

Kit de survie Java

tous ces exemples sont relatifs à un répertoire java/

app/Forest3.java

forest/Tree.java

```
package forest;

/**
 * A simple tree
 */
public class Tree {
    // A unique id
    private int id;
    // Diameter at 1.3m (cm)
    private double dbh;

    /**
     * Constructor
     */
    public Tree (int id, double dbh) {
        this.id = id;
        this.dbh = dbh;
    }

    public int getId () {return id;}

    public double getDbh () {return dbh;}

    public void setDbh (double dbh) {this.dbh = dbh;}

    public String toString () {
        return "Tree id: "+id+" dbh: "+dbh;
    }
}
```

un type Tree
un Tree a un id et un dbh
quelques accesseurs

un programme (méthode main)
création d'un objet Forest
appel de sa méthode addTrees ()
écriture dans le terminal

app/Forest1.java

```
package app;

import forest.*;

/**
 * A program to create a little forest and print
 * it to the terminal
 */
public class Forest1 {

    // Entry point
    public static void main (String[] args) {
        Forest f = new Forest ("Forest 1");
        f.addTrees (5);

        System.out.println (f.toString ());
    }
}
```

forest/XYZTree.java

```
package forest;

/**
 * A tree with a location
 */
public class XYZTree extends Tree{
    // tree location
    private double x;
    private double y;
    private double z;

    /**
     * Constructor
     */
    public XYZTree (int id, double dbh,
                   double x, double y, double z) {
        super (id, dbh);
        this.x = x;
        this.y = y;
        this.z = z;
    }

    public double getX () {return x;}
    public double getY () {return y;}
    public double getZ () {return z;}

    public String toString () {
        return super.toString () + " xyz: (" + x
            + ", " + y + ", " + z + ")";
    }
}
```

un type supplémentaire
hérite de Tree, ajoute une localisation
redéfinition de toString ()

```
package app;

import java.util.*;
import java.io.*;
import forest.*;

/**
 * A program to create a little forest save it in
 * a file. Relies on XYZTree instances with coordinates.
 */
public class Forest3 {

    // Entry point
    public static void main (String[] args) throws Exception {
        Forest f = new Forest ("Forest 3");
        List trees = f.getTrees ();

        double xSize = 10;
        double ySize = 7;

        Random random = new Random ();
        for (int i = 0; i < 7; i++) { // 7 trees
            double dbh = random.nextDouble () * Forest.MAX_DBH; // 0-30cm
            double x = random.nextDouble () * xSize; // x in 0-10
            double y = random.nextDouble () * ySize; // y in 0-7

            Tree t = new XYZTree (i, dbh, x, y, 0); // z = 0
            trees.add (t);
        }

        // Relies on the save () static method in Forest2
        Forest2.save (f, "forest3.txt");
    }
}
```

un troisième programme
création d'une Forêt, ajout de
7 arbres localisés dans un
rectangle de taille 10x7

forest/Forest.java

```
package forest;

import java.util.*;

/**
 * A forest
 */
public class Forest {
    public static final double MAX_DBH = 30; // cm
    // The trees in the forest
    private List trees;
    // Forest name
    private String name;

    /**
     * Constructor
     */
    public Forest (String name) {
        this.name = name;
        trees = new ArrayList ();
    }

    public void addTrees (int number) {
        Random random = new Random ();
        for (int i = 0; i < number; i++) {
            double dbh = random.nextDouble () * MAX_DBH; // 0-30cm
            Tree t = new Tree (i, dbh);
            trees.add (t);
        }
    }

    public int getMaxTreeId () {
        int maxId = 0;
        for (int i = 0; i < trees.size (); i++) {
            Tree t = (Tree) trees.get (i);
            maxId = Math.max (maxId, t.getId ());
        }
        return maxId;
    }

    public List getTrees () {return trees;}

    public String toString () {
        int n = trees.size ();
        String s = "Forest: " + name + " (" + n + " trees)" + "\n";
        for (Iterator i = trees.iterator (); i.hasNext ();) {
            Tree t = (Tree) i.next ();
            s += t.toString () + "\n";
        }
        return s;
    }
}
```

un type Forest
un nom
une liste de Tree

une constante

une liste

nombres aléatoires

boucle for avec indice

boucle for sur Iterator

un deuxième programme (méthode main)
appel d'une méthode save ()
save () est static : on peut l'appeler depuis main

app/Forest2.java

```
package app;

import forest.*;
import java.io.*;

/**
 * A program to create a little forest save it in
 * a file
 */
public class Forest2 {

    // Entry point
    public static void main (String[] args) throws Exception {
        Forest f = new Forest ("Forest 2");
        f.addTrees (10);

        Forest2.save (f, "forest2.txt");
    }

    /**
     * Saves the given forest in a file with the given
     * fileName
     */
    static private void save (Forest f, String fileName) throws Exception {

        try {
            BufferedWriter out = new BufferedWriter (new FileWriter (fileName));

            out.write (f.toString ());
            out.newLine ();

            out.close ();
        } catch (Exception e) {
            System.out.println ("Could not write in file: " + fileName);
            throw e;
        }
    }
}
```

```
> cd java
> javac forest/*.java
> javac app/*.java

> java app.Forest1
Forest: Forest 1 (5 trees)
Tree id: 0 dbh: 8.663007223551755
Tree id: 1 dbh: 4.39927403267964
Tree id: 2 dbh: 22.458451145909137
Tree id: 3 dbh: 12.897102940711283
Tree id: 4 dbh: 22.101207369016

> java app.Forest2
> java app.Forest3
```

compilation

exécution

```
forest3.txt
Forest: Forest 3 (7 trees)
Tree id: 0 dbh: 4.918207932115591 xyz: (3.5995570420746645, 5.171868530132126, 0.0)
Tree id: 1 dbh: 15.328032720311354 xyz: (3.7096119041058975, 0.10301231106553532, 0.0)
Tree id: 2 dbh: 12.682695475913748 xyz: (6.668291787914095, 5.298347403362921, 0.0)
Tree id: 3 dbh: 17.697447032877413 xyz: (0.8992867763825396, 1.451138860537621, 0.0)
Tree id: 4 dbh: 24.594047290730263 xyz: (0.9721391027739423, 0.16771794867285728, 0.0)
Tree id: 5 dbh: 28.815406066454617 xyz: (0.21924290667956425, 6.6317581448157465, 0.0)
Tree id: 6 dbh: 20.482315440350554 xyz: (9.916508029816278, 6.4599980232954195, 0.0)
```