Sa se implementeze o aplicatie care numara liniile de cod din toate fisierele .java dintr-un folder impreuna cu toate subfolderele sale.

Programul poate avea urmatorii parametri:

**-ignoreSpaces**: la contorizare se vor ignora liniile goale (cele pentru care metoda trim() returneaza un sir gol)

**-ignoreComments**: la contorizare se vor ignora liniile care fac parte dintr-un comentariu java

-**countAccessSpecifiers:** contorizeaza numarul de aparitii al specificatorilor de acces din toate fisierele java si afiseaza un raport despre numarul lor

Intotdeauna pe ultima pozitie a listei de parametri se va afla folderul in care vor fi cautate fisierele .java.

Daca acesta lipseste va fi folosit folderul curent.

Outputul va fi de forma:

**12 folders**

**231 files**

**3212 lines of code**

In cazul specificarii parametrului -countAccessSpecifiers la outputul de mai sus se va adauga o secventa de forma:

**Access specifiers report:**

**20 public (40%)**

**0 protected (0%)**

**30 private (60%)**

Orice combinatie de argumente la apelarea programului este valida. Cateva exemple valide de apelare a programului:

**CountLines -checkJavaFile e:\workspace**

**CountLines -ignoreSpaces -ignoreComments e:\workspace**

**CountLines e:\workspace\xxx\_test1**

**CountLines**

**Programul trebuie sa trateze eventualele cazuri de eroare (parametri invalizi, folder inexistent) si sa afiseze un mesaj de eroare corespunzator.**

Exemplu de cod pentru lucrul cu fisierele:

File f = **new** File("."); // "." = directorul curent

File[] files = f.listFiles();

**for** (File file : files) {

System.*out*.println(file.getAbsolutePath() + " isDirectory=" + file.isDirectory());

}