Butelca Radu-Andrei
Grupa 143
README:
Am implementat clasa de baza IO ce contine functii virtuale de afisare si de citire, pe urma clasa Helper pentru verificare de inputuri.
Clasa animal este derivata din IO si mosteneste clasa Vertebrate; (nu am implementat clasa Nevertebrate deoarece nu aveam ce sa adaug ca si variabila un atribut ce nu il au vertebratele).
Din clasa nevertebrate reies 4 clase: Pesti, Mamifere, Reptile si Pasari;
Am realizat clasa template AtlasZoologic ce contine o lista de tip pointer de animale cu metodele aferente cerintei si specializarea ei pentru tipul Pesti cu metoda ce returneaza numarul de pesti rapitori cu lungime mai mare decat 1.
Punctul 5 din barem mi s-a parut putin fortat din prisma faptului ca trebuie inclusa o bucla de tip trycatch: parerea mea fiind ca try-catch ar trebui implementate doar in cazuri exceptionale, inputurile nefiind parte din acestea. De aceea, prin ajutorul clasei Helper am facut in asa fel incat la anumite variabile asa-zise inputuri sa fie sugerate in consola.
Upcasting am utilizat in interiorul meniului, prin declararea de obiecte si const_cast in clasa Animale cu scopul de a incrementa numarul de animale descoperite(ceea ce se intampla foarte rar, de aici si motivul utilizarii).
Ca design pattern am folosit Singleton in clasa Menu, ideea de baza fiind ca aceasta clasa este necesar suficient sa fie declarata o singura data.
INPUT folosit:
0
5g
5
Maimuta
Tf
Terestru
21303
0
Delfin
Acvatic
32401

Somon

Fds

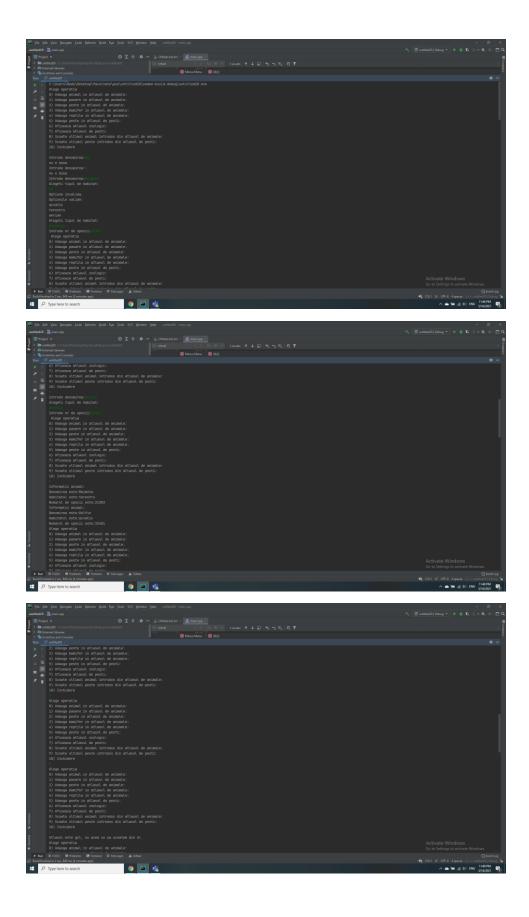
Acvatic

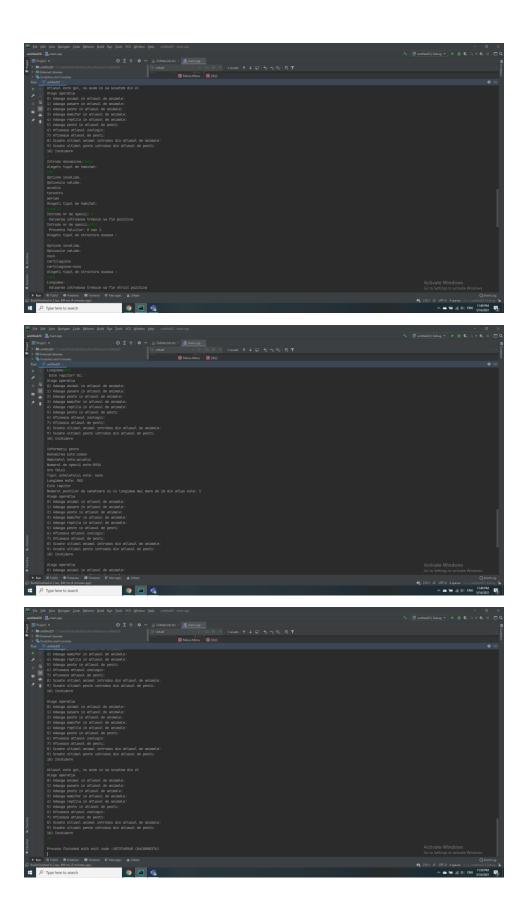
-4

Fd

Osos

-4





Mentiune: Pentru clase IO si Menu m-am inspirat in realizarea lor de la tutorialul de programare orientata pe obiecte 2021, cu mentiunea ca si cei de acolo s-au inspirat de la randul lor de la tutorele lor de laborat, sursa nefiind valabila pentru mine.