tab_p[2]==> exo03.Personne@194df86|Inconnu|Inconnu|0|X
Frapper la touche <entree> pour continuer
    
```

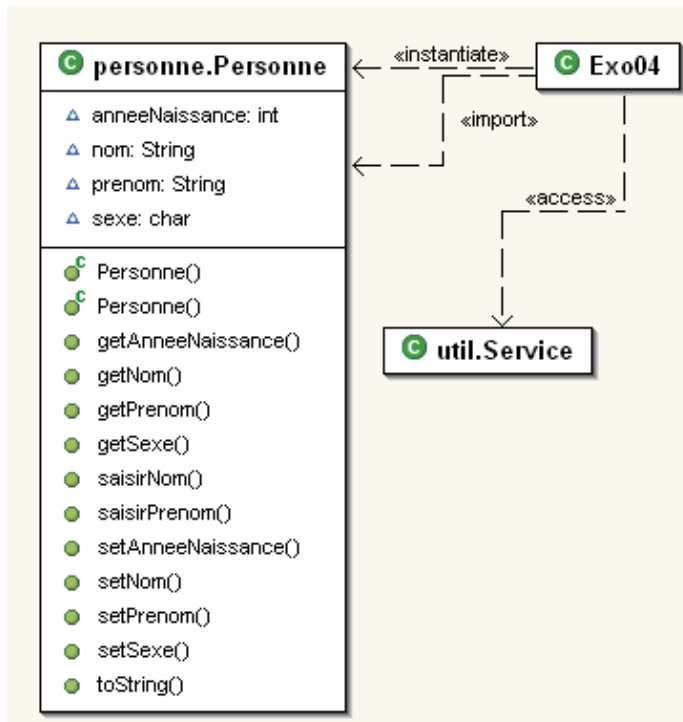
5. Relancer avec les arguments : victor zoe "deux mille" f , tester et analyser.

```
Exo3
====
java.lang.NumberFormatException: For input string: "deux mille"
  at
  java.lang.NumberFormatException.forInputString(NumberFormatException.java:48)
  at java.lang.Integer.parseInt(Integer.java:468)
  at java.lang.Integer.parseInt(Integer.java:518)
  at exo03.Exo03.main(Exo03.java:38)
Exception in thread "main"
```

Sur System .err

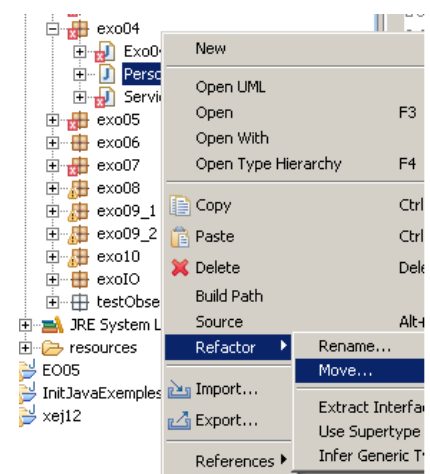
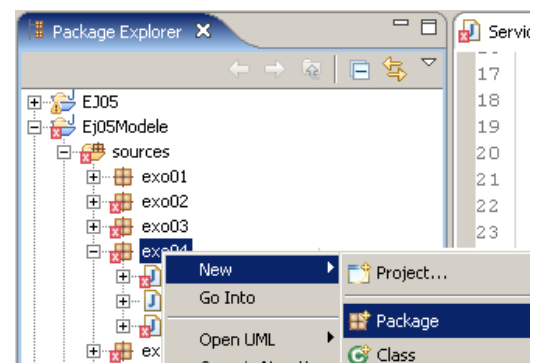
Exo04 Package et encapsulation.

Classes en jeu :



A réaliser :

1. Classe Service : implémenter la méthode initialMajuscule
2. Classe Personne : implémenter les méthodes saisirNom et saisirPrenom
3. Classe Exo04 : compléter la méthode main
4. Tester avec Exo04
5. Mettre la classe Personne dans le package exo04.personne. Mettre la classe Service dans le package exo04.util. Utiliser l'outil création de package et refactor d'Eclipse pour faire ces modifications.
6. Tester avec Exo04 et faire les modifications si nécessaire.



Sur System .out

Exo4

====

Ajout d'une 4ème Personne

tab_p[0]==> exo04.personne.Personne@1add2dd|Victor|Paul Emile|1910|g

tab_p[1]==> exo04.personne.Personne@eee36c|Burma|Nestor|1950|M

tab_p[2]==> exo04.personne.Personne@194df86|Lafarge|Virginie|1892|f

tab_p[3]==> exo04.personne.Personne@defa1a|Inconnu|Inconnu|0|X

Frapper la touche <entree> pour continuer

Nom ? **Voltaire**

Prénom ? **zadig**

tab_p[0]==> exo04.personne.Personne@1add2dd|Victor|Paul Emile|1910|g

tab_p[1]==> exo04.personne.Personne@eee36c|Burma|Nestor|1950|M

tab_p[2]==> exo04.personne.Personne@194df86|Lafarge|Virginie|1892|f

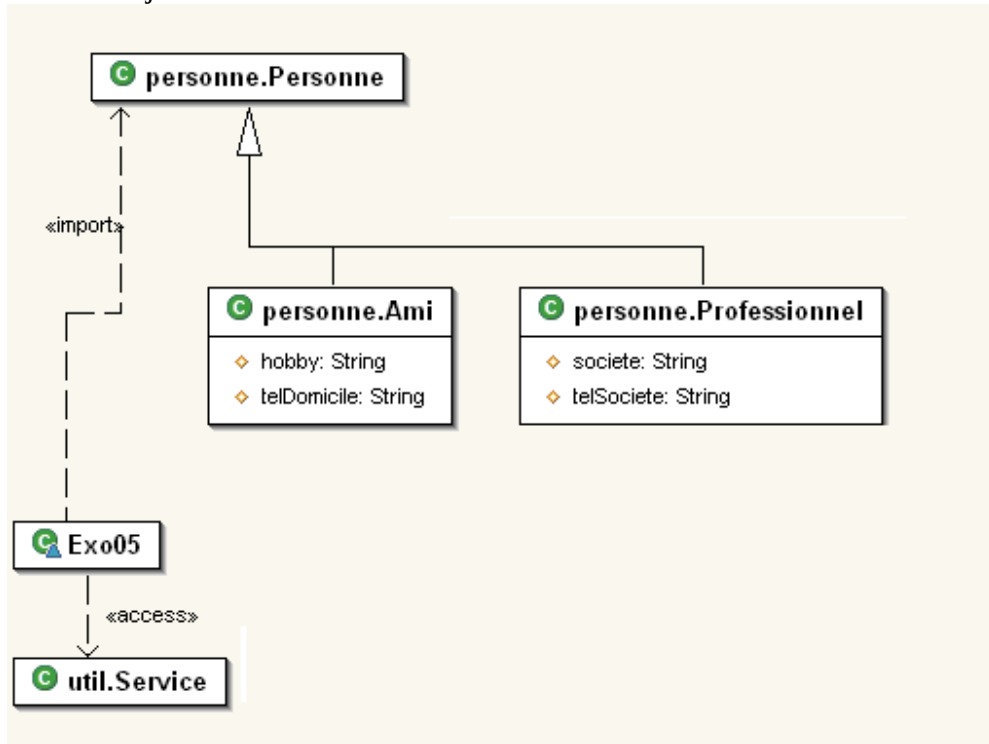
tab_p[3]==> exo04.personne.Personne@defa1a|**Voltaire**|**Zadig**|0|X

Frapper la touche <entree> pour continuer

Exo4 terminé !

Exo05 Héritage et polymorphisme.

Classes en jeu :



A réaliser :

1. Compléter la classe Ami conformément au diagramme et aux commentaires du source.
2. Faire de même pour la classe Professionnel.
3. Tester avec Exo05. Répondre aux questions posées dans le source.

Sur System.out

Exo5

```
====
tab_p[0]==> exo05.personne.Personne@defa1a|Zola|Emile|1880|M
tab_p[1]==> exo05.personne.Ami@f5da06|Inconnu|Inconnu|0|X|Inconnu|Inconnu
tab_p[2]==>
exo05.personne.Professionnel@bd0108|Lafarge|Rosalie|1892|F|01.50.55.76.88|SNCF
Frapper la touche <entree> pour continuer
```

```
On essaye saisirNom et saisirPrenom sur un Ami
Nom ? debieu
Prénom ? fernand
```

```
On affiche le resultat
tab_p[0]==> exo05.personne.Personne@defa1a|Zola|Emile|1880|M
tab_p[1]==> exo05.personne.Ami@f5da06|Debieu|Fernand|0|X|Inconnu|Inconnu
tab_p[2]==>
exo05.personne.Professionnel@bd0108|Lafarge|Rosalie|1892|F|01.50.55.76.88|SNCF
Frapper la touche <entree> pour continuer
```

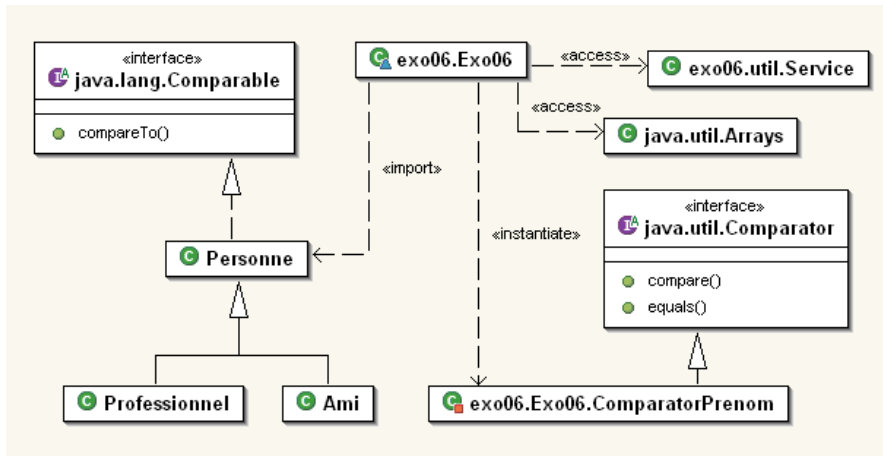
```
On utilise la méthode println (Object) de PrintStream
exo05.personne.Personne@defa1a|Zola|Emile|1880|M
exo05.personne.Ami@f5da06|Debieu|Fernand|0|X|Inconnu|Inconnu
exo05.personne.Professionnel@bd0108|Lafarge|Rosalie|1892|F|01.50.55.76.88|SNCF
Frapper la touche <entree> pour continuer
```

```
On fait la même chose mais tab_p[1] == null
exo05.personne.Personne@defa1a|Zola|Emile|1880|M
null
exo05.personne.Professionnel@bd0108|Lafarge|Rosalie|1892|F|01.50.55.76.88|SNCF
Frapper la touche <entree> pour continuer
```

Exo5 terminé !

Exo06 Interface et classe interne.

Classes en jeu :



A réaliser :

1. Modifier la classe Personne afin qu'elle implémente java.lang.Comparable. Implémenter la méthode boolean compareTo(Object o) de sorte que la comparaison se fasse sur les attributs nom. Si l'argument o n'est pas une Personne, lancer l'exception suivante.


```

throw new ClassCastException ( "L'objet comparé n'est pas une " +
    " Personne. C'est un:\n" +
    o.getClass( ).getName( ) );

```
2. Tester avec Exo06 qui utilise la méthode Arrays.sort(Object [] t) pour trier le tableau de référence de Personne.
3. Retirer les commentaires //XXXX 1 à //XXXX 6 d'Exo06 et implémenter la classe interne statique ComparatorPrenom (implements java.util.Comparator) . S'inspirer de la classe interne anonyme qui implémente Comparator pour comparer sur l'attribut anneeNaissance.
4. Tester avec Exo06. La méthode Arrays.sort(Object [] t, Comparator c) est alors utilisée.
5. Analyser les fichiers classes générés avec javap.

Sur System.out

Ex06

====

Af fhage du tableau de Personne avant sort

```
tab[0]exo06.personne.Personne@defa1a Sartre Victor 2000 M
tab[1]exo06.personne.Ami@11a698a Burma Nestor 1954 M 01.60.96.80.69 Le cinoche
tab[2]exo06.personne.Professionnel@107077e Lafarge Rosalie 1892 F 01.50.55.76.88 SNCF
tab[3]exo06.personne.Ami@7ced01 Dubois Robert 1950 M 01.22.33.44.55 La peinture
tab[4]exo06.personne.Ami@1ac04e8 Dupond Zoe 1951 F 08.22.46.87.92 Le sport
Frappier la touche <entree> pour continuer
```

Af fhage après trisur nom

```
tab[0]exo06.personne.Ami@11a698a Burma Nestor 1954 M 01.60.96.80.69 Le cinoche
tab[1]exo06.personne.Ami@7ced01 Dubois Robert 1950 M 01.22.33.44.55 La peinture
tab[2]exo06.personne.Ami@1ac04e8 Dupond Zoe 1951 F 08.22.46.87.92 Le sport
tab[3]exo06.personne.Professionnel@107077e Lafarge Rosalie 1892 F 01.50.55.76.88 SNCF
tab[4]exo06.personne.Personne@defa1a Sartre Victor 2000 M
Frappier la touche <entree> pour continuer
```

Af fhage après trisur prénom

```
tab[0]exo06.personne.Ami@11a698a Burma Nestor 1954 M 01.60.96.80.69 Le cinoche
tab[1]exo06.personne.Ami@7ced01 Dubois Robert 1950 M 01.22.33.44.55 La peinture
tab[2]exo06.personne.Professionnel@107077e Lafarge Rosalie 1892 F 01.50.55.76.88 SNCF
tab[3]exo06.personne.Personne@defa1a Sartre Victor 2000 M
tab[4]exo06.personne.Ami@1ac04e8 Dupond Zoe 1951 F 08.22.46.87.92 Le sport
Frappier la touche <entree> pour continuer
```

Af fhage après trisur annéeNaissance

```
tab[0]exo06.personne.Professionnel@107077e Lafarge Rosalie 1892 F 01.50.55.76.88 SNCF
tab[1]exo06.personne.Ami@7ced01 Dubois Robert 1950 M 01.22.33.44.55 La peinture
tab[2]exo06.personne.Ami@1ac04e8 Dupond Zoe 1951 F 08.22.46.87.92 Le sport
tab[3]exo06.personne.Ami@11a698a Burma Nestor 1954 M 01.60.96.80.69 Le cinoche
tab[4]exo06.personne.Personne@defa1a Sartre Victor 2000 M
Frappier la touche <entree> pour continuer
```

On utilise maintenant un tableau d'Object

Af fhage après mélange

```
t0[0]exo06.personne.Professionnel@107077e Lafarge Rosalie 1892 F 01.50.55.76.88 SNCF
t0[1]exo06.personne.Personne@defa1a Sartre Victor 2000 M
t0[2]exo06.personne.Ami@11a698a Burma Nestor 1954 M 01.60.96.80.69 Le cinoche
t0[3]exo06.personne.Ami@7ced01 Dubois Robert 1950 M 01.22.33.44.55 La peinture
t0[4]exo06.personne.Ami@1ac04e8 Dupond Zoe 1951 F 08.22.46.87.92 Le sport
Frappier la touche <entree> pour continuer
```

Af fhage après trisur prénom

```
t0[0]exo06.personne.Ami@11a698a Burma Nestor 1954 M 01.60.96.80.69 Le cinoche
t0[1]exo06.personne.Ami@7ced01 Dubois Robert 1950 M 01.22.33.44.55 La peinture
t0[2]exo06.personne.Professionnel@107077e Lafarge Rosalie 1892 F 01.50.55.76.88 SNCF
t0[3]exo06.personne.Personne@defa1a Sartre Victor 2000 M
t0[4]exo06.personne.Ami@1ac04e8 Dupond Zoe 1951 F 08.22.46.87.92 Le sport
Frappier la touche <entree> pour continuer
```

Af fhage après modification de t0[2] en référence de Color.BLUE

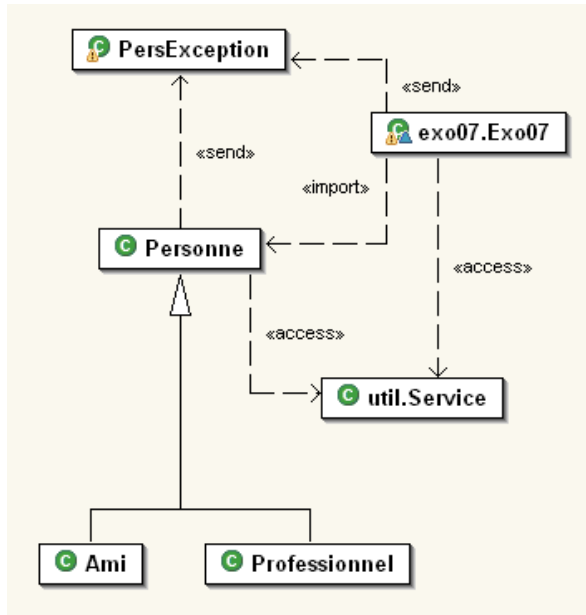
```
t0[0]exo06.personne.Ami@11a698a Burma Nestor 1954 M 01.60.96.80.69 Le cinoche
t0[1]exo06.personne.Ami@7ced01 Dubois Robert 1950 M 01.22.33.44.55 La peinture
t0[2]java.awt.Color[r=0,g=0,b=255]
t0[3]exo06.personne.Personne@defa1a Sartre Victor 2000 M
t0[4]exo06.personne.Ami@1ac04e8 Dupond Zoe 1951 F 08.22.46.87.92 Le sport
Frappier la touche <entree> pour continuer
```

On essaie de trier sur prénom

```
java.lang.ClassCastException: L'argument o2 de compare n'est pas une Personne. Sa classe est:
java.awt.Color
at    exo06.Exo06$ComparatorPrenom.compare(Exo06.java:130)
at    java.util.Arrays.mergeSort(Arrays.java:1278)
at    java.util.Arrays.sort(Arrays.java:1219)
at    exo06.Exo06.main(Exo06.java:113)
```

Exo07 Exception.

Classes en jeu :



A réaliser:

1. Dans la classe **Personne**, modifier les méthodes **setNom** et **setPrenom** afin qu'elles fassent un contrôle sur la taille de l'argument. Si sa longueur est supérieure à **PersException.LONGUEUR_MAX**, on lancera une **PersException** construite sur l'argument responsable du problème.
2. Implémenter la nouvelle méthode de **Personne** : **protected String saisir(String prompt)** qui fait un contrôle sur la String saisie par l'utilisateur et réaffiche le prompt tant que la longueur n'est pas correcte.
3. Dans **Personne**, **Ami** et **Professionnel** corriger les méthodes qui sont impliquées par les modification précédentes.
4. Tester avec **Exo07**

Sur System.out

Exo7

====

exo07.personne.Ami@defa1a||22|M|01.10.67.89.87|La peinture

=====

Commencer par faire des saisies de plus de 10 caractères

Nom ?aaaaaaaaaaaaaaaaaa

Nom ?zoe

Prénom ? victor

exo07.personne.Ami@defa1a|Zoe|Victor|22|M|01.10.67.89.87|La peinture

=====

On instancie un Ami avec un nom trop long dans un bloc try

l'exception sera interceptée et traitée ...

Frapper la touche <entree> pour continuer

===> PersException <===

La string fautive est:

Mon nom est beaucoup trop long

Sa longueur de 30 caractères est trop longue

=====

On instancie un Ami avec un prénom trop long en dehors d'un bloc try

l'exception sera interceptée et non traitée ...

Frapper la touche <entree> pour continuer

A l'extérieur d'un bloc try exception non interceptée ...

exo07.personne.PersException: ===> PersException <===

at exo07.personne.Personne.setPrenom(Personne.java:49)

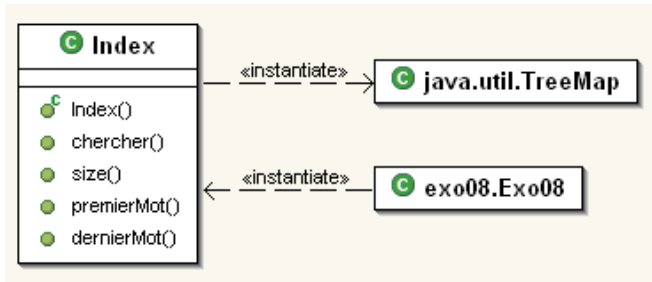
at exo07.personne.Personne.<init>(Personne.java:25)

at exo07.personne.Ami.<init>(Ami.java:19)

at exo07.Exo07.main(Exo07.java:55)

Exo08 Map, casting et java.IO.

Classes en jeu :



A réaliser :

1. Classe `Index`. Implémenter le constructeur `Index(String fileName)` et les méthodes `String chercher(String key)`, `int size()`, `String premierMot()`, `String dernierMot()`
2. Tester avec `Exo08`.

Sur System.out

Quel test voulez-vous faire :

A index=new Index("./resources/FicText/CorbeauEtRenard.txt")

B index=new Index("./resources/FicText/Zola.txt")

C index=new IndexNOK("./resources/FicText/CorbeauEtRenard.txt")

D index=new IndexNOK("./resources/FicText/Zola.txt")

A

Taille de l'index : 99

Nombre d'occurrences indexées : 140

Premier mot de l'index : a

Dernier mot de l'index : êtes

Durée de la création : 10 m s

saisir VIDER pour terminer et vider l'index sur System.out)

saisie EXIT pour terminer directement

Mot recherché:

corbeau

2|7|35|68|121|

Mot recherché:

VIDER

a:64|

alléché:23|

apprenez:100|

arbre:10|

aux:105|

beau:44|

bec:15|85|

belle:79|

bien:115|

bois:63|

bon:98|

bonjour:32|

ce:29|

celui:108|

cés:62|65|

cette:112|

confus:124|

corbeau:2|7|35|68|121|

de:61|73|107|138|

des:59|

dit:96|

doute:119|

du:34|

dépens:106|

en:13|93|

et:3|75|95|123|

flatteur:103|

fontaine:140|

fromage:17|117|

honteux:122|

vaut:114|

vit:104|

voix:80|

votre:48|53|

vous:37|41|55|

y:134|

à:26|52|

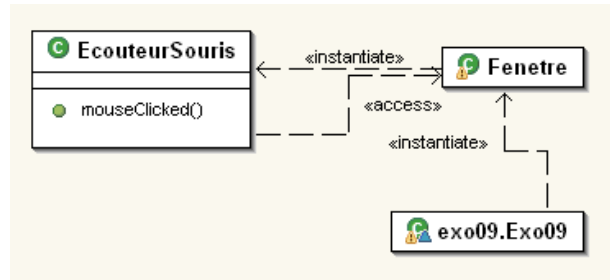
écoute:111|

êtes:38|56|

Exo08 terminé

Exo09 AWT , Frame , Listener et Event.

Classes en jeu :



A réaliser :

Sources :

1. Classe Fenetre :
Compléter le constructeur Fenetre(String titre) (commentaires Etape 1)
Tester avec Exo09
2. Classe Fenetre :
Voir commentaires Etape 2.
Tester avec Exo09
3. Créer une classe EcouteurSouris implémentant MouseListener.
Redéfinir mouseClicked (MouseEvent e) afin d'écrire les coordonnées de la souris dans la barre de titre de la Fenetre. Utiliser, pour connaître la Fenetre créatrice de l'événement, la méthode getSource() sur l'événement.
Ajouter une instance de cette classe , en tant que MouseListener , à la fenetre.
4. Classe Fenetre :
Redéfinir la méthode paint(Graphics g)
Passer la couleur de g en rouge.
Dessiner un cercle de rayon 25 centré sur x et y (attributs de Fenetre).

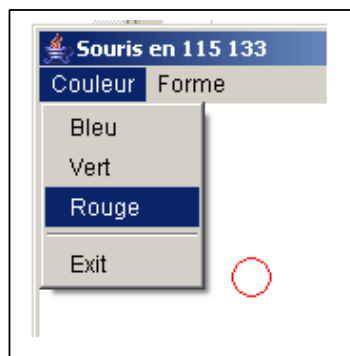
Classe EcouteurSouris :

Modifier la méthode mouseClicked afin que la méthode paint de la fenetre soit appelée et que le cercle soit tracé à l'emplacement du click de souris. Pour invoquer la méthode paint il faut appeler repaint() sur la fenetre.

Tester avec Exo09

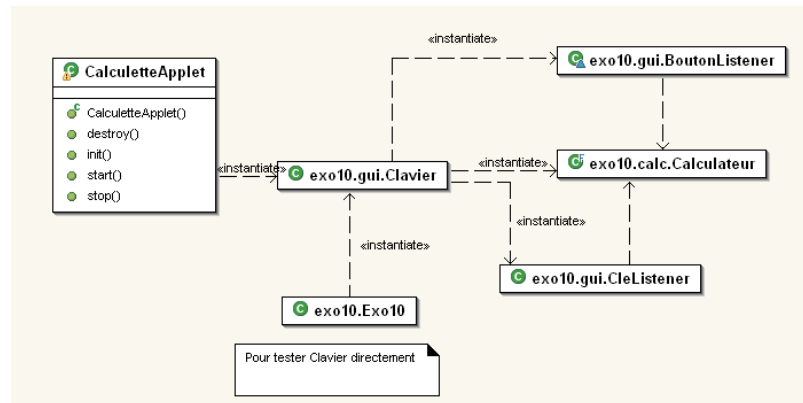
5. Faire , à votre façon, le menu suivant afin que le cercle soit tracé avec la couleur choisie. Un Menu s'écoute avec un ActionListener .

Tester avec Exo09



Exo10 Applet.

Classes en jeu :



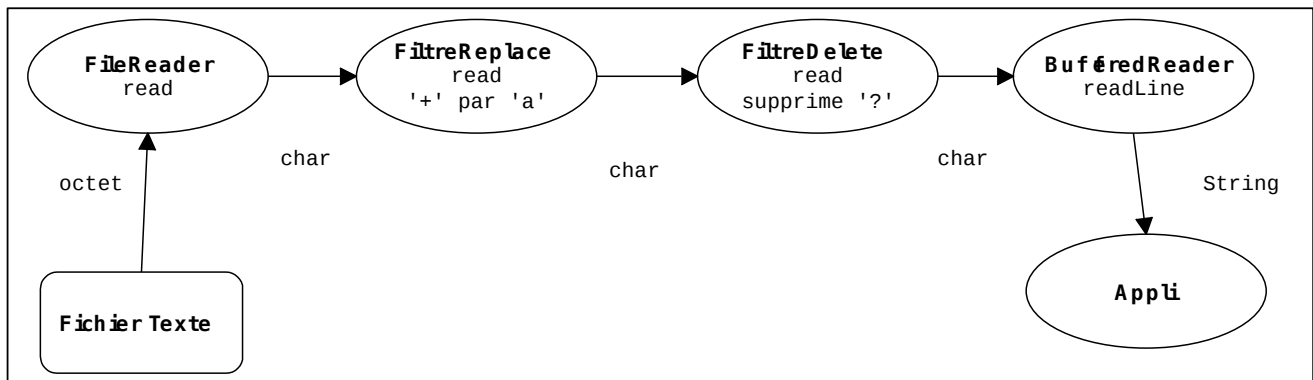
A réaliser :

1. Développer une classe `exo10.applet.CalculetteApplet` qui soit à la fois une Applet et une Application Console.
Elle devra avoir le même aspect que l'application Exo010 et utiliser, comme unique composant la classe `exo10.gui.Clavier`
2. Tester comme une applet en ouvrant dans un navigateur avec racine/Calculette.html
3. Tester comme une applet dans une fenêtre dos avec `appletViewer`.
4. Tester comme une application console dans eclipse.



Exo11 java.IO, FilterReader et pattern décorateur.

Classes en jeu :



A réaliser:

1. Développer une classe `exo11.filtre.FiltreDelete` dérivant de `FilterReader` avec un constructeur `FiltreDelete(char srcDel, Reader in)`
`srcDel`: caractère à supprimer
`in`: Reader à filtrer

Redéfinir la méthode `int read()` (voir doc page suivante)

Redéfinir la méthode `int read(char [] cbuf, int off, int len)` (voir doc page suivante)

2. Développer une classe `exo11.filtre.FiltreReplace` dérivant de `FilterReader` avec un constructeur `FiltreReplace(char srcIn, char srcOut, Reader in)`
`srcIn`: caractère à remplacer
`srcOut`: caractère remplaçant
`in`: Reader à filtrer

Redéfinir la méthode `int read()`

Redéfinir la méthode `int read(char [] cbuf, int off, int len)`

3. Tester avec `exo11.Exo11` qui lit le fichier `"/resources/FicText/CorbeauEtRenardExo11.txt"` à l'aide d'un `BufferedReader` construit sur un `FiltreDelete` pour supprimer les caractères '?' et un `FiltreReplace` pour remplacer les '+' par des 'a'.
`Exo11` utilise la méthode `readLine` d'un `BufferedReader`. Cette dernière utilise la méthode `read(char [] cbuf, int off, int len)` du `FilterReader` sur lequel il est construit.

Doc JDK java.io.FilterReader

java.io
Class FilterReader
java.lang.Object
└─ java.io.Reader
 └─ java.io.FilterReader

public int read()
 throws IOException

Read a single character.

Overrides:
read in class Reader

Returns:
The character read, as an integer in the range 0 to 65535 (0x00-0xffff, or -1 if the end of the stream has been reached

Throws:
IOException - If an I/O error occurs

public int read(char[] cbuf,
 int off,
 int len)
 throws IOException

Read characters into a portion of an array.

Parameters:
cbuf - Destination buffer
off - Offset at which to start storing characters
len - Maximum number of characters to read

Returns:
The number of characters read, or -1 if the end of the stream has been reached

Throws:
IOException - If an I/O error occurs

Le c????o???rbe+u et le ren+rd

M+itre Corbe+u, sur un +rbre perché,
Ten+it en son bec un from+ge.
M+itre Ren+r????d, p+r l'????odeur +lléché,???
Lui tint à peu près ce l+ng+ge :
"Hé ! bonjour, Monsieur du Corbe+u.
Que vous êtes joli ! que vous me semblez be+u !
S+ns mentir, ?????????????? si votre r+m+ge
Se r+pporte à votre plum+ge,
Vous ??????êtes le Phé?n?i?x des hôtes de ces bois."
A ces mots le Corbe+u ne se sent p+s de joie ;
Et pour montrer s+ belle voix,
Il ouvre un l+rge bec, l+i?s?s?e tomber s+ proie.
Le Ren+rd s'en s+isit, et dit : "Mon bon Monsieur,
Apprenez que tout fl+ttteur
Vit +ux dépens de celui qui l'écoute :
Cette leçon v+ut bien un f????rom+ge, s+ns do??ute. "
Le Corbe+u, honteux et confus,
Jur+, m+is un peu t+rd, qu'on ne l'y pre????ndr+it
plus.

Je+n de l+ font+ine

Le corbeau et le renard

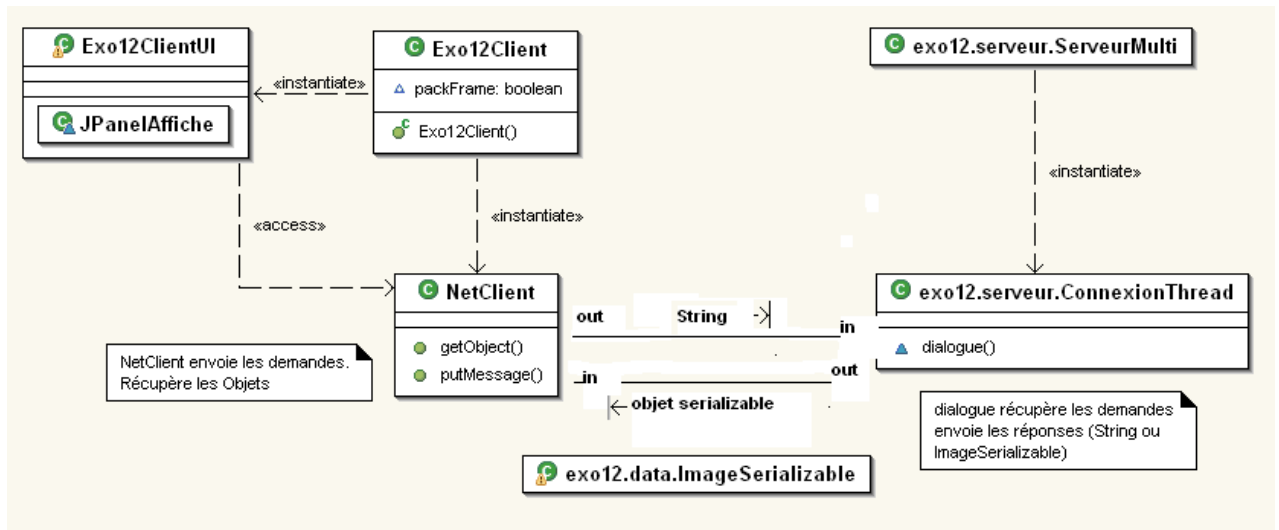
Maître Corbeau, sur un arbre perché,
Tenait en son bec un fromage.
Maître Renard, par l'odeur alléché,
Lui tint à peu près ce langage :
"Hé ! bonjour, Monsieur du Corbeau.
Que vous êtes joli ! que vous me semblez beau !
Sans mentir, si votre ramage
Se rapporte à votre plumage,
Vous êtes le Phénix des hôtes de ces bois."
A ces mots le Corbeau ne se sent pas de joie ;
Et pour montrer sa belle voix,
Il ouvre un large bec, laisse tomber sa proie.
Le Renard s'en saisit, et dit : "Mon bon Monsieur,
Apprenez que tout flatteur
Vit aux dépens de celui qui l'écoute :
Cette leçon vaut bien un fromage, sans doute. "
Le Corbeau, honteux et confus,
Jura, mais un peu tard, qu'on ne l'y prendrait plus.

Jean de la fontaine

Exo11 terminé

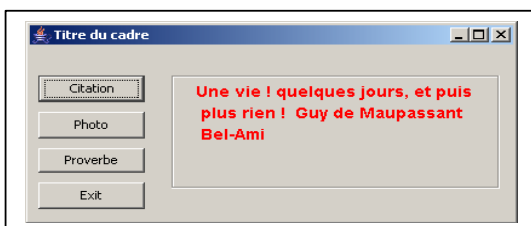
Exo12 Thread, java.net , javax.swing , java.io.Serializable.

Classes en jeu :



A réaliser :

1. Modifier la classe ConnexionThread afin que la méthode dialogue tourne dans un thread
2. Tester en lançant exo12.seveur.ServeurMulti et 2 clients simultanément (exo12.client.Exo12Client) . L'adresse ip du serveur est à passer en argument aux clients



```

System.out de ServeurMulti
Creation de la List photos: 9 photos
Création de la List citations: 78 citations
Création de la Liste proverbes: 16 proverbes
Serveur démarré à l'adresse: 10.0.3.1
Connexion du client 1
Je suis dans le run du client 1
Client 1 demande de Citation
Client 1 demande de Photo
Client 1 demande de Proverbe
Client 1 demande de Citation
    
```


Exo01 Outils du JDK en mode commande.....	3
Exo02 Récupération des arguments de la ligne de commande. Manipulation de tableau. Utilisation d'un objet.....	4
Exo03 Implémentation de la classe Personne Constructeur, getter, setter, redéfinition de méthode Tableau d'objets Méthode statique.....	5
Exo04 Package et encapsulation.....	8
Exo06 Interface et classe interne.....	16
Exo07 Exception.....	22
Exo08 Map, casting et IO.....	28
Exo09 AWT , Frame , Listener et Event.....	33
Exo10 Applet.....	37
Exo11 java.IO, FilterReader et pattern décorateur.....	45
Exo12 Thread, java.net , javax.swing, java.io.Serializable.....	48

Exo01 Outils du JDK en mode commande

1. Ouvrir une fenêtre DOS
2. Définir le PATH afin d'utiliser le bon JDK
`PATH=%JAVA_HOME%\bin;%PATH%`
3. Se positionner dans le répertoire "racine" du projet EJ05Modele
`cd ...\\EJ05Modele`
4. Compiler `exo01/HelloWorld.java` de sorte que la classe `exo01.HelloWorld` se retrouve dans le répertoire "racine"\\bin
`javac -d ./classes -classpath ./bin ./src/exo01/HelloWorld.java`
5. Exécuter `exo01.HelloWorld`
`java -classpath ./bin exo01.HelloWorld`
 Ceci est un Hello World Java !
6. Examiner `exo01.HelloWorld` (avec `javap`)
`javap -private ./bin/exo01/HelloWorld`
 Compiled from "HelloWorld.java"

```
public class exo01.HelloWorld extends java.lang.Object{
    public exo01.HelloWorld();
    public static void main(java.lang.String[]);
}
```
7. Exécuter une classe inexistante (`exo01.helloworld` par exemple) et examiner le résultat
`java -classpath ./bin exo01.HelloWorld`
 Exception in thread "main" java.lang.NoClassDefFoundError: exo01/helloworld (wrong name: exo01/HelloWorld)

```
at java.lang.ClassLoader.defineClass0(Native Method)
at java.lang.ClassLoader.defineClass(ClassLoader.java:539)
at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:123)
at java.net.URLClassLoader.defineClass(URLClassLoader.java:251)
at java.net.URLClassLoader.access$100(URLClassLoader.java:55)
at java.net.URLClassLoader$1.run(URLClassLoader.java:194)
at java.security.AccessController.doPrivileged(Native Method)
at java.net.URLClassLoader.findClass(URLClassLoader.java:187)
at java.lang.ClassLoader.loadClass(ClassLoader.java:289)
at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:274)
at java.lang.ClassLoader.loadClass(ClassLoader.java:235)
at java.lang.ClassLoader.loadClassInternal(ClassLoader.java:302)
```

Exo02 Récupération des arguments de la ligne de commande.

Manipulation de tableau.

Utilisation d'un objet.

```

/*
 * Exo02 Récupération des arguments de la ligne de commande.
 * Manipulation de tableau.
 * Utilisation d'un objet.
 *
 * class exo02.RecupParam
 */

package exo02;

public class RecupParam {

    public static void main(String[] args) {
        if (args.length == 0){
            System.out.println("Vous n'avez pas passé d'argument");
        }
        else {
            System.out.println("Vous avez passé " + args.length
                + " argument(s)");

            /*
             * instanciation de monService afin de pouvoir utiliser la méthode
             * pause
             */
            Service monService = new Service();
            monService.pause();

            for (int i = 0; i < args.length; i++) {
                System.out.println("argument[" + i + "]: " + args[i]);
            }
            System.out.println("Exo2 terminé !");
        }
    }
}

/*
 * class exo02.Service
 */
package exo02

import java.io.IOException;

public class Service {

    public void pause(){
        byte [] rep =new byte[1];
        System.out.println("Frapper la touche <entree> pour continuer");
        try{
            System.in.read(rep);
            System.in.skip(System.in.available());
        }
        catch(IOException e){
            e.printStackTrace();
            System.exit(1);
        }
    }
}

```

Exo03 Implémentation de la classe Personne

Constructeur, getter, setter, redéfinition de méthode

Tableau d'objets

Méthode statique.

```

/**
 * Exo03 Implémentation de la classe Personne.
 * Constructeur, getter, setter, redéfinition de méthodes.
 * Tableau d'objets.
 * Méthode statique.
 *
 * class exo03.Personne
 */
package exo03;

public class Personne{

    String      nom;
    String      prenom;
    int         anneeNaissance;
    char        sexe;

    public      Personne(){
        this("Inconnu", "Inconnu", 0, 'X');
    }

        public Personne( String nom, String prenom, int anneeNaissance, char sexe ){
            this.setNom(nom);
            this.setPrenom(prenom);
            this.setAnneeNaissance(anneeNaissance);
            this.setSexe(sexe);
        }

        public int getAnneeNaissance() {
            return      anneeNaissance;
        }

        public void setAnneeNaissance(int anneeNaissance) {
            this.anneeNaissance = anneeNaissance;
        }

        public String getNom() {
            return      nom;
        }

        public String getPrenom() {
            return      prenom;
        }

        public void setPrenom(String prenom) {
            this.prenom      = prenom;
        }

        public char getSexe() {
            return      sexe;
        }

        public void setSexe(char sexe) {
            this.sexe = sexe;
        }

        public void setNom(String nom) {
            this.nom = nom;
        }
}

```

```

        public String toString() {
            return super.toString() + '|' + nom + '|' + prenom + '|' + anneeNaissance + '|' +
                sexe;
        }
    }

/**
 * Exo03 Implémentation de la classe Personne.
 * Constructeur, getter, setter, redéfinition de méthodes.
 * Tableau d'objets.
 * Méthode statique.
 *
 * class exo03.Service
 *
 */
package exo03;

import java.io.IOException;

public class Service {

    //pause devient static
    public static void pause(){
        byte [] rep =new byte[1];
        System.out.println("Frapper la touche <entree> pour continuer");
        try{
            System.in.read(rep);
            System.in.skip(System.in.available());
        }
        catch(IOException e){
            e.printStackTrace();
            System.exit(1);}
    }
}

```

```

/**
 * Exo03 Implémentation de la classe Personne.
 * Constructeur, getter, setter, redéfinition de méthodes.
 * Tableau d'objets.
 * Méthode statique.
 *
 * class exo03.Exo03
 */

package exo03;

/**
 *
 * Dans la classe Personne:
 * Modification du constructeur par défaut par appel au constructeur à 4 arguments
 *
 * Dans la classe Service
 * Modification de Service.pause afin que'elle soit static
 *
 * Dans la classe Exo3
 * Passage d'arguments à la méthode main
 * Utilisation de tableau de Personne
 * Utilisation de la méthode parseInt de la classe Integer
 * Utilisation de la méthode charAt de la classe String
 */

public class Exo03 {
  public static void main (String []args){
    System.out.println("Exo3\n====");

    //Evaluation du nombre d'arguments passés à main
    if (args.length!=4) {
      System.out.println("veuillez saisir sur la ligne de commande\n" +
        "Nom prenom anneeNaissance sexe\n"+
        "Exo3 terminé!");
      System.exit(1);}

    Personne [] tab_p;
    tab_p = new Personne[3] ;

    //Instanciation de tap_p[0] à partir de args
    tab_p[0]=new Personne(args[0],args[1],Integer.parseInt(args[2]),args[3].charAt(0));

    /* ou en créant un Integer temporaire
    tab_p[0]=new Personne(args[0],args[1],new Integer(args[2]).intValue(),args[3].charAt(0));
    */

    tab_p[1]=new Personne("Burma","Nestor",1950,'M');
    tab_p[2]=new Personne();

    for(int i=0;i<tab_p.length;i++) System.out.println("tab_p["+i+"]==> "+tab_p[i].toString());

    Service.pause();
    System.out.println ("Exo3 terminé!");

  }
}

```

Exo04 Package et encapsulation.

```

/**
 * Exo04 Package et encapsulation.
 *
 * class exo04.personne.Personne
 *
 */
package exo04.personne;

import exo04.util.Service;

public class Personne {

    String nom;
    String prenom;
    int anneeNaissance;
    char sexe;

    public Personne(){
        this("Inconnu", "Inconnu", 0, 'X');
    }

    public Personne( String nom, String prenom, int anneeNaissance, char sexe ){
        this.setNom(nom);
        this.setPrenom(prenom);
        this.setAnneeNaissance(anneeNaissance);
        this.setSexe(sexe);
    }

    public int getAnneeNaissance() {
        return anneeNaissance;
    }

    public void setAnneeNaissance(int anneeNaissance) {
        this.anneeNaissance = anneeNaissance;
    }

    public String getNom() {
        return nom;
    }

    public String getPrenom() {
        return prenom;
    }

    public void setPrenom(String prenom) {
        this.prenom = prenom;
    }

    public char getSexe() {
        return sexe;
    }

    public void setSexe(char sexe) {
        this.sexe = sexe;
    }

    public void setNom(String nom) {
        this.nom = nom;
    }

    public String toString() {
        return super.toString() + '|' + this.nom + '|' + this.prenom + '|' + this.anneeNaissance +
            '|' + this.sexe;
    }
}

```



```

public void saisirNom(){
    System.out.print("Nom ? ");
    this.setNom(Service.initialMajuscule(Service.saisirString()));
}

public void saisirPrenom(){
    System.out.print("Prénom ? ");
    this.setPrenom(Service.initialMajuscule(Service.saisirString()));
}

}

/**
 * Exo04 Package et encapsulation.
 *
 * class exo04.util.Service
 *
 */

package exo04.util;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Service {

    public static void pause(){
        byte [] rep =new byte[1];
        System.out.println("Frapper la touche <entree> pour continuer");
        try{
            System.in.read(rep);
            System.in.skip(System.in.available());
        }
        catch(IOException e){
            e.printStackTrace();
            System.exit(1);
        }
    }

    // A partir d'une String quelconque renvoie une String semblable en minuscule avec l'initiale
    en majuscule
    public static final String initialMajuscule(String s)
    {
        return (s==null || s.length() == 0 )
        ? s
        : s.substring(0,1).toUpperCase() + s.substring(1).toLowerCase();
    }

    // retourne une String saisie au clavier
    // si exception édite un message sur la sortie standard et renvoie une référence null
    public static final String saisirString()
    {try
    {BufferedReader d = new BufferedReader(new InputStreamReader(System.in));
    return d.readLine();
    }
    catch(IOException e){
        System.err.println("Exception dans Service.saisirString"+ e.getMessage());
        return null;
    }
    }
}

```

```

/**
 * Exo04 Package et encapsulation.
 *
 * class exo04.Exo04
 *
 */

package exo04;

import exo04.personne.*;
import exo04.util.*;

public class Exo04 {
    public static void main (String []args) {
        System.out.println("Exo4\n===");

        if (args.length != 4 ) {
            System.out.println("veuillez saisir sur la ligne de commande\n" +
                "Nom prenom anneeNaissance sexe");
            System.out.println ("Exo4 terminé !");
            System.exit(1);}

        Personne [] tab_p = new Personne[4]; // allocation d'un tableau de 4 références de Personne

        tab_p[0]=new Personne(args[0], args[1], Integer.parseInt(args[2]), args[3].charAt(0));
        tab_p[1]=new Personne("Burma", "Nestor", 1950, 'M');
        tab_p[2]=new Personne("Lafarge", "Virginie", 1892, 'F');

        // Ajout d'une 4ème personne avec le constructeur par défaut puis sur cette référence
        // appel de saisirNom et saisirPrenom
        System.out.println("Ajout d'une 4ème Personne");
        tab_p[3]=new Personne();

        for(int i=0;i<tab_p.length;i++) System.out.println("tab_p["+i+"]==>
"+tab_p[i].toString());
        Service.pause();

        tab_p[3].saisirNom();
        tab_p[3].saisirPrenom();

        for(int i=0;i<tab_p.length;i++) System.out.println("tab_p["+i+"]==> "+tab_p[i].toString());
        Service.pause();
        System.out.println ("Exo4 terminé !");

        System.exit(0);
    }
}

```

Exo05 Héritage et polymorphisme.

```
/**
 * Exo05 Héritage et Polymorphisme.
 *
 * class exo05.personne.Personne
 */

package exo05.personne;

import exo05.util.Service;

public class Personne {

    String nom;
    String prenom;
    int anneeNaissance;
    char sexe;

    public Personne(){
        this("Inconnu","Inconnu",0,'X');
    }

    public Personne( String nom, String prenom, int anneeNaissance, char sexe ){
        this.setNom(nom);
        this.setPrenom(prenom);
        this.setAnneeNaissance(anneeNaissance);
        this.setSexe(sexe);
    }

    public int getAnneeNaissance() {
        return anneeNaissance;
    }

    public void setAnneeNaissance(int anneeNaissance) {
        this.anneeNaissance = anneeNaissance;
    }

    public String getNom() {
        return nom;
    }

    public String getPrenom() {
        return prenom;
    }

    public void setPrenom(String prenom) {
        this.prenom = prenom;
    }

    public char getSexe() {
        return sexe;
    }

    public void setSexe(char sexe) {
        this.sexe = sexe;
    }

    public void setNom(String nom) {
        this.nom = nom;
    }

    public String toString() {
        return super.toString() + '|' + this.nom + '|' + this.prenom + '|' + this.anneeNaissance +
            '|' + this.sexe;
    }
}
```

```

public void saisirNom(){
    System.out.print("Nom ? ");
    this.setNom(Service.initialeMajuscule(Service.saisirString()));
}

public void saisirPrenom(){
    System.out.print("Prénom ? ");
    this.setPrenom(Service.initialeMajuscule(Service.saisirString()));
}

}

/**
 * Exo05 Héritage et Polymorphisme.
 *
 * class exo05.personne.Ami
 *
 */

package exo05.personne;

public class Ami extends Personne {
    protected String telDomicile;
    protected String hobby;

    public Ami(){
        super();
        telDomicile="Inconnu";
        hobby="Inconnu";
    }

    public Ami(String nom,String prenom,int anneeNaissance,char sexe,String telDomicile,String
hobby){
        super(nom, prenom, anneeNaissance, sexe);
        this.setTelDomicile(telDomicile);
        this.setHobby(hobby);
    }

    public String getHobby() {
        return hobby;
    }

    public void setHobby(String hobby) {
        this.hobby = hobby;
    }

    public String getTelDomicile() {
        return telDomicile;
    }

    public void setTelDomicile(String telDomicile) {
        this.telDomicile = telDomicile;
    }

    public String toString(){
        return super.toString() + '|' + telDomicile + '|' + hobby ;
    }
}

```

```
/**
 * Exo05 Héritage et Polymorphisme.
 *
 * class exo05.personne.Professionnel
 *
 */

package exo05.personne;

public class Professionnel extends Personne {
    protected String telSociete;
    protected String societe;

    public Professionnel(){
        super();
        this.telSociete="Inconnue";
        this.societe="Inconnue";
    }

    public Professionnel(String nom,String prenom,int anneeNaissance,char sexe,String
telSociete,String societe){
        super(nom, prenom, anneeNaissance, sexe);
        this.setTelSociete(telSociete);
        this.setSociete(societe);
    }

    public String getSociete() {
        return societe;
    }

    public void setSociete(String societe) {
        this.societe = societe;
    }

    public String getTelSociete() {
        return telSociete;
    }

    public void setTelSociete(String telSociete) {
        this.telSociete = telSociete;
    }

    public String toString(){
        return super.toString() + '|' + telSociete+ '|' + societe ;
    }
}
```

```

/**
 * Exo05 Héritage et Polymorphisme.
 *
 * class exo05.util.Service
 *
 */

package exo05.util;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Service {

public static void pause(){
    byte [] rep =new byte[1];
    System.out.println("Frapper la touche <entree> pour continuer");
    try{
        System.in.read(rep);
        System.in.skip(System.in.available());
    }
    catch(IOException e){
        e.printStackTrace();
        System.exit(1);}
    }

//A partir d'une String quelconque, renvoie une String semblable en minuscule avec l'initiale en
majuscule
public static final String initialeMajuscule(String s)
{
    return (s==null || s.length() == 0 )
        ? s
        : s.substring(0,1).toUpperCase() + s.substring(1).toLowerCase();
}

//retourne une String saisie au clavier
//si exception édite un message sur la sortie standard et renvoie une référence null
public static final String saisirString()
{try
{BufferedReader d = new BufferedReader(new InputStreamReader(System.in));
return d.readLine();
}
catch(IOException e){
    System.err.println("Exception dans Service.saisirString"+ e.getMessage());
    return null;}
}
}

```

```

/**
 * Exo05 Héritage et Polymorphisme.
 *
 * class exo05.Exo05
 *
 */

package exo05;

import java.io.IOException;

import exo05.personne.Ami;
import exo05.personne.Personne;
import exo05.personne.Professionnel;
import exo05.util.Service;

class Exo05 {
    public static void main (String []args)throws IOException{
        System.out.println("Exo5\n====");
        Personne [] tab_p=new Personne[3]; //allocation de 3 références de Personnes;

        tab_p[0]=new Personne("Zola","Emile",1880,'M');
        tab_p[1]=new Ami(); //Pour voir les valeurs par défaut
        tab_p[2]=new Professionnel("Lafarge","Rosalie",1892,'F',"01.50.55.76.88","SNCF");

        for(int i=0;i<tab_p.length;i++) System.out.println("tab_p["+i+"]==> "+tab_p[i].toString());

        Service.pause();

        System.out.println("\n\n essaye saisirNom et saisirPrenom sur un Ami");
        tab_p[1].saisirNom();
        tab_p[1].saisirPrenom();

        System.out.println("\n\n affiche le résultat");
        for(int i=0;i<tab_p.length;i++) System.out.println("tab_p["+i+"]==> "+tab_p[i].toString());
        Service.pause();

        System.out.println("\n\n utilise la méthode println (Object) de PrintStream");
        for(int i=0;i<tab_p.length;i++) System.out.println(tab_p[i]);
        Service.pause();

        System.out.println("\n\n fait la même chose mais tab_p[1] == null");
        tab_p[1]=null;
        for(int i=0;i<tab_p.length;i++) System.out.println(tab_p[i]);
        Service.pause();

        System.out.println ("Exo5 terminé !");

        System.exit(0);
    }
}

```

Exo06 Interface et classe interne.

```

/**
 * Exo06 Interface et classe interne.
 *
 * class exo06.personne.Personne
 */

package exo06.personne;

import exo04.util.Service;

public class Personne implements Comparable {

    String nom;
    String prenom;
    int anneeNaissance;
    char sexe;

    public Personne(){
        this("Inconnu", "Inconnu", 0, 'X');
    }

    public Personne( String nom, String prenom, int anneeNaissance, char sexe ){
        this.setNom(nom);
        this.setPrenom(prenom);
        this.setAnneeNaissance(anneeNaissance);
        this.setSexe(sexe);
    }

    public int getAnneeNaissance() {
        return anneeNaissance;
    }

    public void setAnneeNaissance(int anneeNaissance) {
        this.anneeNaissance = anneeNaissance;
    }

    public String getNom() {
        return nom;
    }

    public String getPrenom() {
        return prenom;
    }

    public void setPrenom(String prenom) {
        this.prenom = prenom;
    }

    public char getSexe() {
        return sexe;
    }

    public void setSexe(char sexe) {
        this.sexe = sexe;
    }

    public void setNom(String nom) {
        this.nom = nom;
    }

    public String toString() {
        return super.toString() + '|' + this.nom + '|' + this.prenom + '|' + this.anneeNaissance +
        '|' + this.sexe;
    }
}

```



```

public void saisirNom(){
    System.out.print("Nom ? ");
    this.setNom(Service.initialeMajuscule(Service.saisirString()));
}

public void saisirPrenom(){
    System.out.print("Prénom ? ");
    this.setPrenom(Service.initialeMajuscule(Service.saisirString()));
}

public int compareTo(Personne o){
    return this.nom.compareTo(o.nom);
}

public int compareTo(Object o) {
    if (o instanceof Personne) {
        return this.compareTo((Personne) o );
    }
    throw new ClassCastException("L'objet comparé n'est pas" +
        " une Personne, c'est un:\n "+
        o.getClass().getName());
}

}

```

```

/**
 * Exo06 Interface et classe interne.
 *
 * class exo06.util.Service
 *
 */

package exo06.util;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Service {

public static void pause(){
    byte [] rep =new byte[1];
    System.out.println("Frapper la touche <entree> pour continuer");
    try{
        System.in.read(rep);
        System.in.skip(System.in.available());
    }
    catch(IOException e){
        e.printStackTrace();
        System.exit(1);}
}
}

```

```
//A partir d'une String quelconque renvoie une String semblable en minuscule avec l'initiale en
majuscule
public static final String initialeMajuscule(String s)
{
return (s==null || s.length() == 0 )
? s
: s.substring(0,1).toUpperCase() + s.substring(1).toLowerCase();
}

//retourne une String saisie au clavier
//si exception edite un message sur la sortie standard et renvoie une référence null
public static final String saisirString()
{try
{BufferedReader d = new BufferedReader(new InputStreamReader(System.in));
return d.readLine();
}
catch(IOException e){
System.err.println("Exception dans Service.saisirString"+ e.getMessage());
return null;}
}

// =====
// Mise en forme d'une string src dont les champs sont séparés par un caractère sep
// sep séparateur de champ
// tab1 position de la première tabulation
// dtab largeur de la tabulation
// On utilise un StringBuffer pour de meilleures performances sur l'insertion en fin.
public static final String miseEnForme(String src,char sep,int tab1, int dtab)
{
StringBuffer s=new StringBuffer(src.length()*2);
char c;
int posC=0;
int posT=tab1;
for (int i=0;i<src.length();i++)
{c=src.charAt(i);
if (c!=sep){
s.append(c);
posC++;
if(posC==posT)posT+=dtab;
}
else
{for(int j=0;j<(posT-posC);j++) s.append(" ");
posC=posT;
posT+=dtab;
}
}
return s.toString();
}
// =====
}
```

```

/**
 * Exo06 Interface et classe interne.
 *
 * class exo06.Exo06
 */

package exo06;

import java.awt.Color;
import java.io.IOException;
import java.util.Arrays;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;

import exo06.personne.Ami;
import exo06.personne.Personne;
import exo06.personne.Professionnel;
import exo06.util.Service;

class Exo06 {
    public static void main (String []args)throws IOException{
        System.out.println("Exo6\n====");

// on instancie un tableau de 5 références de Personne

        Personne [] tab=new Personne[5];

        tab[0]=new Personne("Sartre","Victor",2000,'M');
        tab[1]=new Ami("Burma","Nestor",1954,'M',"01.60.96.80.69","Le cinoche");
        tab[2]=new Professionnel("Lafarge","Rosalie",1892,'F',"01.50.55.76.88","SNCF");
        tab[3]=new Ami("Dubois","Robert",1950,'M',"01.22.33.44.55","La peinture");
        tab[4]=new Ami("Dupond","Zoe",1951,'F',"08.22.46.87.92","Le sport");

//Affichage du contenu du tableau d'InterfaceTriable avant sort
        System.out.println("Affichage du tableau de Personne avant sort");
        for(int i=0;i<tab.length;i++) System.out.println(Service.miseEnForme("tab["+i+"]" +
tab[i].toString(),'|',43,8));
        Service.pause();
//=====

//Affichage du contenu du tableau d'InterfaceTriable après sort sur les membres nom
        System.out.println("Affichage après tri sur nom");

        Arrays.sort(tab);
        for(int i=0;i<tab.length;i++) System.out.println(Service.miseEnForme("tab["+i+"]" +
tab[i].toString(),'|',43,8));
        Service.pause();

//=====
//Affichage du contenu du tableau d'InterfaceTriable après sort sur les prénoms en
//utilisant une classe interne static ComparatorPrenom
        System.out.println("Affichage après tri sur prénom");

        ComparatorPrenom compPrenom=new ComparatorPrenom();
        Arrays.sort(tab,compPrenom);
        for(int i=0;i<tab.length;i++) System.out.println(Service.miseEnForme("tab["+i+"]" +
tab[i].toString(),'|',43,8));
        Service.pause();

```

```

//=====
//Affichage du contenu du tableau d'InterfaceTriable après sort sur anneeNaissance en
//utilisant une classe interne static anonyme
System.out.println("Affichage après tri sur anneeNaissance");

Arrays.sort(tab, new Comparator(){

    public int compare(Object o1, Object o2) {
        if(! (o1 instanceof Personne))
            throw new ClassCastException("L'argument o1 de compare n'est pas une Personne");
        if(! (o2 instanceof Personne))
            throw new ClassCastException("L'argument o2 de compare n'est pas une Personne");

        return ((Personne)o1).getAnneeNaissance() - ((Personne)o2).getAnneeNaissance();
    }

});

for(int i=0;i<tab.length;i++) System.out.println(Service.miseEnForme("tab["+i+"]" +
tab[i].toString(), '|', 43, 8));
Service.pause();

//=====
// On fait la même chose avec un tableau d'object
Object [] t0= new Object[5];
for (int i = 0; i < t0.length; i++) {
    t0[i]=tab[i];
}
List list = Arrays.asList(t0); //Obtention d'une liste
Collections.shuffle(list); //Mélange de la liste
t0=list.toArray(); //Récupération du tableau

System.out.println("On utilise maintenant un tableau d'Object");
System.out.println("Affichage après mélange");

for(int i=0;i<tab.length;i++) System.out.println(Service.miseEnForme("t0["+i+"]" +
t0[i].toString(), '|', 43, 8));
Service.pause();

System.out.println("Affichage après tri sur prénom");

Arrays.sort(t0, compPrenom);
for(int i=0;i<tab.length;i++) System.out.println(Service.miseEnForme("t0["+i+"]" +
t0[i].toString(), '|', 43, 8));
Service.pause();

System.out.println("Affichage après modification de t0[2] en référence de Color.BLUE");
t0[2]=Color.BLUE;
for(int i=0;i<tab.length;i++) System.out.println(Service.miseEnForme("t0["+i+"]" +
t0[i].toString(), '|', 43, 8));
Service.pause();

System.out.println("On essaie de trier sur prenom");
// System.out.flush();
Arrays.sort(t0, compPrenom);
for(int i=0;i<tab.length;i++) System.out.println(Service.miseEnForme("t0["+i+"]" +
t0[i].toString(), '|', 43, 8));
Service.pause();

System.out.println ("Exo6 terminé !");

}

```

```
//=====
private static class Com paratorPrenom implements Com parator {
    public int compare(Object o1, Object o2) {
        if(! (o1 instanceof Personne))
            throw new ClassCastException(
                "L'argument o1 de compare n'est pas une Personne. Sa classe est:\n"+
                o1.getClass().getName());
        if(! (o2 instanceof Personne))
            throw new ClassCastException(
                "L'argument o2 de compare n'est pas une Personne. Sa classe est:\n"+
                o2.getClass().getName());

        return ((Personne)o1).getPrenom().compareTo(((Personne)o2).getPrenom());
    }
}
}
```

Exo07 Exception.

```
/**
 * Exo07 Exception.
 *
 * class exo07.personne.Personne
 */

package exo07.personne;

import util.Service;

public class Personne implements Comparable {

    String nom;
    String prenom;
    int anneeNaissance;
    char sexe;

    public Personne() throws PersException {
        this("Inconnu", "Inconnu", 0, 'X');
    }

    public Personne( String nom, String prenom, int anneeNaissance, char sexe ) throws
    PersException{
        this.setNom(nom);
        this.setPrenom(prenom);
        this.setAnneeNaissance(anneeNaissance);
        this.setSexe(sexe);
    }

    public int getAnneeNaissance() {
        return anneeNaissance;
    }

    public void setAnneeNaissance(int anneeNaissance) {
        this.anneeNaissance = anneeNaissance;
    }

    public String getNom() {
        return nom;
    }

    public String getPrenom() {
        return prenom;
    }

    public void setPrenom(String prenom)throws PersException {
        if (prenom.length()> PersException.LONGUEUR_MAX){
            throw new PersException(prenom);
        }
        this.prenom = prenom;
    }

    public char getSexe() {
        return sexe;
    }

    public void setSexe(char sexe) {
        this.sexe = sexe;
    }
}
```

```

public void setNom(String nom) throws PersException {
    if (nom.length() > PersException.LONGUEUR_MAX){
        throw new PersException(nom);
    }
    this.nom = nom;
}

public String toString() {
    return super.toString() + '|' + this.nom + '|' + this.prenom + '|' + this.anneeNaissance +
'|' + this.sexe;
}

public int compareTo(Object o) {
    if (o instanceof Personne) {
        return this.compareTo((Personne) o );
    }
    throw new ClassCastException("L'objet comparé n'est pas" +
        " une Personne, c'est un:\n "+
        o.getClass().getName());
}

public int compareTo(Personne o){
    return this.nom.compareTo(o.nom);
}

protected String saisir(String prompt){
    String s;
    for(;;){
        System.out.print(prompt);
        s=Service.initialeMajuscule(Service.saisirString());
        if (s.length() <= PersException.LONGUEUR_MAX ) return s;
    }
}

public void saisirNom(){
    nom=this.saisir("Nom ?");
}

public void saisirPrenom(){
    prenom=this.saisir("Prénom ? ");
}

}

```

```
/**
 * Exo07 Exception.
 *
 * class exo07.personne.Am i
 */

package exo07.personne;

public class Am i extends Personne {
    protected String telDomicile;
    protected String hobby;

    public Ami() throws PersException{
        super();
        telDomicile="Inconnu";
        hobby="Inconnu";
    }

    public Ami(String nom,String prenom,int anneeNaissance,char sexe,String telDomicile,String
hobby) throws PersException{
        super(nom,prenom,anneeNaissance,sexe);
        this.setTelDomicile(telDomicile);
        this.setHobby(hobby);
    }

    public String getHobby() {
        return hobby;
    }

    public void setHobby(String hobby) {
        this.hobby = hobby;
    }

    public String getTelDomicile() {
        return telDomicile;
    }

    public void setTelDomicile(String telDomicile) {
        this.telDomicile = telDomicile;
    }

    public String toString(){
        return super.toString() + '|' + telDomicile + '|' + hobby ;
    }
}
```



```
/**
 * Exo07 Exception.
 *
 * class exo07.personne.Professionnel
 */

package exo07.personne;

public class Professionnel extends Personne {
    protected String telSociete;
    protected String societe;

    public Professionnel() throws PersException{
        super();
        this.telSociete="Inconnue";
        this.societe="Inconnue";
    }

    public Professionnel(String nom,String prenom,int anneeNaissance,char sexe,String
telSociete,String societe) throws PersException{
        super(nom, prenom, anneeNaissance, sexe);
        this.setTelSociete(telSociete);
        this.setSociete(societe);
    }

    public String getSociete() {
        return societe;
    }

    public void setSociete(String societe) {
        this.societe = societe;
    }

    public String getTelSociete() {
        return telSociete;
    }

    public void setTelSociete(String telSociete) {
        this.telSociete = telSociete;
    }

    public String toString(){
        return super.toString() +'|' + telSociete+ '|' + societe ;
    }
}
```

```
/**
 * Exo07 Exception.
 *
 * class exo07.personne.PersException
 */

package exo07.personne;

public class PersException extends Exception {
    static final int LONGUEUR_MAX=10;
    String stringFautive;

    // Pour l'exemple: "====> PersException <====" est stocké dans l'attribut message et
    // donc récupérable par getMessage()
    public PersException(){
        super("====> PersException <====");
    }

    public PersException (String s){
        this();
        stringFautive=s;
    }

    public String getStringFautive(){
        return stringFautive;
    }

}
```

```

/**
 * Exo07 Exception.
 *
 * class exo07.Exo07
 */

package exo07;

import java.io.IOException;

import util.Service;

import exo07.personne.Ami;
import exo07.personne.PersException;
import exo07.personne.Personne;

class Exo07 {
public static void main (String []args)throws IOException, PersException
{
    System.out.println("Exo7\n====");

    Ami x=new Ami ("", "", 22, 'M', "01.10.67.89.87", "La peinture");
    System.out.println(x.toString());

    System.out.println("\n=====");
    System.out.println("Commencer par faire des saisies de plus de 10 caracteres pour voir");
    x.saisirNom();
    x.saisirPrenom();

    System.out.println(x.toString());

// dans un bloc try exception interceptée
    System.out.println("\n=====");
    System.out.println("\nOn instancie un Ami avec un nom trop long dans un bloc try"+
        "\n\nl'exception sera interceptée et traitée ...");
    try
    { Service.pause();

        Personne y= new Ami("Mon nom est beaucoup trop long",
            "zoe", 10, 'g', "01.10.67.89.87", "La musique");
    }
    catch (PersException e)
    { System.out.println( e.getMessage() + "\nLa string fautive est:\n"+
        e.getStringFautive()+
            "\nSa longueur de " + e.getStringFautive().length()+
            " caractères est trop longue ");
    }

// à l'extérieur d'un bloc try l'exception remonte et sa méthode getMessage sera appelée
    System.out.println("\n=====");
    System.out.println("\nOn instancie un Ami avec un prénom trop long en dehors d'un bloc try"+
        "\n\nl'exception sera interceptée et non traitée ...");
    Service.pause();
    System.out.println( "\nA l'exterieur d'un bloc try exception non interceptée ...");
    Ami z= new Ami("zoe", "Mon prénom est beaucoup trop long", 10, 'C', "10.26.34.56.18", "La
lecture");
    Service.pause();

    System.out.println ("Exo7 terminé !");
}
}

```

Exo08 Map, casting et IO

Comparaison de performance entre **Index** et **IndexNOK**

```

SurSystem .out
Quel test voulez-vous faire:
A index=new Index("./resources/FicText/CorbeauEtRenard.txt")
B index=new Index("./resources/FicText/Zola.txt")
C index=new IndexNOK("./resources/FicText/CorbeauEtRenard.txt")
D index=new IndexNOK("./resources/FicText/Zola.txt")
B
Taille de l'index : 11023
Nombre d'occurrences indexées : 134743
Premier mot de l'index : a
Dernier mot de l'index : zola
Durée de la création : 741 m s
saisir FIN pour terminer et vider l'index sur System.out)
saisir EXIT pour terminer directement
Mot recherché:
EXIT
Exo08terminé
  
```

```

SurSystem .out
Quel test voulez vous faire:
A index=new Index("./resources/FicText/CorbeauEtRenard.txt")
B index=new Index("./resources/FicText/Zola.txt")
C index=new IndexNOK("./resources/FicText/CorbeauEtRenard.txt")
D index=new IndexNOK("./resources/FicText/Zola.txt")
D
Taille de l'index: 11023
Nombre d'occurrences indexées: 134743
Premier mot de l'index: a
Dernier mot de l'index: zola
Durée de la création: 8092 m s
saisir FIN pour terminer et vider l'index sur System.out)
saisir EXIT pour terminer directement
Mot recherché:
EXIT
Exo08terminé
  
```

```

/*
 * Exo08 Map, casting et java.IO.
 *
 * class exo08.index.Index
 */
  
```

```
package exo08.index;
```

```

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintStream;
import java.io.StreamTokenizer;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Set;
import java.util.TreeMap;
  
```

```

public class Index {

    long dureeCreation; //durée de la création en ms

    TreeMap map;
    long totalOccurs;

    protected Index(){
        //Ne fait rien c'est juste pour que IndexNOK(String fileName)
        //puisse l'appeler
    }
    public Index (String fileName )throws IOException{
        dureeCreation = System.currentTimeMillis();

        map=new TreeMap();

        BufferedReader bf=new BufferedReader(
            new                               InputStreamReader(
            new                               FileInputStream(
            new                               File(fileName)));

        StreamTokenizer st=new StreamTokenizer(bf);
        st.lowerCaseMode(true);// tout est vu en minuscule
        st.whitespaceChars('.', '.'); // . devient un séparateur
        st.whitespaceChars('"', '"'); // " devient un séparateur
        st.whitespaceChars('\'', '\''); // ' devient un séparateur

        long pos=0;
        int i=0;
        String s;
        for(;;){
            i=st.nextToken(); //type du prochain token
            if (i==StreamTokenizer.TT_EOF)break; //fin de fichier: FIN
            if (i==StreamTokenizer.TT_WORD){ //un mot: TRAITEMENT
                pos++;
                s=st.sval; //recupération du token

                if(map.containsKey(s)){
                    ((StringBuffer) map.get(s)).append(pos).append('|');
                }
                else {
                    map.put(s, new StringBuffer().append(pos).append('|'));
                }
            }
        }
        totalOccurs=pos;

        dureeCreation = System.currentTimeMillis() - dureeCreation;
    }

    public long getDureeCreation(){
        return this.dureeCreation;
    }
    public void vider(PrintStream ps){
        Set set = map.entrySet();
        Iterator iter = set.iterator();
        for(;iter.hasNext();){
            Map.Entry me=(Entry) iter.next();
            ps.println((String) me.getKey()+':'+me.getValue());
        }
    }

    public String chercher(String key){
        Object o= map.get(key);
        return o==null?null:((StringBuffer)o).toString();
    }
}

```

```

public int size(){
    return map.size();
}

public String premierMot(){
    return (String) map.firstKey();
}

public String dernierMot(){
    return (String)map.lastKey();
}

public long getTotalOccurs(){
    return totalOccurs;
}
}

/*
 * Exo08 Map, casting et java.IO.
 *
 * class exo08.index.IndexNOK
 */

package exo08.index;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.StreamTokenizer;
import java.util.TreeMap;

//Class non correcte, les positions sont stockées sous forme de String
//à chaque nouveau mot, il y a donc création et suppression de String
//temps d'exécution beaucoup plus long (environ 10 fois plus coûteux pour un fichier
//de 140000 mots

public class IndexNOK extends Index {

    public IndexNOK (String fileName )throws IOException{
        dureeCreation = System.currentTimeMillis();
        map=new TreeMap();

        BufferedReader bf=new BufferedReader(
            new
            new
            new
            InputStreamReader(
            FileInputStream(
            File(fileName))););

        StreamTokenizer st=new StreamTokenizer(bf);
        st.lowerCaseMode(true);// tout est vu en minuscule
        st.whitespaceChars('.', '.'); // . devient un séparateur
        st.whitespaceChars('"', '"'); // " devient un séparateur
        st.whitespaceChars('\'', '\''); // ' devient un séparateur

        long pos=0;
        int i=0;
        String s;
        for(;;){
            i=st.nextToken(); //type du prochain token
            if (i==StreamTokenizer.TT_EOF)break; //fin de fichier: FIN
            if (i==StreamTokenizer.TT_WORD ){ //un mot: TRAITEMENT
                pos++;
                s=st.sval; //recupération du token

                if(map.containsKey(s)){

```

```

        String x= (String) map.get(s);
        x = x + pos + '|'; //Perte de temps car création d'une String
        map.put(s,x); //à chaque fois
    }
    else {
// Les String , contrairement au StringBuffer, ne sont pas modifiables donc
// l'opérateur + crée une nouvelle String !
        map.put(s,""+ pos + '|');

    }

}
}
totalOccurs=pos;
this.dureeCreation = System.currentTimeMillis() - dureeCreation;

}

public String chercher(String key) {

    Object o= map.get(key);
    return o==null?null:o.toString();
}
}

/*
 * Exo08 Map, casting et java.IO.
 *
 * class exo08.Exo08
 */

package exo08;

import java.io.IOException;

import exo08.index.Index;
import exo08.index.IndexNOK;

import util.Service;

public class Exo08 {

    public static void main(String[] args) throws IOException{

        System.out.println("Quel test voulez-vous faire: \n"+
        "A index=new Index(\"./resources/FicText/CorbeauEtRenard.txt\")\n" +
        "B index=new Index(\"./resources/FicText/Zola.txt\")\n" +
        "C index=new IndexNOK(\"./resources/FicText/CorbeauEtRenard.txt\")\n" +
        "D index=new IndexNOK(\"./resources/FicText/Zola.txt\")");
        char choix='\0';
        String s = Service.saisirString();
        if( s.length() > 0 ) choix = s.toUpperCase().charAt(0);
        Index index=null;

```

```

switch (choix){
  case 'A':
    index=new      Index("./resources/FicText/CorbeauEtRenard.txt");
    break;
  case 'B':
    index=new      Index("./resources/FicText/Zola.txt");
    break;
  case 'C':
    index=new      IndexNOK("./resources/FicText/CorbeauEtRenard.txt");
    break;
  case 'D':
    index=new      IndexNOK("./resources/FicText/Zola.txt");
    break;
  default:
    System.out.println("Choix      incorrect");
    System.exit(0);
    break;
}

    System.out.println("Taille de l'index : "+index.size());
    System.out.println("Nombre d'occurences indexées : "+index.getTotalOccurs());
    System.out.println("Premier mot de l'index : "+index.premierMot() );
    System.out.println("Dernier mot de l'index : "+index.dernierMot() );
    System.out.println("Durée de la création : "+ index.getDureeCreation()+ " ms");

String rech;
    System.out.println("saisir FIN pour terminer et vider l'index sur System.out)\n"+
        "saisir EXIT pour terminer directement");
for(;;){
    System.out.println("Mot recherché : ");
    rech=Service.saisirString();
    if      (rech.equals("FIN")) {
        index.vider(System.out);
        break;
    }
    if      (rech.equals("EXIT")){
        break;
    }
    rech=index.chercher(rech);
    if (rech==null) System.out.println("mot inconnu dans l'index") ;
    else      System.out.println(rech) ;
}
System.out.println("\nExo08 terminé");
}
}

```


Exo09 AWT , Frame , Listener et Event.

```

/**
 * Exo09 AWT , Frame , Listener et Event.
 *
 * class exo09.gui.Fenetre
 */
package exo09.gui;

import java.awt.*;
import java.awt.event.*;

//C'est Fenetre elle-même qui écoute les actions des menus et qui fait le travail.

public class Fenetre extends Frame implements ActionListener{
  int x=-1;
  int y;
  Color couleur= Color.red;
  String forme="carre";
  int taille=20;
  TextDialog td;

  public static final String ACTION_BLEU="ACT_BLEU";
  public static final String ACTION_ROUGE="ACT_ROUGE";
  public static final String ACTION_VERT="ACT_VERT";

  public static final String ACTION_EXIT="ACT_EXIT";

  public static final String ACTION_CARRE="ACT_CARRE";
  public static final String ACTION_CERCLE="ACT_CERCLE";
  public static final String ACTION_TAILLE="ACT_TAILLE";

  boolean go=false;

  public Fenetre (String titre)
  {
    super(titre);
    setSize(300,300);
    setVisible(true);

    //etape 2
    addWindowListener(new WindowAdapter() {
      public void windowClosing(WindowEvent e){System.exit(0);}
    });

    //On s'ajoute un EcouteurSouris
    addMouseListener(new EcouteurSouris());

    //Fabrication du menu
    initMenu();

    //Fabrication de la boite de dialogue
    td=new TextDialog("Saisir taille");
    td.setSize(200,80);
  }
  // =====
  public void actionPerformed(ActionEvent e) {
    if (e.getActionCommand().equals (Fenetre.ACTION_BLEU) ) couleur=Color.blue;
    else if (e.getActionCommand().equals (Fenetre.ACTION_VERT)) couleur=Color.green;
    else if (e.getActionCommand().equals (Fenetre.ACTION_ROUGE)) couleur=Color.red;
    else if (e.getActionCommand().equals (Fenetre.ACTION_EXIT)) System.exit(0);
    else if (e.getActionCommand().equals (Fenetre.ACTION_CARRE)) forme="carre";
    else if (e.getActionCommand().equals (Fenetre.ACTION_CERCLE)) forme="cercle";
    else if (e.getActionCommand().equals (Fenetre.ACTION_TAILLE)) td.setVisible(true);

    repaint();
  }
}

```

```
// =====
// Redéfinition de la méthode paint pour dessiner un cercle ou un carré à
// l'emplacement du clic
public void paint (Graphics g)
{
    if (x!=-1) {
        int dx=taille/2;
        g.setColor(couleur);
        if (forme.equals("cercle")) g.drawOval( x-dx,y-dx,taille,taille);
        else if (forme.equals("carre")) g.drawRect(x-dx,y-dx,taille,taille);
    }
}

// =====
void initMenu(){
    MenuBar menuBar=new MenuBar();
    setMenuBar(menuBar);
    Menu menu=new Menu("Couleur");

    MenuItem menuItem=null;
    menuItem = new MenuItem("Bleu");
    menuItem.setActionCommand(Fenetre.ACTION_BLEU);
    menu.add(menuItem);

    menuItem = new MenuItem("Vert");
    menuItem.setActionCommand(Fenetre.ACTION_VERT);
    menu.add(menuItem);

    menuItem = new MenuItem("Rouge");
    menuItem.setActionCommand(Fenetre.ACTION_ROUGE);
    menu.add(menuItem);

    menu.addSeparator();

    menuItem = new MenuItem("Exit");
    menuItem.setActionCommand(Fenetre.ACTION_EXIT);
    menu.add(menuItem);
    menu.addActionListener(this);

    menuBar.add(menu);

    menu =new Menu("Forme");

    menuItem = new MenuItem("Carre");
    menuItem.setActionCommand(Fenetre.ACTION_CARRE);
    menu.add(menuItem);

    menuItem = new MenuItem("Cercle");
    menuItem.setActionCommand(Fenetre.ACTION_CERCLE);
    menu.add(menuItem);

    menu.addSeparator();

    menuItem = new MenuItem("Taille");
    menuItem.setActionCommand(Fenetre.ACTION_TAILLE);
    menu.add(menuItem);

    menu.addActionListener(this); //obligatoire il ne s'agit pas du même menu
    menuBar.add(menu);
}
}
```

```
// =====
//   TextDialog est interne à la classe Fenetre
//   elle connaît implicitement l'instance qui l'a créée, Fenetre.this
//   elle accède donc directement à l'attribut taille de l'instance qui l'a créée.
class TextDialog extends Dialog {
    TextField tf;

    TextDialog(String titre) {
        super(Fenetre.this,titre);

        setLayout(new BorderLayout());
        setBackground(Color.yellow);
        tf=new TextField("Taille",20);
        add(tf,BorderLayout.CENTER);
        Button b=new Button("OK");
        add(b,BorderLayout.SOUTH);

        b.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e) {
                try{
                    taille=Integer.parseInt(tf.getText());
                    setVisible(false);
                    Fenetre.this.repaint(); //repaint de la Fenetre parente
                }
                catch(NumberFormatException ex){setTitle("Il faut saisir un nombre");}
            }
        });
    }
}

/**
 * Exo09 AWT , Frame , Listener et Event.
 *
 * class exo09.gui.EcouteurSouris
 */

package exo09.gui;

import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

public class EcouteurSouris extends MouseListener{
    public void mouseClicked(MouseEvent e){
        Fenetre f = (Fenetre) e.getSource();
        f.setTitle("Souris en " + e.getX() + " " + e.getY());
        f.x=e.getX();
        f.y=e.getY();
        f.repaint();
    }
}
```

```
/**
 * Exo09 AWT , Frame , Listener et Event.
 *
 * class exo09.Exo09
 */

package exo09;

import exo09.gui.Fenetre;

class Exo09 {
    public static void main (String []args) {
        System.out.println("Exo9\n====");
        Fenetre e = new Fenetre("Belle Fenêtre pour Exo9");
    }
}
```

Exo10 Applet.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<!-- .../Projet/Calcullette.html -->
</head>
<body>
<h1> Voici une belle applet </h1>
<applet code="exo10.applet.CalculletteApplet" codebase="./bin"
height="220"
width="300">
</applet>
<h1> Et encore une autre </h1>
<applet code="exo10.applet.CalculletteApplet" codebase="./bin"
height="250"
width="350">
</applet>

<h1> Maintenant lancer Calcullette.html &agrave; l'aide d'appletviewer </h1>
</body>
</html>

```

```

/**
 * Exo10 Applet.
 *
 * class exo10.applet.CalculletteApplet
 */

package exo10.applet;

import java.applet.Applet;
import java.awt.BorderLayout;
import java.awt.Frame;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;

import exo10.gui.Clavier;

public class CalculletteApplet extends Applet {

    public CalculletteApplet(){
        System.out.println("Constructeur");
    }

    public void init() {
        this.setLayout(new BorderLayout());
        this.add(new Clavier(),BorderLayout.CENTER);
        System.out.println("init");
    }

    public void start() {
        System.out.println("start");
    }

}

```

```

public void stop() {
    System.out.println("stop");
}

public void destroy() {
    System.out.println("destroy");
}

public static void main(String[] args){
    CalculetteApplet applet= new CalculetteApplet();
    applet.init();
    Frame calculetteFrame=new Frame();
    calculetteFrame.add(applet, BorderLayout.CENTER);
    calculetteFrame.setSize(400,250);
    calculetteFrame.setVisible(true);

    calculetteFrame.addWindowListener(
        new WindowAdapter(){
            public void windowClosing(WindowEvent e){System.exit(0);}
        });
}
}

/**
 * Exo10 Applet.
 *
 * class exo10.calc.Calculateur
 *
 */

package exo10.calc;

public final class Calculateur {
    float x=0;
    float y=0;
    float z=0;
    float t=0;
    float saveX;

    boolean partieFrac = false;
    float puissFrac =1;
    /**
     * operFlag est mis à true après chaque opération. Cela signifie que si l'on utilise setX on
     * démarre un nouvel x
     * setX(char) permet de saisir un nombre dans x . operFlag est mis à false
     * le registre x est initialisé à 0 puis complété avec les caracteres frappés
     */
    boolean operFlag=false;

    public void operFlagTrue(){operFlag=true;}
    public void operFlagFalse(){operFlag=false;}

    public void plus() {x+=y; y=z; z=t; operFlagTrue();}
    public void moins() {x=y-x; y=z; z=t; operFlagTrue();}
    public void mult() {x*=y; y=z; z=t; operFlagTrue();}
    public void div() {x=y/x; y=z; z=t; operFlagTrue();}

    public void clearAll() {t=0; z=0; y=0; clearX(); }
    public void clearX () {operFlagTrue(); x=0; partieFrac=false; puissFrac=1;}
    public void push() {t=z; z=y; y=x; operFlagTrue();}
    public void rotateDown() {saveX=x; x=y; y=z; z=t; t=saveX; operFlagTrue();}
    public void rotateUp() {saveX=x; x=t; t=z; z=y; y=saveX; operFlagTrue();}
    public void echXY() {saveX=x; x=y; y=saveX; operFlagTrue();}
    public void invX() {x=1.F/x; operFlagTrue();}
    public void chSigne() {x=-x; operFlagTrue();}

```

```

//Permet de compléter x par la droite en fournissant 1 caractère numérique ou un . (point
décimal)
public void setX(char c) {
  if(operFlag) {clearX(); operFlag=false;}
  if( Character.isDigit(c))
    {if (partieFrac==false) {x*=10; x+=Integer.parseInt(String.valueOf(c));}
    else {puissFrac/=10; x+=(Integer.parseInt(String.valueOf(c))*puissFrac);}
  }
  else if (c == '.') partieFrac=true;
  else ; //on ignore tous les autres caractères
}
//

public void setX(float x){this.x=x;}

public float getX(){return x;}
public String toString() {return Float.toString(x);}
public String [] getRegistre(){
  String [] tString=new String[4];
  tString[0]=Float.toString(x);
  tString[1]=Float.toString(y);
  tString[2]=Float.toString(z);
  tString[3]=Float.toString(t);
  return tString;
}
}

/**
 * Exo10 Applet.
 *
 * class exo10.gui.Clavier
 */

package exo10.gui;

import java.awt.BorderLayout;
import java.awt.Button;
import java.awt.Color;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.Panel;

import exo10.calc.Calculateur;

//la classe clavier contient un panel pour les chiffres
//et un pour les opérateurs.

public class Clavier extends Panel {

  Calculateur calculateur;;
  Panel panelAffiche;
  Panel panelChiffre;
  Panel panelOperateur;
  Panel panelClear;

  String [] vChiffre={"7","8","9","4","5","6","1","2","3","0"};
  String [] vOperateur={".", "Push (p)", "+", "-", "*", "/", "1/x (i)", "RotateUp", "RotateDown",
    "ChangeXY (e)", "ClearAll (C)", "ClearX (c)", "Signe X (s)"};
  String [] actionOperateur={".", "p", "+", "-", "*", "/", "i", "U", "D", "e", "C", "c", "s"};

  Button [] bChiffre=new Button[10];
  BoutonCouleur [] bOperateur= new BoutonCouleur[13];
  LabelCouleur [] lRegistre;
  BoutonListener boutonListener;
  CleListener cleListener;

```

```

//=====
public Clavier(){

//Changement de layout
setLayout(new BorderLayout());

calculateur=new Calculateur();

//Création du panel panelAffiche avec ses composants
lRegistre=new LabelCouleur[4];

lRegistre[0]=new LabelCouleur("0",Color.pink);

for (int i=0,r=100;i<4;i++,r+=33) lRegistre[i]= new LabelCouleur("0", 255,r,255 );

panelAffiche=new Panel();
panelAffiche.setLayout(new GridLayout(4,1));
for (int i=3;i>-1 ;i-- ) panelAffiche.add(lRegistre[i]);
add(panelAffiche,BorderLayout.NORTH );

//Construction de panelChiffre avec GridLayout 4 lignes 3 Colonnes
panelChiffre=new Panel();
panelChiffre.setLayout(new GridLayout(4,3));

//Ajout des boutons pour les chiffres au panelChiffre;
for(int i=0;i< vChiffre.length;i++)
    panelChiffre.add(bChiffre[i]=new Button(vChiffre[i]));

//Construction des 6 boutonCouleur pour les opérateurs
//et ajout des 2 premiers a panelChiffre
for (int i=0;i<bOperateur.length;i++) {bOperateur[i]=new
BoutonCouleur(vOperateur[i],Color.yellow);
    bOperateur[i].setActionCommand(actionOperateur[i]);}

for(int i=0;i<2;i++) panelChiffre.add(bOperateur[i]);

//contruction de panelOperateur avec GridLayout 4 lignes 2 Colonnes
//et ajout des 8 derniers bouton opérateur
panelOperateur =new Panel();
panelOperateur.setLayout(new GridLayout(4,2));
for(int i=2;i<10;i++)panelOperateur.add(bOperateur[i]);

//Ajout des panels Chiffre au centre et Operateur à l'est du clavier
add("Center",panelChiffre);
add("East",panelOperateur);

//Ajout du bouton clear au sud du clavier
panelClear=new Panel();
panelClear.setLayout(new GridLayout(1,3));
for (int i=10;i<13;i++) panelClear.add(bOperateur[i]);

add("South",panelClear);

//Ajout d'un écouteur de bouton et de clavier sur tous les boutons.
boutonListener=new BoutonListener(lRegistre,calculateur);
cleListener=new CleListener(lRegistre,calculateur);

for(int i=0;i<bChiffre.length;i++){
    bChiffre[i].addActionListener(boutonListener);
    bChiffre[i].addKeyListener(cleListener);
}

for(int i=0;i<bOperateur.length ;i++) {
    bOperateur[i].addActionListener(boutonListener);
    bOperateur[i].addKeyListener(cleListener);
};

}

```



```
}  
//=====  
class BoutonCouleur extends Button{  
    BoutonCouleur(String texte,Color couleur){  
        super(texte);  
        setBackground(couleur);  
    }  
}  
//=====  
class LabelCouleur extends Label{  
    LabelCouleur(String texte,Color couleur){  
        super(texte,Label.RIGHT);  
        setBackground(couleur);  
    }  
  
    LabelCouleur(String texte,int r,int g, int b) {  
        this(texte,new Color(r,g,b));  
    }  
}  
//=====
```

```

/**
 * Exo10 Applet.
 *
 * class exo10.gui.BoutonListener
 *
 */

package exo10.gui;

import java.awt.event.*;
import java.awt.*;

import exo10.calc.Calculateur;

class BoutonListener implements ActionListener {
  Label [] affiche;
  Calculateur calc;

  public BoutonListener(Label [] affiche,Calculateur calc) {
    this.affiche=affiche;
    this.calc=calc;
  }

  public void actionPerformed(ActionEvent e){
//      récupération du premier caractere sur le bouton
    char c=e.getActionCommand().charAt(0);
    switch (c)
    {
      case '+': //pour exemple : on insère dans la queue d'événement un KeyEvent correspondant à
la touche +
        //c'est le keyListener qui traite alors l'événement
        EventQueue evtq=Toolkit.getDefaultToolkit().getSystemEventQueue() ;
        evtq.postEvent(new KeyEvent((Component) e.getSource(),401,0,KeyEvent.VK_SHIFT
,KeyEvent.VK_EQUALS, '+') );
        break;
      case '-':calc.moins();break;
      case '*':calc.mult();break;
      case '/':calc.div();break;
      case 'e':calc.echXY();break;
      case 'i':calc.invX();break;
      case 'U':calc.rotateUp();break;
      case 'D':calc.rotateDown();break;
      case 'C':calc.clearAll();break;
      case 'c':calc.clearX();break;
      case 's':calc.chSigne();break;
      case 'p':calc.push();break;
    }
    if(Character.isDigit(c) || c =='.') calc.setX(c);

    for (int i=0;i<affiche.length;i++) {affiche[i].setText(calc.getRegistre()[i]);}
  }
}

```

```

/**
 * Exo10 Applet.
 *
 * class exo10.gui.CleListener
 *
 */

package exo10.gui;

import java.awt.Label;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;

import exo10.calc.Calculateur;

public class CleListener implements KeyListener {
    Label [] affiche;
    Calculateur calc;

    public CleListener(Label [] affiche,Calculateur calc) {
        this.affiche=affiche;
        this.calc=calc;
    }

    public void keyTyped(KeyEvent e) {
        //Ne reçoit pas les touches du type shift flèche haute etc. on travaille donc dans KeyPressed()
        //Pour info les touches "a b + =" se traitent avec getKeyChar() car si l'on frappe sur "shift +"
        //on obtient avec getKeyCode() le même code que "=" car il s'agit de la même touche du clavier
    }

    public void keyPressed(KeyEvent e) {

        char c=e.getKeyChar();

        switch (c)
        {
            case 'c': calc.clearX();break;
            case 'C': calc.clearAll();break;
            case '+': calc.plus();break;
            case '-': calc.moins();break;
            case '*': calc.mult();break;
            case '/': calc.div();break;
            case 'e':
            case 'E':calc.echXY();break;
            case 'i':
            case 'I':calc.invX();break;
            case 'p':
            case 'P': calc.push();break;
            case 's':
            case 'S':calc.chSigne();break;
            case 'x':
            case 'X': System.exit(1);break;
        }

        switch (e.getKeyCode()){
            case KeyEvent.VK_UP: calc.rotateUp();break;
            case KeyEvent.VK_DOWN: calc.rotateDown();break;
        }

        if(Character.isDigit(c) ||
            c== '.' ) calc.setX(c);

            for (int i = 0; i < affiche.length; i++) {affiche[i].setText(calc.getRegistre()[i]);}
        }

    public void keyReleased(KeyEvent e) {
    }
}

```

```
/**
 * Exo10 Applet.
 *
 * class exo10.Exo10
 */

package exo10;

import java.awt.Dimension;
import java.awt.Frame;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;

import exo10.gui.Clavier;

public class Exo10 {

public static void main(String[] args){
    Frame [] calculetteFrame=new Frame[3];
    for (int i=0;i<3;i++){
        calculetteFrame[i]=new Frame("Calcullette "+ i);
        calculetteFrame[i].add("Center",new Clavier());
        calculetteFrame[i].setBounds(i*50,i*150,400,250);
        calculetteFrame[i].setVisible(true);

        calculetteFrame[i].addWindowListener(
            new WindowAdapter(){
                public void windowClosing(WindowEvent e){System.exit(0);}
            });
    }
}
}
```

Exo11 java.IO, FilterReader et pattern décorateur.

```

/**
 * Exo11 java.IO, FilterReader et pattern décorateur.
 *
 * class exo11.filtre.FiltreDelete
 *
 */

package exo11.filtre;

import java.io.FilterReader;
import java.io.IOException;
import java.io.Reader;

public class FiltreDelete extends FilterReader {
    char srcDel=0; //caractere à supprimer

    public FiltreDelete(char srcDel,Reader in){
        super(in);
        this.srcDel=srcDel;
    }

    public int read() throws IOException {
        int ret=0;
        ret = super.read();
        if ( ((char) ret) == srcDel ) return read();
        else return ret;
    }

    public int read(char[] cbuf, int off, int len) throws IOException {
        int ret=0;
        int dest=0;

        ret=super.read(cbuf, off, len);

        if (ret == -1) return ret;

        for ( int i = 0; i<ret ;i++){
            if (cbuf[i]!= srcDel) {
                cbuf[dest]=cbuf[i];
                dest++;
            }
        }
        return dest;
    }
}

```

```

/**
 * Exo11 java.IO, FilterReader et pattern décorateur.
 *
 * class exo11.filtre.FiltreReplace
 *
 */

package exo11.filtre;

import java.io.FilterReader;
import java.io.IOException;
import java.io.Reader;

public class FiltreReplace extends FiltreReader{

    char srcIn=0;          //caractere à remplacer
    char srcOut=0;        //caractere remplaçant

    public FiltreReplac(char srcIn, char srcOut, Reader in){
        super(in);
        this.srcIn=srcIn;
        this.srcOut=srcOut;
    }

    public int read() throws IOException {
        int ret=0;
        ret = super.read();
        if ( ((char) ret) == this.srcIn ) return srcOut;
        else return ret;
    }

    public int read(char[] cbuf, int off, int len) throws IOException {
        int ret=0;
        ret=super.read(cbuf, off, len);
        if (ret == -1) return ret;

        for ( int i = 0; i<ret ;i++){
            if (cbuf[i]== srcIn) {
                cbuf[i]=srcOut;
            }
        }
        return ret;
    }
}

```

```

/**
 * Exo11 java.IO, FilterReader et pattern décorateur.
 *
 * class exo11.Exo11
 *
 */

package exo11;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;

import exo11.filtre.FiltreReplace;
import exo11.filtre.FiltreDelete;

public class Exo11 {

    public static void main(String[] args) {

//Retirer les ? et remplacer les + par a

        String fileName="./resources/FicText/CorbeauEtRenardExo11.txt";
        BufferedReader bf=null;
        String ligne;
        try {

            bf=new          BufferedReader(
                new          FiltreDelete('?',
                    new          FiltreReplace('+','a',
                        new          FileReader(fileName)));

            while          (true){
                ligne=bf.readLine();
                if          (ligne==null)break ;
                System.out.println(ligne);
            }
            } catch (FileNotFoundException e) {
                e.printStackTrace();
            } catch (IOException e){
                e.printStackTrace();
            }

            System.out.println("\nExo11 terminé");
        }
    }
}

```

Exo12 Thread, java.net , javax.swing, java.io.Serializable.

```

/**
 * Exo12 Thread, java.net , javax.swing et sérialization.
 *
 * class exo12 .serveur.ConnexionThread
 *
 */

package exo12 .serveur;

import java.util.*;
import java.io.*;
import java.net.*;

public class ConnexionThread implements Runnable{

private List photos;
private List citations;
private List proverbes;

private Iterator iPhotos;
private Iterator iCitations;
private Iterator iProverbes;

private int numClient;
private Socket client;
private BufferedReader in;
private ObjectOutputStream out;

private Thread thread;

public ConnexionThread (Socket client,int numClient,List photos,List citations,List proverbes)
{
    try{
        this.client=client;
        this.photos=photos;
        this.citations=citations;
        this.proverbes=proverbes;

        iPhotos=this.photos.iterator();
        iCitations=this.citations.iterator();
        iProverbes=this.proverbes.iterator();

        this.numClient=numClient;
        out=new ObjectOutputStream(this.client.getOutputStream() );
        in=new BufferedReader(new InputStreamReader(client.getInputStream() ));

        System.out.println("Connexion du client "+this.numClient);

        thread =new Thread(this);
        thread.start(); //remplace l'appel à dialogue();
    }
    catch (java.io.IOException e) {e.printStackTrace() ;}
}

void dialogue(){
    String sIn;
    try{
        for(;true;){

```



```

        sIn=in.readLine();
        if(sIn.equals("Photo")) out.writeObject(iPhotos.hasNext() ? iPhotos.next():initItPhotos());
        else if (sIn.equals("Citation")) out.writeObject(iCitations.hasNext() ? iCitations.next():
initItCitations());
        else if (sIn.equals("Proverbe")) out.writeObject(iProverbes.hasNext()? iProverbes.next():
initItProverbes());

        else if(sIn.equals("Exit" ) {System.out.println("Client " + numClient + " C'est terminé");
            in.close();
            client.close() ;
            return;}

        System.out.println("Client " + numClient + " demande de " + sIn);
    }
}
catch (java.net.SocketException e) {System.out.println("Connexion rompue par le client "+
numClient);
            return;
        }
catch(java.io.IOException e){e.printStackTrace();
return;}
}

public void run() {

    System.out.println("Je suis dans le run du client "+ num Client);
    dialogue();
}

private Object initItProverbes(){ iProverbes=this.proverbes.iterator();return
iProverbes.next();}
private Object initItCitations(){iCitations=this.citations.iterator();return iCitations.next();
}
private Object initItPhotos(){iPhotos=this.photos.iterator();return iPhotos.next() ;}

}

```

```

/**
 * Exo12 Thread, java.net , javax.swing et sérialization.
 *
 * class exo12 .serveur.ServeurMulti
 *
 */

package exo12 .serveur;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.net.InetAddress;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.ArrayList;
import java.util.List;

import exo12 .data .ImageSerializable;

//idem ServeurMono mais crée des ConnexionThread

public class ServeurMulti{

private List photos;
private List citations;
private List proverbes;

private ServerSocket serverSocket;

public ServeurMulti() {

    try {

        //Construction du vector d'ImageSerializable photos

        File dirPhotos = new File("./resources/exo10/Photos/YannArtusBertrand");
        File [] fp = dirPhotos.listFiles();
        photos =new ArrayList(fp.length);
        for (int i = 0; i < fp.length; i++) {photos.add(new ImageSerializable(fp[i])) ;}
        System.out.println("Creation de la List photos: "+ fp.length+ " photos");

        //Construction du vector de String citations
        citations = new ArrayList();
        BufferedReader fCitation= new BufferedReader(
            new
                FileReader("./resources/exo10/Data/Citations.txt"));
        for(String s;(s=fCitation.readLine())!=null;citations.add(s));
        System.out.println("Création de la List citations: "+ citations.size() + " citations");

        //Construction du vector de String proverbes
        proverbes = new ArrayList();
        BufferedReader fProverbe= new BufferedReader(new
            FileReader("./resources/exo10/Data/Proverbes.txt"));
        for(String s;(s=fProverbe.readLine())!=null;proverbes.add(s));
        System.out.println("Création de la Liste proverbes: "+ proverbes.size()+ " proverbes");

        //Creation du serveur de sockets
        serverSocket=new ServerSocket(11111);
        System.out.println("Serveur démarré à l'adresse: "+
            InetAddress.getLocalHost().getHostAddress());

        //Attente d'une connexion dans une boucle
        Socket client;
        int numClient;
    }
}

```

```
        for (numClient=1; numClient<15;numClient ++) {
            client=serverSocket.accept();

            //Création d'une Connexion qui va faire son travail de dialogue
            //il n'y aura pas de Connexion simultanée car il n'y pas de nouveau Thread crée
            new ConnexionThread(client,numClient,photos,citations,proverbes);
        }

    }
    catch(java.io.IOException e){
        e.printStackTrace();
        System.exit(0);}
}

public static void main(String [] args){
    new ServeurMulti();
}
}
```

```
/**
 * Exo12 Thread, java.net , javax.swing et sérialization.
 *
 * class exo12.data.ImageSerializable
 *
 */

package exo12.data;

import java.io.Serializable;

import java.awt.Toolkit;
import java.awt.Image;
import java.io.File;
import java.io.FileInputStream;

public class ImageSerializable implements Serializable {

private byte [] buffer;

/**
 * Constructeur à partir d'un File codé en jpeg
 */
public ImageSerializable(File fileImage) {
    try {
        buffer=new byte [ (int ) fileImage.length()];
        new FileInputStream(fileImage).read(buffer);
    }
    catch (java.io.FileNotFoundException e) { e.printStackTrace() ; }
    catch (java.io.IOException e) {e.printStackTrace();}
}

public Image getImage(){
    return Toolkit.getDefaultToolkit().createImage(buffer);
}

}
```

```

/**
 * Exo12 Thread, java.net , javax.swing et sérialization.
 *
 * class exo12.client.Exo12Client
 *
 */

package exo12.cclient;

import javax.swing.UIManager;

import java.awt.*;
import java.io.IOException;
import java.net.ConnectException;

public class Exo12Client {
    boolean packFrame = false;

    //Construire l'application
    public Exo12Client(String adrServeur) {
        NetClient client=null;
        try {
            client=new NetClient(adrServeur,11111);
        } catch (ConnectException e) {
            System.out.println("Pas de serveur disponible");
            System.exit(0);
        } catch (IOException e) {
            System.out.println("Problèmes pour établir les flux entre client et serveur");
            System.exit(0);
        }
        Exo12ClientUI frame = new Exo12ClientUI(client);
        //Valider les cadres ayant des tailles prédéfinies
        //Compacter les cadres ayant des infos de taille préférées - ex. depuis leur disposition
        if (packFrame) {
            frame.pack();
        }
        else {
            frame.validate();
        }
        //Centrer la fenêtre
        Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
        Dimension frameSize = frame.getSize();
        if (frameSize.height > screenSize.height) {
            frameSize.height = screenSize.height;
        }
        if (frameSize.width > screenSize.width) {
            frameSize.width = screenSize.width;
        }
        frame.setLocation((screenSize.width - frameSize.width) / 2, (screenSize.height -
frameSize.height) / 2);
        frame.setVisible(true);
    }

    //Méthode main
    public static void main(String[] args) {
        if (args.length==0){
            System.out.println("Vous devez passer l'adresse du serveur en argument");
            System.exit(0);
        }

        try {
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        new Exo12Client(args[0]);
    }
}

```

```

}

/**
 * Exo12 Thread, java.net , javax.swing et sérialization.
 *
 * class exo12.client.Exo12ClientUI
 *
 */

package exo12.cclient;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

import exo12.data.ImageSerializable;

public class Exo12ClientUI extends JFrame {

//Composants graphiques
  JPanel contentPane;
  JPanel panelAffiche = new JPanelAffiche();
  JButton jButtonPhoto = new JButton();
  JButton jButtonProverbe = new JButton();
  JButton jButtonExit = new JButton();
  JButton jButtonCitation = new JButton();
  GridBagLayout gridBagLayout1 = new GridBagLayout();

//Composants techniques autres
  private NetClient client;
  Object objet=null;
  String stringAffiche="String originale";

//Construire le cadre
  public Exo12ClientUI(NetClient client) {
    this.client=client;
    enableEvents(AWTEvent.WINDOW_EVENT_MASK);
    try {
      jbInit();
    }
    catch(Exception e) {
      e.printStackTrace();
    }
  }

//Initialiser le composant
  private void jbInit() throws Exception {
    contentPane = (JPanel) this.getContentPane();
    contentPane.setLayout(gridBagLayout1);
    this.setSize(new Dimension(407, 325));
    this.setTitle("Titre du cadre");
    contentPane.setMinimumSize(new Dimension(407, 325));
    panelAffiche.setBorder(BorderFactory.createEtchedBorder());
    jButtonPhoto.setMaximumSize(new Dimension(79, 25));
    jButtonPhoto.setMinimumSize(new Dimension(79, 25));
    jButtonPhoto.setPreferredSize(new Dimension(79, 25));
    jButtonPhoto.setMnemonic('0');
    jButtonPhoto.setText("Photo");
    jButtonPhoto.addActionListener(new Exo10ClientUI_jButtonPhoto_actionAdapter(this));
    jButtonProverbe.setMaximumSize(new Dimension(79, 25));
    jButtonProverbe.setSelected(false);
    jButtonProverbe.setText("Proverbe");
    jButtonProverbe.addActionListener(new Exo10ClientUI_jButtonProverbe_actionAdapter(this));
    jButtonExit.setMaximumSize(new Dimension(79, 25));
    jButtonExit.setMinimumSize(new Dimension(79, 25));
    jButtonExit.setPreferredSize(new Dimension(79, 25));
    jButtonExit.setMnemonic('0');
    jButtonExit.setSelected(false);
    jButtonExit.setText("Exit");
  }

```

```

jButtonExit.addActionListener(new Exo10ClientUI_jButtononExit_actionAdapter(this));
jButtonCitation.setMaximumSize(new Dimension(79, 25));
jButtonCitation.setMinimumSize(new Dimension(79, 25));
jButtonCitation.setPreferredSize(new Dimension(79, 25));
jButtonCitation.setMnemonic('0');
jButtonCitation.setText("Citation");
jButtonCitation.addActionListener(new Exo10ClientUI_jButtononCitation_actionAdapter(this));
contentPane.add(panelAffiche, new GridBagConstraints(1, 0, 1, 4, 1.0, 1.0
    ,GridBagConstraints.CENTER, GridBagConstraints.BOTH, new Insets(28, 15, 29, 11), 271,
254));
contentPane.add(jButtononCitation, new GridBagConstraints(0, 0, 1, 1, 0.0, 0.0
    ,GridBagConstraints.CENTER, GridBagConstraints.NONE, new Insets(28, 7, 0, 7), 9, 0));
contentPane.add(jButtononPhoto, new GridBagConstraints(0, 1, 1, 1, 0.0, 0.0
    ,GridBagConstraints.CENTER, GridBagConstraints.NONE, new Insets(9, 7, 0, 7), 9, 0));
contentPane.add(jButtononExit, new GridBagConstraints(0, 3, 1, 1, 0.0, 0.0
    ,GridBagConstraints.NORTH, GridBagConstraints.NONE, new Insets(9, 7, 10, 7), 9, 0));
contentPane.add(jButtononProverbe, new GridBagConstraints(0, 2, 1, 1, 0.0, 0.0
    ,GridBagConstraints.WEST, GridBagConstraints.NONE, new Insets(9, 7, 0, 7), 9, 0));
}

//Redéfini, ainsi nous pouvons sortir quand la fenêtre est fermée
protected void processWindowEvent(WindowEvent e) {
    super.processWindowEvent(e);
    if (e.getID() == WindowEvent.WINDOW_CLOSING) {
        System.exit(0);
    }
}

void jButtononCitation_actionPerformed(ActionEvent e) {
    client.putMessage("Citation");
    objet=client.getObject();
    panelAffiche.repaint();
}

void jButtononPhoto_actionPerformed(ActionEvent e) {
    client.putMessage("Photo");
    objet=client.getObject();
    panelAffiche.repaint();
}

void jButtononProverbe_actionPerformed(ActionEvent e) {
    client.putMessage("Proverbe");
    objet=client.getObject();
    panelAffiche.repaint();
}

void jButtononExit_actionPerformed(ActionEvent e) {
    System.exit(0);
}
}
//=====
class JPanelAffiche extends JPanel{

protected void paintComponent(Graphics g) {

    super.paintComponent(g);
    if(objet==null){//on a encore rien demandé
        return;
    }

    int hPanel=getHeight(); //hauteur du canvas
    int lPanel=getWidth(); //largeur du canvas

    if ( objet instanceof String) {
        String s =(String )objet;
        Font f=new Font("SansSerif",Font.BOLD,14);
        g.setFont(f);
        g.setColor(Color.red);
        int lf= g.getFontMetrics().stringWidth(s) ;
    }
}
}

```

```
// Petite manip pour aller à la ligne
int ie;
e1: for(int y=20;;y+=20) {

    for (int i=1;i<s.length();i++) {
        if( g.getFontMetrics().stringWidth(s.substring(0,i)) > lPanel - 20 ){
            ie=s.substring(0,i).lastIndexOf(" ");
            g.drawString(s.substring(0,ie),20,y);
            s=s.substring(ie);
            continue e1;
        }
    }
    g.drawString(s,20,y);
    break;
}

else if (objet instanceof ImageSerializable) {
    Image image= ((ImageSerializable) objet).getImage();
    MediaTracker tracker;
    tracker=new MediaTracker(this);
    tracker.addImage(image,0);
    try {
        tracker.waitForID(0);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }

    int lImag=image.getWidth(this); //largeur et hauteur de image
    int hImag=image.getHeight(this);
    System.out.println("hPanel:" + hPanel + " lPanel:"+lPanel);
    System.out.println("hImag:" + hImag + " lImag:"+lImag);

    float rlilg=(float)lImag/lPanel; //rapport des largeurs
    float rhihg=(float)hImag/hPanel; //rapport des hauteurs
    float r=rililg>=rhihg?rililg:rhihg;

    //largeur et hauteur d'affichage
    int lCible=(int)(lImag/r);
    int hCible=(int)(hImag/r);

    g.drawImage(image,0,0,lCible,hCible,null);
}
}
}
//=====
class Exo10ClientUI_jButtonCitation_actionAdapter implements java.awt.event.ActionListener {
    Exo12ClientUI adaptee;

    Exo10ClientUI_jButtonCitation_actionAdapter(Exo12ClientUI adaptee) {
        this.adaptee = adaptee;
    }
    public void actionPerformed(ActionEvent e) {
        adaptee.jButtonCitation_actionPerformed(e);
    }
}

class Exo10ClientUI_jButtonPhoto_actionAdapter implements java.awt.event.ActionListener {
    Exo12ClientUI adaptee;

    Exo10ClientUI_jButtonPhoto_actionAdapter(Exo12ClientUI adaptee) {
        this.adaptee = adaptee;
    }
    public void actionPerformed(ActionEvent e) {
        adaptee.jButtonPhoto_actionPerformed(e);
    }
}
}
```



```
class Exo10ClientUI_jButtonProverbe_actionAdapter implements java.awt.event.ActionListener {
    Exo12ClientUI adaptee;

    Exo10ClientUI_jButtonProverbe_actionAdapter(Exo12ClientUI adaptee) {
        this.adaptee = adaptee;
    }
    public void actionPerformed(ActionEvent e) {
        adaptee.jButtonProverbe_actionPerformed(e);
    }
}

class Exo10ClientUI_jButtonExit_actionAdapter implements java.awt.event.ActionListener {
    Exo12ClientUI adaptee;

    Exo10ClientUI_jButtonExit_actionAdapter(Exo12ClientUI adaptee) {
        this.adaptee = adaptee;
    }
    public void actionPerformed(ActionEvent e) {
        adaptee.jButtonExit_actionPerformed(e);
    }
}
```

```
/**
 * Exo12 Thread, java.net , javax.swing et sérialization.
 *
 * class exo12.client.NetClient
 *
 */

package exo12.cclient;

import java.io.*;
import java.net.*;

public class NetClient {

private Socket socket;
private PrintStream out;
private ObjectInputStream in;

public NetClient(String host,int port) throws java.net.ConnectException ,
                                                    java.io.IOException {

    socket=new Socket(host,port);
    out=new PrintStream(socket.getOutputStream() );
    in=new ObjectInputStream(socket.getInputStream() );

}

public void putMessage(String message) {
    out.println(message);
}

public Object getObject() {
    try {
        return in.readObject() ;
    }
    catch (java.io.IOException e) {return "Le serveur a disparu. Quitter l'application" ;}
    catch (java.lang.ClassNotFoundException e) {
        return "ClassNotFoundException. Quitter l'application";}

}

}
```