class Animal {

private String identitate;

public Animal() { this.setIdentitate("animal"); }

public String getIdentitate() { return identitate; }

public void setIdentitate(String identitate) { this.identitate = identitate; }

}

class Lup extends Animal {

public Lup() {

super();

super.setIdentitate("lup");

}

}

void afiseazaAnimale(ArrayList<? super Lup> lista){

lista.forEach(x -> {

System.*out*.println(x.getIdentitate());

});

}

Ce se va afisa in urma executarii urmatoarelor linii de cod?

ArrayList<Animal> lista = new ArrayList<>();

Animal a1 = new Animal();

Animal a2 = new Lup();

Lup a3 = new Lup();

lista.add(a1);

lista.add(a2);

lista.add(a3);

*afiseazaAnimale*(lista);

1. animal, animal, lup
2. animal, lup, lup
3. animal, animal, animal
4. eroare de compilare

Raspuns corect d. Eroarea de compilare apare deoarece in functia “afiseazaAnimale” avem ca parametru un container de <? super Lup>, ceea ce este un tip generic delimitat inferior, asadar putem apela doar functii apartinand clasei sau interfetei din limita superioara, aceasta fiind clasa Object care nicidecum nu are o implementare pentru functia “getIdentitate()”.