```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package javalab1;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.nio.file.Files;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashSet;
import java.util.Objects;
import java.util.Set;
import java.util.TreeSet;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author Damian Darczuk
 */
public class DiskFile implements Serializable, Comparable<DiskFile> {

    public enum Type {
        Katalog,
        Plik;
    }

    private Set<DiskFile> files;
    private Date date;
    private String name;
    private Type type;

    public DiskFile() {
    }


    public DiskFile(String sciezka, int tryb) {
        File file = new File(sciezka);

        this.name = file.getName();
        this.date = new Date(file.lastModified());

        if (file.isFile())
             this.type = type.Plik;
        else if (file.isDirectory()) {
            this.type = type.Katalog;

            File[] filesArray = file.listFiles();

            if (tryb == 1) {
                files = new TreeSet();
                //tryb = 3;
                if(filesArray != null)
                for(File f : filesArray) {
                    files.add(new DiskFile(f.getAbsolutePath(),tryb) );
                }

            }
            else if (tryb == 2) {
                files = new HashSet();
                //tryb = 3;

                for(File f : filesArray) {
```

```
                files.add(new DiskFile(f.getAbsolutePath(),tryb) );
                }

            }

         }

    }
    public void wypisz() {
         wypisz(1);
    }

    public void wypisz(int level) {

        for (int n = 0; n < level; n ++)
            System.out.print("-");
        System.out.printf("%-30.30s",this.getName());

        String k = "K", p = "P";
        if (this.getType() == type.Katalog)
             System.out.printf("\t%-20s\t", k);
        else
            System.out.printf("\t%-20s\t", p);

        SimpleDateFormat sdf = new SimpleDateFormat("yyy-MM-dd");
        String formatedDate = sdf.format(date);
        System.out.println(formatedDate);
            level ++;
        if (type.Katalog == this.type) {
            if(this.files!=null)
                for(DiskFile f : this.files) {
                        f.wypisz(level);
                }
        }
    }

    Type getType() {
        return this.type;
    }

    String getName() {
        return this.name;
    }

     @Override
    public int hashCode() {
        return Objects.hashCode(this.name + this.date);
    }

    public boolean equals(DiskFile obj) {

        if (this.upperCaseCounter() == obj.upperCaseCounter())
             return true;
        else
            return false;
    }


    @Override
    public boolean equals(Object obj) {
        if (obj == null) {
            return false;
        }
        if (getClass() != obj.getClass()) {
            return false;
        }
        final DiskFile other = (DiskFile) obj;
        if (!Objects.equals(this.name, other.name)) {
            return false;
        }
        if (this.date != other.date) {
            return false;
        }
        return true;
    }

    private int upperCaseCounter() {
        int numberUpperCase = 0;
        String string = this.getName();
        for(int i = 0; i < string.length(); i++) {
            if (Character.isUpperCase(string.charAt(i)))
        {
                    numberUpperCase++;
        }
        }
        return numberUpperCase;
    }



     @Override
    public int compareTo(DiskFile o) {

         if (this.upperCaseCounter() > o.upperCaseCounter())
            {
                return -1;
            }
            else if(this.upperCaseCounter() < o.upperCaseCounter()){
                return 1;
            }
            else return name.compareTo(o.name);
    }

}
```