

# Tema laborator 7

## Mostenire I.

In aceasta tema veti exersa “mecanica” mostenirii. Va fi un exemplu simplu ce pune in evidenta ideea de overriding, de mostenire a membrilor si metodelor.

Se va implementa o ierarhie simpla de animale. Fiecare clasa va implementa cateva operatii de baza. Vetii face diverse operatii cu aceste “animale”.

### 1 Materiale aditionale ce trebuie studiate

Cititi si intelegeti cursul cu mostenirea.

Cititi si intelegeti laboratorul 7 Mostenire de la d-nul prof Joldos

Folositi cu incredere google-ul si java tutorials.

<http://docs.oracle.com/javase/tutorial/java/IandI/subclasses.html>

### 2 Practica

#### 2.1 Ierarhia de clase

Creati un proiect nou.

Creati urmatoarea ierarhie de clase:

Clasa Animal ce au metodele

String faSunet(int tarie) //Implicit se returneaza “Necunoscut”

String misca(int distanta) //Implicit se returneaza “Miscat pe distanta “ urmat de valoarea distantei

String getNume() //Implicit se returneaza “nesetat”

Clasa Caine care mosteneste clasa Animal

Cainele are un nume (de ex Grivei sau Belzebut). Acest nume se poate seta printr-un constructor. Atentie, trebuie ca aceasta clasa sa accepte si constructorul implicit.

Pe langa un constructor implicit,

Suprascrieti metoda faSunet astfel incat sa returneze “HAM”. Numarul de A din HAM trebuie sa fie egal cu taria (ex pt tarie == 5 se scrie HAAAAAM). Trebuie afisat cel putin un A in HAM.

Suprascrieti metoda geNume astfel incat sa returneze “Sunt un caine numit “ urmat de numele cainelui (Ex: “Sunt un caine numit Baiazid”)

Clasa PisicaSalbatica, extinsa din Animal.

Metoda faSunet returneaza miau, cate un a, pentru fiecare tarie/10 (ex: pentru tarie == 20 va afisa miaau)

Metoda getNume returneaza Pisica Salbatica

Clasa Pasare, extinde Animal

Pasare are un nume, pastrat intr-o variabila ce trebuie sa fie accesibila doar din clasele copil (numita numeSpecie)

faSunet() va returna Cip Cirip

getNumere returneaza numele pastrat in variabila numeSpecie

Clasa mai are o metoda abstracta, accesibil doar copiilor si nu este vizibila din exterior: int calculeazaDistanta(int distanta) care returneaza distanta parcursa, in functie de specie.

Metoda misca(): Returneaza un string "Am zburat pe distanta: " dupa care sa afiseze distanta calculata cu calculeazaDistanta()

Clasa Cuc ce extinde Pasare

Din constructorul implicit se va apela constructorul clasei parinte setdandu-se numele "Cuc"

La faSunet() se returneaza "Cucu"

Se implementeaza si metoda abstracta calculeazaDistanta(). Cucul zboara pe maxim 10m, dupa care trebuie sa se odihneasca. Deci, metoda nu va returna mai mult de valoarea 10.

Clasa Vultur extinde Pasare

Variabila numeSpecie va fi setata la "Vultur"

Vulturul se poate deplasa mult mai rapid, deci, pentru fiecare unitate primita la input vulturul va zbura de 2 ori mai mult. (ex: misca(10) va afisa "Am zburat pe distanta 20")

## 2.2 Operatii pe clase

In main() faceti o lista ce contine obiecte de tip Animal in care puneti vulturi, caini pisici, etc.

Faceti o metoda, una care apeleaza pentru fiecare animal din colectie getNume(), apoi alta care apeleaza misca() pentru o colectie si inca una pentru faSunet()

Observati ca se apeleaza metoda corecta, in functie de tipul animalului.

Ideea de a combina metoda abstracta si apoi o metoda concreta ce se bazeaza pe metoda abstracta va fi exploatata in laboratorul viitor.

## 3 Evaluare

Executati toti pasii din laborator.