

14. Rotate Image (90, 180, 270)

Tema proiectului a fost rotirea unei imagini cu 90, 180 sau 270 de grade. Proiectul a fost realizat in Eclipse, cu Java.

Mod de rulare: După rulare, utilizatorului i se va cere numele imaginii care trebuie rotită (ex: imagine.bmp), apoi i se va cere calea unde se afla imaginea respectivă (ex: C:\Users\saioc\Desktop), apoi calea unde se va salva imaginea după rotire și în final valoarea unghiului de rotație (care poate fi doar 90, 180 sau 270). După introducerea corectă a datelor aplicația va crea o nouă imagine cu numele rotated_(Nume_vechi).bmp, unde se va regăsi imaginea rotită.

Structura: Este alcătuită din 2 pachete, primul “Test” este pachetul de testare a aplicației, în care se găsește main-ul, în care se află secvențele de citire a datelor de la tastatura, initializarea obiectelor necesare și aplicarea funcțiilor pentru citirea, scrierea și rotirea imaginii.

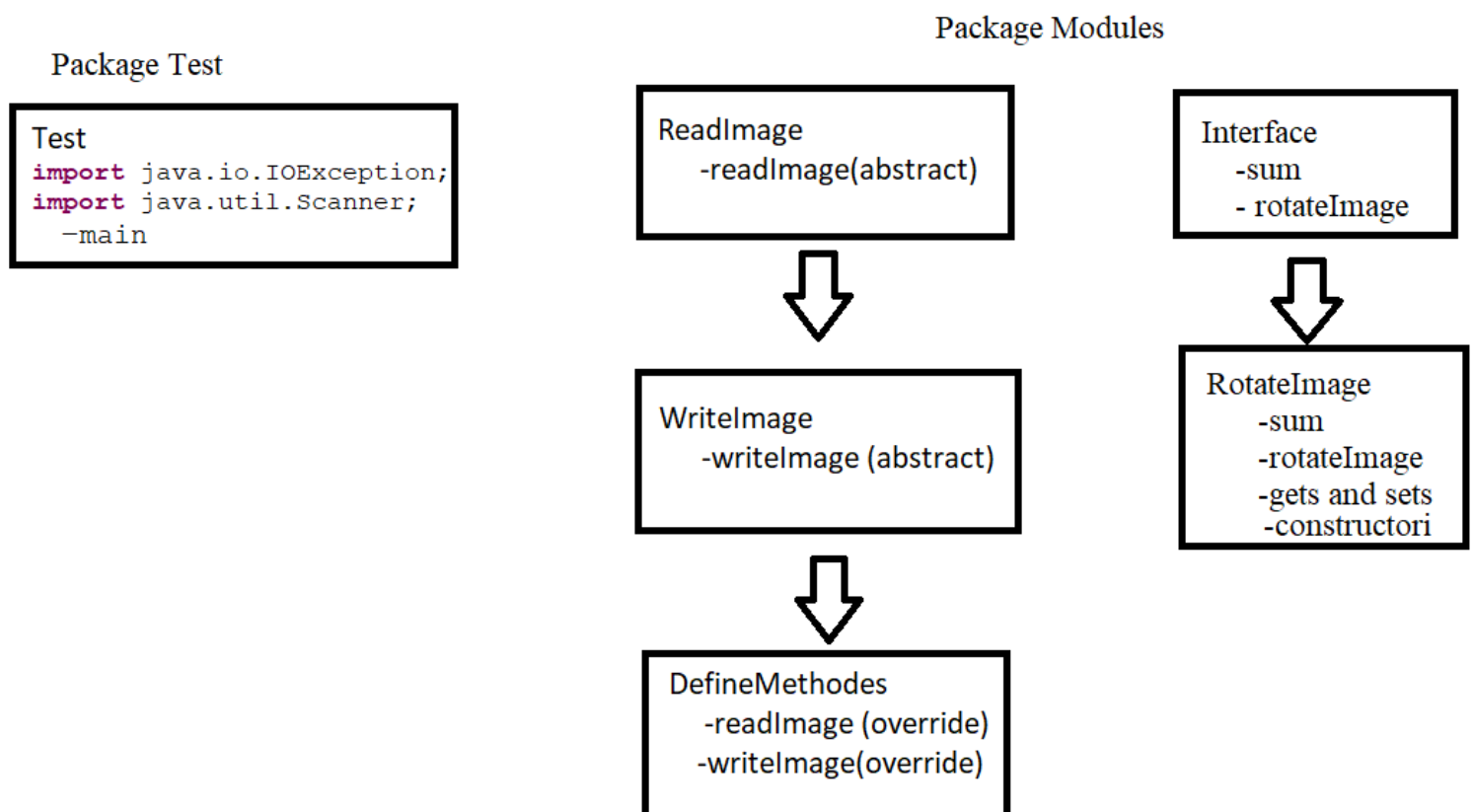
Cel de-al doilea pachet, numit “Modules” conține mai multe clase după cum urmează:

Pentru implementarea celor 3 module de moștenire avem clasa abstracta ReadImage care conține o singura metodă abstractă readImage. Aceasta este moștenită de clasa WriteImage care conține metoda abstractă writeImage, acesta fiind nivelul 2 de moștenire. Iar la nivelul 3 de moștenire, WriteImage este moștenită de clasa

DefineMethodes în interiorul căreia se suprascriu și se implementează cele 2 metode menționate anterior.

În acest pachet “Modules” mai avem clasa Interface, care definește 2 metode sum și rotateImage, care vor fi implementate în clasa RotateImage, clasă care implementează interfață.

Se menționează faptul că metoda sum este didactică, implementată doar cu scopul includerii varargs în aplicație, metoda returnând o simplă sumă de variabile.



Concluzii:

Timpii de rulare ai aplicației au fost aproximativ aceiași pentru aceeași poza, timpii fiind direct influențați de dimensiunea pozei care trebuie rotită. Am observat însă că, indiferent de dimensiunea pozei, se respecta regula în ceea ce privește comparația timpilor de execuției între ei: cel mai rapid este algoritmul de rotire, urmat de cel de citire a pozei, iar scrierea pozei într-un fișier nou consumă cel mai mult timp.

Bibliografie:

1. <https://stackoverflow.com/questions/42778773/how-to-implement-rotate-image-algorithm-and-scale-image-algorithm/42778991>
2. <http://www.java2s.com/example/java/2d-graphics/rotate-image-270.html>
3. <https://www.youtube.com/watch?v=RLHG1dR3Tsl>