Tableaux et Entrées-Sorties

Valeur d'accueil et reconversion en informatique 1 (VARI1)

Daniel Porumbel

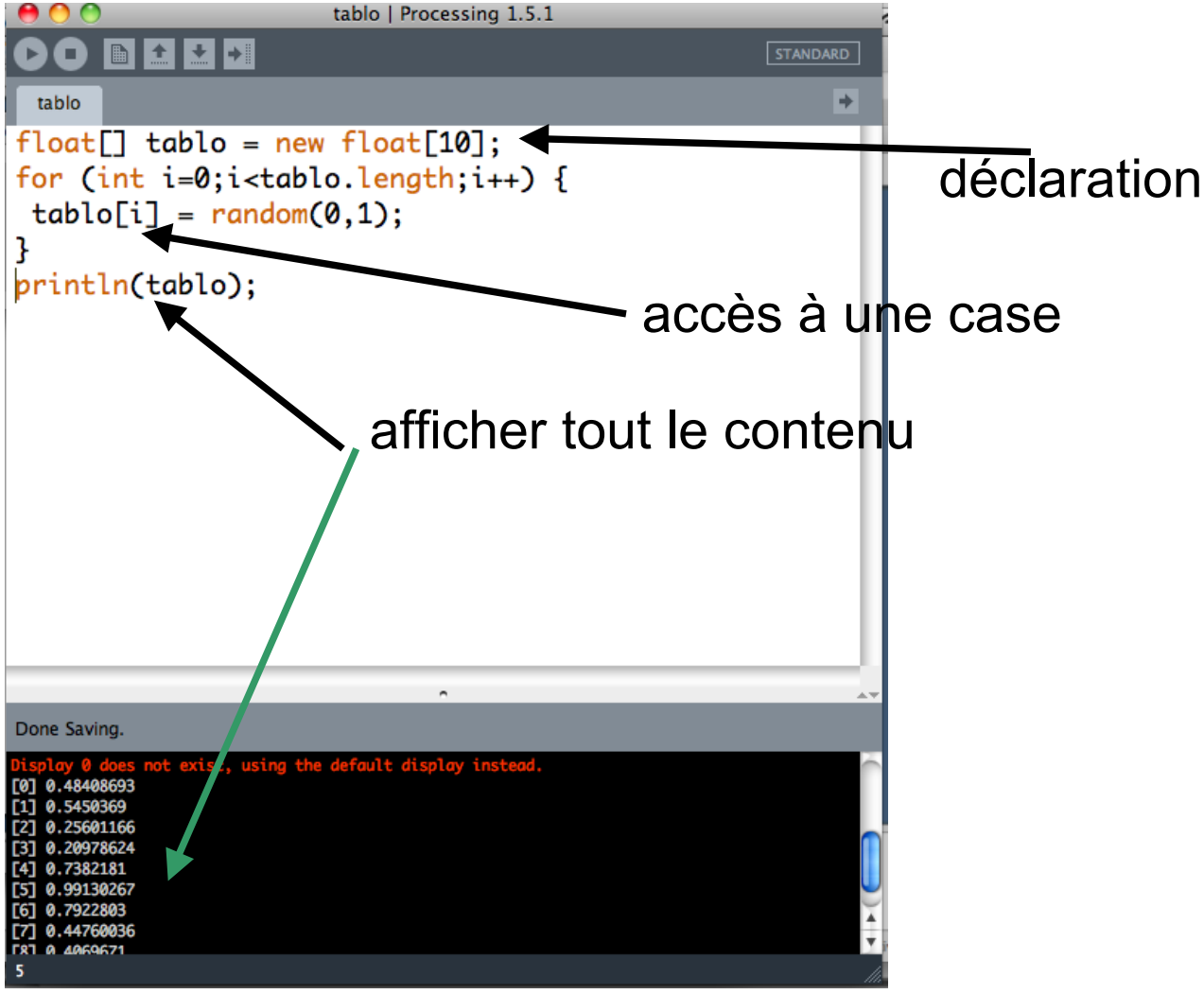
nombreaux slides dus à

Pierre Cubaud

Plan

- Tableaux
  - déclaration et utilisation
  - dimension >1 : matrices, etc.
- Tris et boucles
  - visualisations intuitives
  - le nombre de calculs la complexité
- Entrées-sorties
  - lire un fichier dans un tableau de `String`
  - dessiner la Terre et la France,

(1) Les tableaux



déclaration $\neq$ allocation mémoire

```
float[] tab;              //declaration
tab = new float[10];//alloc mem
```

Le nombre de cases peut dépendre d'une variable

```
tab = new float[n*m];//alloc mem
```

on peut initialiser les tableaux à la main :

```
int[] tab = {1,2,3,5};
```

Comment calculer la valeur maximale ?

Exemple graphique

tablo

```
float[] tablo = new float[100];
for (int i=0;i<tablo.length;i++) {
 tablo[i] = random(0,1);
}
size(100,6*tablo.length);
fill(0);
int y=0;
for (int i=0;i<tablo.length;i++) {
 rect(0,y,100*tablo[i],3);
 y += 6;
}

save("tablo.png");
```

Exemple du tri bulle vu dans le cours Architecture :

- Rechercher le plus petit élément parmi les 10 cases

000	001	002	003	004	005	**006**	007	008	009
12	5	7	100	34	9	**1**	7	9	10

- Faire l'échange avec la première case

000	001	002	003	004	005	**006**	007	008	009
1	5	7	0	34	9	**12**	7	9	10

- Recommencer à partir de la deuxième case, etc…

tribulle

```
int[] tablo = {12,5,7,100,34,9,1,9,10};

for (int i=0;i<tablo.length;i++) {
 int lemin = tablo[i];
 int m=i;
 for (int j=i+1;j<tablo.length;j++) {
   if (tablo[j]<lemin) {m=j;lemin = tablo[m];}
 }
 tablo[m] = tablo[i];
 tablo[i] = lemin;
}

println(tablo);
```

Affectation entre tableaux

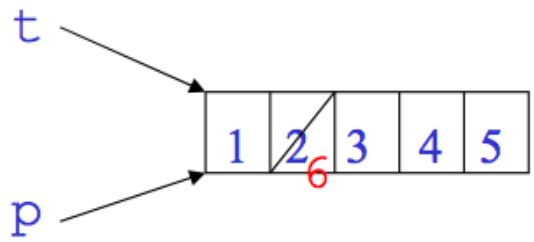
```
int[] t ;
int[] p = {1,2,3,4,5};
t = p;
```

t et p désignent le même tableau

```
            println("t[1]="+t[1]);
            println("p[1]="+p[1]);
t[1] = 6;
            println("nouveau t[1]="+t[1]);
            println("nouveau p[1]="+p[1]);
```

résultat

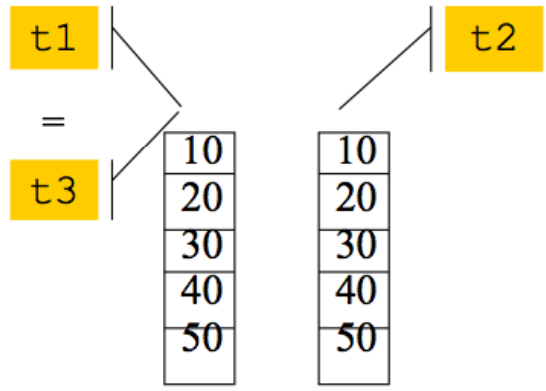
```
t[1]=2
p[1]=2
nouveau t[1]=6
nouveau p[1]=6
```



Comparaison de tableaux

La comparaison entre 2 tableaux porte sur leur référence et non sur leur contenu.

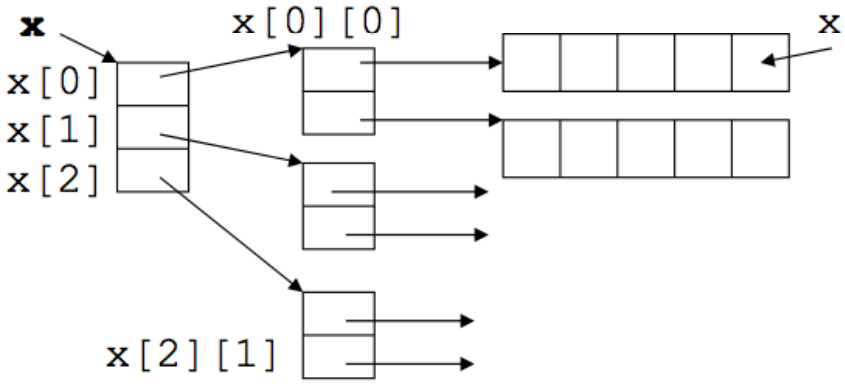
```
int[] t1 = new int[5]{10,20,30,40,50};
int[] t2 = new int[5]{10,20,30,40,50};
if(t1==t2)
                println(t1=t2);
else
                println(t1 t2);
int[] t3 = t1;
if(t3==t1)
                println(t3=t1);
else
                println(t3 t1);
```



Tableaux multi-dimensionnels

- Un tableau à 2 dimensions est un tableau à 1 dimension dont chaque élément est lui-même un tableau à une dimension
- Un tableau à 3 dimensions est un tableau à 1 dimension dont chaque élément est lui-même un tableau à 2 dimensions
- Exemple :

  Soit le tableau `x = new int[3][2][5];`



`x.length=3`

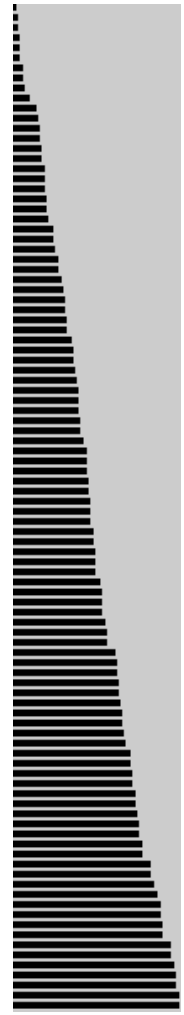
`x[0].length=2`

`x[2][1].length=5`

Fonction de tri toute faite

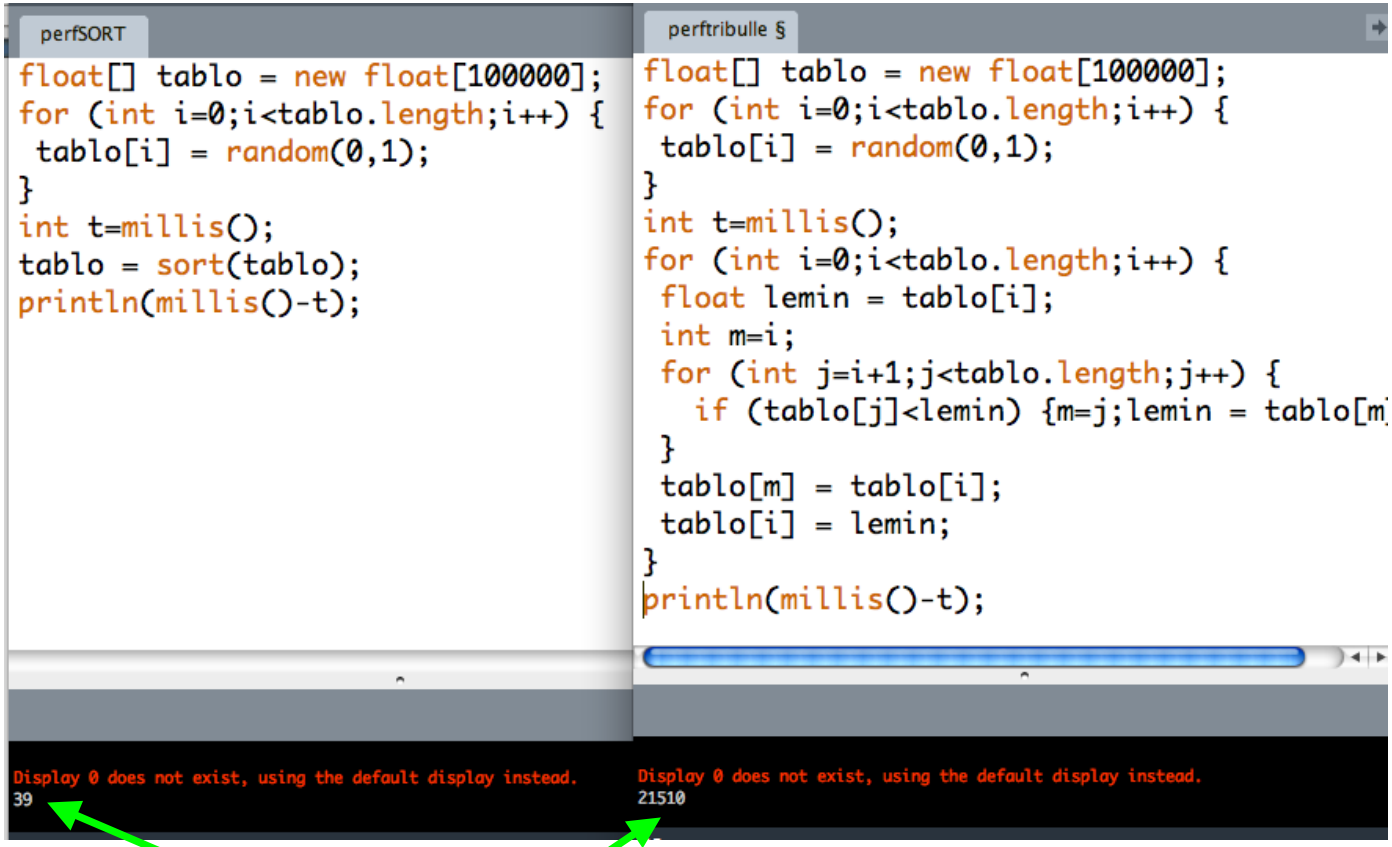
tabloSORT

```
float[] tablo = new float[100];
for (int i=0;i<tablo.length;i++) {
 tablo[i] = random(0,1);
}

tablo = sort(tablo);

size(100,6*tablo.length);
fill(0);
int y=0;
for (int i=0;i<tablo.length;i++) {
 rect(0,y,100*tablo[i],3);
 y += 6;
}
save("tabloSORT.png");
```



Rque : sort() fonctionne aussi avec les tableaux de chaines

temps de calcul du tri : mesure avec millis()



perfSORT

```
float[] tablo = new float[100000];
for (int i=0;i<tablo.length;i++) {
 tablo[i] = random(0,1);
}
int t=millis();
tablo = sort(tablo);
println(millis()-t);
```

```
Display 0 does not exist, using the default display instead.
39
```

perftribulle §

```
float[] tablo = new float[100000];
for (int i=0;i<tablo.length;i++) {
 tablo[i] = random(0,1);
}
int t=millis();
for (int i=0;i<tablo.length;i++) {
 float lemin = tablo[i];
 int m=i;
 for (int j=i+1;j<tablo.length;j++) {
   if (tablo[j]<lemin) {m=j;lemin = tablo[m
 }
 tablo[m] = tablo[i];
 tablo[i] = lemin;
}
println(millis()-t);
```

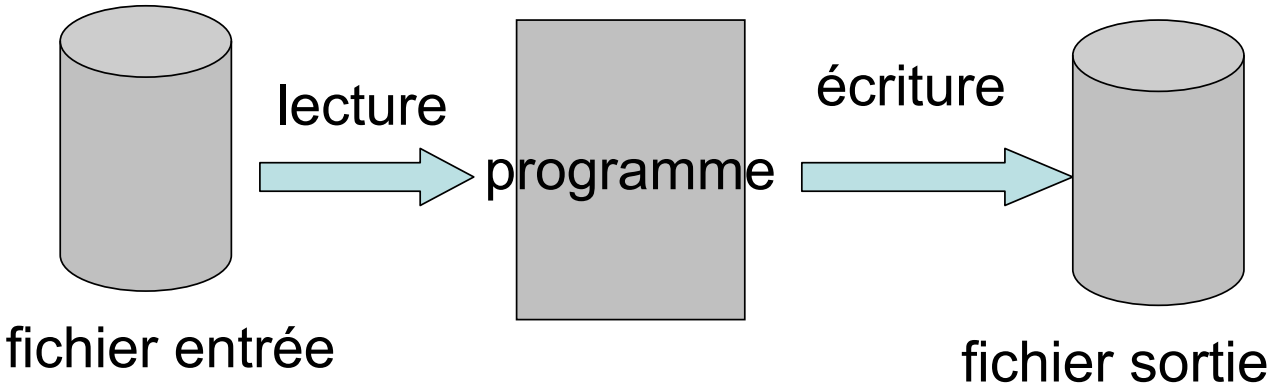
```
Display 0 does not exist, using the default display instead.
21510
```

difficile de lutter avec sort !

nombre de cases	tri bulle	tri sort()
1000	2	5
10000	233	7
100000	21510	39

à revoir en NFP136…

(2) Les fichiers



- ouverture du fichier (open)
- utilisation (lecture ou écriture)
- vidage du tampon d'E/S (fluch)
- fermeture du fichier (close)

- dépendance du système d'exploitation
- flots d'octets ≠ flots de caractères (texte)
- disque = accès lent

Les fichiers avec Processing version >2

en entrée:

Files
BufferedReader
createInput()
createReader()
loadBytes()
loadJSONArray()
loadJSONObject()
loadStrings()
loadTable()
loadXML()
open()
parseXML()
selectFolder()
selectInput()

en sortie:

Files
beginRaw()
beginRecord()
createOutput()
createWriter()
endRaw()
endRecord()
PrintWriter
saveBytes()
saveJSONArray()
saveJSONObject()
saveStream()
saveStrings()
saveTable()
saveXML()
selectOutput()

On ne va pas tout voir !!

Exemple lecture d'entiers

Soit un fichier `abc.txt` :

```
12
19
```

Exemple lecture d'entiers

Soit un fichier `abc.txt` :

```
12
19
```

Le code suivant permet de lire le contenu dans deux variables

```
String[] lignes= loadStrings("abc.txt");
int x = int(lignes[0]);
println(x);
int y = int(lignes[1]);
println(y);
```

Exemple complet : tracer le monde

Nous avons un fichier `lemonde.csv` :

```
...
-5.6619487,54.554604,Europe
-6.197885,53.867565,Europe
141.00021,-2.600151,Asia
142.73524,-3.2891529,Asia
...
32.94696,35.386703,Africa
33.66723,35.373215,Africa
```

Objectif

- lire toutes ces ligner et tracer le contour du monde !

Étape 1 : lecture du fichier

```
...
-5.6619487,54.554604,Europe
-6.197885,53.867565,Europe
141.00021,-2.600151,Asia
142.73524,-3.2891529,Asia
...
32.94696,35.386703,Africa
33.66723,35.373215,Africa
```

On va lire ces données dans un tableau :

```
String lignes[] ;
lignes = loadStrings("lemonde.csv");
```

Le nombre de ligne est *lignes.length*. On peut découper chaque ligne *ligne*[*i*] en deux grâce à la fonction *split*.

étape 1 :
- chargement du fichier texte dans un tableau
- on récupère le nombre de lignes du fichier

étape 2 :
décomposition de la ligne lue : split(chaine,separateur)

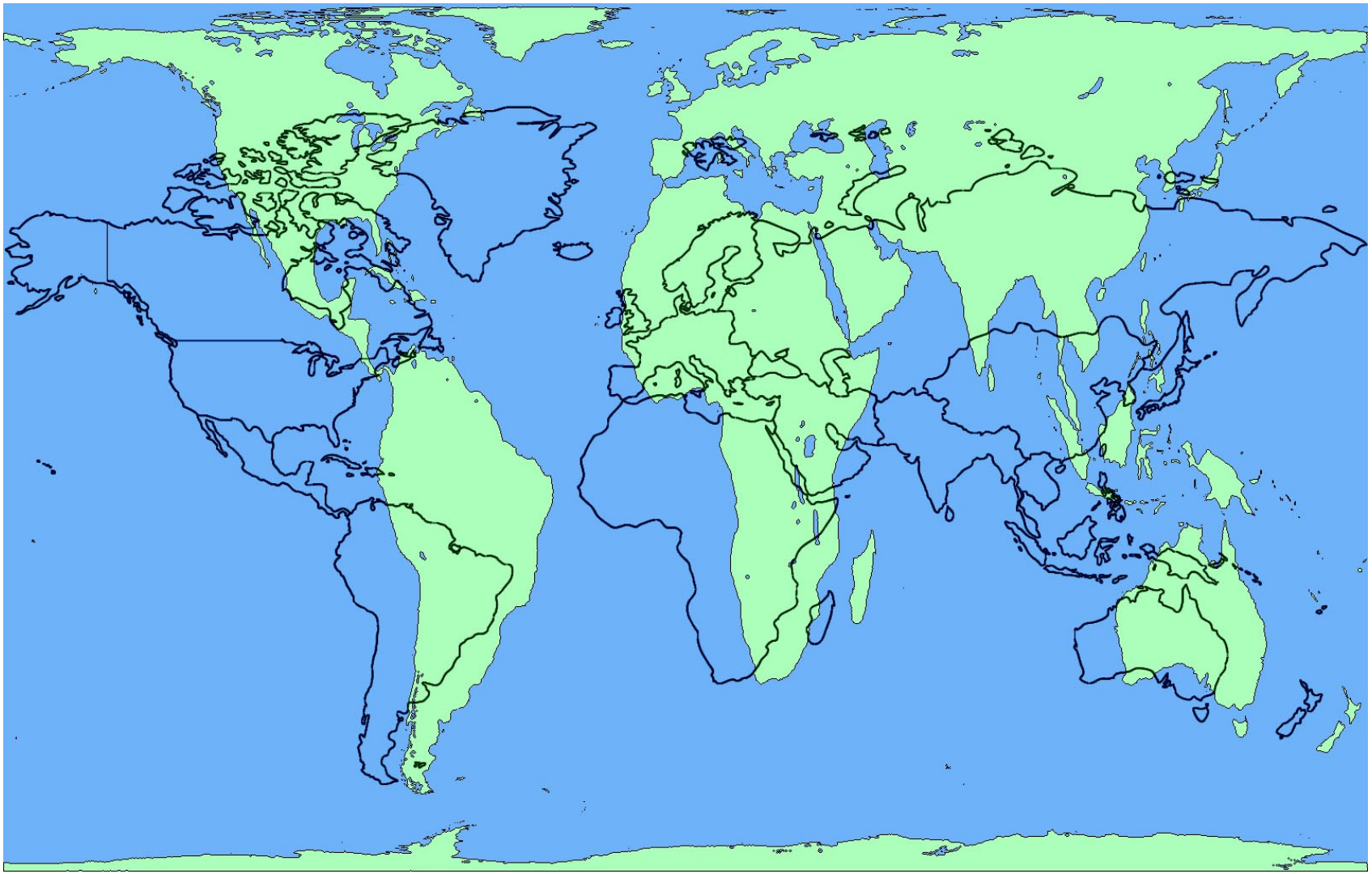
étape 3 :
map() effectue une conversion linéaire

loadString() peut échouer :
il ne faut jamais supposer dans un programme que les ressources (fichier, etc) existent
=> mécanisme des exceptions vu plus tard

Un squellete de code

```
String lines[] ;
lines = loadStrings("lemonde.csv");
size(1000,800);
background(0);
for (int i=0; i < ..... ; i++) {
   .... vals = split(lines[i],",");
   if(vals.length>=3) {
     float longitude = float(vals[0]);
     float latitude = float(vals[1]);
     stroke(100,000,200);
     point(4*longitude+500,4*latitude+400);
     //4* pour zoomer, 500,400=center
   }
}
```

L'impact de la projection de Mercator



Aucune projection n'est parfaite : L'Amérique du Sud semble plus petite que le Groenland ; en réalité, elle est 8 $\times$ plus grande

Réalisation de la projection de Mercator

Il faut appliquer une formule sur la latitude :

```
latitude = 50*log(tan(PI/4)
           +radians(latitude)/2);
```

Réalisation de la projection de Mercator

Il faut appliquer une formule sur la latitude :

```
latitude = 50*log(tan(PI/4)
           +radians(latitude)/2);
```

- Comment tracer l'émisphère Sud en vert ?

Réalisation de la projection de Mercator

Il faut appliquer une formule sur la latitude :

```
latitude = 50*log(tan(PI/4)
           +radians(latitude)/2);
```

- Comment tracer l'émisphère Sud en vert ?
- Comment tracer l'Europe en bleu ?
  - On utilise la fonction `match(String a,String b)`

Produire un fichier : `saveStrings(...)`

Un simple exemple :

```
String[] lignes = new String[3];
lignes[0] = "première ligne";
lignes[1] = "deuxième ligne";
lignes[2] = "troisème ligne";
saveStrings("test.txt",lignes);
```

- utile lorsqu'on peut tout mémoriser dans un tableau
- on doit connaître le nombre de lignes à l'avance !
- Si on replacait la première ligne par
  `String[] lignes = new String[5]` ?

Produire un fichier : `createWriter(...)`

```
PrintWriter fic;
fic=createWriter(nomFichier);
fic.println(....);
....
fic.flush();//vidage tampon E/S
fic.close();//fermer le fichier
```

Produire un fichier : `createWriter(...)`

```
PrintWriter fic;
fic=createWriter(nomFichier);
fic.println(....);
....
fic.flush();//vidage tampon E/S
fic.close();//fermer le fichier
```

Écrire l'europe dans un fichier `europe.csv`

Programmes entrée-sortie

1. Générer 100 nombres aléatoires (entre 1 et 1000) et stocker-les dans un fichier `a.txt`

Programmes entrée-sortie

1. Générer 100 nombres aléatoires (entre 1 et 1000) et stocker-les dans un fichier `a.txt`
2. Calculer l'addition de tous les nombres stockés dans un fichier `a.txt` et écrire le résultat dans un fichier `b.txt`
   - le fichier `a.txt` comporte un nombre par ligne

Programmes entrée-sortie

1. Générer 100 nombres aléatoires (entre 1 et 1000) et stocker-les dans un fichier `a.txt`
2. Calculer l'addition de tous les nombres stockés dans un fichier `a.txt` et écrire le résultat dans un fichier `b.txt`
   - le fichier `a.txt` comporte un nombre par ligne
3. Trier les nombres stockes dans un fichier `a.txt` et écrire le tableau trié dans un fichier `c.txt`